

Dead Reckoning in Mobile Ad Hoc Networks

Aarti Agarwal
Department of ECECS
University of Cincinnati
Cincinnati, OH 45221

Samir R. Das
Computer Science Department
SUNY at Stony Brook
Stony Brook, NY 11794

Abstract—A predictive model-based mobility tracking method, called dead reckoning, is proposed 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. This technique is applied to solve the unicast routing problem by modeling link costs using both link lifetime and geographic distance from the destination to the link egress point. This method presents a much superior routing performance compared to either DSR or AODV, two other popular routing protocols, particularly in terms of delivery fraction and routing load.

I. INTRODUCTION

Routing protocols in mobile ad hoc networks must adapt to frequently changing topologies without generating significant routing overhead. It is conceivable that availability of location information on the nodes of the network may make the routing process simpler. Several location-based routing protocols have been proposed (see, for example, [1], [2]) 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 network. However, this issue has not received enough attention. Location update can be purely opportunistic or passive (e.g., in LAR [1]), where location information is simply piggybacked on packets ordinarily exchanged by nodes. This makes the information stale very easily if the source-destination pair does not communicate very often, or if the destination node moves very fast. Either of this also makes the “paging area” (geographic area where the destination can be located at the time of communication) very large increasing routing overhead. On the other hand, protocols with proactive updates (e.g., SLURP [3], GLS [4], DREAM [5], and GRSS[6]) take a very simple approach; a node updates its location when it moves outside a pre-defined region (SLURP) or moves a certain distance (GLS) or simply periodically (DREAM and GRSS). Such updates are generally dependent on node speed. The updates are sent to an appropriate set of nodes based on the characteristics of the protocol used. These are the set of the nodes that are to be “paged” if the node in question needs to be “queried” (to establish route, for example).

In most practical systems, a mobile node typically travels with a destination in mind. Even otherwise, its mobility pattern is unlikely to be a purely Brownian motion or a model representing memory-less random walk. We thus anticipate that in many situations, a mobile node’s location and velocity

in future will be correlated with its current location and velocity – an observation also made in [7] in connection with PCS systems. Thus, location tracking schemes based on predictive models using current and past location and velocity of a mobile node are worth exploring.

In this paper we take this approach using a method called *dead reckoning*, an idea borrowed from an ancient navigation technique. Here, a mobile node samples its own location continuously or periodically and constructs a model of its movement. The model can be simple or sophisticated, deterministic or stochastic, depending on the predictive ability of the mobile node. The node disseminates its current location as well as this model (together called the *dead reckoning model* or DRM) in the network. Every other node uses this information to track the location of this node. Very little location update cost is incurred if the model’s prediction is accurate. In addition of location tracking the model gives the ability for predicting the mobile’s future trajectory that is useful in many network layer applications such as routing. For example, models for link lifetime can be deduced from the DRM even for remote node pairs. This is useful for constructing more stable routes. In our knowledge dead reckoning has not been used for mobility tracking in ad hoc networks. However, it has long been considered to be a useful technique in other domains, such as distributed interactive simulations (see, for example, [8]).

The rest of the paper is organized as follows. Section II describes the dead-reckoning model that we use to distribute this location information to all nodes in the network. We present an overview of our proposed DRM-based routing protocol in section III. Simulation results are presented in section IV. Related work and conclusions are presented in section V and VI.

II. MOBILITY TRACKING USING DEAD RECKONING

The dead reckoning model or DRM could be a simple first order model giving the node’s velocity (i.e., speed with direction) or could be more complex higher order model involving accelerations (i.e., changes in speed and/or direction). The model could be deterministic or probabilistic. The actual nature of the model depends on the mobile node’s ability to predict its course, by tracking its own movement over time or from inputs from upper layers (e.g., from a path planning application).

Without loss of generality, 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 [9], beacons from an available fixed infrastructure [10], or GPS-less positioning systems such as [11] 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. The location and movement models together form the dead-reckoning model or DRM. Each node maintains a DRM table. Whenever it receives a DRM update from a node it adds or updates an entry for that node's model in its DRM table. The DRM for each in the table has a timestamp denoting when it was last updated. Thus, each node now has a location and movement model for every node in the network. It uses this to predict the location (x_{pred}, y_{pred}) of the nodes at the current time as per the following formula:

$$\begin{aligned} x_{pred} &= x_{mod} + (v_{x_{mod}} * (t_{cur} - t_{mod})) \\ y_{pred} &= y_{mod} + (v_{y_{mod}} * (t_{cur} - t_{mod})), \end{aligned} \quad (1)$$

where, (x_{mod}, y_{mod}) are the (x, y) coordinates and $(v_{x_{mod}}, v_{y_{mod}})$ are the velocity components distributed in the model, t_{mod} = time at which model was computed and t_{cur} = current time. Note that the time the DRM is computed and the time it is inserted in a remote node's DRM table with an update timestamp are different. However, for all practical purposes this difference is small compared to the time-scale of a node's movement, and will not affect the accuracy of the prediction any significantly.

Even 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

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

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.

The threshold essentially determines the allowable error in the location estimates. Small thresholds will increase location

update overhead. Large thresholds may make location estimates too crude to be of any use. The appropriate choice of this threshold is application dependent.

III. ROUTING USING DEAD RECKONING MODEL

In this section we discuss a geographic routing approach as an application of the dead reckoning technique discussed above. The idea is quite straightforward. Since each node is able to track the location of every other node in the network so long as the DRM is available, every node is capable of constructing a topology of the network using the knowledge of the nominal radio range. This will be similar to a link-state database used in a link-state routing protocol that is used to build routing tables using a shortest path algorithm. There is no need for an end-to-end route establishment. We use a more sophisticated version of this basic idea below.

A. Basic Routing Mechanism

Each node computes the topology of the network by computing the current locations (x_{pred}, y_{pred}) of all nodes (as in equation 1) and using the knowledge of the nominal radio range. Two nodes within the nominal radio range are assumed to have a link between them. Next the lifetime of each such link is computed from the knowledge of the DRMs of the nodes on either side of the link.

Assume that (x_i, y_i) and (v_{x_i}, v_{y_i}) are the current (predicted) location and velocity vector respectively for the two nodes ($i = 1, 2$) at the end points of a given link and R is the nominal radio range. Then, from simple geometrical calculations, the lifetime of the link ΔT is given by:

$$\Delta T = \frac{-(ab + cd) + \sqrt{(a^2 + c^2)R^2 - (ad - bc)^2}}{a^2 + c^2}, \quad (2)$$

where,

$$\begin{aligned} a &= v_{x_1} - v_{x_2}, \\ b &= x_1 - x_2, \\ c &= v_{y_1} - v_{y_2}, \\ d &= y_1 - y_2. \end{aligned}$$

This lifetime is used to assign a cost to the link. Instead of using link lifetimes to determine route lifetime and then choosing the most stable route (that may increase route lengths), we design the link cost to have two additive components – one component inversely proportional to the link lifetime and the other proportional to the Euclidean distance of the egress endpoint of the link from the destination node. This way more stable links are favored, but not those that do not go towards the destination. The proportionality constants are tuned for the best overall performance. With this design of link costs, Dijkstra's algorithm is used to choose the next hop on the least cost (shortest) path.

When the DRMs of all nodes are available, the above procedure is very similar to link-state routing as all link costs are available locally. Each node on the route follows the same procedure to compute the next hop, and so on. To save

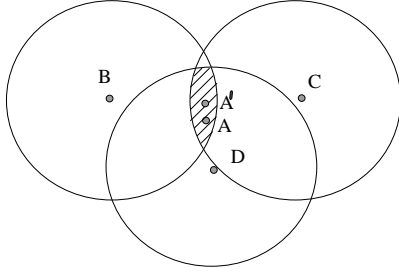


Fig. 1. Node A localizes itself at A' after hearing beacons from B , C and D with their locations. A' is the centroid of the intersection area (shaded). The error AA' is proportional to the square root of the size of the shaded area.

computation, the next hops computed are *cached* until there is any /DRM update.

B. Maintaining Neighborhood Information

Simulation studies revealed that because of the inaccuracies involved in the DRM, or due to the loss of DRM update messages, sometimes the next hop node as computed above may not be an actual neighbor. To avoid this problem, we use periodic beaconing or *hello* messages so that each node can determine its neighbors. Next hop links to non-neighbors are not used to compute routes even when such links are predicted to exist as per the DRM database. The hello messages serve other useful purpose as well. All nodes in the network may not always have access to direct location information. For example, for GPS-based systems, a node may lose the GPS fix temporarily. Or one or more nodes do not carry GPS receivers. However, it is still possible for such nodes to deduce their locations approximately by listening to hello messages of the neighboring nodes, if the hellos contain the location of the originator. For example, if a node hears hellos from n neighbors, it can localize itself in the *centroid* of the circular radio coverage areas of these neighbors. See Figure 1 for an illustration.

Of course, this technique adds errors to location estimates making the dead reckoning models less accurate. The error will typically be proportional to the square root of the intersection area. But it also makes the system fairly robust. In our simulation, we keep the hellos (and count them as overheads), but do not consider situations when only partial location information is available.

C. Imparting Robustness

To make the protocol robust in the face of loss of DRM update packets, we have added a simple query-reply paradigm. If a node gets a packet for a destination for which it does not have a DRM yet, it floods the network with a *DRM request* packet. DRM request always contains the DRM of the originator. The destination node eventually receives this request and responds with *DRM update* unicast back to the

originator. Any intermediate node having a DRM for the destination can also respond and thus quench the flood. We expect that such situations where a node must explicitly make requests for DRM will be rare. Indeed this was the case in our simulations. Explicit requests were seen only in relatively stable networks, where DRM updates were rare. Thus, if the very first DRM update was lost due to collision, e.g., some nodes were left without any DRM for some other nodes for long periods of time. In such cases, the former nodes will resort to DRM requests.

Several other mechanisms are used to impart robustness for data packet forwarding. Packets waiting for DRM update as above are buffered in a *send buffer* (until the buffer overflows). Also, if a next hop is found broken at the time of forwarding, the next best next hop is chosen, up to a maximum of three attempts before the packet is dropped.

D. Optimizations

So far, we have assumed that the DRM updates are flooded in the entire network. This is not too bad when sophisticated and/or accurate models are available, as then floods will be very infrequent. But otherwise, frequent network-wide floods may be a problem. To counter this, several DRM thresholds can be defined. Larger deviations from the DRM are propagated network wide. But smaller deviations are propagated only to a close proximity of the node propagating the update. This is based on the hypothesis that small deviations will not affect routes at large distances; routes will possibly be changing only locally. In our simulation experiments, however, we used a single threshold and network-wide flood. So at best our results underestimate the performance in terms of routing load.

E. Other Approaches

The above routing technique is presented as a case-study illustrating the usefulness of dead reckoning-based mobility tracking. Many other routing approaches are possible. For example, purely geographic routing approaches [2] are perfectly applicable. Note that all geographic routing approaches rely on location information being available (though do not always explicitly mention how they are available) and provide some robustness when information may be stale. Powerful predictive methods can only make them more efficient.

IV. SIMULATION RESULTS

We compare the performance of our geographic routing protocol with that of two popular on-demand routing protocols AODV [12], [13] and DSR [14], [15]. AODV and DSR are chosen as comparison points, as these two protocols are already widely evaluated [16], [17], [18]; recent IETF drafts exist that clearly describe their specifications [12], [15]. Also, simulation codes for these protocols are available in the public domain by the authors of these protocols making the comparisons easy and fair. Both DSR and AODV are source-initiated flood-based protocols, where routes are established as a response to a flooded query initiated by the source. The major

difference between them is that DSR uses source routing and aggressively caches source routes, possibly multiple routes, to reuse later; AODV, on the other hand, is more conservative, and uses a sequence number-based scheme to keep only the most recently learnt route.

We use the popular ns-2 simulator [19] along with the wireless protocol stack model developed at CMU [16]. This model uses the IEEE 802.11 standard [20] as the MAC-layer protocol and uses the first-generation 2 Mbps wavelan radios [21] for the radio model. A free-space propagation model is used. For the given choice of radio parameters the nominal radio range is 250m. MAC-layer feedback is used to know that the link to next hop is broken during data forwarding. Traffic sources are constant bit rate (CBR). The source-destination pairs are spread randomly over the network.

The mobility model uses the *random waypoint* model [16] in a rectangular field. The field configurations are used (i) 2200 m x 600 m field with 100 nodes, and (ii) 3000 m x 1000m field with 250 nodes. In the random waypoint model, each node starts its journey from a random location to a random destination location with a randomly chosen speed (uniformly distributed between 0-20 m/sec). Once the destination is reached, another random destination is targeted after a pause. The pause time is varied that affects the relative speed of the mobile nodes. Note also that a lower pause time indicates higher mobility. In this mobility model, a DRM update can be generated when a node stops for a pause; it can be generated again when the node restarts unless it restarts too soon (before new location samples can be taken).

We simulate 40 CBR sessions and a packet rate of 4 packets/sec. The data packets are 512 bytes in size. The sessions are long-lived. They start towards the beginning of the simulation and stay until the end. Note that given the low capacity of ad hoc networks due to its multihop nature, this presents a relatively high load on the network. Three key performance metrics are evaluated: (i) *packet delivery fraction*: ratio of the packets delivered to the destination to those generated by the CBR sources. (ii) *median delay* of data packets: this includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, propagation and transfer times, and (iii) *normalized routing load*: the number of routing packets “transmitted” per data packet “delivered” at the destination. Each hop-wise transmission of a routing packet is counted as one transmission.

Some initial experiments are conducted to determine a suitable threshold to use in the DRM-based protocol. A threshold of 200m is used. As for the other parameters, an hello interval of 2 sec is used for the DRM-based protocol. Hello packets are counted as a part of the routing load for a fair comparison. The AODV and DSR protocols simulated do not use hellos. Location sampling intervals are also chosen as 2 sec.

The performance results are plotted in Figure 2. As expected, the DRM-based routing presents superior results in both 100 and 250 node networks. Its packet delivery performance is superior to both AODV and DSR (Figure 2(a) and

(b)). Notice the wider performance differential at 0 pause time (continuous mobility). While continuous mobility is a challenging case for traditional routing protocol, performance for DRM-based method actually improves with continuous mobility for random waypoint model because of improved prediction ability. This is because if the pause time is very small, at most one DRM update can be generated for every change of direction, whereas for larger pause two DRM updates will be generated.

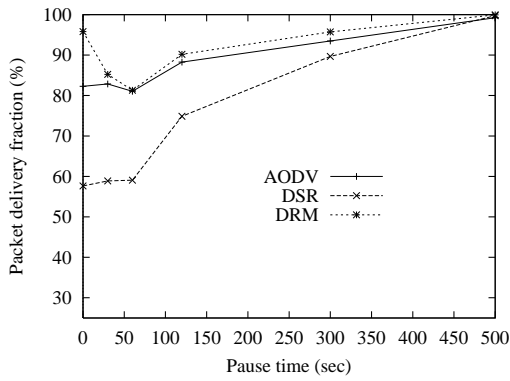
The delays of the three protocols are similar for 100 nodes; but with 250 nodes DRM presents lower delays due to better path selection (less need for buffering etc). See Figure 2(c) and (d). Looking at the subfigures (e) and (f) the normalized routing load is far superior to either DSR and AODV. In fact it is less than half of the best of DSR and AODV for higher mobility. DRM loses the routing load advantages for the static network case (500 sec. pause time). Here DSR and AODV initiates few route discovery floods, typically equal to the number of sources. But in our protocol, every node in the network must still transmit DRM update floods at least once.

Note that DSR packet delivery performance was relatively very poor compared to both AODV and DRM, particularly so for larger network. For large networks all performance metrics are in fact poor for DSR. The DSR protocol version used include fixes developed by us for improving its route caching performance [22] which are similar to those developed by the protocols authors [23]. Without these fixes, the performance was even poorer. The poor performance was traced back to still-remaining stale cache problem, and due to route reply storm, where a large number of nearby nodes reply to a route request increasing congestion hot spots. Similar performance degradation for DSR was also observed when compared with other location-based protocols on large networks [4]. As a side note, both DSR and AODV implementations used here utilize their own flood containment procedures to prevent network-wide flood of route requests that keep the routing load down. On the other hand, the DRM-based protocol floods the entire network; no optimization is used. Thus, the results presented here underestimates the relative performance of the DRM.

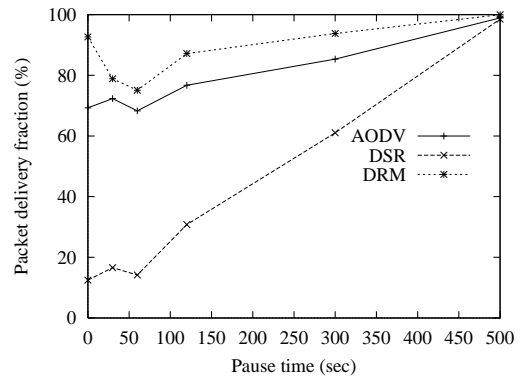
V. RELATED WORK

There is a growing body of literature that use location information for routing in ad hoc networks. Early work started with the LAR and DREAM protocols. LAR [1] limits the route request flood within a zone for protocols like DSR and AODV. The zone is where the destination is most likely to be found. DREAM [5], 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.

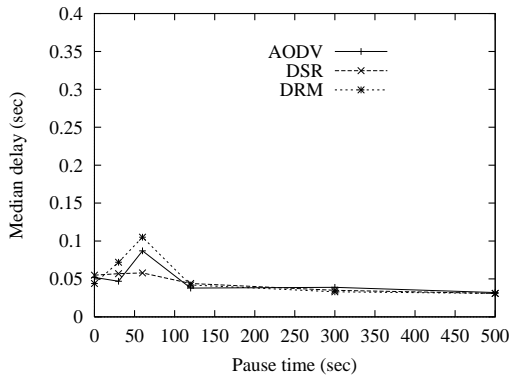
Several techniques were proposed for geographic forwarding where availability of location information is assumed. A very simple forwarding scheme simply forwards the packet to the neighbor closest to the destination [24]. However, some techniques are needed if this greedy method reaches a dead end, i.e., there is no neighbor closer to destination than the current node itself. Karp and Kung in their GPSR technique



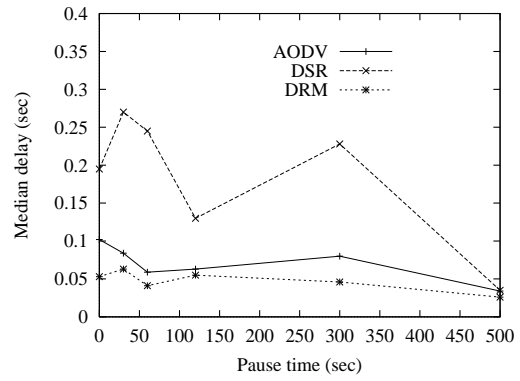
(a) Delivery fraction, 100 nodes



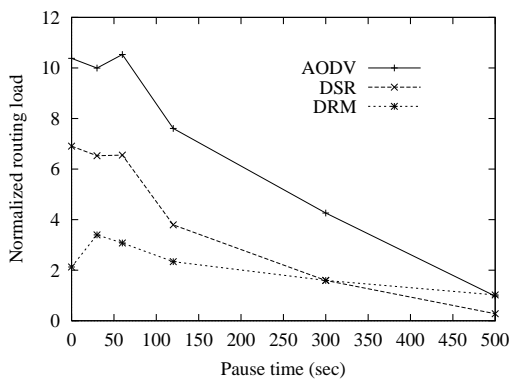
(b) Delivery fraction, 250 nodes



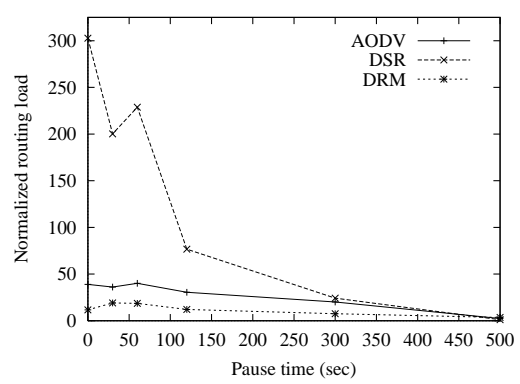
(c) Median delay, 100 nodes



(d) Median delay, 250 nodes



(e) Normalized routing load, 100 nodes



(f) Normalized routing load, 250 nodes

Fig. 2. Performance of DRM-based routing technique relative to AODV and DSR for 100 and 250 node networks with 40 CBR sources.

[2], and Bose et. al. [25] independently, present efficient techniques to recover from such situations.

In [4] authors propose GLS, a location database service, where a mobile node periodically updates a set of location servers with its current location. In GRSS [6] mobile nodes proactively disseminate aggregated forwarding locations as summaries of node IDs in a defined region. Finally Liang and Haas's predictive model-based mobility management technique [7] is very similar to our dead reckoning technique, but has been targeted for cellular PCS networks.

In somewhat similar vein to our routing technique Su [26] have proposed a mobility prediction method that uses the location and mobility information provided by GPS to predict the link expiration time between nodes and perform rerouting when a route breaks. In [3], Woo and Singh present SLURP (Scalable Location Update-based Routing Protocol) based on the concept of Mobile IP.

Though not directly related to mobility tracking, several papers have used models of link or route stability for routing. Early work was done by Toh who proposed the ABR protocol [27], where a notion of associativity or link stability was used for the most stable route selection. More recently Kim, Choi and Toh [28] have used location information to select the most long-lived route among the ones available. They have used this technique as an extension of a source-initiated on-demand routing protocol.

VI. CONCLUSIONS

The strength of the dead reckoning technique lies in its predictive ability, which is somewhat mobility model dependent. However, we have demonstrated its power even with a random memory-less mobility model like random waypoint with where the DRM-based technique delivered superior routing performance compared to popular protocols such as DSR and AODV. We will study other mobility models in our future work as well as extend the DRM technique for other applications or different routing protocols.

ACKNOWLEDGMENT

This work was partially supported by NSF CAREER grant ACI-0096186 and NSF networking research grant ANI-0096264.

REFERENCES

- [1] Y. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in *ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, November 1998, pp. 66–75.
- [2] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *6th International Conference on Mobile Computing and Networking (ACM MOBICOM'00)*, August 2000, pp. 186–194.
- [3] S.-C. M. Woo and S. Singh, "Scalable routing protocol for ad hoc networks," *ACM/Kluwer Wireless Networks (WINET) Journal*, 2001, to appear.
- [4] J. Li, J. Jannotti, D. DeCouto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *6th International Conference on Mobile Computing and Networking (ACM MOBICOM 2000)*, August 2000, pp. 66–84.

- [5] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "Distance routing effect algorithm for mobility (DREAM)," in *4th International Conference on Mobile Computing and Networking (ACM MOBICOM'98)*, October 1998, pp. 66–84.
- [6] P.-H. Hsiao, "Geographical region summary service for geographical routing," *Mobile Computing and Communications Review*, vol. 5, no. 4, 2001.
- [7] B. Liang and Z. Haas, "Predictive distance-based mobility management for pcs networks," in *Proceedings of IEEE Infocom Conference*, 1999, pp. 1377–1384.
- [8] W. Cai, F. B. S. Lee, and L. Chen, "An auto-adaptive dead reckoning algorithm for distributed interactive simulation," in *Proceedings of the 13th Workshop on Parallel and Distributed Simulation (PADS)*, 1999, pp. 82–89.
- [9] J. Caffery and G. Stuber, "Overview of radiolocation in CDMA systems," *IEEE Communications Magazine*, vol. 36, pp. 38–45, April 1998.
- [10] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low cost outdoor localization for very small devices," 2000.
- [11] S. Capkun, M. Hamdi, and J. Hubaux, "GPS-free positioning in mobile ad-hoc networks," in *Proc. Hawaii Int. Conf. on System Sciences*, Jan. 2001.
- [12] C. Perkins, E. Royer, and S. R. Das, "Ad hoc on demand distance vector (AODV) routing," <http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-07.txt>, Nov 2000, iETF Internet Draft (work in progress).
- [13] C. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, Feb 1999, pp. 90–100.
- [14] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile computing*, T. Imielinski and H. Korth, Eds. Kluwer Academic, 1996, ch. 5.
- [15] D. Johnson, D. Maltz, Y.-C. Hu, and J. Jetcheva, "The dynamic source routing protocol for mobile ad hoc networks," <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-05.txt>, March 2001, iETF Internet Draft (work in progress).
- [16] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th International Conference on Mobile Computing and Networking (ACM MOBICOM'98)*, October 1998, pp. 85–97.
- [17] P. Johansson, T. Larsson, N. Hedman, and B. Mielczarek, "Routing protocols for mobile ad-hoc networks - a comparative performance analysis," in *Proceedings of the 5th International Conference on Mobile Computing and Networking (ACM MOBICOM'99)*, August 1999, pp. 195–206.
- [18] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *Proceedings of the IEEE INFOCOM 2000 Conference*, March 2000, pp. 3–12.
- [19] K. Fall and K. V. (Eds.), "ns notes and documentation," 1999, available from <http://www.mash.cs.berkeley.edu/ns/>.
- [20] I.E.E.E., "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE standard 802.11-1997," 1997.
- [21] B. Tuch, "Development of WaveLAN, an ISM band wireless LAN," *AT&T Technical Journal*, vol. 72, no. 4, pp. 27–33, July/Aug 1993.
- [22] M. K. Marina and S. R. Das, "Performance of Route Caching Strategies in Dynamic Source Routing," in *Proceedings of the Int'l Workshop on Wireless Networks and Mobile Computing (WNMC) in conjunction with Int'l Conf. on Distributed Computing Systems (ICDCS)*, Apr 2001, pp. 425–432.
- [23] Y.-C. Hu and D. Johnson, "Caching strategies in On-demand Routing Protocols for Wireless Ad Hoc Networks," in *Proceedings of the IEEE/ACM MOBICOM*, 2000, pp. 231–242.
- [24] G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," ISI, Tech. Rep. ISI/RR-87-180, 1987.
- [25] P. Bose, P. Morin, P. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proceedings of the DIAL-M'99 Workshop*, Aug. 1999.
- [26] W. Su, "Mobility prediction in mobile/wireless networks," Ph.D. dissertation, University of California, Los Angeles, 2000.
- [27] C.-K. Toh, "Associativity-based routing for ad hoc mobile networks," *Wireless Personal Communications*, vol. 4, pp. 103–139, 1997.
- [28] D. Kim, T. Choi, and C. Toh, "Location-aware long-lived route selection in wireless ad hoc network," in *Proceedings of Fall VTC*, Boston, 2000.