

# “Direction” forwarding for highly mobile, large scale ad hoc networks

Yeng-Zhong Lee, Jason Chen, Biao Zhou, Mario Gerla,  
University of California, at Los Angeles  
Computer Science Department  
{yenglee, jvc, zhb, gerla}@cs.ucla.edu

Antonio Caruso  
Institute of Sciences and Technologies information, ISTI  
National Research Council  
Research Area of San Cataldo - Pisa, Italy  
antonio.caruso@isti.cnr.it

**Abstract--** In this paper, we present a novel packet forwarding scheme for wireless ad hoc networks --- “Direction” Forwarding (DFR). Popular routing protocols such as DSDV and AODV use “predecessor” based forwarding, namely, the packet is forwarded to the predecessor on the shortest path from destination, as advertised during the last update. Predecessor forwarding may fail in large scale networks where the routing update rate must be reduced by the need to maintain link O/H below reasonable levels. If nodes are mobile, the routing table entries may become “stale” very rapidly. In other words, the “predecessor” listed in the routing table may have moved away and predecessor based packet forwarding fails!

DFR is designed to overcome the “stale” routing table entry problem. Suppose our ad hoc network is equipped with a geo coordinate system, either global (e.g., GPS) or local (e.g., virtual coordinates locally computed via trilateration). When the routing update arrives, the node remembers not only the predecessor delivering the update, but also the update “direction” of arrival. When a packet must be forwarded to destination, it is first forwarded to the node ID found in the routing table. If the node has moved and ID forwarding fails, the packet is “direction” forwarded to the “most promising” node in the indicated direction. If the network is sufficiently dense, direction forwarding will recover from most “predecessor” ID forwarding failures.

At first glance, DFR seems to combine the features of table based routing and geo-routing. However, direction forwarding differs from geo-routing in that the direction is learned from the routing updates, instead of being computed from the destination coordinates. Thus, DFR does not require destination coordinates, global coordinate system, or Geo Location Server. In the paper we show the application of DFR to a scalable routing scheme, LANMAR. Through simulation experiments we show that DFR substantially enhances LANMAR performance in large, mobile network scenarios.

## I. INTRODUCTION

Ad-hoc wireless networking technology has been gaining increasing visibility and importance in distributed applications that can not rely on a fixed infrastructure but require instant deployment and easy reconfiguration. These networks are typically characterized by limited bandwidth, limited radio range, high mobility and high bit error rate (BER). These characteristics pose challenges to the design and implementation of MANET (Mobile Ad Hoc Net) routing protocols and have motivated extensive research in this area over the past few years.

One of the most challenging research areas that have recently emerged in the design of MANETs is in fact scalability; in particular, scalability of the routing protocols. Conventional proactive routing protocols, such as DSDV[15] and Fisheye[14], rely on periodic exchanges of routing information. They do not scale well because they propagate routing information of all nodes throughout the whole network. With mobility, more frequent updates are required to keep the information up to date, thus producing a large amount of control overhead. In a large scale mobile environment, on-demand routing protocols such as AODV[16] and DSR[10], which generate routing overhead only when there is data traffic to send, and thus have been traditionally considered more suitable for ad hoc wireless networks, also tend to cause heavy overhead due to the large-scale flood search triggered by motion. In the case of 100 nodes and 40 sources with uniform traffic patterns, results have shown that both DSR and AODV generate more routing overhead than actual throughput [5][9].

An important class of MANETs finds applications in disaster recovery, civilian emergencies and battlefield operations. These networks tend to grow large, involving up to thousand nodes. Thus, scalability becomes a critical design issue, especially

---

This work was supported in part by the ONR “MINUTEMAN” project under contract N00014-01-C\_0016, and in part by NSF “WHYNET” project under contract CNS-0335302.

when combined with high node mobility. Recently some efforts have been made to improve the scalability of ad hoc routing protocols. One technique is to utilize geographical information such as in LAR [12] and GPSR[11], which try to utilize geographic information (typically from GPS) to achieve scalability. Such position based routing protocols use physical location information about the participating nodes to make decisions on how to route packets. Traditional techniques such as Link State routing have also been retrofitted for scalability. Link State routing algorithms generally use flooding to distribute network topology information, leading to significant control traffic overhead that reduces the bandwidth available for application data. In order to reduce the control overhead in presence of mobility, recent extension of Link State have explored group mobility patterns such as in the LANMAR scheme[17][18].

Group mobility is common in military and disaster recovery scenarios. For example, during rescue operations, teams of fire-fighters and medical assistants are moving as groups following different patterns. Exploiting such group mobility patterns could greatly simplify routing. In LANMAR routing, a landmark node is elected from each group to represent the whole group. Nodes in the network, but outside a group, do not need to track routes to all nodes in that group. Only one route toward the landmark node needs to be maintained. Such kind of summarized routing will not affect the routing accuracy much since the group nodes are moving together. However, the overhead reduction and thus scalability improvement is significant. For example, if the average group size is  $g$ , the routing table size might be reduced to  $1/g$ . The routing overhead is in turn reduced as the routing packet size is significantly shortened.

To route packets to landmarks, the LANMAR protocol uses “predecessor” based forwarding. Namely, the packet is forwarded to the predecessor on the shortest path from the Landmark, as advertised during the last update. Predecessor forwarding may fail in large scale networks where the routing update rate must be reduced in order to maintain link O/H below reasonable levels. If nodes are highly mobile, the routing table entries may become “stale” very rapidly. In other words, the “predecessor” listed in the routing table may have moved away and no route is found!

In this paper we introduce DFR (Direction Forwarding) to overcome the “stale” next hop problem. Suppose our ad hoc network is equipped with a geo coordinate system, either a global system (e.g., GPS) or a localized system (e.g., virtual coordinates locally computed via trilateration). When the routing update arrives, the node remembers not only the predecessor, but also the “direction” from which the update arrived. When a packet must be forwarded to destination, it is first forwarded to the node ID found in the routing table. If the node has moved and ID forwarding fails, the packet is “direction” forwarded to the “most promising” node in the indicated direction. If the network is sufficiently dense, direction forwarding will recover from most ID forwarding failures.

In DFR, the computation of the direction implies that either the nodes have a directional antenna and are equipped with compass, or there is a local “coordinate” system. The first scheme would work only outdoor because of antenna directionality. Focusing on the second approach, there are many ways to create local position reference systems – many have been published. One way is to place RFIDs in published location and compute one’s relative position with respect to the RFIDs. Another approach is to use GPS (where GPS is available). In our experiments, for simplicity we assume that each node has GPS and it periodically propagates its GPS location to all the direct neighbors.

In the remainder we report on the extension of LANMAR[6] using DFR. The version of DFR used in conjunction with LANMAR is rather sophisticated as it takes advantage of the Link State (Fisheye) local routing procedure of LANMAR. We note however that a basic version of DFR can be retrofitted in any Distance Vector scheme, including On Demand vector schemes such as AODV.

The rest of the paper is organized in the following way. We briefly review some related work in section II and give an introduction of the original LANMAR routing in section III. In section IV, we present the protocol DFR routing scheme in details. Intensive performance evaluations are presented in section V and then we present conclusions and future works in section VI.

## II. RELATED WORK

A considerable body of literature has addressed research on ad hoc routing protocols. The early entries include DSR, DSDV, and AODV. These protocols work well in small or medium size ad hoc networks. However, much larger ad hoc networks emerge in several application scenarios, such as in military or disaster recovery situations. Scalability becomes an important issue in such situations. To improve routing scalability, several techniques have been recently proposed such as OLSR[4], LANMAR[6][17][18], ZRP[7], TBRPF[13], and Fisheye. For example, Fisheye propagates link state packets with different frequencies to nodes inside vs. outside its fisheye scope respectively. OLSR reduces the control packets by selecting

only part of the neighbor nodes for packet forwarding. TBRBF reduces the update overhead in a Link State routing scheme by maintaining and sharing a tree for update broadcast. R-DSDV[3] introduces congestion control to the original DSDV[15] routing, thus limiting control overhead. LANMAR uses a landmark node to represent a group of mobile nodes. ZRP[7] combines both proactive routing (in local zone) and on-demand routing (to remote nodes). The control overhead of on-demand routing protocols can also be reduced by repairing a broken route locally at the node, which experiences the link breakage as is done in WAR[2].

Geographical information assisted routing protocols such as GPSR[11] and TBF[19] try to use geographic information (typically from GPS) to achieve scalability. GPSR makes a greedy forwarding decision using only information about the position of direct neighbor and data packet destination in the network topology. Packets are forwarded to a next hop node which yields the most progress toward the destination. TBF (Trajectory Based Forwarding) embeds a trajectory with route stamped in the packet header and then lets intermediate nodes forward packets to those nodes that lie along the trajectory. It is apparent that none of the above proposed schemes uses Directional Forwarding. The closest to this concept is geo-forwarding as in GPSR. However, Direction Forwarding differs from geo-routing in that the direction is learned from the routing updates, instead of being computed from the destination coordinates. Thus, DFR does not require destination coordinates, nor a global coordinate system, nor a Geo Location Server. Moreover, DFR does not get stuck in dead ends like geo-routing does; thus, it does not require “perimeter re-routing”.

### III. LANMAR ROUTING OVERVIEW

Since in this study LANMAR is the protocol “hosting” DFR, we provide here a brief overview. LANMAR (Landmark Ad Hoc Routing) protocol is a proactive routing protocol. It uses the notion of landmarks to keep track of logical subnets. A logical subnet consists of nodes that have a common interest and move together as a “group”. A representative of the subnet, i.e., a “landmark” node, is dynamically elected in each subnet. LANMAR uses an IP like address consisting of a group ID (or subnet ID) and a host ID, i.e. <GroupID, HostID>. The LANMAR protocol is supported by two complementary, cooperating routing schemes: (a) a local, “myopic” proactive routing scheme operating within a limited scope centered at each node and exchanging route information about nodes up to only a few hops away; and (b) a “long haul” distance vector routing scheme (referred to as LMDV) that propagates the path to each elected landmark into the entire network. As a result, each node maintains two routing tables: local routing table with direct routes to nearby destinations and landmark table with routes to all the landmarks. To relay a packet, a node first queries a route (i.e., next hop ID) to the destination in its local routing table. If the destination is found in the local table, the packet will be directly forwarded to the next hop. Otherwise, the subnet ID of the destination is read from the packet header and the packet will be instead routed towards the corresponding landmark.

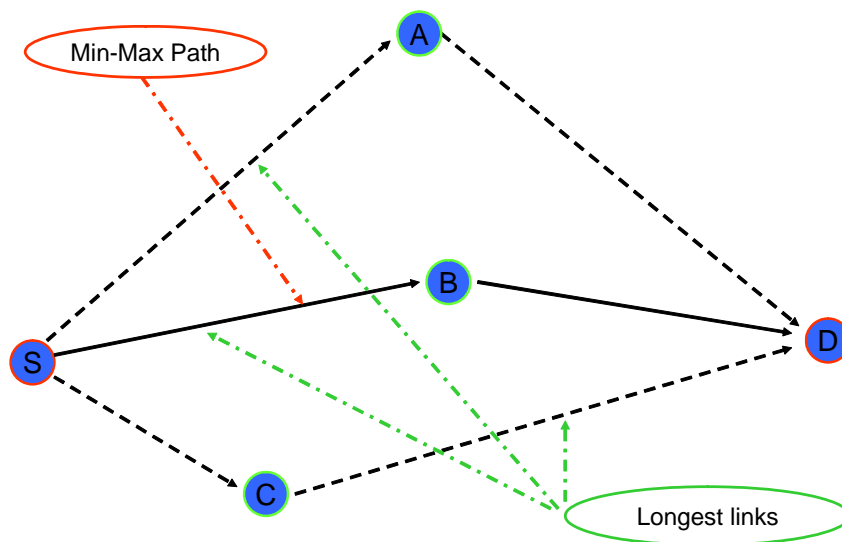


Figure 1. Next-hop selection using an enhanced shortest path algorithm

In our LANMAR implementation, we use Fisheye routing as the local scoped routing protocol. Fisheye routing (FSR)[14] is basically a Link State (LS) routing. It features gradually reduced propagation frequencies of LSs to nodes further and further away. Since we only run the local routing within a few hops (scope limited), we only use one frequency in Fisheye routing. That is each node broadcasts its topology information at most  $K$  hops away. Here,  $K$  is the scope limit of the local routing. Now, with the proactive routing within the local scopes, each node will have accurate routing information to all nodes at most  $K$  hops away from it. In FSR, scope size  $K$  is set to 2 and neighbor's coordinates are piggybacked on the link state update routing packets so that each node is able to obtain up to 2 hop neighbors' coordinates. And in order to reduce update routing packet size, each routing packet only includes its direct neighbor information instead of all neighbor's neighbor information.

FSR Routing table in a scope is evaluated based on the current link state table using a shortest path algorithm, like Dijkstra's algorithm. Each route entry consists of the destination, the next node to the destination and number of hops to the destination. During a route evaluation instance, all the entries are initially removed. New route entries are recorded in the table starting with one-hop neighbors as destination nodes. Then, consider the shortest path algorithm iteration at a given value of number of hops  $h$ , where information from the topology table is utilized to evaluate routes to a destination  $h$  hops away. In LS, the shortest path algorithm selects the first available path with shortest distance to the destination. If an entry for a destination exists in the LS routing table, no further action is taken. In other words, since the destination node in question is already present in the routing table, a potential alternate route isn't considered, since both the routes have an equal number of hops,  $h$  to the destination. However, in enhanced FSR, since a node knows the coordinates of all neighbors up to 2 hops away, our claim is that an alternate route may be a better one in terms of "longevity". We herewith propose an enhancement to the existing shortest path algorithm employed in Link State made possible by local coordinates. Recall that node  $S$  knows all its neighbors' coordinates within its scope. For simplicity, assume that the destination  $D$  is within the scope and in fact it is two hops away from the source. Generally, there are multiple paths between  $S$  and  $D$  with length equal to 2 hops. Which one should we choose? To this end, we note that in each path one link is longer (in geographic terms) than the other. The best path is the path for which the longest link is the minimum over all available paths as shown in Figure 1. The reason for this choice is that in a mobile network long links are the ones most likely to fail first when the length exceeds radio range. The enhanced LS routing scheme selects such a min-max path exploiting the geo-coordinate information. The scheme can be easily generalized to the case in which the destination is several hops away, outside of scope. In this case the alternate path optimization is carried out over the first two hops of the path. Basically we do "two hop" lookout and trace the path two hops at a time. This scheme has shown improved delivery ratios in high mobility scenarios.

#### IV. "DIRECTION" FORWARDING PROTOCOL

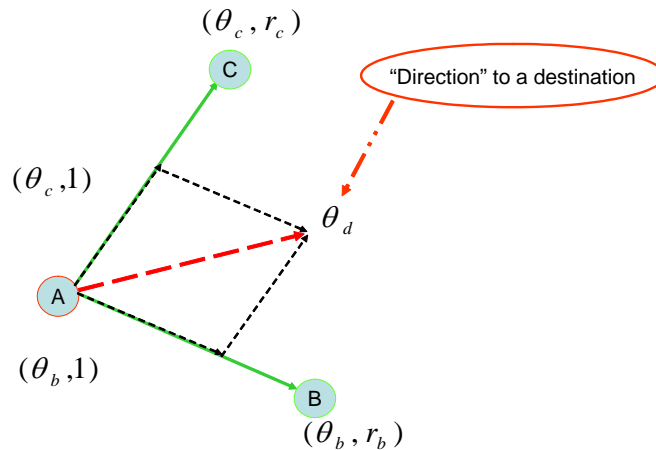


Figure 2. Computation of the "direction"

##### A. Direction forwarding computation

In direction forwarding we apply the Distance Vector concept, and each intermediate node selects the predecessor with minimum hop distance. The main difference: the intermediate node saves not only the ID, but also the "direction" to the predecessor on the path. If there is only one update with minimum hop distance to the destination, the direction to the

predecessor is also the “direction” of the destination.

1) *Computation of the direction to the predecessor.*

The computation of the direction implies that a node knows the coordinates of a predecessor. Once a node receives an update for a destination with min hop distance from a neighbor as a predecessor, the node simply reads the GPS coordinates in the packet header or consults a cache named “Neighbor coordinates cache” to get the coordinates of the predecessor. If GPS is not available, a virtual coordinate system is used. The Neighbor coordinates cache is maintained with information extracted from local routing update packets. The “direction” to the predecessor is computed based on the node’s current coordinates  $(X_1, Y_1)$  and the predecessor coordinates  $(X_2, Y_2)$ . From elementary geometry we get:

$$r = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

$$\theta = \tan^{-1}\left(\frac{Y_2 - Y_1}{X_2 - X_1}\right) \quad (1, 2)$$

In this polar coordinate space the node is at the origin point. The radial coordinate  $r$  is the distance from the node to the predecessor and the polar angle  $\theta$  is the radian from the  $x$ -axis that is used as the direction of the predecessor node.

2) *Update the direction to a destination*

A node may receive more than one distance vector update packet from different “predecessors” with same hop distance and sequence number for the same destination. Thus, we need to aggregate the updates to get an accurate direction of the destination. To illustrate the procedure, in Figure 2, first, node A gets a distance vector update from node B for destination D and computes the forwarding direction  $\theta_b$  by using the equation (1, 2). Later, node A gets another update from node C and computes a new forwarding direction  $\theta_c$ . The two directions can be combined by using the addition of unit vectors  $(\theta_b, 1)$  and  $(\theta_c, 1)$  leading to:

$$\theta_d = \frac{\theta_b + \theta_c}{2}$$

Here, order of addition does not matter since unit vectors are used to combine the “directions”. If a distance vector update packet with a new sequence number or same sequence number but smaller hop distance to the destination is received, the direction of the destination will be reset.

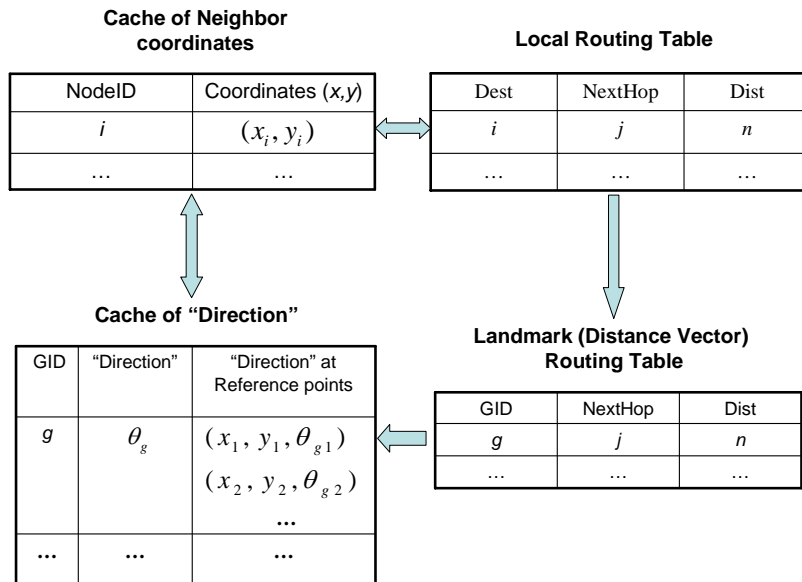
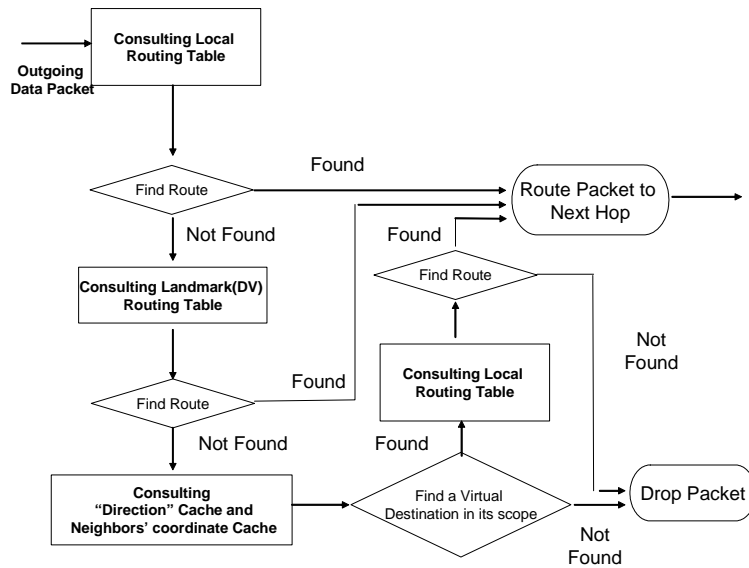


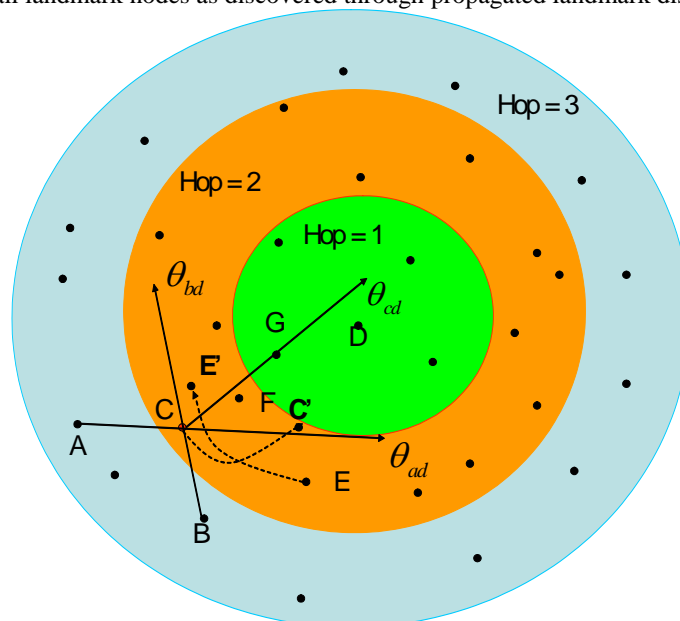
Figure 3. Illustration of the caches and routing tables



**Figure 4. Illustration of the procedure for routing a data packet**

### B. Packet Forwarding Procedure

In this subsection, we describe in detail the routing tables and caches maintained at each node and the procedure of data packet forwarding. After DFR computation is performed, each node will have one neighbor coordinates cache, one “Direction” cache, and two routing tables (local and Landmark). The neighbor coordinates cache keeps the coordinates of all its neighbors within its scope as discovered by the local link state routing protocol. The “Direction” cache keeps the direction to all remote nodes (i.e. Landmarks) computed from DV update and from neighbor coordinates cache. This cache is refreshed and its entries are expired after a pre-specified timeout. The refresh time of these entries is related to the mobility of the neighbors. If the nodes seldom change their position, long refresh time can be used, the “Direction” cache is not so frequently refreshed. The two routing tables - Local proactive and Landmark- provide complete routing information i.e., “direction” to all destination nodes. The local routing table is built by the local scoped proactive routing protocol (e.g. Fisheye in our implementation). It provides the exact routing information to any nodes within the local scope of current node. The landmark distance vector routing table contains routing information to all landmark nodes as discovered through propagated landmark distance vectors.



**Figure 5. "Direction" Forwarding method**

The two caches and two routing tables are illustrated in Figure 3, and the procedure for routing a data packet is illustrated in Figure 4. When a data packet needs to be routed, the node will first consult the local routing table as it provides accurate and up-to-date routing information to nearby nodes. If a route is found and the next hop is along the “direction” to the packet destination, the packet is routed directly. If the destination node is a remote node not in the local routing table, the node then checks the DV routing table for an available next-hop for the destination node. If a next-hop exists, then the packet is routed towards the landmark node of that group via the landmark routing table. Once the packet reaches the remote group, it will then be forwarded via the local routing table either by that landmark node or any other node in the same group. If there is no next-hop for the destination (e.g. the link to the next-hop is broken), then the “direction” cache is used for forwarding the packet.

Most geo-routing protocols select the most forward node within radius (MFR) closest to the packet’s destination as a next hop. In DFR, the selection of the most forward node along the “direction” to the destination may lead to a loop. Consider the situation in Figure 5. Suppose node A has data packets destined for node D. Now, the predecessor of node A, C has moved to C’; note that C was also the predecessor of node B. Node A finds that node B is the most forward node according to DFR direction, and transmits to B. Once the data packet is received at node B, node B computes the “directional” next hop to D and sends the data packets back to node A. Thus, a loop is formed between node A and node B. To avoid such loops, we have modified our scheme for finding the next “direction” hop by using a “virtual destination” method. Basically, we do “look ahead” i.e., we look at the direction computed by the predecessor in the previous hop. When Node A gets a distance vector routing update from a predecessor, say node C, node A consults the Cache of Neighbor Coordinates to get node C’s current location and saves the location as a reference point. Also, node A saves the “previous hop” direction to destination D that was computed at the predecessor node C (the information is piggybacked in the distance vector update packet from C). In Figure 5, that would be the direction from C to G. When a next-hop to destination D is needed at node A, first, node A finds a 2-hop neighbor as a “virtual destination” which is the most forward node originating from the reference point (i.e. location of C) in the direction from C to D. Then, node A uses the virtual destination to find a next-hop to the destination by consulting its local routing table. If more than one reference point exist (i.e., other predecessors besides C), node A computes a virtual destination for each reference point. Then node A selects an optimal virtual destination that is the most forward node within radius among all computed virtual destinations. Back to the example in Figure 5, the virtual destination from A is node G, and the next hop is node E’. Directional forwarding will fail when there is no neighbor left in the desired direction. This occurrence however is unlikely if the network is reasonably dense and a node has at least six neighbors on average.

## V. PERFORMANCE EVALUATION

### A. Simulation Environment

We use QualNet® simulator [1], a packet level simulator to evaluate the proposed DFR routing scheme. The main purpose is to verify its scalability and flexibility in various scenarios. In our simulations, standard IEEE 802.11 radios are adopted with a channel rate of 2Mbps and transmission range of 367 meters. Randomly generated UDP based Constant Bit Rate (CBR) traffic is used for evaluation. The routing protocol selected for comparison is the original LANMAR routing, which has been proved to deliver outstanding performance in large scalable MANETS [17] [18]

### B. Group Mobility Model

To explore group mobility, we first need to have an abstract model to represent such mobility behaviors shown in Figure 6. We assume that group affiliation is static and each node knows in which group it is. For the group mobility pattern, we view the nodes in the same group share a same group mobility vector. This mobility vector decides the moving behavior of the group as a whole. Moreover, nodes within the group will have their internal random moves. We view the internal moves as random moves under the constraint of the group boundary.

The mobility of the group as a whole is also following the random waypoint mobility model. Since the dimensions of the group are fixed, we use the central point of the group area as the reference point of the current position of the group. Then each group chooses its next destination for the groups central point randomly following the random waypoint mobility model. To guarantee that the whole group area is within the boundary of the simulation field, we keep the selected positions of the group central pointer adequately distant from the edges of the simulation field.

So, in our experiments each node in a group has two components in its mobility vector - the individual component and the group component. A node randomly picks a destination within the group and moves towards that destination at a fixed speed. Once the node reaches the waypoint, it pauses 10 second, and then selects another destination randomly and moves towards it. The group component of mobility is also based on the mobility model as the individual component. Group field size is set to 750m by 750 m and all groups are deployed in a 2250m by 2250m network field. Mobility speeds used in this study are 0, 4, 8,

12, and 16m/s and mobile nodes are equally distributed in 9 groups. We use a short pause time of 10 seconds to ensure frequent topology changes.

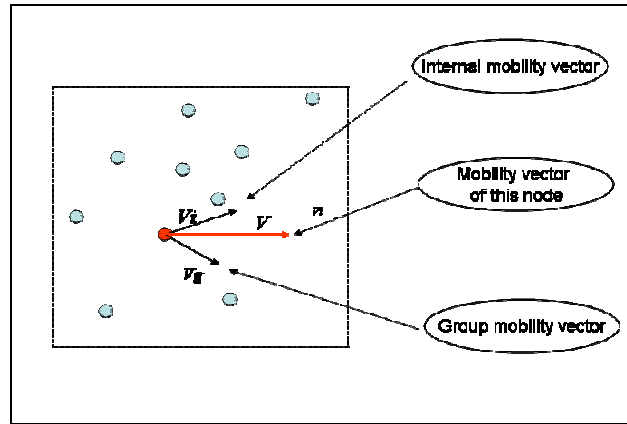


Figure 6: Modeling of group mobility

### C. Traffic pattern

The Constant Bit Rate (CBR) traffic pairs are spread randomly over the network. The number of source-destination pairs is varied in the simulations from 20 pairs to 80 pairs thus varying the offered traffic load. The size of data payload is 512 bytes and the inter-arrival time of the data packets on each source/destination connection is 1 second to model an interactive environment.

### D. Metrics

Metrics selected include 1) Data packet delivery ratio: the ratio between the number of data packets received and those originated by the sources; 2) Average end-to-end packet delay: the average time from when the sources generate the data packets to when they reach the destination nodes; 3) Routing overhead: the total routing control overhead in bits/sec; and 4) Aggregated throughput: the aggregate of the throughput of all CBR connections, computed at each destination node.

### E. Performance as a function of speed

In this set of experiments, we investigate the scalability of the DFR scheme under various network sizes and motion speeds. We fix the number of CBR pairs as 40 per experiment. We run simulations with two different network sizes (225 nodes and 360 nodes) to see the impact of size on performance. For the larger network size, we keep the network field size as 2250 by 2250 meters. The simulation results are presented from Figure 7 to Figure 11.

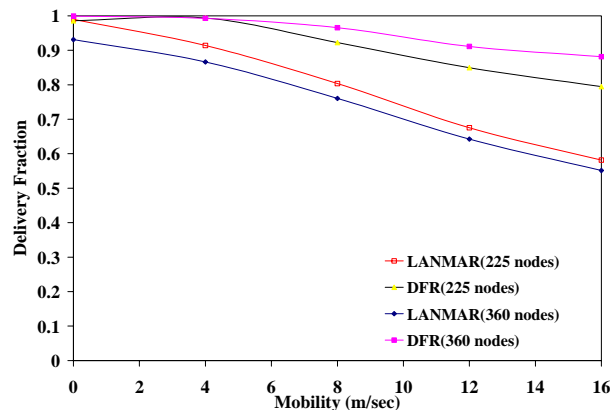


Figure 7: Delivery ratio vs. speed (including packets lost due to disconnected destination)



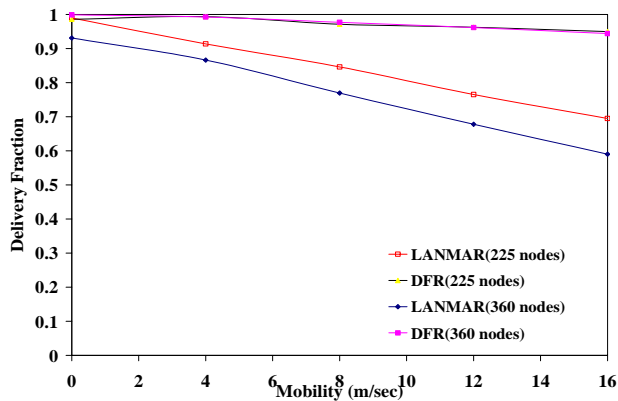


Figure 8: Delivery ratio vs. speed (excluding packets lost due to disconnected destination)

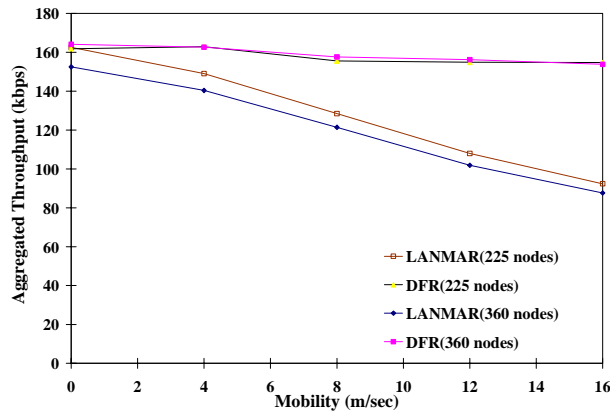


Figure 9: Aggregated Throughput vs. speed

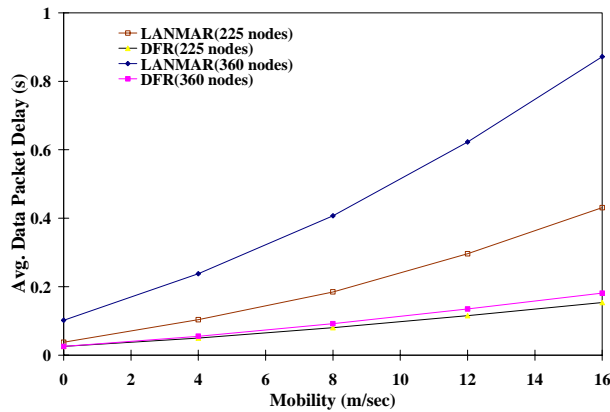


Figure 10: Average data packet delay vs. speed

First, let us compare Figure 7 and Figure 8 results. In Figure 7, there is no distinction between packet delivery failure due to routing failure and topology disconnection. The latter clearly is not the responsibility of routing. We have run another experiment in which for each packet dropped by the simulator, we have checked whether the destination was connected or not, and have not counted the drop in the latter case. The results in Figure 8 show, when the network disconnected cases are discounted, that DFR is above 95% delivery even for 15 m/sec speeds. This is an enormous improvement with respect to the conventional LANMAR that delivers only 60% in a 225 node topology! The aggregate throughput results confirm these findings

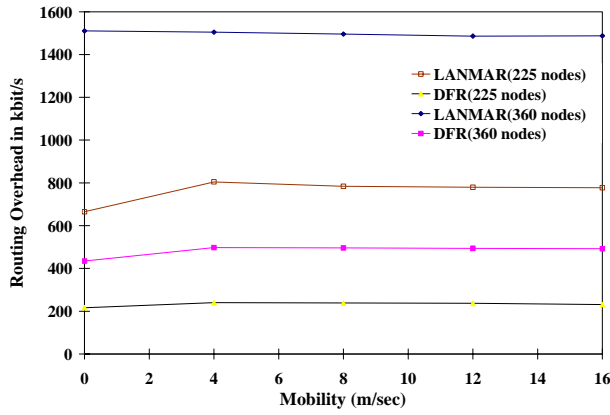


Figure 11: Routing overhead (per node) vs. speed

#### F. Performance under Varying Loads

In this set of experiments, we further investigate the scalability of the proposed DFR routing, 225 mobile nodes are deployed in a 2250mX2250m field. Nodes within each group are also uniformly distributed with their group area. Mobility speed is fixed as 8 meters per second. We vary the traffic load (e.g. number of CBR pairs) to investigate the performance of DFR routing as well as original LANMAR routing. The simulation results are presented in Figure 12 to Figure 14.

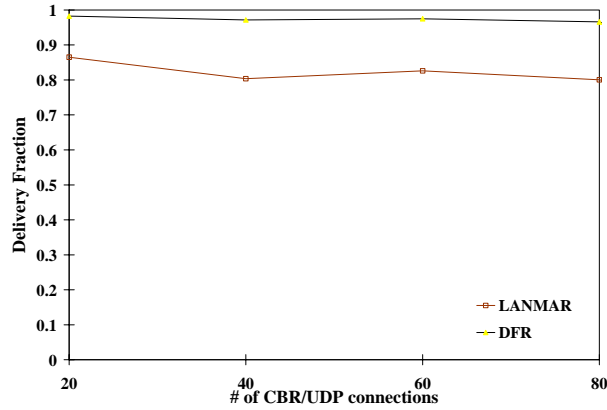


Figure 12: Delivery ratio vs. traffic load

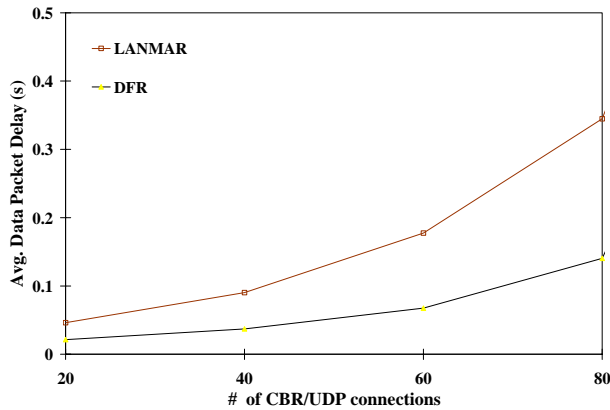
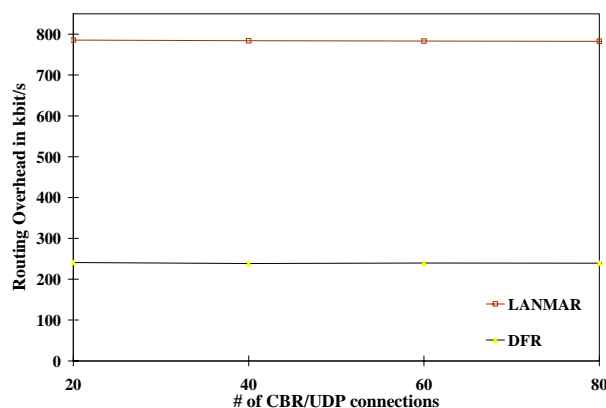


Figure 13: Average data packet delay vs. traffic load



**Figure 14: Average routing overhead on each node vs. traffic load**

The higher efficiency of DFR+LANMAR shows also in the variable load experiments. From Figure 12, we observe that DFR+LANMAR routing always shows better packet delivery ration no matter the traffic load. Remarkable is the lower delay due to lower overhead (see Figure 13); and the much lower overhead (see Figure 14)

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we have identified a vulnerability to high motion in the conventional “predecessor forwarding” scheme used in virtually all popular routing schemes, such as Distance Vector, LANMAR and AODV. We have proposed a more robust forwarding scheme called DFR (Direction Forwarding), which is based on direction of arrival (of the updates) rather than on predecessor ID. We have tested the new scheme on LANMAR and have reported impressive performance improvements. The delivery ratio jumps from 60% to 95% in a moderately dense network at 15 m/s.

In future work we plan to incorporate the DFR scheme in other routing protocols (e.g., AODV, ODMRP, DSDV, etc). We also plan to explore local coordinate techniques (instead of relying on GPS), starting from published techniques, targeting methods that do not depend on GPS and that can work also indoor. One option is to disseminate RFIDs in “published” location and compute one’s relative position with respect to the RFIDs. Another way is to use TOA data as well as trilateration.

## References

- [1] Qualnet simulator. Available at <http://www.qualnet.com>.
- [2] I. D. Aron and S. K. S. Gupta. A witness aided routing protocol for mobile ad hoc networks with unidirectional links. Proceedings of the First International Conference on Mobile Data Access, MDA99, Dec. 1999.
- [3] A. Boukerche, S. K. Das, and A. Fabbri. Analysis of randomized congestion control with dsdv routing in ad hoc wireless networks. Journal of Parallel and Distributed Computing (JPDC), 61, 2001.
- [4] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). RFC 3626, Oct. 2003.
- [5] S. R. Das, C. E. Perkins, and E. M. Royer. Performance comparison of two on-demand routing protocols. Proceedings of IEEE INFOCOM’00, Mar. 2000.
- [6] M. Gerla, X. Hong, and G. Pei. Landmark routing for large ad hoc wireless networks. Proceeding of IEEE GLOBECOM 2000, Nov. 2000.
- [7] Z. Haas and M. Pearlman. The zone routing protocol (ZRP) for ad hoc networks. Internet Draft draft-ietf-manet-zonezrp-02.txt, June 1999.
- [8] X. Hong and M. Gerla. Dynamic group discovery and routing in ad hoc networks. Proceedings of the First Annual Mediterranean Ad Hoc Networking Workshop (Med-hoc-Net 2002), Sept. 2002.
- [9] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen. Scalable routing strategies for ad-hoc wireless networks. IEEE Journal on Selected Areas in Communications (JSAC), 17(8), Aug. 1999.
- [10] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. Mobile Computing, edited by T. Imielinski and H. Korth, Chapter 5, 1996.
- [11] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. Proceedings of ACM Mobi-Com’00, Aug. 2000.
- [12] Y. B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. Proceedings of ACM/IEEE MOBICOM98, Oct. 1998.
- [13] R. Ogier, F. Templin, and M. Lewis. Topology dissemination based on reverse-path forwarding (TBRPF). Internet Draft draft-ietf-manet-tbrpf-11.txt, Oct. 2003.
- [14] G. Pei, M. Gerla, and T. W. Chen. Fisheye state routing in mobile ad hoc networks. Proceeding of ICDCS 2000 workshops, Apr. 2000.
- [15] C. Perkins and P. Bhagwat. Highly dynamic destinationsequenced distance-vector routing (DSDV) for mobile computers. Proceeding of the ACM SIGCOMM’94, Sep. 1994.
- [16] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. Proceedings of IEEE WMCSA’99, Feb. 1999.
- [17] G. Pei, M. Gerla and X. Hong, “LANMAR: Landmark Routing for Large Scale Wireless Ad-Hoc Networks with Group Mobility,” in Proceedings of IEEE/ACM MobiHOC 2000, Boston, MA, Aug. 2000, pp. 11-18.
- [18] X. Hong, M. Gerla, Y. Yi, K. Xu, and T. Kwon, “Scalable Ad Hoc Routing in Large, Dense Wireless Networks Using Clustering and Landmarks,” in Proceedings of ICC 2002, New York City, New York, April 2002.
- [19] D. Niculescu and B. Nath, “Trajectory Based Forwarding and Its Applications,” in Proceedings of Mobicom 2003, San Diego, CA, Sept 2003.