“DIRECTION” FORWARDING FOR HIGHLY MOBILE, LARGE SCALE AD HOC NETWORKS

Mario Gerla[1], Yeng-Zhong Lee[1], Biao Zhou[1], Jason Chen[1], Antonio Caruso[2]
[1]University of California, at Los Angeles Computer Science Department; [2]Institute of Sciences and Technologies Information, ISTL, National Research Council, Reasearch Area of San Catalado – Pisa, Italy

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 the 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. However, if nodes are mobile, routing table entries may become “stale” very rapidly. DFR is designed to overcome the “stale” routing table entry problem. 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. 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, a global coordinate system, or a 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.

Key words: Wireless Networks; MANET; routing; Geographic Routing Protocol; Location System; mobility; scalability

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.

1. 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 [11] and Fisheye [10], rely on periodic exchanges of routing information. They do not scale well because they propagate routing information of all nodes throughout the entire 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 [12] and DSR [6], 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 [3][5].

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 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 [8] and GPSR [7], 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 [13][14].

In this paper we introduce DFR (Direction Forwarding) [15] 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 the remainder we report on the extension of LANMAR [4] 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. In section 2, we present the protocol DFR routing scheme in details. Intensive performance evaluations are presented in section 3 and then we present conclusions in section 4.

2. “DIRECTION” FORWARDING PROTOCOL

2.1 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.

2.1.1 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:

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

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

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 θ is the radian from the x-axis that is used as the direction of the predecessor node.

2.1.2 Update the direction to a destination

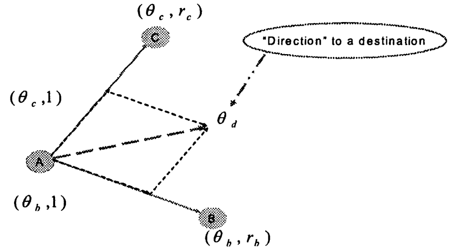


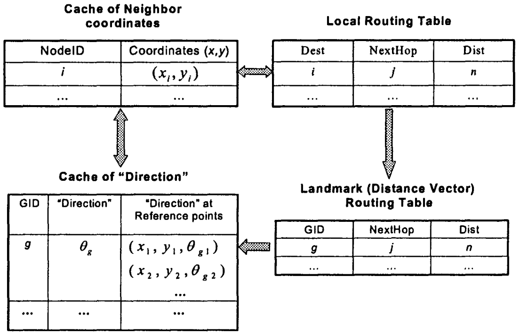
Figure 1. Computation of the "direction"

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 1, first, node A gets a distance vector update from node B for destination D and computes the forwarding direction θ_b by using the equation (1, 2). Later, node A gets another update from node C and computes a new forwarding direction θ_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.

2.2 Packet Forwarding Procedure



Cache of Neighbor coordinates

NodeID	Coordinates (*x*,*y*)
i	(x_i, y_i)
...	...

Local Routing Table

Dest	NextHop	Dist
i	j	n
...	...	...

Cache of "Direction"

GID	"Direction"	"Direction" at Reference points
g	θ_g	(x_1, y_1, θ_{g1}) (x_2, y_2, θ_{g2}) ...
...	...	...

Landmark (Distance Vector) Routing Table

GID	NextHop	Dist
g	j	n
...	...	...

Figure 2. Illustration of the cache and routing tables

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.

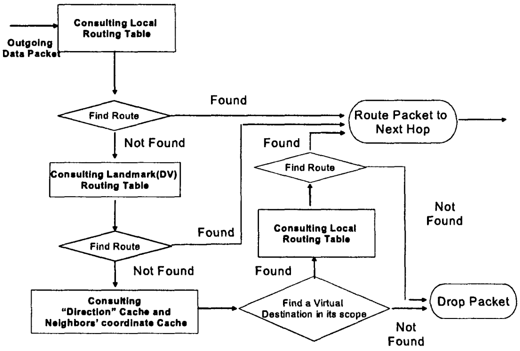

Outgoing Data Packet
Consulting Local Routing Table
Find Route
Found
Not Found
Consulting Landmark(DV) Routing Table
Find Route
Found
Not Found
Consulting "Direction" Cache and Neighbors' coordinate Cache
Find a Virtual Destination in its scope
Found
Consulting Local Routing Table
Find Route
Found
Route Packet to Next Hop
Not Found
Drop Packet
Not Found

Figure 3. Illustration of the procedure to route data packets

The two caches and two routing tables are illustrated in Figure 2, and the procedure for routing a data packet is illustrated in Figure 3. 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, then the "direction" cache is used for forwarding the packet.

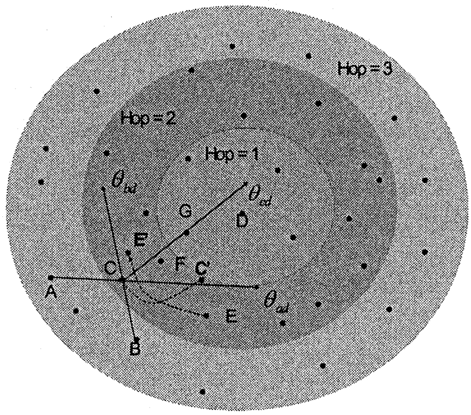


Figure 4. "Direction" Forwarding method

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 4. 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 4 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 4, 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.

3. PERFORMANCE EVALUATION

3.1 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 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. The routing protocol selected for comparison is the original LANMAR routing, which has been proved to deliver outstanding performance in large scalable MANETS [13][14].

3.2 Comparison with Original LANMAR

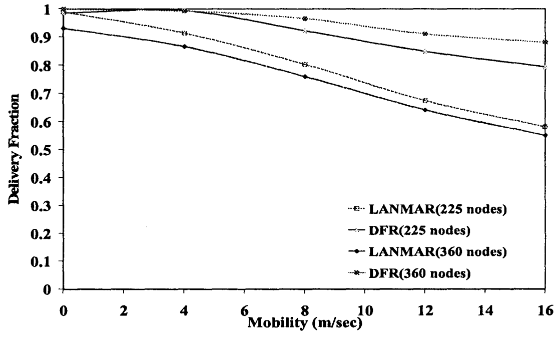


Figure 5. Delivery ratio vs. speed (including packets lost due to disconnected destination)

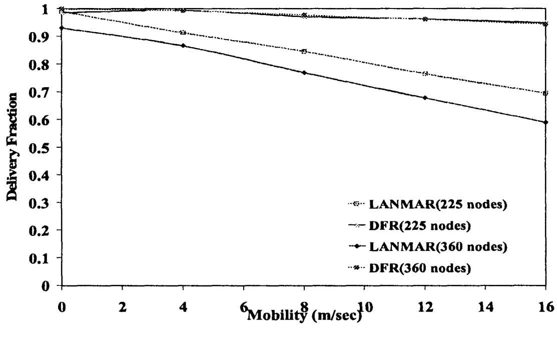


Figure 6. Delivery ratio vs. speed (excluding packets lost due to disconnected destination)

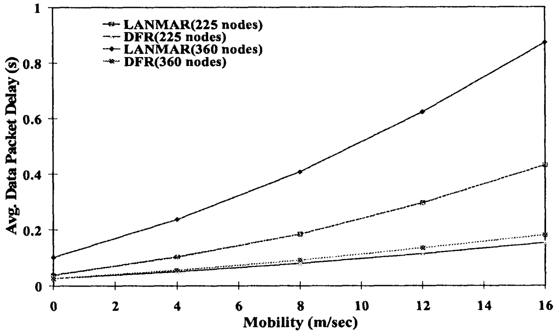


Figure 7. Average data packet delay vs. speed

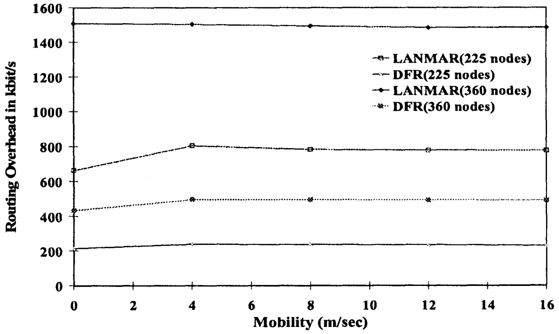


Figure 8. Routing overhead(per node) vs. speed

In this set of experiments, we investigate the scalability of the DFR scheme under various network sizes and movement 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 5 to Figure 8.

First, let us compare Figure 5 and Figure 6. In Figure 5, 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 5 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! Remarkable is the lower delay (see Figure 7) due to lower overhead (see Figure 8).

4. CONCLUSIONS

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.

REFERENCES

[1] Qualnet simulator. Available at http://www.qualnet.com.

[2] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). RFC 3626, Oct. 2003.

[3] 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.

[4] M. Gerla, X. Hong, and G. Pei. Landmark routing for large ad hoc wireless networks. Proceeding of IEEE GLOBECOM 2000, Nov. 2000.

[5] 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.

[6] 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.

[7] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. Proceedings of ACM Mobi-Com'00, Aug. 2000.

[8] Y. B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. Proceedings of ACM/IEEE MOBICOM98, Oct. 1998.

[9] R. Ogier, F. Templin, and M. Lewis. Topology dissemination based on reverse-path forwarding (TBRPF). Internet Draf draft-ietf-manet-tbrpf-11.txt, Oct. 2003.

[10] G. Pei, M. Gerla, and T. W. Chen. Fisheye state routing in mobile ad hoc networks. Proceeding of ICDCS 2000 workshops, Apr. 2000.

[11] C. Perkins and P. Bhagwat. Highly dynamic destinationsequenced distance-vector routing (DSDV) for mobile computers. Proceeding of the ACM SIGCOMM'94, Sep. 1994.

[12] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. Proceedings of IEEE WMCSA'99, Feb. 1999.

[13] 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"00, Aug. 2000.

[14] 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, April 2002.

[15] Y Lee, J. Chen, B. Zhou, M. Gerla, & Antonio Caruso. "'Direction' forwarding for highly mobile, large scale ad hoc networks" (full version), Technical Report TR050026, CSD, UCLA, 05.