An Efficient Table Driven Routing Algorithm for Wireless Ad Hoc Networks

Woo-Jin Choi, Sirin Tekinay
New Jersey Center for Wireless Telecommunications
New Jersey Institute of Technology
Newark, NJ, USA 07102-1982
Email: wxc7456@njit.edu, tekinay@adm.njit.edu

Abstract— **In this paper we propose a table-driven routing algorithm for ad hoc networks using geolocation information of mobiles. Assuming that each mobile is aware of its mobility status** [1] **(i.e., position, speed and direction of movement), mobiles may exchange this information for the purpose of routing. A high speed mobile on a highly directed path doesn't need to update its mobility status as long as it maintains its speed and direction. Each mobile can accurately estimate the geolocation of others using the speed and direction information. On the other hand, a slow-moving mobile doing random walk doesn't need to update its geolocation as long as it stays in a small local area. Based on this observation, we classify mobiles into Local and Global, and let the mobile choose the geolocation update scheme that minimizes the frequency of update for the corresponding mobility class. Each mobile may estimate the network topology using the geolocation information of other mobiles, but the geolocation information alone cannot provide accurate view of the network because two adjacent mobiles may not be able to establish communication due to the radio communication impairment such as fading, jamming or interference from other radio sources. The proposed routing algorithm dynamically adapts to the physical layer impairment and provide loop-free minimum hop route.**

I. INTRODUCTION

The most distinctive characteristic of ad hoc network is that the network topology changes dynamically, and routing paths need to be updated very frequently. There has been significant effort to address this complexity, and many researchers have proposed cost effective dynamic routing algorithms. In the table-driven routing algorithms [1][2][3], each node maintains a routing table that contains routing information to every other node, and data packets are relayed from node to node using the distributed routing tables. These routing tables need to be updated often enough to maintain accurate view of the network topology. On the other hand, the on-demand routing algorithms initiate route discovery when there is need to establish communication between mobile nodes; therefore, the route search delay of the on-demand routing algorithm is usually greater than that of the table-driven routing algorithm. The Dynamic Source Routing (DSR) algorithm [4] is an on-demand routing algorithm in which a source node initiates route search by flooding control packets. Since there is no need to maintain up-to-date routing information, the control overhead is smaller than that of the table-driven routing algorithm. Ad Hoc On-Demand Distance Vector routing algorithm (AODV) [5] is an on-demand routing algorithm based on Destination-Sequenced Distance-Vector routing algorithm (DSDV) [1]. AODV is a hybrid of the on-demand and table-driven routing algorithms. It minimizes the number of broadcasts for the routing table update by creating routes on demand. The Dynamic Source Routing algorithm using Global Positioning System (DSR-GPS) is proposed in [6]. In DSR-GPS, packets are delivered using the source route that is determined from the geolocation information of mobiles in the network. Two mobile nodes whose geographical distance is smaller than the predetermined radio range are assumed to have radio communication link between them. In practice, the geographical distance doesn't guarantee the radio connectivity, and if the route based on geolocation information doesn't exist due to the radio communication impairment, DSR-GPS needs to employ other route search mechanism.

In this paper, we introduce a robust table-driven routing algorithm based on a bandwidth efficient geolocation update scheme. Assuming that mobiles are aware of their own mobility status (i.e., position, speed and direction), they exchange this information and maintain Mobility Status Table (MST) for the purpose of routing. Since each mobile knows where the other mobiles were and where they were headed, they can estimate the geolocation of others at any time. No matter how far, how long, and (or) how fast a mobile travels in the network, the mobile doesn't need to update its geolocation as long as it maintains its speed and direction. Let's consider a one-dimensional Brownian motion [9] for example. At each time unit Δt, a particle takes a step of size Δx to the forward direction with probability p and to the reverse direction with probability q, and it stays at the current position with probability $1-p-q$. Assuming that Δt and Δx are small, and the particle starts moving from position x=0 at time t=0, the probability density for the displacement of the particle at time t becomes

$$p_{X(t)}(x,t) = \frac{1}{\sqrt{\pi f t}} e^{-(x-vt)^2/ft}$$

where $v = (p-q)\Delta x/\Delta t$ is the drift velocity and $f = 2((1-p)p + (1-q)q + 2pq)(\Delta x)^2/\Delta t$ is the diffusion coefficient. The diffusion coefficient represents the rate of increase in location uncertainty. In the conventional routing algorithms, high velocity means high mobility which

---

[1] We assume that the mobiles are aware of their mobility status using GPS [7] or other self-location techniques [8].

0-7803-7005-8/01/$10.00 © 2001 IEEE

requires large control overhead; however, a mobile on a highly directed path has a small diffusion coefficient, and the frequency of the geolocation update can be made small even if the drift velocity is quite large. Consider a commercial airliner. It has a small diffusion coefficient because it usually maintains its speed and direction. If the airliner travels on the designated air route, we can accurately estimate the position of the airliner at any time if we know the departure time, speed and air route. On the other hand, a slowly meandering mobile such as pedestrian walking in the street doesn't need to update its geolocation as long as it stays in a small local area where the position estimation error is bounded to a constant. The speed and direction are efficient and valuable information for position estimation of the mobiles that travel long distance without changing their course; however, for a slow-moving mobile doing random walk, the speed and direction are not efficient means to advertise the geolocation information. Based on this observation, we dynamically classify mobiles into *Local* and *Global* mobility classes, and let the mobile choose the geolocation updating scheme that is designed to minimize the update frequency for the corresponding mobility class. Consider the following scenario that exemplifies the mobility classification. A local bus making stops when there are people waiting at local bus stations has diffusion coefficient of $0.44Km^2/min$ assuming that the bus stations are 1Km apart, the speed of the bus is 60Km/hr and the probability that people are waiting at a local bus station is 1/3 (i.e., $p = 2/3$, $q = 0$). An express bus traveling at the same speed doesn't make stops at the local bus stations, and the diffusion coefficient of the express bus is zero because $p = 1$ at the local bus station. Now, how do we know whether a bus is express or local if there is no sign hanging on the window? We can observe the stop pattern of the bus and make a probabilistic decision. If the bus didn't make stops at N consecutive local bus stations, then we say the bus is express with a probability of error p_{local}^N. We may also look at the diffusion coefficient and say the bus is express if the diffusion coefficient is smaller than a predetermined threshold. In general, the mobiles in the network have diverse mobility patterns, and the mobility classification needs be optimized such that the geolocation update frequency of each individual mobile is minimized.

Each mobile in the network maintains MST, and they estimate the topology of the network for the purpose of routing; however, the actual network topology could be different from the estimated topology because MST is independent of the physical link state of the network. This independence reduces control overhead, but a distributed routing algorithm without the knowledge of actual link state may cause routing loops resulting inefficient routes. We let the data packets carry Negative Link State (NLS) information (i.e., the IDs of inactive radio links between two mobiles whose geographical distance is smaller than the predetermined radio range) so that the data packets can inform the network about the inactive radio links. The proposed routing algorithm is robust and loop-free.

The rest of this paper is organized as follows. In section 2, we introduce a bandwidth efficient geolocation update scheme. In section 3, we introduce a robust routing algorithm using MST. In section 4, we present simulation study. Finally, Section 5 concludes the paper.

II. MOBILITY CLASSES AND GEOLOCATION UPDATING

A mobile on a highly directed path such as aircraft or vehicle on highway doesn't need to update its mobility status pertaining to geolocation estimation as long as it maintains its speed and direction. We may classify such mobile as *Global.* A meandering pedestrian walking in the street doesn't need to update its geolocation as long as he/she remains in a small local area. We may classify such mobile as *Local.* We prescribe the geolocation update schemes for *Local* and *Global* mobility classes.

A. Geolocation Update Scheme A (for Local mobile)

Consider a trajectory of a *Local* mobile that changes the speed and direction frequently. If the mobile updates its speed and direction whenever they are changed, the mobile generates a lot of control messages even if it stays in the same local area. We let the *Local* mobile update its geolocation using its position coordinates when the distance between the current position and the position where the mobile performed the latest geolocation update becomes R. Therefore, R is the maximum position error for a *Local* mobile.

B. Geolocation Update Scheme B (for Global mobile)

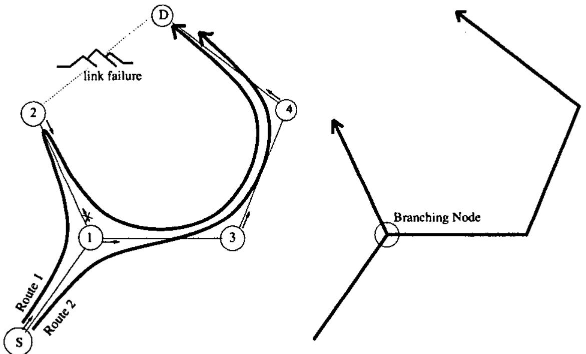
Consider the trajectory of a *Global* mobile. We let the *Global* mobile update its geolocation using its mobility status information (i.e., position, speed and direction). Assume that the mobility status of the *Global* mobile was updated at time t_0. If the mobile changes its speed and direction at time $t_1 (> t_0)$, the distance between the true position and the *nominal position* (i.e., the position estimated by the network using the latest mobility status update) begins to diverge. Since the position estimation is based on the latest mobility status update, the mobile is also aware of its own *nominal position.* The mobile updates its mobility status when this distance becomes a predetermined value R. Therefore, the maximum position error for the *Global* mobile is R.

C. Dynamic Mobility Classification Rule

We may classify a mobile as *Global* if the trajectory of the mobile is "predictable". Otherwise, we may classify the mobile as *Local.* Since the goal of the mobility classification is to reduce the geolocation update frequency, we classify mobile based on its geolocation update frequency.

Proposition 1: A mobile is classified as *Global* if the geolocation update frequency for scheme A is greater than the geolocation update frequency for scheme B. Otherwise, the mobile is classified as *Local.*

Since the decision to update one's geolocation is solely based on the trajectory of the mobile (independent of the network topology), the mobile can calculate the update frequencies of scheme A and B simultaneously. Each mobile



(a) (b)

Fig. 1. Route Discovery

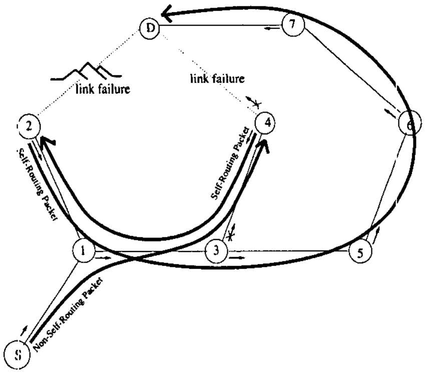


Fig. 2. Route Maintenance

dynamically classifies its mobility and changes its geolocation update scheme by comparing the calculated update frequencies of scheme A and scheme B.

D. Maximum Position Error

The maximum position error, R, is subject to optimization. If we decrease R, we can maintain accurate view of the network topology; however, the frequency of geolocation update will increase. If we increase R, the frequency of geolocation update decreases, and the inaccurate geolocation information degrades the performance of the routing algorithm. In this paper, we assume that the maximum position error is given as a system parameter.

III. MST BASED ROUTING

Based on the efficient geolocation update scheme above, we introduce a table-driven routing algorithm. Each mobile maintains MST to route packets in a distributed manner. The mobile estimates the topology of the network with the assumption that the adjacent mobile nodes within the radio range have communication link between them. The actual topology of the network may be different from the estimated topology because MST is independent of the link state of the network. A packet traveling on a route based on the estimated topology may encounter inactive radio link on the way to its destination. We let the data packet carry Negative Link State (NLS) information so that the mobiles that participate in routing may learn the actual topology of the network. A source node initiates route discovery to destination node by sending a self-routing [2] data packet whose route is determined using the MST at the source node. Each node, which is visited by the self-routing packet, sets a beacon that directs other non-self-routing data packets with the same source and destination pair. Each beacon has a timer, and the beacon is deleted if the timer expires before the next data packet arrives. The timer is reset if a data packet arrives while the beacon still exists. If a data packet arrives at a node, and the node doesn't have beacon, the node makes the packet self-routing with the route determined using its MST. If the node has a beacon, and the radio link to the next hop node is inactive, the node calculates a new route using its local MST and makes the packet self-routing. In this case, the packet is tagged with NLS information of the inactive radio link. Consider the network in Figure 1a. The link between node 2 and node D is failed not because the two mobile nodes are out of radio range but because there is an obstacle between them. When there is a packet arrival at the source node S with destination node D, node S determines the shortest path (i.e., with minimum number of hops) to node D based on the estimated topology. Since node S is not aware of the link failure between node 2 and node D, the shortest path determined using the estimated topology is S-1-2-D. Node S makes the packet self-routing and sends the packet to node 1. Node S sets a beacon which points to the next hop node, node 1. The beacon is shown as a small arrow in Figure 1a. When node 1 receives the packet, it creates a beacon and sends the packet to node 2 using the route specified in the packet header. When the packet arrives at node 2, node 2 discovers that the packet cannot be delivered using the route specified in the packet header. Node 2 calculates the shortest path from node S to node D and locates the node that made a "wrong" routing decision, namely *branching node*. In this example, node 1 is branching node that should have routed the packet to node 3 instead of node 2 (see Figure 1b). Node 2 makes the packet self-routing using the route that passes the branching node. The route consists of two parts. The first part of the route is the shortest path from node 2 to the branching node, and the second part of the route is the shortest path from the branching node to node D. The first part is obtained from the received packet header, and the second part is obtained from the estimated topology using MST. For the network in the figure, the corresponding route from node 2 to node D is 2-1-3-4-D. Node 2 attaches NLS information about the inactive radio link between node 2 and node D and transmits the packet to node 1. The route of the self-routing data packet is labeled as Route 1 in Figure 1a. Each node on Route 1 updates or creates the beacon for the source and destination pair. For the subsequent packet arrivals at node S, node S simply forwards the packets to the node that the beacon is pointing at. These packets are

---

[2] The self-routing packet is delivered using the route specified in the packet header.

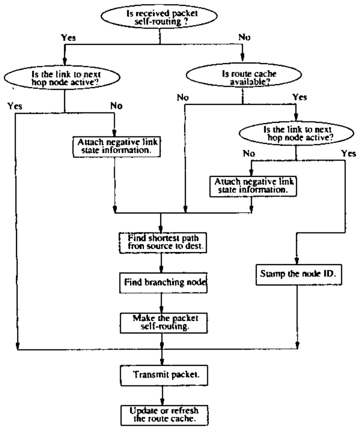


Fig. 3. Flow chart of the MST based routing

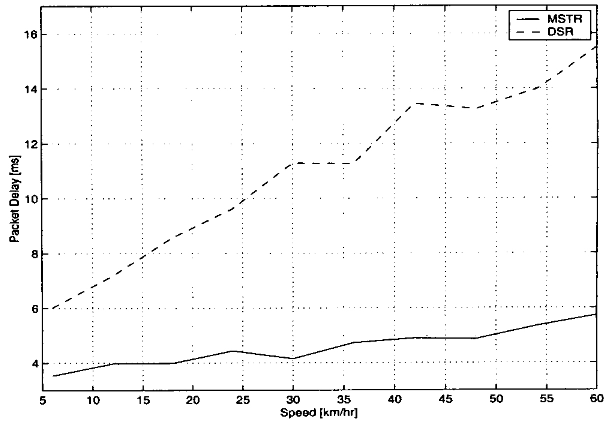


Fig. 4. Packet Delay

not self-routing, and they just follow the beacons until they reach the destination. The route of these packets is shown in the figure as Route 2. For route maintenance purposes, each non-self-routing packet keeps record of the nodes that the packet visits. In case of link failure, the node that is responsible for rerouting the packet will determine the branching node using this record. Consider the network in Figure 2. The shortest path found in the previous route discovery phase is now failed because the link between node 4 and node D is not active any more. When node 4 receives a non-self-routing packet, it determines the shortest path from node S to node D and locates the branching node. Since the nodes don't keep the old NLS information, node 4 doesn't "remember" the link failure between node 2 and node D. Therefore, the shortest path from node S to node D is again S-1-2-D. Since the packet is delivered to node 4 using route S-1-3-4, node 1 becomes the branching node. Node 4 makes the packet self-routing using route 4-3-1-2-D, and attaches NLS information about the inactive radio link between node 4 and node D. When node 2 receives the self-routing packet, node 2 cannot deliver the packet using the route specified in the packet header. Node 2 determines the shortest path from node S to node D using the MST and the NLS information in the packet. Since node 2 is now aware of link failure between node 4 and node D, the shortest path from the source to destination is S-1-3-5-6-7-D where the branching node is node 1. Node 2 makes the packet self-routing using the route 2-1-3-5-6-7-D and transmits the packet. Each node along the path again updates its beacon. The small arrows that are crossed out indicate the updates. Non-self-routing packets will follow the beacons on the way to the destination. The flow chart of the proposed algorithm is shown in Figure 3.

IV. SIMULATION STUDY

In this section, we simulated the MST based routing (MSTR) protocol and compared it to the Dynamic Source Routing (DSR) protocol. In DSR, the source node S initiates route discovery to node D by broadcasting a route query packet and waiting for the route reply from node D. When node S finds a route to node D, DSR specifies the route in the packet header and transmits the packet to the next hop node on the route. We employ a simple route maintenance scheme for DSR. When an existing route fails due to a radio link failure, the node with the inactive radio link sends a link failure message to node S. When node S receives the message, it initiates another route search by broadcasting route query packet. Route caching is not implemented so that the packet delay performance is independent of the system load. Our network simulator is implemented using C++. There are 300 mobiles in the network, and each node has a radio range of 2Km and moves freely in a 10Km x 10Km square shaped area. A node moves with a designated speed toward a randomly chosen direction (i.e., the angle of direction is uniformly distributed between 0 and 2π). The node changes the direction after traveling for the time duration that is also randomly distributed (i.e., uniformly distributed between 0 and 1 min). When a node reaches the network boundary, it bounces back with the same angle as the incoming direction. Each node is modeled by a FIFO queue with infinite buffer space. An radio link is modeled as a server whose service time is the packet transmission time where the speed of transmission is 1Mbps. A source node generates data packet every 1ms for a destination node, and the packet size is fixed to 1Kb. In Figure 4, we plot the average delay for routing a data packet from a source to a destination where the speed of the mobile varies from 6Km/hr to 60Km/hr. As we increase mobile speed, the packet delay in DSR increases roughly 2ms for every 10Km/hr. The packet delay in MSTR increases about 0.4ms for every 10Km/hr. As the mobile speed increases, the life-time of route decreases, and the frequency of rerouting increases. The reason for having small packet delay in MSTR is that the data packet, which needs to be rerouted, becomes a probe packet that searches a new minimum hop route using the distributed routing table (i.e., MST), and there is no need for the destination node to inform the source node about the newly discovered route.

V. CONCLUSIONS

In the conventional routing algorithms, we expect large control overhead for routing packets to high speed mobiles. In this paper, we proposed a geolocation based routing algorithm in which the control overhead for a high speed mobile is not necessarily larger than the control overhead for a slow-moving mobile. A high speed mobile on a highly directed path doesn't need to update its geolocation as long as it maintains its speed and direction. Each mobile in the network locally computes the network topology using the geolocation information of other mobiles. Through simulation analysis, we have shown that the packet delay in MSTR is smaller than the packet delay in DSR. The performance gain is expected to increase as the speed of mobile increases.

REFERENCES

[1] C. E. Perkins, P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Comp. Commun. Rev.,* Oct. 1994, pp.234-244

[2] S. Murthy, J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks,* Oct. 1996, pp. 183-197

[3] C. C. Chiang, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," *Proc. IEEE SICON '97*, Apr. 1997, pp.197-211

[4] D. B. Johnson, D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks," *Mobile Computing, T. Imielinski and H. Korth, Eds., Kluwer*, 1996, pp.153-181

[5] C. E. Perkins, E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. and Apps.*, Feb. 1999, pp.90-100

[6] S. Basagni, I. Chlamtac, V.R. Syrotiuk, "Dynamic Source Routing for Ad Hoc Networks Using the Global Positioning System," *Proc. Wireless Communications and Networking Conference*, 1999, vol.1, pp.301-305.

[7] E.D. Kaplan, Ed., *Understanding GPS : principles and applications,* Artech House, Boston, MA, 1996.

[8] T. S. Rappaport, J. H. Reed, B. D. Woerner, "Position Location Using Wireless Communications on Highways of the future," *IEEE Communications Magazine*, pp. 33-41, Oct. 1996.

[9] C. Rose, R. Yates "Location uncertainty in mobile networks: a theoretical framework," *IEEE Communications Magazine*, Feb. 1997, Vol.35 Issue:2, pp. 94-101