Enhanced Route Maintenance for Dynamic Source Routing in Mobile Ad Hoc Networks *

Bin Xiao, Qingfeng Zhuge, Edwin H.-M. Sha
Department of Computer Science
University of Texas at Dallas
Richardson, Texas 75083
bxiao, qfzhuge, edsha@utdallas.edu

Chantana Chantrapornchai
Department of Mathematics
Silpakorn University
Nakorn Pathom, Thailand 73000
ctana@su.ac.th

Abstract

Dynamic Source Routing protocol (DSR) is an on-demand routing protocol suited for ad-hoc networks, where nodes are mobile and transmission range of a node is limited. In DSR, for a packet to be delivered, there are two main steps: route discovery and route maintenance. In this paper, we present an algorithm to enhance the route maintenance process. In the algorithm, when the predetermined route to send a packet is broken, the packet needs not to be sent back to the source node to restart the route discovery. We propose a heuristic to reroute the packet which is done by a node who discovers the broken link. The node uses information stored locally and neighbor's previous knowledge to help decide the new route. The experiments show that our approach can reduce the error during the delivery due to the broken link upto 90% over the original DSR in the environments where node's mobility and transmission power are varied.

1 Introduction

The ad hoc networks are becoming popular in many commercial and military applications, for example mobile conference outside the office, battlefield and natural disaster communications [1]. In the network, there are a set of mobile hosts without the base station. Every mobile node can communicate with each other directly or by using multi-hop wireless links. Because the ad hoc networks eliminate the complexities of infrastructure setup and administration, it enables mobile hosts to create and join the networks anywhere, anytime. Thus, each node in the network virtually acts as a router, which can find a route for a data packet and can forward the data packet for other nodes. A simple ad hoc network may be look like as shown in Figure 1. Mobile nodes A, B, C are in the transmission range of each other. As a result, they are virtually connected to each other. When node C moves out of B's range (but it still is in the range of node A), the network topology becomes as shown in Figure 2. Node C still can send data packet to node B via node A who acts as a router. Because of host mobility, a routing protocol should be aware of changes in network topology at all times. Also, the routing protocol must consider limitations in available bandwidth and transmission power of a node. Thus, finding a route for a packet from a source to a destination in the ad hoc networks becomes a challenging problem.

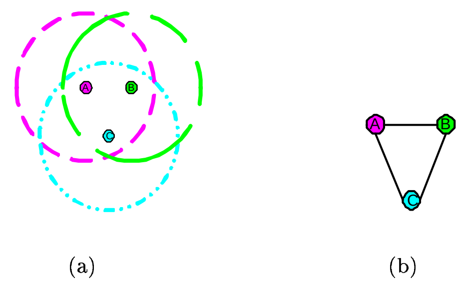

Figure 1: (a) Mobile nodes with transmission range. (b) Equivalent topology.

With all constraints, routing protocols for ad hoc networks fall into two main categories: *proactive* and *reactive* which differ on how a route is discovered.A host running a proactive protocol (e.g. DSDV [2]) will maintain consistent, up-to-date routing information from itself to every other node in the network. When there is a change in network topology, each node

*This work is partially supported by TI University Program, NSF EIA-0103709, Texas ARP 009741-0028-2001, USA, and also by NECTEC, NT-B-06-4D-16-509, Thailand.

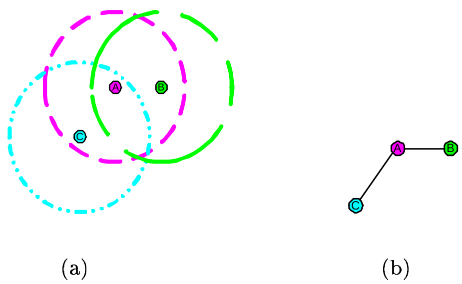

Figure 2: (a) Node C moves out of B's range. (b) Equivalent topology.

should update its routing table and propagate the update information throughout the network in order to maintain a consistent network view [1]. Route discovery would become very easy by just searching route from the tables. However, this method causes a lot of network traffic since it has to periodically checking the topology changes. Also, there are cases, where most of the acquired routing information expire before it is ever used. In the reactive protocol, the route is determined only when needed. Hence, reactive protocols reduce an overwhelming amount of network traffic incurred in the proactive protocol. The reactive protocol has two steps for a packet delivery: *route discovery* and *route maintenance*. Route discovery is a process where a source node determines a route to send a packet. Route maintenance is a process which monitors the correctness of the routing operation. Compared to the proactive protocol, there are more delays in determining a route. However, the protocol introduces much less network traffic and the space to keep routing table is less since not all routing paths are maintained. Hence, the reactive protocol is more attractive for the ad hoc network when considering limited power and bandwidth.

Two prominent reactive protocols, Dynamic Source Routing (DSR) [3], Ad Hoc On Demand Distance Vector Routing (AODV) [4], have been studied. Compared between the two, DSR has a much better performance in a less "stressful" environment such as smaller number of nodes and lower load and/or lower node mobility [5]. Another benefit of DSR is that it consistently generates less routing traffic than AODV. Some other reactive protocols such as WAR [6], ABR [7] and TORA [8] include a local error recovery mechanism, where a route discovery can be performed by a local node. However, these mechanisms are closely related with their route discovery process is and can not be applied to DSR.

In this paper, we focus on the route maintenance for DSR. In [3], whenever there is a broken link in the route for a packet delivery, it will send an error message back to the source node and let the source node initiate another route discovery procedure. This protocol is called an *end-to-end error recovery*. Nevertheless, an end-to-end error recovery method will result in a waste of wireless bandwidth as well as the delay of packet delivery. In our approach, since the original route in the delivered packet contains valuable information, a node can fix a broken route using information from its local routing table and the information in the original packet header. Therefore, a packet delivery can be continued and the performance can be improved. Our simulation result shows that up-to 75% success rate can be achieved by continuing the packet delivery using our approach whenever a broken link occurs.

2 Dynamic Source Routing Protocol

Dynamic source routing protocol is a reactive protocol, which is based on the concept of source routing. The header in a delivery packet specifies the route for a packet to be traversed to reach its destination. When a node along the route receives a packet, it checks the route specified in the packet header and decides to which node the packet should be transmitted. Once the packet reaches its destination, the data packet is successfully delivered [3].

Routing table cache and neighbors are important information for each mobile node to find a route for a route inquiry. In the wireless network, neighbor's content of a node should be refreshed periodically. The content contains information on what other hosts the node can directly communicate with. Routing table cache records different paths from the host (source node) to different destinations. A node and its neighbors can update their cache using *promiscuous* receiving mode on its wireless network interface. When a node tries to find a path for a data packet, it first check whether the destination of packet is its neighbors or is in routing table cache. If a route is found, this route is put in the packet header for transmitting. Otherwise, the sender may attempt to discover the route to send a packet by the *route discovery* protocol.

During the delivery process, the selected route should be monitored. If one of the intermediate nodes along a route is moved out of the others' transmission range, the packet can not reach its destination via this route anymore. Also, the same situation happens if a host along the route is failed or powered off. *Route maintenance* is the process to monitor the validity of the selected route. When the route maintenance detects a problem in sending a packet, it should figure out an alternative solution either by sending the packet back to the source node so that the route discovery is performed again or find another route using information locally.

2.1 Route Discovery

Route discovery is a process for a host in the ad hoc network to discover a route to deliver a packet to any other host. The destination can be just its neighbors or can be reached using multiple hops.

When a mobile node (source node) has a packet to send to a destination, it first checks its neighbors and its routing table cache. If the destination is its neighbor, it will send the packet to the neighbor. If it finds a path to the destination from the cache, it will use this route to send the packet. Otherwise, it initiates a route request and broadcasts the request. This message contains the address of the destination together with a request ID number. Each node receiving a route request will check whether it has seen this request before. If so, it discards the route request message. If not, the node will check whether it knows a route to the destination. If it does not know, it adds its ID to the route request message and then broadcasts it. A route reply message will be generated to the request initiator when the node receiving the request is the destination and the route is created by reversing the route in the route record of the request. The route reply may also be generated by the node who knows a route to the destination. When the source node receives a route reply message, it will transmit the packet using the route information received.

2.2 Route Maintenance

Route maintenance is a procedure that monitors the operation of packet delivery and reports any link failure during the packet delivery. Because of node mobility, a node probably can not deliver data packet to its next node in the selected route. The data-link layer may report the problem in transmitting. The node then generates a route error message. This message is sent to the source node. With the wireless network interface working in a promiscuous mode, all nodes (including the source node and the failed node) will receive the error message. Thus, they will refresh their neighbors and routing table cache. All routes containing the failed node are truncated at that point. The source node receives the error message and may begin a route discovery process again and transmits the packet using the new route.

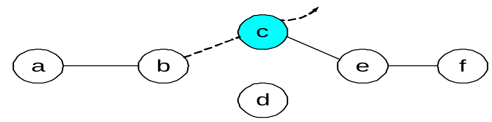

Figure 3: An example where a link from $\mathtt{b}$ to $\mathtt{c}$ is broken.

3 Route Maintenance for DSR

When a route for delivering a packet is broken, the straightforward solution is to send an error message back to the source node. Then the source node initiates another route discovery. In Figure 3, we have route from $\mathtt{a}$ to $\mathtt{f}$: $\mathtt{a} \rightarrow \mathtt{b} \rightarrow \mathtt{c} \rightarrow \mathtt{e} \rightarrow \mathtt{f}$. If host $\mathtt{c}$ moves out of the transmission range of host $\mathtt{b}$, the route is then broken. When node $\mathtt{b}$ transmits a packet to node $\mathtt{c}$, node $\mathtt{b}$ should have a communication channel to forward the packet. Since now the channel does not exist, node $\mathtt{b}$ would assume that the link to node $\mathtt{c}$ is broken when it hears no feedback from node $\mathtt{c}$. Thus, node $\mathtt{b}$ will send an error message to the source node $\mathtt{a}$. When a route error message is received, nodes along the route which previously forwarded the packet will invalidate the packet delivery information from their caches, and all routes which contain this hop must be truncated accordingly. Source node $\mathtt{a}$ will invoke a route discovery process again to construct a new route to send this packet.

The above method for route recovery may be inefficient. Sending an error message back to the source node to invoke a new route discovery may be costly. This process will make the packet delivery time longer. The scenario may become even worse for a larger network and can not be tolerate by many real-time applications. It is desirable to find a new route as soon as possible when the route can no longer be used at the minimized overhead (in terms of bandwidth). Since the packet already has the whole route information from its source to its destination, and there is only one broken link, called $\mathtt{B} \rightarrow \mathtt{C}$, the remaining part of the route after node $\mathtt{C}$ may still be valid. Also, there may still be a route from $\mathtt{B}$ to other nodes after $\mathtt{C}$. If we can find another way to from $\mathtt{B}$ to the packet destination, the delivery process can be continued. Therefore, the new route discovery performed by the source node for the packet is not necessary. By doing so, the overall route discovery time will be decreased, and the whole system performance can be improved.

To achieve a better performance, mobile hosts should have up-to-date route information, which includes any error message introduced by a failed transmission. Because wireless transmissions often use a broadcast message, we assume all mobile hosts to take a full advantage of their promiscuous receiving mode by overhearing all surrounding packet informations. The information the node perceived will be used to update the node's neighbor content and its routing table cache.

In order to find a new route for a link broken $\mathtt{A} \rightarrow \mathtt{B}$, the local route recovery is performed as follows:

1. An error message about the broken link is broadcasted to all node A's neighbors. Node A's neighbors receiving the error message refresh their caches.

2. Node A searches for a new route to any node after B according to the original route from its cache. If the new route is found, the routing information is changed accordingly in the packet header. The packet then is forwarded using the new route. After that, node A sends an error message with the new route to the source node. Any node who overhears this message may update its cache content. If the new route is not found, go to step 3.

3. Node A performs a local route recovery. Node A sends a local inquiry to its neighbors for finding a new route to any node after B according to the original route. If a new route is discovered by any of its neighbors' cache, the route is changed accordingly in the packet header. Then the packet is delivered. Node A sends an error message and a new route to the source node. Any node who overhears this message may update its cache content. If the new route is not discovered by its neighbors, step 4 is taken.

4. An error message is sent to the source node in order to update its cache content. After the source node receives this error message, it will begin a route discovery, to find another route to deliver the packet.

In step 1, the error message ensures that the new route discovered will not consider the broken link $\mathtt{A} \rightarrow \mathtt{B}$. In step 3, one may expand the local routing inquiry by asking neighbors of neighbors of node A and so on.

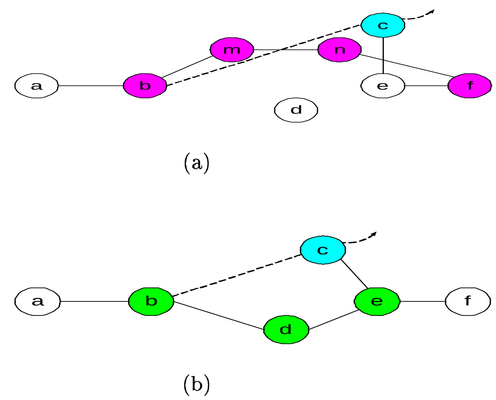

Figure 4: (a) Find a new path in b's cache. (b) Find a new path in the cache of b's neighbor

However, our simulation results reveal that doing this yields very little improvement.

In an example shown in Figure 3, if node c moves out the transmission range of node b, according to step 1, node b's neighbors receives an error message and their caches are updated. In step 2, suppose we can find a new route from $\mathtt{b} \rightarrow \mathtt{m} \rightarrow \mathtt{n} \rightarrow \mathtt{f}$ in b's cache. We deliver the packet using the new route as shown in Figure 4(a).

However, in the case that, b's cache does not contain a route to any node after c according to the original route, we initiate a local route discovery as in step 3. If b's neighbor, say d, find a route to the destination, from $\mathtt{d} \rightarrow \mathtt{e} \rightarrow \mathtt{f}$, d will report the route to node b. Then node b will change the new route to be $\mathtt{b} \rightarrow \mathtt{d} \rightarrow \mathtt{e} \rightarrow \mathtt{f}$ as shown in Figure 4(b) and the packet is delivered using this path.

4 Simulation Model

We evaluate the performance of our route maintenance technique for DSR through an event-driven simulation of an ad hoc network. The performance of route maintenance is measured by the routing and delivering success, delivery error reduction for various mobile node's velocity and transmission range. In the simulation model, we compare the two approaches: the original DSR protocol which discards the packet and lets the source node start the packet delivery again, while our route maintenance technique where a route discovery is performed locally to continue the delivery when a link is broken.

The ad hoc network in our experiments consists of 30 mobile nodes, whose initial positions are chosen from a uniform random distribution over the area of 1000×1000 meters. We assume there are 5,000 packets initiated in the system. Further, we studied the cases when the mobile node velocity and the transmission range changes as in the following two cases:

1. The mobile node's transmission power range is 270 meters, while all nodes' velocity varies from 1 meter/sec to 20 meters/sec.

2. The mobile node's velocity is 10 meters/sec, while all nodes' transmission range is varying from 220 meters to 320 meters.

We also assume that the node's direction θ, which is measured counter-clockwise as an angle compared to the positive x-axis, is uniformly distributed and valued between 0 and 2π. When a node reaches the edge of the simulation region, it is reflected back into the area with its direction set to $-\theta$ (with respect to the horizontal edge) or $\pi - \theta$ (with respect to the vertical edge). The magnitude of the velocity is not changed.

Each node has a store-and-forward queue, with an adequate buffer space to hold the packets for transmission. Each link is modeled using a First-Come-First-Serve queue with a fixed service time. All packets are assumed to be data packets. Data packet already contains some useful control information, such as time-stamped, packet identification, packet search or delivery etc.

A neighbor discovery is based on the reception of "HELLO" beacons that are broadcast by a node. These short beacons (containing only a source address) are transmitted at a fixed interval of T_{beacon} = 0.2 sec. Neighbor connectivity is determined by the reception of the "HELLO" beacons. A reply to this beacon should arrive within $2 \cdot T_{beacon}$. If a node has not received the reply beacon back from its neighbor within this time, the link is assumed to be broken. The node will update its neighbor content accordingly. In our simplified ad hoc network environment, these links are bidirectional.

5 Simulation results

In Table 1, we compare the performance of our route maintenance technique with the original method when different node velocity is assumed. In the simulation, 5,000 packets are initiated for delivery. Every node's power transmission assumes to be 270 meters and velocity is varies. In Column "improved" under

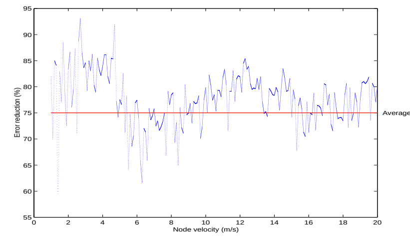

Figure 5: Delivery error reduction at different node's velocity.

" # successful routing", we show the number of packets that are successfully delivered using our approach while Column "original" shows the number of packets that successfully finds their route to the destination using the original DSR. Under Column "# error delivery", we compare the number of packets that never reaches its destination because of a link failure. The number for our route maintenance technique and original DSR are shown under fields "improved" and "original" respectively. Also, in Column "# never find a route", we gives the number of packets initiated by source nodes, which cannot be sent since the source can not find a route during the route discovery. Column "%" under fields "# successful routing" and "# error delivery" shows how better our approach is performed in terms of routing success and the packet delivery when compared to the original DSR. Column "avg. time to rebuild locally" shows the average time used to compute a new route for each case used in our approach. We use a Pentium III 866 MHz to perform this simulation. In general, our approach only takes less than 20 ms to find a new route locally and deliver it to destination when there is a broken link.

Similarly in Table 2, we test the scenario when the node's velocity is fixed to 10 meters/second and we vary the ranges of transmission. From both tables, we can see that our technique gives a better performance by increasing the successful delivery than the original DSR averagely by 75%.

On average, our method gives a 75% reduction on the delivery error over the original DSR as indicated in Figure 5, which is for Case 1 in Section 4. When transmission range is over 300 meters at the node speed at 10 meters/second, we can reduce the error during the delivery by almost 90%, which can be seen from Figure 6.

Node velocity (m/s)	# successful routing			# error delivery			# never find a route	avg. time to rebuild locally (ms)
	original	improved	%	original	improved	%		
1	4550	4573	0.5	28	5	82.1	422	18.9372
10	4413	4528	2.6	144	29	79.9	443	17.8734
20	4557	4750	4.2	241	48	80.1	202	17.4712

Table 1: Performance comparison between our route maintenance technique and the original DSR for different mobile node's velocity with transmission range = 270 meters.

Transmission range (m)	# successful routing			# error delivery			# never find a route	avg. time to rebuild locally (ms)
	original	improved	%	original	improved	%		
220	2033	2086	2.6	109	56	48.6	2858	13.2688
280	4777	4909	2.8	178	46	74.2	45	18.6234
320	4837	4985	3.1	163	15	90.8	0	17.5143

Table 2: Performance comparison between our route maintenance technique and the original DSR for different transmission range with node's velocity = 10 meters/second.

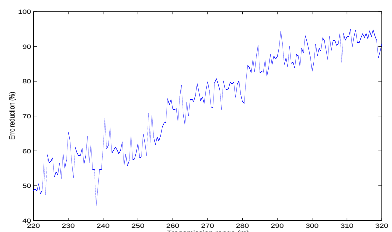

Figure 6: Delivery error reduction at different transmission ranges.

6 Conclusion

In this paper, we propose an enhancement to the original DSR. The approach attempts to fix the broken link locally. A new route may be discovered by any node which detects the problem and the packet can continue to be delivered. The simulation scenarios show that the faster a node's velocity, the higher rate for link broken during the packet delivery. Also, the broader the transmission range is, the easier the route can be found. Our experimental results show that by using our approach the delivery error can be reduced about 75% on average. When transmission range is over 300 meters at the node's speed at 10 meters/second, we can reduce the error during the delivery by almost 90%.

References

[1] S.-L. Wu, S.-Y. Ni, Y.-C. Tseng, and J.-P. Sheu, "Route maintenance in a wireless mobile ad hoc network," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000.*, pp. 3015–3024, 2000.

[2] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pp. 234–244, 1994.

[3] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing* (Imielinski and Korth, eds.), vol. 353, Kluwer Academic Publishers, 1996.

[4] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," pp. 90–100, IEEE WMCSA'99, New Orleans, LA, Feb. 1999.

[5] S. R. Das, C. E. Perkins, and E. E. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *INFOCOM (1)*, pp. 3–12, 2000.

[6] I. D. Aron and S. K. S. Gupta, "A witness-aided routing protocol for mobile ad-hoc networks with unidirectional links," in *MDA*, pp. 24–33, 1999.

[7] C.-K. Toh, "A novel distributed routing protocol to support ad-hoc mobile computing," in *Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications*, pp. 480–486, 1996.

[8] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *INFOCOM (3)*, pp. 1405–1413, 1997.