Route Reutilization Routing in Mobile Ad Hoc Networks

Seungjin Park[1], Seong-Moo Yoo[2] and Fan Qiu[3]
[1]Department of Management, MIS, and CS, University of Southern Indiana
Evansville, IN 47712, USA
[e-mail: park@usi.edu]
[2]Electrical and Computer Engineering Department, University of Alabama in Huntsville
Huntsville, AL 35899, USA
[e-mail: yoos@eng.uah.edu]
[3]Computer Science Department, Vanderbilt University
Nashville, TN 37203, USA
[e-mail: qqfan2002@hotmail.com]
*Corresponding author: Seong-Moo Yoo

Received February 1, 2010; revised March 16, 2010; accepted March 18, 2010; published April 29, 2010

Abstract

Route discovery in wireless mobile networks requires a considerable amount of resources due to the mobility of the hosts. Therefore, it would be wise to utilize the effort already invested in existing paths. This paper proposes an efficient way to reuse, whenever possible, existing paths when a new path is being established. In our proposed algorithm, called Route Reutilization Routing (RRR), the reusability is accomplished by the notion of the dynamic proactive zones (DPZ), through which nearby existing path information is disseminated. By utilizing the information stored in DPZs, RRR can achieve considerable savings over other on-demand routing algorithms that use flooding. The unique feature of the proposed algorithm is that DPZs are created and destroyed dynamically around the existing paths, whereas proactive zones are formed around the nodes throughout the network in other route finding algorithms. Even though using DPZs may not result in the shortest path between source and destination, simulation results show the considerable reduction in traffic needed to find a path and therefore increases the available bandwidth for data transmission.

Keywords: Mobile ad hoc networks, routing, reactive routing, route reutilization

DOI: 10.3837/tiis.2010.04.001

1. Introduction

A network that consists of wireless mobile hosts without any centralized control point or fixed infrastructure is called a *mobile ad hoc network* (MANET). MANETs are of interest because they can be quickly deployed and used without prior arrangements. Applications include fast establishment of military communication and rescue missions where established network is neither feasible nor available [1]. Each mobile host (called a node) in a MANET has a limited transmission range. Therefore, if a node (source node) needs to communicate with another node (destination node) that is not within the source's transmission range, the source node must find intermediate nodes that are willing and able to relay the message to the destination node. The path from the source through the intermediate nodes to the destination is called a route or a path. Finding and maintaining such routes without global information in a MANET are very challenging tasks due to the mobility of the nodes. One popular classification of the routing algorithms is based on the time when the route from the source to the destination is determined. In proactive algorithms [2] each node periodically updates and maintains a Routing Table that contains the routes to all nodes in the network using the well-known algorithm such as link state or distance vector routing. Therefore, when the source node S has a packet to be sent to the destination node D, then S can send the packet immediately along the best path available in the routing table.

On the other hand, nodes in *reactive* algorithms do not maintain any routing information [3][4][5]. Instead, a routing usually consists of two phases: route discovery phase and data delivery phase. When node S has a data for D, S starts finding a path to node D in route discovery phase. Once a route is found, S sends data along the path in data delivery phase. In the route discovery phase S broadcast a control packet called route request packet (REQ) to all its neighbor nodes. If a neighbor node does not have route information to D, it appends its information to the REQ and relays the REQ to all its neighbor nodes, and so on until the REQ reaches D. (This forwarding of requests from one node to all its neighbors is called "flooding" and consumes a lot of time and bandwidth.) If the REQ reaches either D or a node that contains the path information to D, the node sends a control packet called route reply packet (RRY) back to S by reversing the path stored in the REQ. On receiving the RRY, S knows that the path to D has been established, and S sends its data packet along the path. Haas [6] has proposed a hybrid routing protocol called Zone Routing Protocol (ZRP), which takes desirable features of both proactive and reactive protocols. Routing in ZRP consists of Interzone and Intrazone routing. Each node forms an Intrazone around it in which the best routes to all nodes in the zone are proactively maintained. Interzone routing reactively discovers routes to destinations that are beyond the node's Intrazone. Other well-scalable routing schemes based on hierarchical routing can be found in [7][8][9][10][11][12][13][14]. Routing protocols can be improved if additional information about the nodes is available. A set of different routing algorithms have been proposed based on the assumption that each node may obtain its own geographic information via GPS or other service [15][16][17]. LAR [15] uses location information to facilitate a reactive route discovery algorithm by searching only Expected Zone (EZ). Greedy forwarding [16] selects the node geographically closest to the destination as the next hop node. DREAM [17] uses a proactive approach by constantly exchanging location information among nodes in the network. Even though both methods use global flooding to find the destination nodes in the first place, the position information of the nodes can reduce the considerable amount of search space for later search. As mentioned earlier, finding a route is an expensive operation in wireless networks in terms of both time and bandwidth utilization. Therefore, an efficient routing algorithm with low communication overhead may increase throughput for data traffic in the network. Since we have already invested considerable resources to find routes, it would be wise to take full advantage of these existing paths. Unfortunately, most of the routing algorithms described above do not consider this, and as a result, they waste valuable resources in finding new routes even when there already exist nearby routes that may reduce the overhead in route discovery process.

This paper presents a new hybrid routing protocol, called Route Reutilization Routing (RRR), that utilizes the existing paths whenever possible to reduce the communication cost. RRR shares information of an existing route with nodes “near” that route so that when a route request comes near the existing route the request can be forwarded to the existing route and travel that route if it would move the REQ closer to the destination. This is only possible because there is location information of the destination nodes of the existing routes. To control the propagation of this existing route information a specific zone is formed around the path, and dissemination of the path information is limited to the inside this zone.

Although RRR and ZRP [6] are both hybrid algorithms that use routing zones, there are fundamental differences.

1) Routing zones in ZRP are created around each and every *node* in the network all the time, whereas routing zones of RRR are formed around *paths* that are created and destroyed dynamically with time.
2) Routing zones in ZRP needs to be updated constantly (proactive), whereas routing zones in RRR do not need any periodical maintenance (reactive).
3) Routing zones in RRR is designed to help finding paths quickly and efficiently by re-using portions of existing paths, whereas routing zones of ZRP are designed to propagate packets as quickly and efficiently as possible.

Our paper is organized as follows. Section 2 presents necessary background to understand the paper. The proposed algorithm and its simulation results are explained in Section 3, 4, respectively. Conclusion follows in Section 5.

2. Preliminaries

2.1Terminology and Notations

In wireless networks, radio transmission ranges of wireless nodes are limited due to physical and economical reasons. If node B is within a transmission range of node A, then B is said to be a *neighbor* of A. It is also said that there is a link connecting from A to B. For all nodes in the network, if A being a neighbor of B implies B being a neighbor of A, then the network is called *symmetric*, and *asymmetric* otherwise. Often node S wants to communicate with node D that is not a neighbor of S. For a successful transmission between the two nodes there must be a series of *intermediate nodes*, $B_1, B_2, \ldots, B_m$, $m \geq 1$, such that B_i is the neighbor of B_{i-1} and B_{i+1} to relay the packet from S to D. These ordered set of nodes (S, $B_1, B_2, \ldots, B_m$, D) is referred to as a *path* (or *route*) from S to D and denoted as $P(S, B_1, B_2, \ldots, B_m, D)$. Note that every intermediate node has exactly two neighbor nodes in the path. The *distance* between two nodes A and B is defined in two ways: 1) as the number of links (also known as *hops*) between the two nodes, 2). As the physical distance between the two nodes denoted as $|(A, B)| = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2}$.

There is a significant difference between *unicast* (or *send*) a packet and *broadcast* (or *flooding*) a packet in wireless networks. Unicast implies there is a single designated destination node in the source's transmission range for the packet, whereas all nodes are the destinations in broadcasting. When node A wants to *unicast* a packet to node B within its transmission range, A transmits a signal that reaches all nodes in its transmission range. Although it could be broadcast, only B reacts to that packet in unicast communication, whereas in broadcast all nodes in the transmission range react. Therefore, unicast consumes much less time and bandwidth than broadcast. In the proposed algorithm, unicast communication will be used, whenever possible, over broadcast.

2.2 Motivation

To illustrate the advantage of using existing paths, suppose P (A, B, C, D, E) shown in **Fig. 1** is established at time ti. Further suppose that at time tj, tj > ti, node V wants to communicate with node W. If a routing algorithm does not take advantage of P(A, B, C, D, E) , then V should flood a REQ either over the entire network (case when the location information is not available or when the routing algorithm does not utilizes the location information even if it is available) or over a limited area (case when the location information is available and used in the routing algorithm). Note that both cases require broadcast that consumes lots of resources. On the other hand, if a routing algorithm tries to utilize the information of the existing paths, and if node V is aware of the existence of P(A, B, C, D, E) with the information of its direction towards W, then V simply unicasts a REQ to B, and B forwards the REQ to E along P(A, B, C, D, E) (using unicast, of course), and E finally unicasts the REQ to W. Note that all the communications are unicasts, not flooding.

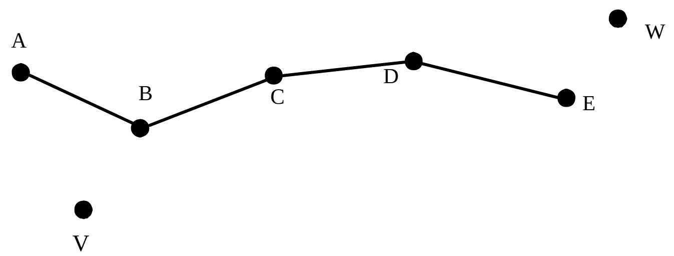


Fig. 1. A network contains a path P (A, B, C, D, E)

From the example given above, it should be clear by now that the utilization of the existing paths could improve the network throughput considerably by reducing broadcast whenever possible. To achieve the improvement, however, there are some problems that should be resolved. They are: 1) How far the path information should be disseminated, and 2) what should be the shape of the disseminated area? Following examples show some insights.

Example 1) Suppose node G wants to find a path to H in **Fig. 2**. Further suppose the distance between G and B is 2. Then, if G uses path P(A, B, C, D, E, F), then the packet would take P(G, L, B, C, , E, H) that is six hops. On the other hand, if G discovers and uses a new path P(G, I, J, K, H) instead of using the existing path P(A, B, C, D, E, F), the distance between G to H is 4 hops. Therefore, P(G, I, J, K, H) may result in less number of hops in data transmission, however, it might require flooding of REQ in path discovery phase. Therefore, in this case it would be beneficial to disseminate the path information to G.

Example 2) Suppose node M in **Fig. 2** has a packet for J. Then, it is obvious from the figure that P(A, B, C, D, E, F) may not be useful for M, since M is too far away from P(A, B, C, D, E, F), This implies that the path information should not be disseminated to node M.

The previous examples entail a very important question: What are the best size and the shape of disseminated area of an existing path for efficient path reutilization? If the information coverage is too large, then it may waste the resources, since the path information distribution increases network traffic. On the other hand, if the coverage of the information is not large enough, then the paths may not be utilized sufficiently. These observations have motivated us to investigate the relationship among the size of the path information coverage, dissemination pattern, and the actual utilization of the paths.

The assumptions made in this paper are 1) Transmission ranges are the same for each node. Therefore, the network is symmetric. 2) Each node knows its location via system such as GPS. 3) Once a path is found, all nodes in the path store the path information that includes IDs and location information of the source node and destination node ID.

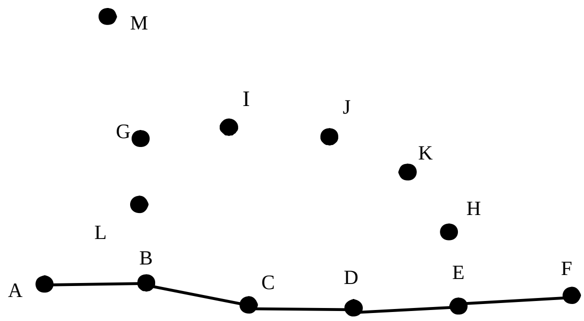


Fig. 2. There is an existing path, P(A, B, C, D, E, F)

2.3 Related Works

This section provides some interesting hybrid routing protocols that consist of both proactive and reactive approaches. More algorithms can be found in [18][19][20][21][22][23][24][25][26][27][28][29].

SHARP [30] is similar to ZRP [6] but there are some fundamental differences. 1) ZRP's main goal is to reduce routing overhead, whereas SHARP concentrates on bound loss rate and control jitter in addition to controlling the overhead of the routing protocols, and 2) a proactive zone is formed around every node in the network in ZRP, whereas SHARP maintains proactive zone only around those nodes that have significant incoming data. HARP [10], like ZRP, is a zone level hierarchical routing protocol. Nodes in HARP maintains only routing information of those nodes that are within its zone, and its neighboring zone. Then, the routing is performed in two levels: intra-zone and inter-zone, depending on the location of the destination nodes. Zone creation and proactive behavior in relation to network properties are provided by DDR [31].

The routing zone of DHR [32] also consists of proactive and reactive zones, however, the main difference from other hybrid algorithms is that the zone is proactively formed around every *path* instead of node. When a path, say P, is formed, the path information is disseminated in the proactive zone (PZ) of P. Therefore, when a node, say A, in this PZ is looking for a path to the destination B that is on P, then A does not have to go through route discovery phase since A knows there is a nearby path to B because all nodes in the PZ has the information of P. The nodes on the path are called *pivots* and the PZ is divided into smaller zones called subPZs such that each subPZ contains exactly one pivot node that is in charge of disseminating path information to its subPZ. The *width* of a subPZ is defined as the number of hops from its pivot node to the farthest node in the subPZ. The shape of a PZ is approximately a triangle such that the width of a PZ is the largest near the source node and becomes smaller as it goes towards the destination node.

Although DHR shows a promising way to reutilize the existing paths, it has following drawbacks.

1) The width of a PZ is proportional to the distance from the destination node. This means that the longer path may have the larger PZ, and therefore, the more nodes in the PZ. If nodes in a PZ want to establish paths towards the destination of the path, they would use the same path, which may cause serious delay due to the heavy traffic on the path.
2) Most importantly, the shape of PZs and subPZs may not take full advantage of existing paths. More detailed discussion will be presented in section 3.
3) In DHR, not only pivot nodes but also nodes in the PZs implement Route
4) Tables,whereas in our proposed algorithm nodes in PZs store the path information in their Path Tables which is much simpler than Route Table. Therefore, considerable space can be saved in RRR.

3. Route Reutilization Routing (RRR)

This section presents a new reactive routing algorithm, called *Route Reutilization Routing* (RRR), which is a reactive source routing, i.e., a path is established dynamically only when the source node has a packet to send to the destination. Further, the final path is determined at the source node. In addition to the Route Table that is commonly used in most routing algorithms, each node in RRR maintains a Path Table. To minimize the unnecessary broadcasts by maximizing the utilization of existing paths, RRR implements two independent modules, Path Utilization Module (PUM) and Routing Module (RM), each is explained in detail below. RM is basically a reactive routing algorithm such as DSR [5], AODV [3], TORA [4], LAR [15], or one that is yet to be developed. RM consists of two phases: *route discovery phase* followed by *data transmission phase*. It is the first phase when PUM is in use. For the fast and efficient path discovery with better route reutilization, PUM at each node maintains a Path Table. Although each node maintains two tables in RRR, Path Table and Routing Table, they are fundamentally different: Routing Table of a node contains the routes that pass “through” the node, and Path Table of a node contains the *information* of the routes that pass “nearby” of the node. For example, nodes A through F in **Fig. 2** contain P(A, B, C, D, E, F) in their Routing Tables, whereas nodes G, L and H may contain the *information* about the P(A, B, C, D, E, F) in their Path Tables. When a new path is discovered, the two tables are utilized such that every node in the path 1) stores the path in its Route Table when it receives the RRY, and then 2) starts disseminating path information to its nearby nodes so that the nodes store the information about the path in their Path Tables for later use. The information represented in Path Table will be explained in Section 3.1. The path information dissemination pattern and its area of coverage are presented in Section 3.2, and path discovery process is shown in Section 3.3.

3.1 PUM and Path Table

The unique feature of RRR is in maintaining Path Table. This section presents the contents of Path Table, and how PUM implements the Path Table to improve the route discovery process. When a new path is found and informed to a nearby node, it should be decided whether the path is significant enough to be stored in the Path Table of the node. For example, if the destination of a newly discovered path is close to a node, it is better for the node not to store the new path information in its Path Table, since it may not be quite useful. For example, the path information of P (A, B, C, D, E, F) is not disseminated to node M in **Fig. 3**, because it is not quite useful to M since M is too close to the destination of P (A, B, C, D, E, F). Instead of storing the whole paths, a row in a Path Table consists of 1) the ID and the location of the source node of the path, 2) the ID and location of the destination node of the path, and 3) the upstream node that leads towards the path. For example, suppose there are two paths P (A, B, C, D, E, F) and P (G, H, C, D, I, J, K), as shown in **Fig. 3**. Then, the Path Table at node L may contain the information of two paths as shown in **Table 1**.

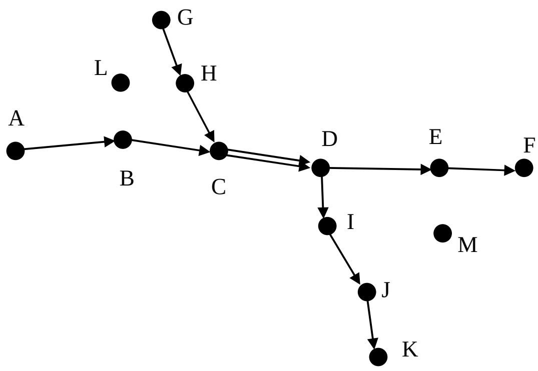


Fig. 3. A network that contains two paths, P (A, B, C, D, E, F) and P (G, H, C, D, I, J, K)

Table 1. Path Table at node L in **Fig. 3**

Source node ID	Source node location	Destination node ID	Destination node location	Upstream node ID
A	(A_x, A_y)	*F*	(F_x, F_y)	*B*
G	(G_x, G_y)	*K*	(K_x, K_y)	*H*

The function of PUM is to guide the route discovery activity in RM via Path Table so that the REQs can discover the paths to the destinations with minimal communication overhead by using the existing paths. The relationship between PUM and RM is described in **Fig. 4**.

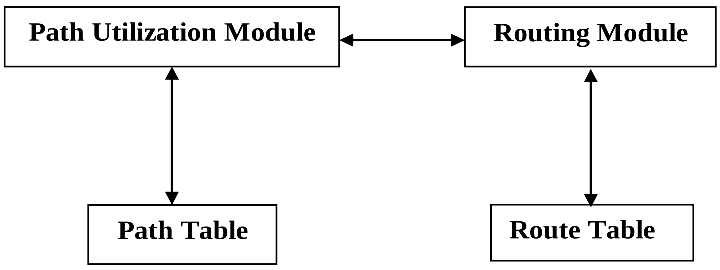


Fig. 4. The relationship between PUM, RM, Route Table, and Path Table

The path information could be stored in Route Table. That is, each node can implement only one table, Route Table. However, because the size of an entry of Path Table becomes much smaller than that of Route Table as the length of the path increases, storage space as well as searching time for a useful path can be reduced considerably by implementing Path Table.

3.2 Path Information Dissemination Pattern and Its Area Coverage

When a new route is discovered, not only should it be decided whether the information of new path be distributed to its nearby nodes, it should also be decided the area coverage of the dissemination, i. e., how many and which nearby nodes should have the path information. In RRR, a zone, called a *Path Disseminated Zone* (PDZ), is formed around a path over which the path information is disseminated. Upon discovering a new path, PUM at each node on the path does the followings.

1) Decides the shape and the size of the PDZ of the path.
2) Produces and propagates a special control packet, called *Path Information Packet* (PIP), to other nodes in the PDZ.
3) When a node receives a PIP, the node stores the path information in its Path Table.

In this section, discussion on the pattern and the area coverage of PDZs are presented.

3.2.1. Path Information Dissemination Range

DHR causes a longer path to draw more packet transmissions, because its PZ becomes bigger. As a result, the longer a path, the more congestion the path may suffer. To overcome this drawback, rectangular shape, as shown in **Fig. 5**, is chosen for PDZs, not only for the relief from the traffic congestion by restricting the size of the width, but also for simplicity and easy maintenance.

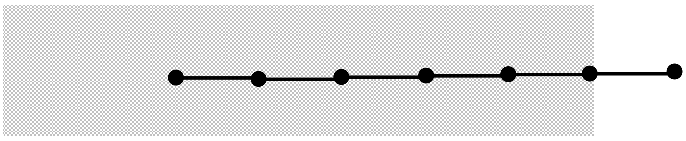


Fig. 5. Path Dissemination Zones (PDZ) chosen for RRR. It is rectangular shape with the same width along the path

3.2.2 Path Information Dissemination Pattern

To minimize the storage usage at each node, RRR allows only every m-th nodes in a newly found path to disseminate the path information to nodes in their PDZs such that they form lines called *branches* of the path. *m* is called *Interbranch Distance*. To propagate the path information, a special type of control packet called *Path Information Packet* (PIP) is prepared and used by every *m*-th nodes. A PIP contains information including the IDs and locations of the source and the destination of the path, the dissemination distance, and the upstream node to the path. Following example illustrates the function of PUM and resulting PDZ with branches. Refer to **Fig. 6**. When the path P(A, B, C, D, E, F, G, H) is discovered by A, all nodes in the path store the path in their Routing Table. Then, PUM at each node on the path decides the width of PDZ with interbranch distance *m*. **Fig. 6** shows a PDZ with *m* = 3 and width = 2, since only every third nodes, i.e., nodes A, D and G, in the path are designated to disseminate the path information to the nodes within two hops from the path. As the first dissemination process, node A (D and G, resp.) prepares a PIP and disseminates it to U and V (K and M, and Q and R, resp.). The PIP contains the ID and the location of the source (i.e., A and (A_X, A_Y)), the ID and the location of the destination (i.e., H and (H_X, H_Y)), upstream node = A, dissemination distance (which becomes the width of the PDZ)= 2. On receiving the PIP, node U (V, K, M, Q, R, resp.) adds the path information to its Path Table, modifies upstream node = U (V, K, M, Q, R, resp.), decreases the dissemination distance by 1, and sends the PIP to T (W, L, N, P, S, resp.). On receiving the PIP, nodes T (W, L, N, P, S, resp.) also adds the path information to its Path Table. When T(W, L, N, P, S, resp.) decreases the dissemination distance by 1, it becomes 0 which indicates the termination of the dissemination of the PIP. **Table 2** shows a new entry in the path table at node L after it received a PIP from K. Other benefit of having m > 1 is that it may help the network traffic being evenly distributed between existing paths.

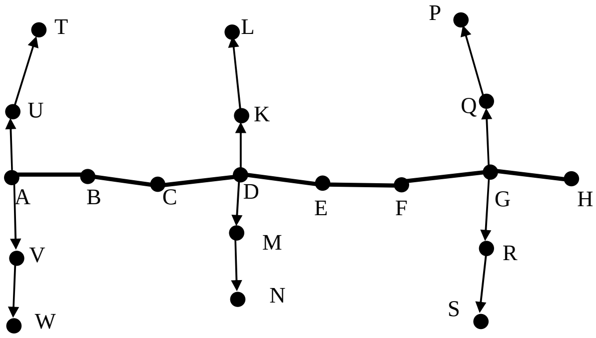


Fig. 6. An example of path dissemination pattern after the discovery of path P (A, B, C, D, E, F, G, H). In this example, Interbranch Distance *m* = 3, dissemination distance = 2, and the arrows show the flow of PIPs in its PDZ

Table 2. A new entry of path table at node L after receiving PIP from D through K

Source ID	Location of the source	Destination ID	Location of the destination	Upstream node
A	(A_X, A_Y)	H	(H_X, H_Y)	K

3.3 Path Discovery in RRR

This section shows how RRR utilizes existing paths when a new path is being established. When the source node S has a data packet to the destination node D, RRR at S invokes RM. RM first checks its Route Table to see if it contains a path to D. If it does, then S unicasts the data packet right away along the path. If there is no entry to D in its Route Table, then RM prepared a REQ and invokes PUM to check its Path Table to see if there is any nearby path that might lead towards D. If there is, then PUM returns to RM the upstream node that leads to the nearby path, and RM sends the REQ to the upstream node. Otherwise, i.e., neither Route Table nor Path Table contains path information to D, RM in S broadcasts the REQ to all its neighbors with its ID appended. If a neighbor node, say N, that received the REQ is the destination D, RM at N (= D) prepares RRY and sends it back to S along the path stored in the REQ. Otherwise, if N has an entry to D in its Route Table, then RM at N prepares RRY with the rest of the paths in the Route Table appended, and sends it to S. If N is not D and does not have entry to D in its Route Table, N invokes PUM to check its Path Table to see if there is any nearby path that may lead to D. If there is, then PUM returns the upstream node that leads to the path, and N sends the REQ to the upstream node. (If there is more than one path that may lead the REQ towards its destination, then the one whose destination is closer to the REQ's destination is chosen. For example, suppose the location of the destination of a REQ is (D_x, D_y), and there are two paths with destinations of (A_x, A_y) and (B_x, B_y), respectively. Then, PUM chooses the smaller of |(D, A)| = $\sqrt{(D_x - A_x)^2 + (D_y - A_y)^2}$ and |(D, B)| = $\sqrt{(D_x - B_x)^2 + (D_y - B_y)^2}$.)

Otherwise, the node appends its information to the REQ, and broadcasts the REQ to all of its neighbors. This process goes on until either TTL of the REQ expires or the REQ reaches D. On receiving the RRY, S sends the data pack to D along the path just found in discovery phase.

This process is summarized below.

At the source node α:

Step 1) α looks up its Route Table. If there is a path entry that contains the destination, then α sends the data packet right away o the next hop in the path; Stop. (For example, if the source node is B and the destination node is F in **Fig. 7**, then B sends its data packet to C, since B contains P(A, B, C, E, F) in its Routing Table.)

Otherwise, go to step 2.

Step 2) α looks up its Path Table. If there is an entry that leads towards the destination, then α sends the REQ to the next hop in the entry; Stop.(For example, if the source node is W and the destination node is U, then W sends the REQ to X to utilize the P(A, B, C, E, F) that leads towards U.)

Otherwise, go to step 3.

Step 3) α starts broadcasting a REQ to all its neighbors, since α does not have any information to the destination. (For example, if the source is V and the destination is D, V does not have any information about D.) Set a waiting duration time for a RRY. If α receives an RRY before the duration time, goes to step 4. Otherwise, waits a random amount of time and goes to step 1.

Step 4) on receiving a RRY, α sends its data packet to the destination along the path that just discovered by the REQ. Store the path in its Route Table and prepare a PIP to disseminate the path information to its nearby nodes.

At intermediate node β:

Step 1) If β is the destination of the REQ, then β prepares and sends a RRY packet to the source node along the path just found by the REQ; Stop.

Otherwise, go to step 2.

Step 2) β looks up its Route Table. If there is a path entry that contains the destination, then β prepares a RRY packet that contains the path stored in the REQ + path from β to the destination, and sends the RRY to the source node by reversing the path in the REQ. (For example, if the source node is J and the destination node is I in **Fig. 7**, then on receiving the REQ containing path (J, U), G prepares a RRY with the path P(P(J, U) + P(G, H, I)), and sends the RRY to U by reversing the path in the REQ.

Otherwise, go to step 3.

Step 3) β looks up its Path Table. If there is an entry that leads towards the destination, then β sends the REQ to the next hop in the entry; Stop. (For example, if the source node is W and the destination node is U, then W sends the REQ to X to utilize the P(A, B, C, E, F) that leads towards U.)

Otherwise, go to step 4.

Step 4) β starts broadcasting a REQ to all its neighbors, since β does not have any information to the destination. (For example, node V does not have any information about the destination D.); Stop.

Example) Refer to **Fig. 7** where there are two paths, P(A, B, C, E, F) and P(G, H, I) with both m = 2. Suppose node S wants to send a packet to node D. Then, S takes the following steps.

1) S prepares a REQ destined to D.
2) S checks its Route Table (RT), but found out that there is no entry to D.

3) S checks its Path Table (PT), but again found no entry to D either.
4) S appends its ID to the REQ and broadcast it to all its neighbor nodes.
5) V receives the REQ and found D ≠ V.
6) V goes through the steps 2 to 4 with V substituting S.
7) L receives the REQ and checks RT and PT, and found that there is a path that leads to D.
8) L appends itself to the REQ, and sends it to upstream node K.
9) On receiving the REQ, K repeats the steps 7 and 8 with K substituting L.
10) On receiving the REQ, C appends its ID to the REQ, and sends it to E.
11) On receiving the REQ, E appends its ID to the REQ, and sends it to F.
12) F goes through the steps 5 and 6 with F substituting V.
13) J goes through the steps 5 and 6 with J substituting V.
14) U goes through the steps 5 and 6 with U substituting V.
15) On receiving the REQ, G sends it to H, and H sends it to I.
16) I goes through steps 5 to 6 with I substituting V.
17) On receiving the REQ, D found that the REQ is destined to itself. Prepares RRY and sends it to S.

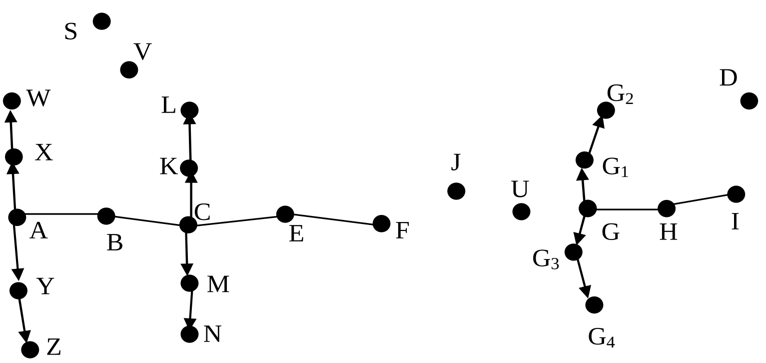


Fig. 7. A network that contains two paths, P(A, B, C, E, F) and P(G, H, I), both with m = 2

3.4 Packet Formats

A REQ in RRR contains fields for the source node ID and its location, the destination node ID and its location, and a list of nodes this REQ passed through thus far. In addition to these fields, a REQ contains "communication mode" field (CMODE in abbr.) that indicates whether the REQ is in BROADCAST or UNICAST mode. A REQ is in BROADCAST mode when flooding is necessary due to the lack of any guidance to the destination. A REQ is in UNICAST mode if it finds a path that brings the packet close to the destination. Therefore, if CMODE = UNICAST, then the Next Hop field contains the node information of the next hop in the path. **Table 3** shows the partial contents of the REQ packet. RRY is similar to REQ except that the CMOD in RRY is always UNICAST.

Table 3. Partial contents of REQ control packet. CMOD indicates the communication pattern, and Next Hop contains the next node, if CMOD = UNICAST

Source ID	Source node location	Destination ID	Destination node location	List of nodes the REQ passed through	Next Hop	CMOD

4. Simulation Results

In our simulation, DSR and RRR were implemented and compared in terms of delivery fraction, average end-to-end delay, and normalized routing load. The main reason why DSR was chosen to be compared with RRR among many routing algorithms was that DSR has proven to be one of the most competitive algorithms in MANET. Since DSR is a pure reactive algorithm and RRR is hybrid, for fair comparison we take into account the number of packet transmissions for proactive zone. Other fundamental difference between RRR and DSR is in use of location information. Since DSR was proposed without location information, GPS has become easier to access and more popular than ever. Therefore, it may not be wise not to take full advantage of it. Note that although the location information may reduce the size of the searching space, it also increases the packet size (and therefore, increase in both network traffic and storage size) since the location information should be included in REQs, RRYs, and PIPs. Because it is not easy to compare two different strategies, it is assumed in our simulation that reduction on searching space and increase in packet sizes level off. To simplify the simulation, we set an Interbranch Distance $m = 1$, and the width of the PDZ =1, i.e., the path information is disseminated only to the nodes that are one hop away from the path. For example, node A in **Table 2** disseminates the path information to nodes U, V only. (Note that since $m = 1$, B, C, E, F also disseminate the path information to their one-hop neighbors.) It should be obvious that the performance of RRR would improve if the width of PDZ > 1 and/or m > 1. The following terms are used in our simulation.

1) Delivery fraction: Data packets received by destination node/Data packets sent by source node.
2) Average end-to-end delay: the average time cost to deliver a single data packet.
3) Normalized routing load: the number of routing packets/number of data packets received.

Here routing packets include: RREQ, RREP and Error packets. For RRR, the PIP packets are also included in routing packets.Our simulation results show that RRR outperforms DSR in most cases.

4.1 Simulation Environment

All simulation results are from simulations using ns2 network simulator. The data packets are generated by constant bit rate generator. Random pairs of nodes are chosen from the network to establish data connections, and the data source is attached to one of them. Then nodes are randomly distributed across the area at the beginning of the simulation. During the simulation, each node will move randomly toward a direction with an unknown speed within a maximum speed that is determined in advanced. Once it reaches a temporary destination, it pauses for a while, and then it moves toward another random destination. In our simulation, the two algorithms were tested with two test scenarios in which the network area, the maximum speed of nodes, and the transmission range are different. Since good algorithm should perform better even under tough situation, we tested the two algorithms under very harsh environment, i.e., the network density, the mobility of the nodes, and the network load are all relatively high.

4.2 Simulation Results

In this test, we change the pause time of the nodes in the network and observe the performance of the two routing protocols, DSR and RRR. Note that the smaller is the value of the pause time, the higher is the mobility of the nodes.

4.2.1 Test Scenario 1

Total simulation time is 800 simulation seconds. The number of nodes in the network is 100. Network area is 300 m × 300 m. Max speed of nodes is 5 m/s. There are 37 data sources with 60 connections. Each source node generates four packets per second with packet size of 512 bytes. Lastly, the transmission range is about 90 m.

1) Delivery Fraction (Data Packets Received/Data Packets Sent)

Fig. 8 shows the delivery fraction, which is the ratio of the data packets received by the destination node to the data packets sent by the source node. Simulation shows that there is not much difference between the two routing algorithms except RRR outperforms DSR when node mobility is high

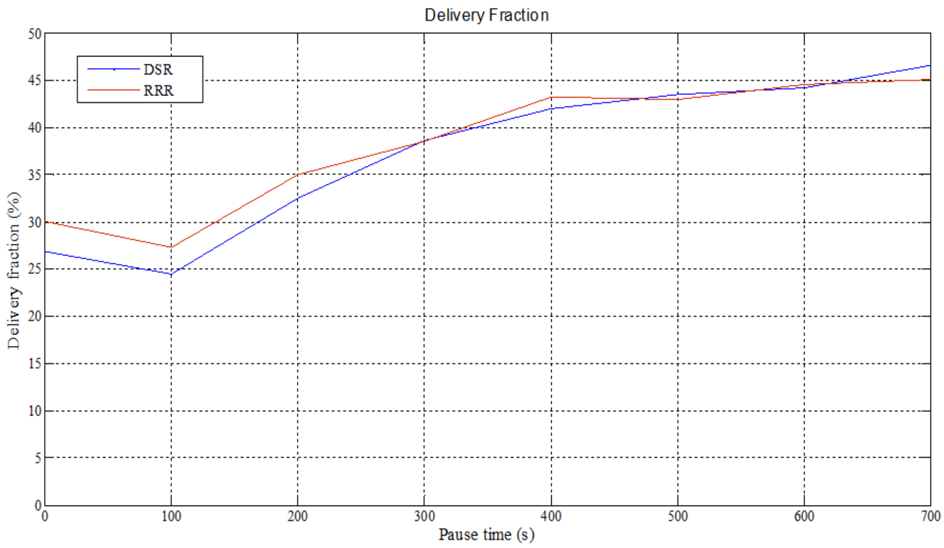


Fig. 8. Delivery fraction (Test scenario 1)

2) Average end-to-end delay

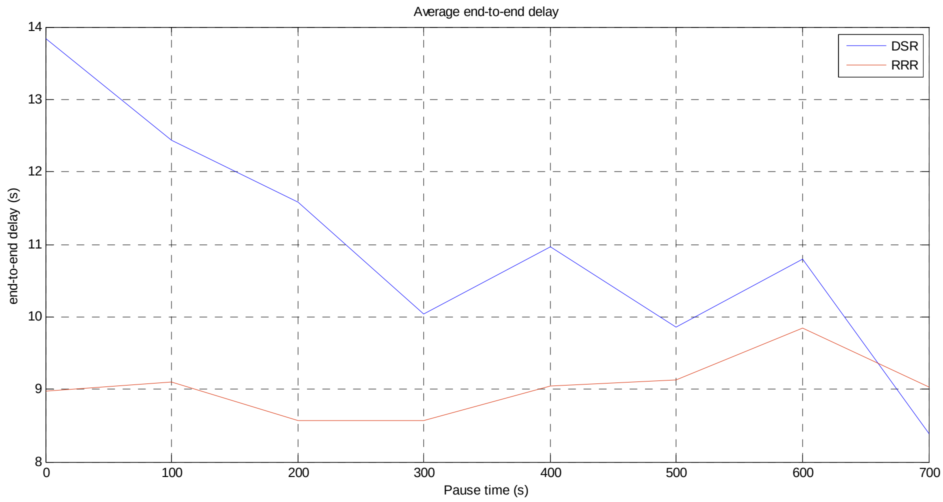


Fig. 9. Average end-to-end delay (Test scenario 1)

From **Fig. 9** we can see that in most cases the performance of RRR is better than DSR in terms of end-to-end delay, especially when the pause time is small. This means when node mobility is high, RRR outperforms DSR. The results indicate that RRR is more adaptive to highly dynamic network. In some situations (e.g. when pause time is between 500 and 700), the average delays are quite close.

3) Normalized routing load (Routing Packets/Data Packets Received)

Fig. 10 shows that RRR outperforms DSR in most cases in terms of the normalized routing load (ratio of total number of routing packets to the number of data packets received by the destination). This means that RRR requires less control packet traffic than DSR. This phenomenon is obvious when the pause time is short, i.e., when the node mobility is high. We may also notice that sometimes RRR costs more than DSR. This is due to the broadcast packets PIP. A node can obtain some routing information from its neighbors in RRR routing. But if these routes are not used when it sends data packets to some other nodes, the network cannot benefit from the dissemination of these PIP packets.

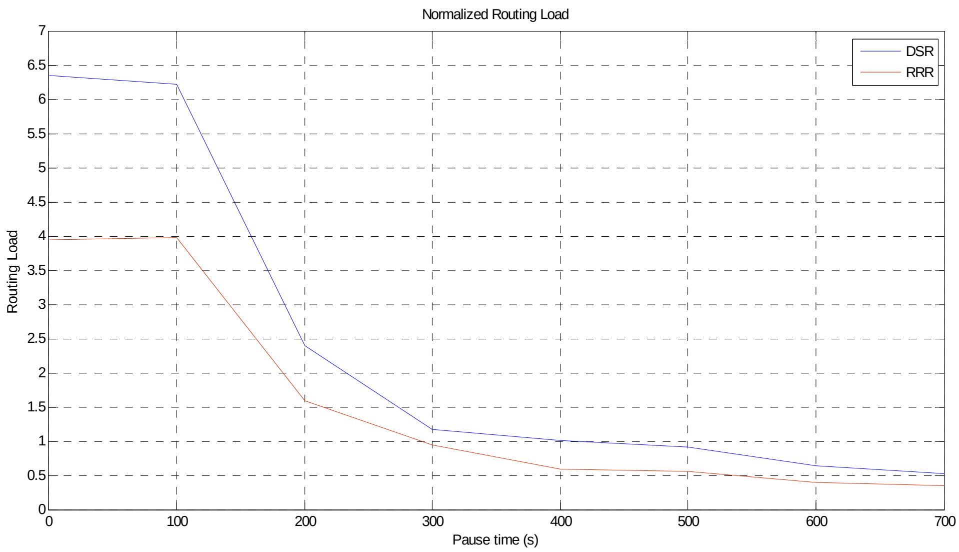


Fig. 10. Normalized Routing Load (Test scenario 1)

4.2.2 Test Scenario 2

Total simulation time is 800 simulation seconds. The number of nodes in the network is 100. Network area is 800m × 500m. Maximum speed of the nodes is 10 m/s. There are 37 data sources with 60 connections. Each source node generates four packets per second with packet size of 512 bytes. Lastly, the transmission range is about 250 m. Since the simulation outcomes in this case are similar to the Test Scenario 1, only results will be presented in **Fig. 11**, **Fig. 12**, and **Fig. 13** without explanation. **Fig. 14** illustrates the proportion of PIP packets to all packets in RRR.

1) Delivery Fraction (Data Packets Received/Data Packets sent)

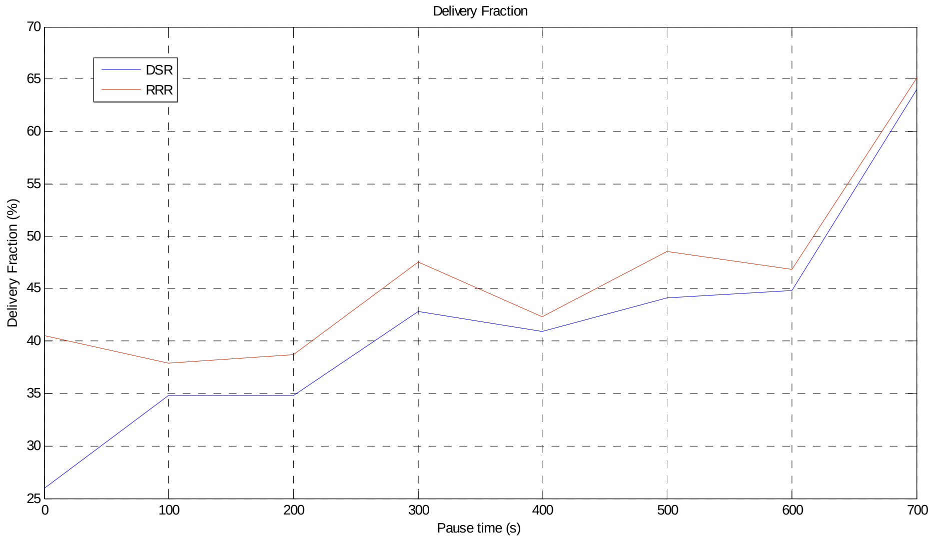


Fig. 11. Delivery Fraction (Test scenario 2)

2) Average end-to-end delay

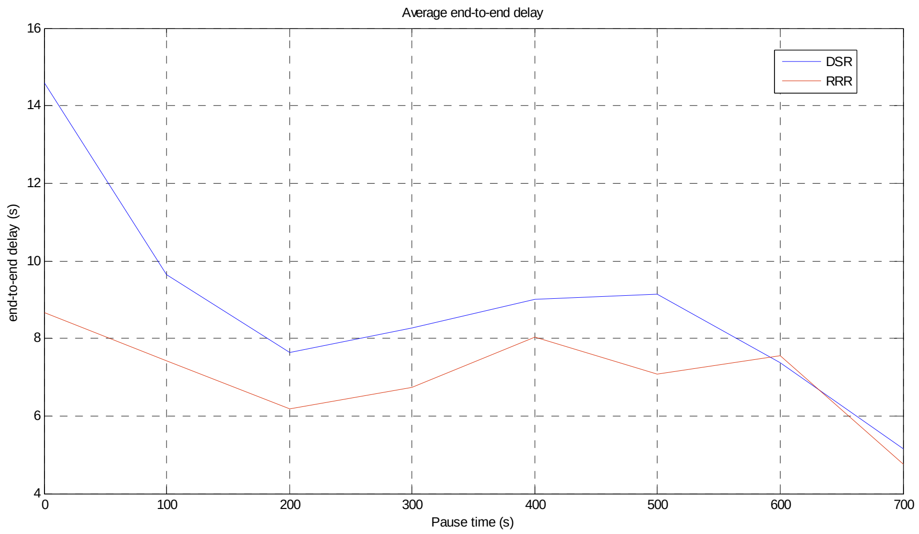


Fig. 12. Average end-to-end delay (Test scenario 2)

3) Normalized Routing Load (Routing packets/Packets received)

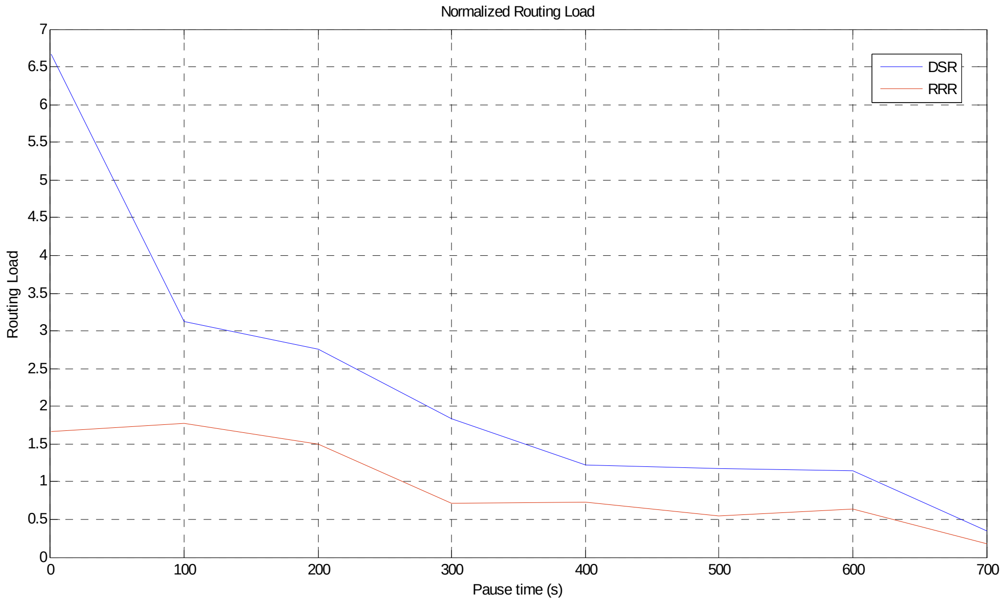


Fig. 13. Normalized Routing Load (Test scenario 2)

4) Fraction of PIP in RRR routing

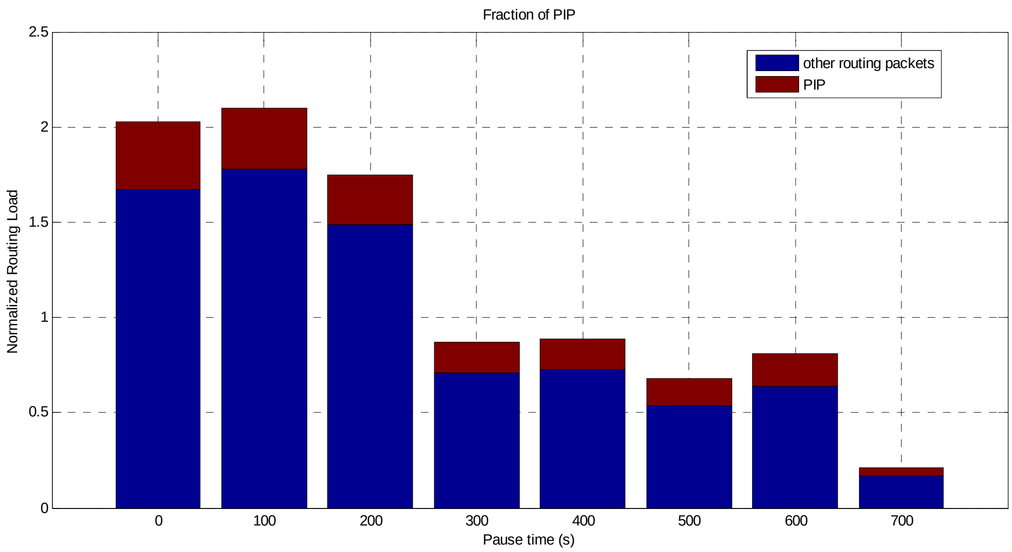


Fig. 14. Fraction of PIP in RRR routing

4.2.3 Network Load (Offered load) vs Performance

To evaluate RRR from another perspective, we change the offered network load and observe the performance of RRR and DSR. Here *network load* = total data sent/simulation duration time. Network setup is the same as Test Scenario 2 above with pause time= 0. Note that the simulation duration time is fixed at 800 simulation seconds in this case, network load is only proportional to the size of the data sent. Simulation results presented **Fig. 15**, **Fig. 16** and **Fig. 17** show that RRR outperforms DSR under most of the network loads, especially when the amount of data sent is large.

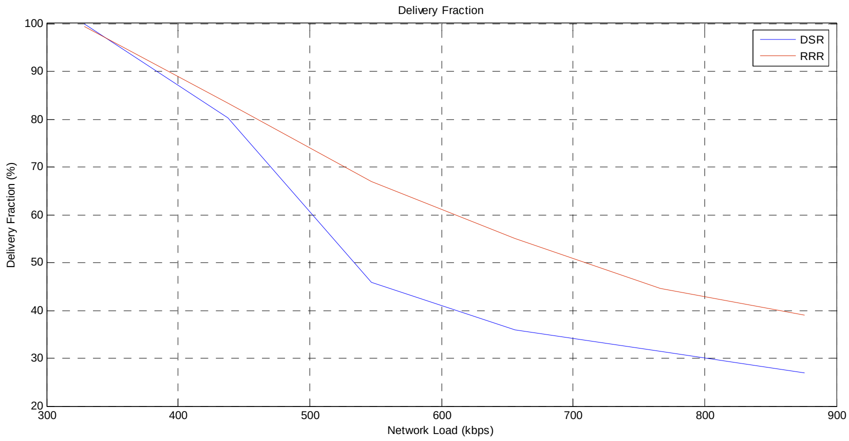


Fig. 15. Delivery Fraction (varying network load)

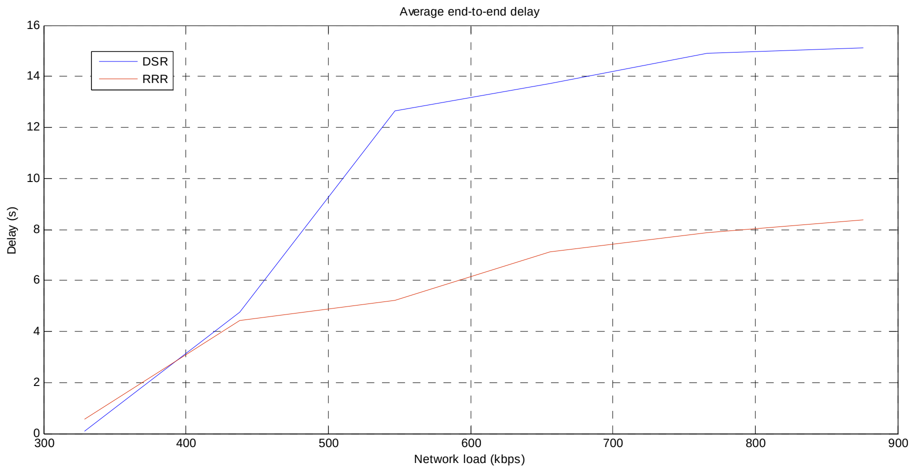


Fig. 16. Average end-to-end delay (varying network load)

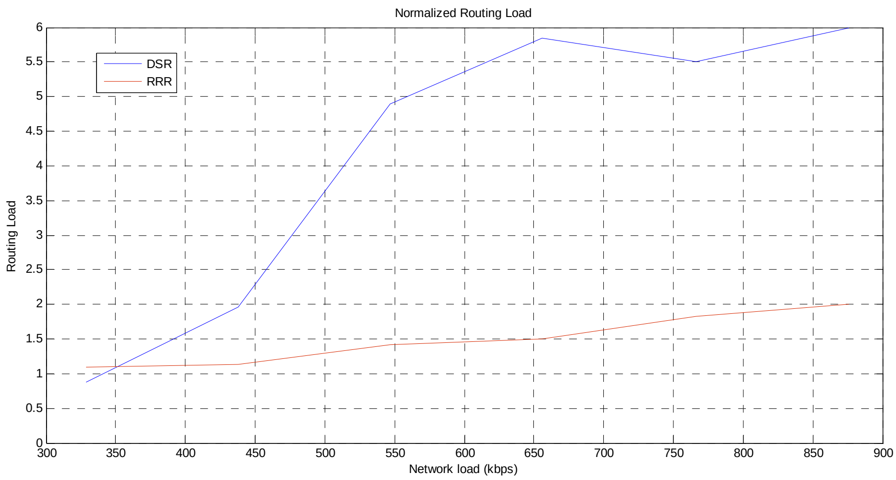


Fig. 17. Normalized Routing Load (varying network load)

5. Discussion and Conclusion

We have proposed an efficient routing algorithm called Route Reutilization Routing (RRR) that is a source and reactive routing. The unique feature of RRR is in the formation and utilization of Path Disseminated Zone (PDZ). When a new path is discovered, the path is stored in the Path Tables of nodes in the path, and the same time, a PDZ is formed around the path through which the information of the path is distributed. If a node in a PDZ has a REQ to broadcast (i.e., if it is involved in the process of discovering a new path), and if the path may help the REQ to reach its destination as quickly and efficiently as possible, then the REQ is guided to use the path. As a result, significant amount of bandwidth can be saved.

Unlike most proactive zones that are static, a PDZ is formed and destroyed dynamically as necessary. Therefore, the lifespan, the shape and the size of PDZs may lay a huge impact on the overall performance of RRR. Another issue can be the pattern of information dissemination within the PDZs. Also, note that since it is assumed that the links connecting nodes are bidirectional, both directions of a path can be used.

Since RRR encourages using the existing paths instead of building a new one, it may cause some heavy traffic in certain paths. To resolve the problem, the width of PDZs, regardless of the length of the paths, remains the same so that it prevents the longer paths from attracting excessive traffic. Also, if a node receives more REQs than some predefined threshold value, it may delay the broadcast of the REQ on purpose so that it reaches the destination later than the REQs along other paths. By the time the REQ reaches the destination, the REQ could be ignored, since the destination already received a REQ, and generated and transmitted a RRY. This strategy may prevent traffic congestion by evenly distributing the traffic among the paths in the same direction.

Note that although the discussion on the destroying of PDZs is not presented in this paper for the simplicity of the algorithm, it can be easily added by introducing Time To Live (TTL) field in Path Table. TTL value of a path is reset to predefined value when 1) it is first established, and 2) it is used in a data packet delivery. When the TTL of a path expires, the path will be removed from the table.

References

[1] C. Perkins, "Ad Hoc Networking," *Addison-Wesley*, 2001.

[2] C. Perkins, and P. Bhagwat, "Highly-dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *SIGCOM'94*, pp.234-244, 1994.

[3] C. Perkins, E. Royer and S. Das, "Ad hoc on demand distance vector (AODV) routing," *IETF*, 1999.

[4] V. Park and M. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. of IEEE Infocom*, pp.1405-1413, 1997.

[5] D. Johnson and D. Maltz, "Dynamic source routing in ad-hoc wireless networking," *Mobile Computing*, pp.153-181, 1996.

[6] J. Haas, "A new routing protocol for the reconfigurable wireless network," in *Proc. of 6th Int. of IEEE Conf. on Personal Communications*, pp.562-566, 1997.

[7] K. K. Kasera and R. Ramanathan, "A location management protocol for hierarchically organized multihop mobile wireless networks," *IEEE ICUPC '97*, pp.158-162, 1997.

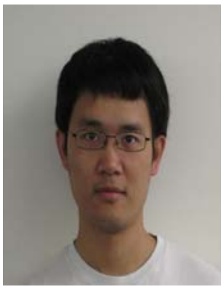
[8] G. Pei, M. Gerla and T Chen, "Fisheye state routing: A routing scheme for ad hoc wireless networks," in *Proc. of IEEE Conf. on Communications*, pp.70-74, June 2000.

[9] A. Iwata, C Chiang, G. Pei, M. Gerla and T.Chen, "Scalable routing strategies for ad hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol.17, no.8, pp.1369-1379, Aug. 1999.

[10] N. Nikaein, C. Bonnet and N. Nikaein, "HARP - hybrid ad hoc routing protocol," in *Proc. of Int. Symposium on Telecommunications 2001*, 2001.
[11] M. Joa-Ng and I-Tai Lu, "A peer-to-peer zone-based two-level link state routing for mobile ad hoc net-works," *IEEE on Selected Areas in Communications*, vol.17, no.8, pp.1415-1425, 1999.
[12] C. C. Chiang, T. C. Tsai, W. Liu and M. Gerla, "Routing in clustered multihop, mobile wireless networks with fading channel," *The Next Millennium, The IEEE SICON*, pp.197-211, 1997.
[13] 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*, pp.1369-1379, 1999.
[14] P. Sinha, R. Sivakumar and V. Bharghaven, "CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm," *IEEE INFOCOM*, 1999.
[15] Y. Ko and N. Vaidya, "Location-aide routing in mobile ad hoc networks," in *Proc. of 4th Annual ACM/IEEE Conf. on MobiCom '98*, pp.66-75, 1998.
[16] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. of 6th Annual ACM/IEEE Conf. on MobiCom '00*, pp.243-254, 2000.
[17] S.Basagni, I.Chlamtac and V.Syrotiuk, "A distance effect algorithm for mobility (DREAM)," in *Proc. of 4th Annual ACM/IEEE Conf. on MobiCom '98*, pp.76- 84, 1998.
[18] S. Hsu, and H. Chang, "A Hybrid Efficient Routing Protocol for Sink-Oriented Ad Hoc Senor Networks," International Journal of Mobile Communcations, vol.7, no.6, pp.695-714, 2009.
[19] H. Cheng and J. Cao, "A Design framework and taxonomy for hybrid routing protocols in mobile adhoc networks," *Communications Surveys & Tutorials IEEE*, pp.62-73, 2008.
[20] T. Dargaghi, A.M Rahmani, and A. Khademzadeh, "A hybrid reactive and proactive approaches based on neighbors' condition of nodes for mobile adhoc networks," in *Proc. of 4th Int. Conf of ICIAFS*, pp.425-430, 2008.
[21] S. Liu, T. Fevens, and A.E Abdallah, "Hybrid Position-based routing algorithms for 3D Mobile adhoc networks," in *Proc. of 4th Int. Conf of MSN*, pp.177-186, 2008.
[22] W. Chen, K. Yang, and A. Achilleos, "RZRP: A Pure reactive zone-based routing protocol with location based predictive caching scheme for wireless mobile Ad Hoc networks," in *Proc. of Int. Conf of Wireless Communications and Mobile Computing*, pp.505-510, 2008.
[23] L.C.P Albini, M.N Lima, A. Brawernab, S. Chessa, and P. Maestrini, "Virtual structure effects in two hybrid routing protocols for ad hoc networks," in *Proc. of 3rd Int. Conf of ISWPC*, pp.436-440, 2008.
[24] Nakagawa, Hiroshi, Ohta, Tomoyuki, Ishida, Kenji, Kakuda, and Yoshiaki, "A hybrid routing with location information for mobile ad hoc networks," in *Proc. of 8th Int. Symposium on Autonomous Decentralized systems*, pp.129-136, 2007.
[25] X. Niu, Z. Tao, G. Wu, C. Huan, and L. Cui, "Hybrid Cluster Routing: An efficient routing protocol for Mobile Ad hoc networks," in *Proc. of Int. IEEE Conf on Communications*, pp. 3554-3559, 2006.
[26] A. Nilsson, J.J. Garcia-Luna-Aceves, and M. Sphon, "Routing in hybrid ad hoc networks using service points," in *Proc. of 58th Int. IEEE Conf. on Vehicular Technology*, pp.2078-2082, 2003.
[27] P. Sholander, A. Yankopolus, P.Cocolli, and S.S. Tabrizi, "Experimental comparison of hybrid and proactive MANET routing protocols", *MILCOM proceedings*, pp.513-518, 2002.
[28] H. Cheng, J. Cao, and X. Fan, "GMZRP: Geography-aided Multicast Zone Routing Protocol in Mobile Ad Hoc Networks," *Mobile Networks and Applications*, vo.14, no.2, pp.165-177, 2009.
[29] J. Wang, E. Osagie, P. Thulasiraman, and R. Thulasiram, "HOPNET: A Hybrid Ant Colony Optimization Routing Algorithm for Mobile Ad Hoc Network," *Ad Hoc Networks*, vol.7, no.4, pp.690-705, 2009.
[30] V. Ramasubramanian, Z. Haas, and E. Sirer, "SHARP: A hybrid adaptive rioting protocol for mobile ad hoc networks," *MobHoc* (2003), pp.303-314, 2003.
[31] N. Nikaein, H. Labiod, and C. Bonnet, "DDR- distributed dynamic routing algorithm for mobile ad hoc networks," *MobiHoc 2000*, pp.19-27, 2000.
[32] S. Park and B. VanVoorst, "Dynamic hybrid routing (DHR) in mobile ad Hoc networks," in *Proc. of Int Conf. on Parallel Processing (ICPP)*, pp.401- 408, 2002.

Seungjin Park received the BE degree in civil engineering from Hanyang University, Korea, in 1973, MS degree in computer science from University of Texas at Arlington, in 1986, and PhD degree in computer science from Oregon State University, in 1993. He is currently an assistant professor in the Department of MNGT, MIS, and CS, University of Southern Indiana. His research interests include parallel algorithms, wireless ad hoc networks, and sensor networks.

Seong-Moo Yoo is an Associate Professor of Electrical and Computer Engineering at UAH. Before joining UAH, he was an Assistant Professor at Columbus State Univ., Columbus, Georgia – USA. He earned MS and PhD degrees in Computer Science at the University of Texas at Arlington. He was a Fulbright Scholar, Kazakhstan 2008-2009. His research interests include computer network security, wireless network routing, and parallel computer architecture. He has co-authored over 70 scientific articles in refereed journals and international conferences.



Fan Qiu is a Ph.D student of Computer Science at Vanderbilt University, Nashville, Tennessee, USA. He earned his MSE in Computer Engineering at the University of Alabama in Huntsville. His research interests are distributed systems and mobile ad hoc networks