RBR: REFINEMENT-BASED ROUTE MAINTENANCE PROTOCOL IN WIRELESS AD HOC NETWORKS

JainShing Liu,[1] and Chun-Hung Richard Lin [2]
[1] *Providence University, R.O.C.,* [2] *National Sun Yat-Sen University, R.O.C.*

Abstract In this paper, we propose a so-called refinement-based routing protocol that uses dynamic route redirection to provide proactive route selection and maintenance to on-demand routing algorithms so that the benefits of both types of routing algorithms can be combined and their drawbacks minimized. Experimental results demonstrate that adding the refinement-based routing protocol to AODV significantly reduces the number of broken paths and the end-to-end packet latency when compared with the pure on-demand routing protocol, AODV.

Keywords: route maintenance, ad hoc wireless network

1. Introduction

As the popularity of mobile computing increases, cooperative communication using wireless devices is attracting interest. A Mobile Ad-Hoc Network (MANET) is a collection of such mobile devices, denoted as nodes, without the required intervention of any centralized access point or existing infrastructure. Each node in the network is equipped with a wireless transmitter and a receiver, and can act as both a host and a router forwarding packets to other nodes.

An important issue for achieving efficient resource utilization in the network is how to update route information depending on a change of network topology and connectivity. Since mobility in MANET causes frequent, unpredictable and drastic changes to the topology, it is especially important for nodes to be able to adapt to such changes and find an efficient route between communicating source and destination. To provide routes in such dynamic environments, many routing protocols have been proposed over the last few years. These protocols can be broadly classified onto three categories, namely, proactive, reactive, and hybrid.

On the one hand, pro-active routing protocols are proposed as a means to dynamically adjust an on-going route when network topology changes, or e-

quivalently to provide different routes at different points of time so as to control the dynamics of a MANET. However, when providing such adaptability, a pro-active routing protocol uses a large portion of network capacity to keep the routing information current, which is an inefficient use of bandwidth to flood control messages to the network.

On the other hand, on-demand routing protocols only maintain the active paths to those destinations to which data must be sent and thus significantly reduce routing overheads. These have recently attracted more attention than the pro-active protocols. However, on-demand routing protocols accommodate route changes only when an active route is disconnected. They cannot adapt to the change of network topology even if another route with less hops becomes available by the movement of intermediate nodes, unless any link in-between is disconnected. To alleviate this problem, several preemptive-based routing protocols were proposed [1][2][3][4]. Among these, Preemptive Routing [1] is in principle, a preventive mechanism, which in advance initializes a route discovery when links of a path are likely to be broken, and then hands off the dangerous path to the shortest new path just found. Such an approach reduces a handoff delay experienced by the data packets if the path-break predicted actually happens. However, it is no help in reducing routing overheads because, whenever a path is suspected to be broken, this method will flood the route-request packets to the network. This may even increase such overheads.

Router Handoff [2] deals with the path-broken problem by finding an alternate node in the vicinity of a potential link break, and then handing off the route to this node. However, the approach involves no optimizations for an active path; that is, a router handoff cannot shorten a path since such an approach only replaces an intermediate node with another which connects the replaced node's upstream node and its downstream node, with stronger links.

OR2 [3] is an adaptive path tuning scheme for mobile ad hoc networks. This method reduces the hop count of an active route while data packets are sent without link disconnection. However, it targets the "one-hop path shortening" between nodes that are adjacent with a single intermediate node, and provides no "N-hop path shortening".

Recently, a so called P_rAODV protocol [4] is proposed as an extension to AODV [5]. This method invokes a path rediscovery routine from the source according to the received information about the path which is contained in the route reply packet or according to a warning message received at the source. Moreover, instead of monitoring the signal power of the receiving packet, it monitors the transfer time for the hello packets to predict the link break. However, when a source node receives warning messages and decides to rediscover the path, it still needs to flood the route-request packets to the whole network, increasing the possible routing overhead as that noted for the preemptive routing [1].

In contrast to the above research, "RBR" provides path optimizations that simultaneously reduce the routing overheads for a broken path and the end-to-end delays of an active route. To achieve the first goal of reducing routing overheads, when a path breaks, RBR does not flood the route-request packets into the network; instead, it repairs the broken path using only local broadcast messages. Regarding the second goal of decreasing delays, RBR progressively shortens a path by introducing a suitable redirector to the path with fewer hop counts. We have incorporated the two mechanisms into the AODV routing protocol. Experimental results demonstrate that adding refinement-based routing to AODV significantly reduces the number of broken paths and the end-to-end packet latency, with only small extra control overheads when compared with the pure on-demand routing protocol, AODV.

This paper is organized as follows. In Sections 2 and 3, two major mechanisms involved in RBR are introduced. Then, performance evaluation results are given in Section 4. Finally we draw up our conclusions in Section 5.

2. Passive Probe Route Redirection

This section describes the details of the proposed Passive Probe Route Redirection P-PR_2, the first component of the proposed Refinement-Based Routing Protocol. First, we introduce the concept of "vicinity" of two nodes and that of the "redirector" of a path in our RBR. Then, we explain the design of P-PR_2 on the basis of overheard data, recorded in a so-called Overhear Table without the aid of source route information.

Vicinity

To argue for the concept of "nearness" of two nodes and that of "redirector" of a path more formally for our RBR, we introduce the notions of vicinity and bypassed hop count based on the observation of the distance relationship between two nodes. For this purpose, let us define the following notations:

- $D_{(AB)}$: The distance estimated from B to A.
- $P_{(A)}$: The vicinity of A.
- $R_{uf}(A)$: The upper-stream adjacent node of A in relation to flow f.
- $R_{df}(A)$: The down-stream adjacent node of A in relation to flow f.
- $HC_f(K)$: The bypassed hop count of a Redirector K in relation to flow f.

In this context, we assume $D_{(AB)} = D_{(BA)}$. Based on this, B is said to be in the vicinity of A, or $B \in P_{(A)}$, implying that $A \in P_{(B)}$ as well. Furthermore, suppose that a flow, f, now traverses A, B, C and D in this order, and X

and Y are two nodes not in f. Using the above definitions, the relationship between the four nodes involved in f can be represented as $A = R_{uf}(B) = R_{uf}(R_{uf}(C))(= R^2_{uf}(C)) = R_{uf}(R_{uf}(R_{uf}(D)))(= R^3_{uf}(D))$.

As shown in Fig. 1, when C moves away from B, and $C \in P_{(B)}$ is no longer true, there is a possibility that X or Y can act as a redirector to repair the broken path. In this scenario, X may be $R_{df}(B)$ and $R_{uf}(C)$ while Y may be $R_{df}(B)$ and $R_{uf}(D)$. Comparing the two candidates, X and Y, it can be seen that X can only connect to the direct down-stream node of B, i.e., $C = R_{df}(B)$ while Y can connect to the more descendent down-stream node of B, i.e., $D = R^2_{df}(B)$. This can be represented more precisely as $HC_f(X) = 1$ and $HC_f(Y) = 2$. Consequently, if X and Y compete to be a redirector for the broken link, $\overline{BC}$, simultaneously, and B can choose Y as the result, then the need of reinitializing a route discovery will be avoided and a shorter repaired path, when compared with that done by X, can be obtained. From this, we can develop the scheme described in the next section.

Design of P-PR_2

Passive Probe Route Redirection is, in principle, a preventive, preemptive approach to deal with path-breaks. We set two design goals to P-PR_2: repairing a broken path, and minimizing the number of additional control packets. On the one hand, the first goal is obvious in the context of the aforementioned problem. On the other hand, as regards the second goal, we aim at a scheme that can result in only a small extra routing overhead for repairing a broken path. In particular, we do not allow a path to be broken, which wastes huge bandwidth when flooding routing packets to find a new path, whenever there exists at least one redirector, in the vicinity of a broken link which can repair it. This is an important consideration for an ad hoc network since nodes in the network need to reduce their power consumption whenever possible. We design a scheme in which control packets are transmitted only when a node determines that a path should be changed based on when the alarm is given that there is a link-break. We call the scheme P-PR_2 and it is the first component of RBR. In what follows, we first introduce the Overhear Table used in P-PR_2, and then discuss our solutions for the so-called Route Redirection Reply Storm Problem.

Overhear Table. In P-PR_2, each node makes use of its *Overhear Table*. This table contains information about the status of each neighbor overheard in promiscuous receive mode. In particular, without the aid of source route information, an overhear table under AODV can be implemented with the following fields:

- The identity of the overheard node.

- The overhear time. It aids the flush work triggered when the overheard data is out of date.
- The path source and the path destination: two fields used to identify a flow.
- The time to live (TTL). This represents the sequence of overheard nodes in the same route.
- The estimated distance: a value obtained with received power and the transmitted power announced by the sender. With this distance and the hop count derived from TTL, a node can decide on its preference to be a redirector.

With this table, let us now explain the fundamental messages passed among the nodes in a MANET, which involves three control messages (packets): P-PR_2_$RRREQ$ (Route Redirection Request), P-PR_2_$RRREP$ (Route Redirection Reply), and P-PR_2_$RRACK$ (Route Redirection Acknowledgement).

At first, when movement of an intermediate node or the destination causes a link to break, or when heavy traffic in an area makes a link useless, a node that uses this link to its next-hop, retransmitting RTS more than m_{rt} times, changes its state from normal to probe and locally broadcasts a P-PR_2_$RRREQ$. Upon receiving the P-PR_2_$RRREQ$, each neighboring node triggers its own P-PR_2 reply procedure. That is, the node that receives P-PR_2_$RRREQ$ first checks if itself can be a redirector by searching the initiating node and the downstream nodes in the same path, in its own overhear table. If both are found, the bypassed hop counts between the sender and the found downstream nodes, HCs, will be calculated, and the distances from the node to these downstream nodes, d_rs, and the distance from the node to the initiating node, d_s, will be all retrieved. These values are then used to decide the preference of this node to be a redirector in the path, which helps to solve the Route Redirection Reply Storm Problem to be discussed in the next section. According to the preference value, the node randomly delays for a period of time, and then sends a P-PR_2_$RRREP$ to the initiator. By receiving the P-PR_2_$RRREP$, the initiator decides which redirector will be its new next-hop, sending its decision with a P-PR_2_$RRACK$ to the redirector, which makes the redirector update its routing table and completes the redirection procedure.

Preventing Route Redirection Reply Storms. The ability of nodes to reply to a P-PR_2_$RRREQ$ on the basis of information in their Overhear Tables can result in a possible Route Redirection Reply "storm" in some cases. That is, when a node broadcasts a P-PR_2_$RRREQ$, each neighbor may attempt to send a P-PR_2_$RRREP$, thereby wasting bandwidth and possibly increasing the number of network collisions in the area.

This is illustrated by an example in Fig. 2, the same one given in Fig. 1 but with more details. As shown in this figure, when link $\overline{BC}$ is likely broken or congested, nodes X, Y, Z and W at this time will receive B's P-PR_2_$RRREQ$ for the path from S to I, and each has some overheard nodes cached for this route. Normally, they all attempt to reply from their own Overhear Tables and all send their P-PR_2_$RRREP$ at about the same time because they all receive the broadcast P-PR_2_$RRREQ$ at about the same time. Such simultaneous replies from different nodes may create packet collisions among some or all of these replies and may cause local congestion in the wireless network.

When a node puts its network interface into promiscuous receive mode, it can delay sending its own P-PR_2_$RRREP$ for a short period of time and listen to see if the initiating node begins using a redirected route first. That is, this node should delay sending its own P-PR_2_$RRREP$ for a random period, which is now obtained as

$$T = \delta \times (M - HC + \frac{d_s}{TR} \times r_1 + \frac{d_r}{TR} \times r_2), \quad (1)$$

where δ is a small constant delay, M is the maximum number of hop counts that can be bypassed, HC is the bypassed hop count, d_s and d_r are the distance to the initiating node, and that to the found downstream node, respectively, TR is the node's transmission range, and r_1 and r_2 are two random numbers between 0 and 1. This delay effectively randomizes the time at which each node sends its P-PR_2_$RRREP$. That is, in the probability sense, all nodes sending P-PR_2_$RRREP$ messages giving larger bypassed hop counts, HCs, will send these replies earlier than all nodes sending P-PR_2_$RRREP$ messages giving smaller HCs. In addition, a node's relative location, represented as (d_s, d_r), serves as a perturbation factor in this randomization. More precisely, a node with location near both the initiating node and the downstream node may reply sooner than the others.

Provided with the delay period, a neighboring node promiscuously receives all packets, looking for P-PR_2_$RRACK$ from the initiator of this P-PR_2_$RRREQ$. If such an ACK, destined to another node, is overheard by this neighboring node during the delay period, the node may infer that the initiator of the P-PR_2_$RRREQ$ has already received a P-PR_2_$RRREP$ giving an equally good or better redirected route. In this case, this node cancels its own P-PR_2_$RRREP$ for this P-PR_2_$RRREQ$.

In this example, Y receives the P-PR_2_$RRREQ$ and checks its Overhear Table, finding that it can overhear three nodes, B, C, and D, in this path. Y then chooses D as its next-hop for this path since it gives a larger bypassed hop count, 2, compared with that provided by C, 1. Given this value, 2, and two distances (to the initiator B, and the chosen next-hop, D, respectively), 150 and 180, Y can compute its delay period according to equation 6, sending a P-PR_2_$RRREP$ at the indicated time if no other replies are overheard during

this period. Additionally, since the hop count given by Y is larger than those provided by X, Y is very likely to send its reply successfully if its location is not too far from the initiator and the chosen next-hop when compared with that of its competitor, X.

For reference, the control message flow summarizing the salient features of P-PR_2 is given in Fig. 3. This figure mainly depicts the control messages exchanged between the initiator, B, and the successful redirector Y in the example. In addition, the original next-hop, C, is also shown, which explicitly indicates the possibility that C can also give a response for the P-PR_2_$RRREP$ even though this does not happen in this example.

3. Active Probe Route Redirection

This section describes the details of A-PR_2, the second component of RBR. The goal of A-PR_2 is that without the aid of source routing information, by using the Overhear Table only, it can automatically shorten a path if one or more of its intermediate hops becomes unnecessary. This is particularly useful when the path is unavoidably lengthened by P-PR_2 as the redirectors found at that time have their own HC_fs equal to 1 at most. The Active Probe Route Redirection (A-PR_2) is designed in such a way that even MANET is operating with a distance vector routing algorithm.

More precisely, like P-PR_2, A-PR_2 uses the same Overhear Table, filled with overheard path information obtained under the promiscuous receive mode, to check if a node itself can be a redirector to shorten an active path without the need for reinitializing a route-discovery procedure. That is, whenever a node discovers in its Overhear Table that there exists at least two nodes, say i and j, in the same path, having their TTLs, $TTL(i)$ and $TTL(j)$, with difference equal to or greater than 3, i.e., $|TTL(i) - TTL(j)| >= 3$, or $HC_f >= 3$, the node will trigger an Active Route Redirection Request, A-PR_2_$RRREQ$, relayed to the upstream neighboring node, say i, to redirect the next-hop of the route to this node. Upon receiving the A-PR_2_$RRREQ$, node i makes a decision as to whether the request should be responded to or not based on certain criteria. If the decision is positive, an A-PR_2_$RRREP$ will be sent to the requestor (redirector).

One of the possible criteria is the number of hop counts to be shortened in the path if several A-PR_2_$RRREQ$ messages are received from different nodes at the same time. In this case, a redirector providing a shorter path, i.e., a larger HC_f, is preferred. In addition, the criteria should be considered with those redirections that just took place in the near past in mind. That is to say, if node i changes its next-hop to a node, say k, as its new next-hop for a redirection request delivered by k already, the following request from the node

in question should be rejected as long as the request suggests a HC_f is not an improvement on the previous one.

Fig. 4 illustrates an example demonstrating the control flow in A-PR_2, which involves only two control messages: A-PR_2_$RRREQ$ and A-PR_2_$RRREP$. Let us assume that $A = R_{uf}(B)$, $B = R_{uf}(C)$, $C = R_{uf}(D)$, and $D = R_{uf}(E)$ in relation to flow f. Moreover, suppose that C' has overheard data packets transmitted from A to B and data packets transmitted from D to E, simultaneously. Based on the overheard information, C' determines that it may be a redirector between A and D; that is, $R_{uf}(C') = A$ and $R_{df}(C') = D$ may be established. In this case, this node, C', sends an A-PR_2_$RRREQ$ to A, and upon receipt of this request, A updates its routing table and sends an A-PR_2_$RRREP$ back to C' if the criteria mentioned previously are satisfied. In particular, to ensure that the downstream node D can also be connected to C', C' has the option to ask D to reply to the same A-PR_2_$RRREQ$, which is shown with dot-lines in the figure. Finally, when A-PR_2_$RRREP$ is received, C' can update its own routing table accordingly. Note that since this route is now changed, the refreshed routing information can be piggybacked from the initiating node toward the path endpoints to inform other nodes on the path of the fact that the path length may be changed, which is the same issue considered in P-PR_2.

4. Performance Evaluations

In this section, we report on theoretical analysis and simulation studies (with GloMoSim [6] in order to come to an understanding of the potential improvements in performance obtained by the proposed *Refinement-Based Routing*.

Theoretical Analysis

Some definitions and basic results are given first. Let the number of nodes in the network be N, the transmission range of each node be R, and the average path length of all possible traffic be $\bar{L}$. The average hop count from source to destination can be approximated as $H = \frac{\bar{L}}{R}$, and the number of hops to discover a route is $2H$ on average.

Overhead of repairing a broken link. Consider the control overheads that AODV may involve. Supposing that AODV's RREQ packets can reach all nodes in the network, the overhead of flooding such messages will be N. Further, if each of the H hops has the same broken probability and the number of routes affected by a link breakage is ψ on average, then the number of RERR packets for the link break can be estimated as $\psi \cdot \frac{1+2+,...,+H-1}{H}$.

In pure AODV, a broken link causes RERRs sent to the sources affected. Assuming all the sources initiate route rediscovery after receiving such R-

ERRs, the average number of control overhead, $Pkt^f_{AODV_p}$, and the average delay, $Del^f_{AODV_p}$, for the broken link can be obtained with $Pkt^f_{AODV_p} = Pkt^f_{RERR} + Pkt^f_{RREQ} + Pkt^f_{RREP} = \psi \cdot (\frac{1+2+,\ldots,+H-1}{H} + N + H) \approx \frac{\psi \cdot (3H+2N)}{2}$ and $Del^f_{AODV_p} = Del^f_{RERR} + Del^f_{RREQ} + Del^f_{RREP} = \frac{1+2+,\ldots,+H-1}{H} + H + H \approx \frac{5H}{2}$, respectively, where $Pkt^f_{RERR}(Del^f_{RERR})$, $Pkt^f_{RREQ}(Del^f_{RREQ})$, and $Pkt^f_{RREP}(Del^f_{RREP})$ are the number of RRER packets (the delay of RRER) sent to the sources, the number of RREQ packets (the delay of RREQ) sent to find new routes to destinations, and the number of RREP packets (the delay of RREP) replied to the sources, respectively.

Equipping AODV with local repair function results in the upstream node of a broken link being able to find the new routes to the destinations affected by itself. Thus, the control overhead and the corresponding delay can be reduced as $Pkt^f_{AODV_{lr}} = Pkt^f_{RERR} + Pkt^f_{RREQ} + Pkt^{f'}_{RREP} = \psi \cdot (\frac{1+2+,\ldots,+H-1}{H} + N + \frac{H}{2}) \approx \psi \cdot (H+N)$, and $Del^f_{AODV_{lr}} = Del^{f'}_{RREQ} + Del^{f'}_{RREP} = \frac{H}{2} + \frac{H}{2} = H$, respectively, where $Pkt^{f'}_{RREP}$ is the number of RREP packets replied to the upstream node, and $Del^{f'}_{RREQ}$ and $Del^{f'}_{RREP}$ are the delay to reach the destinations and the delay to reach the upstream node, respectively.

Now consider the control overhead of RBR for repairing a broken link. Unlike the AODV's approaches, which flag an error and re-initiate a route-discovery procedure at the source or at an intermediate node where the route is broken, P-PR_2 repairs a broken link with only local broadcasts to find a suitable redirector, and thus, its control overhead and the corresponding delay are future reduced as $Pkt^f_{RBR} = Pkt^f_{P-PR_2\_RRREQ} + Pkt^f_{P-PR_2\_RRREP} + Pkt^f_{P-PR_2\_RRACK} = 1 + 1 + 1 = 3$, and $Del^f_{RBR} = Del^f_{P-PR_2\_RRREQ} + Del^f_{P-PR_2\_RRREP} + Del^f_{P-PR_2\_RRACK} = 1+1+1 = 3$, respectively, where $Del^f_{P-PR_2\_RRREQ}$, $Del^f_{P-PR_2\_RRREP}$ and $Del^f_{P-PR_2\_RRACK}$ are regarded as the same and all given with 1 unit of time for analytical simplicity.

Overhead of shortening an active route. Consider the Automatic Route Shortening (ARS) mechanism of DSR at first. According to the Internet draft [7], in DSR, nodes promiscuously listen to packets, and if a node receives a packet found in the Flow Table but the MAC-layer (next hop) destination address of the packet is not this node, the node determines whether the packet was sent by an upstream or downstream node by examining the Hop Count field in the DSR Flow State header. If the Hop Count field is less than the expected Hop Count at this node, the node will adds an entry for the packet to its Automatic Route Shortening Table, and returns a Gratuitous Route Reply to the source of the packet.

According to this, the communication overhead and packet delay of ARS could be $Pkt^s_{DSR} = Pkt^s_{G_{Route_Reply}} = \frac{H}{2}$, and $Del^s_{DSR} = Del^s_{G_{Route_Reply}} = \frac{H}{2}$, where $Pkt_{G_{Route_Reply}}$ represents the number of Gratuitous Route Reply unicasting from the intermediate node to the source, and $Del_{G_{Route_Reply}}$ is the corresponding delay. On the other hand, RBR requires only three control messages to shorten an active path, and costs $Pkt^s_{RBR} = Pkt^s_{A-PR_2\_RRREQ} + Pkt^s_{A-PR_2\_RRREP} = 1 + 1 = 2$, and $Del^s_{RBR} = Del^s_{A-PR_2\_RRREQ} + Del^s_{A-PR_2\_RRREP} = 1 + 1 = 2$.

In above, we assume that if RBR causes path lengths changed in the run time, the routing protocol under consideration, e.g., AODV, can record this fact and piggyback these changes to the downstream nodes. Therefore, no additional control overhead should be considered beyond that of RBR itself.

Simulation

In this subsection, we simulate the RBR-enhanced method and the original AODV to obtain their possible performance differences in mobile environment. For an unbiased comparison, scenarios similar to the previous work [1] were simulated with and without the use of RBR. However, instead of only 35 nodes, we experimented using scenarios with a set of 100 nodes in an area of 2000 square meters. In addition to this, we created the following environment. The transmission range was 250 meters. The mobility adopted was the random waypoint model. In this experiment, MOBILITY-WP-MIN-SPEED was 0 meters/sec, the maximum speed, MOBILITY-WP-MAX-SPEED, of 10 meters/sec was taken as our low speed, and that of 20 meters/sec as our high speed. The MAC layer adopted was IEEE 802.11 designed to note a link-break in advance.

With above, we examined the impact of dynamic channel quality change caused by mobility to the performance of RBR. For this aim, every Constant Bit Rate (CBR) flow is randomly selected from among 35 nodes, for each speed (low, medium, and high). Each CBR source sent one 512-byte packet every second to its destination for a duration of 1000 seconds of simulation time. In total 100 simulations for each speed and each number of flows were carried out, and the performance metrics measured, that is, the end-to-end delay, jitter, throughput, control overhead, and number of path-breaks, were averaged to show any possible performance differences that may exist between the original AODV and our RBR.

Figs. 5 and 6 show the number of path break and the packet latency (end-to-end delay) for both mobility scenarios. As shown in these figures, the number of broken paths of RBR is only 0.33% (0.38%) of that of AODV in the low (high) mobility scenario. The end-to-end delay is improved by RBR up to 30% in both mobility scenarios. In addition, as shown in both figures, the

higher the mobility, the higher the number of broken paths and the higher the packet latency is.

Other information of the experiment is given in Figs. 7 and 8. Fig. 7 shows that the two methods provide similar throughputs. The maximum difference between these throughputs is no greater than 1%. Fig. 8 shows the control overhead, i.e., the number of control messages sent by each method. The control messages include routing packets (AODV's RREQ, RREP, and RERR), route redirection requests (P-PR_2_$RRREQ$ and A-PR_2_$RRREQ$), route redirection replies (P-PR_2_$RRREP$ and A-PR_2_$RRREP$), and route redirection acknowledgements (P-PR_2_$RRACK$). As shown in this figure, the number of control message of RBR is only half of that of AODV in both mobility scenarios. Observing Figs. 5 and 8, it is evident that RBR results in fewer path-breaks and greater savings in control overheads.

5. Conclusion

In this paper, we investigate adding proactive route selection and maintenance to on-demand routing algorithms to combine the benefits of both types of routing algorithms while minimizing their drawbacks [1]. Evaluation results show that adding the refinement-based mechanisms to AODV significantly reduces the number of broken paths as well as the end-to-end packet latency when compared with the pure on-demand routing protocol, AODV.

References

[1] T. Goff, N.B. Abu-Ghazaleh, D. S. Phatak and R. Kahvecioglu. "Preemptive routing in ad hoc networks," Proceedings of ACM MobiCom, 2001.

[2] Srianth Perur, Abhilash P., Sridhar Iyer. "Router handoff: A preemptive route repair strategy for AODV," IEEE Intl. Conference on Personal Wireless Computing, December 2002.

[3] Masato Saito, Hiroto Aida, Yoshito Tobe, Yosuke Tamura and Hideyuki Tokuda. "OR2: A path tuning algorithm for routing in ad hoc networks," Proceedings of IEEE Conference on Local Computer Networks, pages 560-567, Nov. 2001.

[4] A. Boukerche and Liqin Zhang. "A preemptive on-demand distance vector routing protocol for mobile and wireless ad hoc networks," The 36th Annual of Simulation Symposium, pages 73-80, April 2003.

[5] Charles Perkins and Elizabeth Royer. "Ad hoc on-demand distance vector routing," Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications.

[6] Global Mobile Information Systems Simulation Library. http://pcl.cs.ucla.edu/projects/glomosim.

[7] David B. Johnson, David A. Maltz and Yih-Chun Hu. "The dynamic source routing protocol for mobile ad hoc networks (DSR)," IETF MANET Working Group INTERNET-DRAFT, draft-ieft-manet-dsr-09.txt, 15 April 2003.

[1]This work was supported by the National Science Council, Republic of China, under grant NSC92-2213-E-126-004.

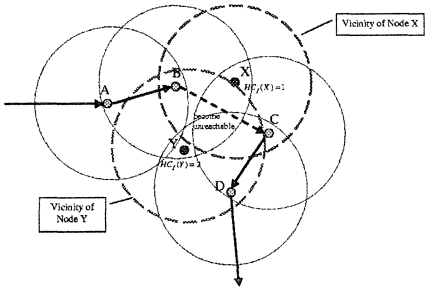

Fig. 1. Vicinity of node

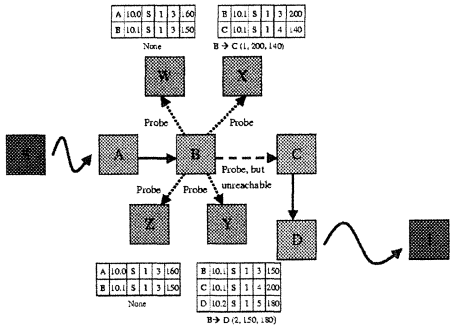

Fig. 2. Route redirection reply storm

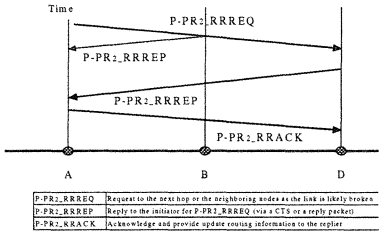

Fig. 3. P-*PR2* control message flow

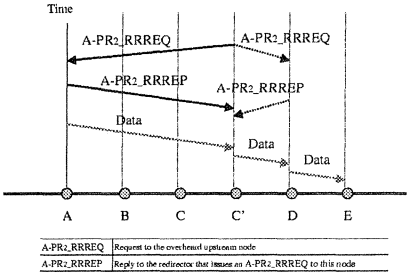

Fig. 4. A-*PR2* control message flow

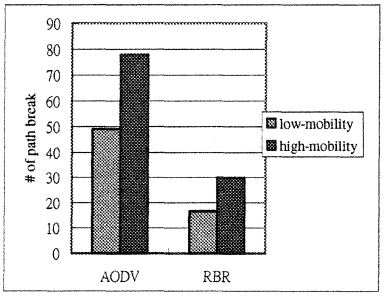

Fig. 5. The number of path break

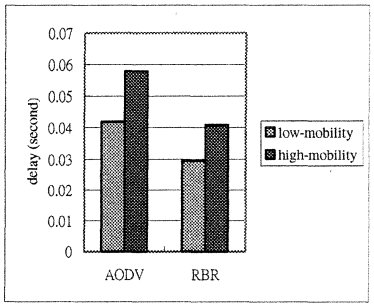

Fig. 6. The end to end delay

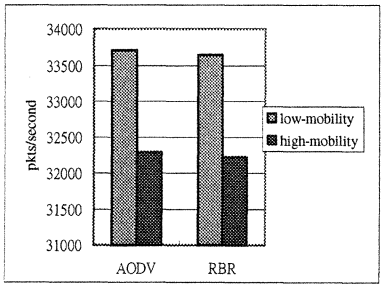

Fig. 7. The throughput

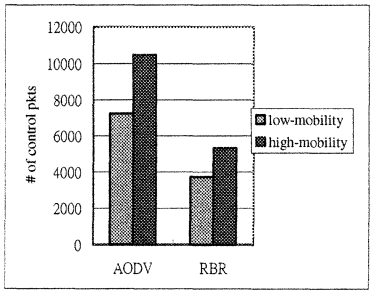

Fig. 8. The number of control message