2012 26th IEEE International Conference on Advanced Information Networking and Applications

Efficient Route Discovery and Repair in Mobile Ad-hoc Networks

Po-Jen Chuang, Po-Hsun Yen, and Ting-Yi Chu
Department of Electrical Engineering, Tamkang University
Tamsui, New Taipei City, Taiwan 25137, R. O. C.
E-mail: pjchuang@ee.tku.edu.tw

***Abstract*—Building proper routing protocols for a Mobile Ad-hoc Network (MANET) is a critical challenge because (1) flooding route requests often results in broadcast storm (especially when nodes or connections increase) and (2) re-broadcasting route discovery packets to rebuild or repair a broken path (caused by node mobility) will consume more control packets. To solve such broadcast storm and path damage problems, this paper mixes the ideas of multipoint relaying and 2-hop route repair into a new routing protocol suitable for large-scale MANETs. Simulation results show that the new protocol can effectively repair damaged routes at reduced bandwidth consumption.**

Keywords-mobile ad-hoc networks (MANETs); on-demand routing protocols; local repair; multipoint relaying.

I. INTRODUCTION

How to build efficient and desirable transmission routes for resource-limited MANETs is a crucial issue. Most routing protocols focus on route validation and transmission delay. For reactive routing protocols which flood route requests, broadcasting can be particularly critical as massive broadcast packets – resulting from increased hops and connections – may seriously degrade network efficiency. For improvement, this paper combines both the broadcast mechanism and routing protocol to form stable transmission routes for large-scale, high-mobility MANETs.

The basic idea is incorporating the concept of *Multipoint Relaying* (MPR) [1] into a properly modified AODV routing protocol [2]. In MPR, each node will periodically broadcast neighbor nodes, build a two-hop adjacent node list, and based on the list calculate the two-hop connected dominating sets (TCDS) – when to broadcast a route request. Receiving the request, only nodes in TCDS need to re-broadcast it. Bandwidth consumption is thus saved. As nodes in a highly mobile MANET tend to change positions, an existing route may soon fail due to invalid information. To repair the route, AODV and MPRDV [3] choose to re-send route requests (RREQ) – which is indeed inefficient for a highly mobile network. AOMDV [4] adopts a different multi-path approach. When sending a RREQ to build a route, it also gives the position of the first node that receives the RREQ so that a middle node can build an independent alternative path – through which transmission can be carried on whenever a current path fails. This *multipath* approach may save route rebuilding time and yet degrade network performance because (1) the alternative path can fail easily when path length grows and (2) the source node also needs to send the dropped packet by the alternative path. MMDV [5] tries to add the MPR broadcasting into AOMDV to raise the packet arrival ratio by multiple paths but takes more control packets than AODV in building routes.

To attain more efficient transmission for a highly active MANET, we bring in *the local repairing mechanism* to fix and work with MPRDV. The main purpose is to reduce the number of broadcast packets effectively and repair damaged paths quickly. The key design of our scheme is as follows. When sending a RREQ or RREP (route reply), a middle node will include the position of the last broadcasting node and record the positions of the last two nodes. After a path is constructed, if a node along it gets invalid, we can locate alternative hops from the nodes in the two-hop adjacent list and notify the last broadcasting node of the next two valid hop positions – to repair the path. Such a local repairing mechanism can ensure path validity when nodes become invalid, without having to rebroadcast a RREQ. Major advantages of our new protocol include (1) rapid path damage detection and instant repair, (2) reducing the needed packet amounts and (3) lifting repair success ratios.

II. BACKGROUND STUDY

2.1 Ad-hoc On-demand Distance Vector routing (AODV)

Composed of Route Request, Route Reply and Route maintenance, AODV [2] lets nodes build/repair a path only when necessary – to reduce the extra cost of building routes in a dynamic network topology.

- ***Route request***

To send a packet to a destination, a node first searches its routing table for usable information. If there is valid information, the node will send out the packet with the next hop indicated in the table; otherwise, it will flood the RREQ packet. A RREQ carries source IP, destination IP and broadcast ID (each node maintains a broadcast ID which will increase each time when a RREQ is sent – to mark the event). After sending a RREQ, the source will set a timer and wait for a RREP. Sending a RREQ creates not only a route to the destination but also a reverse route for the destination to return the RREP to the source.

- ***Route reply***

1550-445X/12 $26.00 © 2012 IEEE
DOI 10.1109/AINA.2012.120

In sending a RREQ, if there is a valid route to the destination in the routing table of a middle node, the middle node will return a RREP to the source; otherwise, the destination must return the RREP by the end. Different from a RREQ, a RREP is transmitted in unicast, i.e., the middle nodes will build a *forward route* to the destination. After the source receives the RREP, it then transmits the data packet by the built forward route to the destination.

- ***Route maintenance***

Route maintenance includes maintaining information in the routing table and managing the route breaking problem. Maintaining the information in the routing table can be easy: When a path is not used or updated in a period time, simply remove it from the table. When a route breaks, two measures can be taken. One is to return a RERR (route error) to the source and clear the information of the broken route along the way. Receiving such a RERR, the source will attempt to rebuild a new route to the destination. The other is to make use of the local repair concept. If the broken point nears the destination node, the middle node (at the broken point) will broadcast the RREQ to search for an alternative route. If the destination finally gets the RREQ, it will send back a RREP and meanwhile complete the route repair job, to save route reconstruction time.

2.2 Multi-path Routing Protocols

A multi-path routing protocol is designed to balance the overhead of each route so as to maintain the quality of service. It uses the concept of fault tolerance to handle the route information change/destruction problem which results from node mobility in a MANET. The main idea is to build multiple paths between a pair of source-destination nodes during route searching and replying. Thus, when a current path fails, packets can be sent via an alternative path – to save the cost of searching for a new route. A number of multi-path routing protocols have been introduced in recent years [4-11]. The Ad-hoc On-demand Multipath Distance Vector routing (AOMDV) [4] stands as a major example.

AOMDV searches a route by the sequence number in a RREQ – like AODV – but aided by two more parameters, the hop counts and the id of the first adjacent node to receive the RREQ first. Besides, the next hop in the routing table is replaced by a path table. When a node sends a RREQ, the first receiving neighbor node will put its own id into the RREQ and broadcast further on. When a middle node receives the RREQ, it will compare the hop counts in the RREQ with its own hop counts to see if it has ever received this RREQ from the source. If not and its hop counts are bigger than the hop counts in the RREQ: The middle node will record this path information and transfer the RREQ to the next hop. Otherwise, the hop counts will be changed by the smaller hop counts. Through this process, a middle node will be able to record many loop-less paths. Meanwhile, the destination node will reply a RREP to all

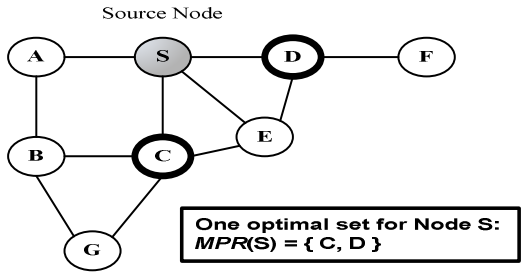

Fig. 1. An MPR set selection example.

received RREQ and eventually to the source node by these different paths – a multi-path route is then completed.

After the route is built, the path with the least hops will be taken as the main transmission path. If a node detects that it gets off the transmission range from an adjacent node, it will check if there exist any paths with invalid next hops in the routing table: If yes, it will delete such paths. When a node realizes the alternative paths of a transmission route are all deleted, it will broadcast a RERR packet together with the destination id. Any node receiving the RERR will check if it has the route items specified in the message. If yes and the next hop of any path in the route item is also the source node of a RERR, delete the path. The same approach will be employed to see if any other route runs out of valid alternative paths: If so, send out a RERR again.

AOMDV is able to pick up an alternative path from the built multiple paths to continue packet transmission when encountering path damages. It reduces packet drop ratios and route repair time. There are also disadvantages. For instance, the destination node needs to reply a RREP to each RREQ, consuming more control packets than AODV in a single route search. It also produces higher network congestion because each node needs to save multiple (but mostly unused) paths for alternative routing and for sending a RERR when detecting any invalid path (to prevent upper nodes from using it). Besides, its required RERR packet is longer and the delivery ratio is higher.

2.3 Local Route Repair Mechanisms

Route repair is another way to restore a transmission path which fails due to node mobility. AODV has an original but simple route repair mechanism: If a node detects the next hop of a transmission path breaks, it will start route repair only when the hop counts in the routing packet exceeds the hop counts between itself and the destination. The repair process is as follows. The node rebroadcasts a RREQ to the destination: If not receiving a RREP in a certain period of time, it will drop the packet and send a RERR to the source node. This process has apparent problems. By flooding a RREQ, all nodes (except the destination and the node that replies the RREP) will receive and transfer the RREQ. The needed bandwidth comes close to that for route reconstruction. Such a route repair approach may raise packet arrival ratios in some circumstances but

consume more bandwidth and degrade network performance when network sizes grow. A number of routing protocols with local repair mechanisms have been introduced in recent years [12-17] to fix the problem. PATCH [12] is a typical example. In PATCH, when a node encounters a broken path, the TTL of a RREQ will be set to 2 (to generate a good chance for finding the original next 2-hop – if the next hop is not broken – and reduce the RREQ and RREP amounts).

2.4 The MANET Broadcast Issues

As mentioned, a MANET contains a large number of mobile nodes that communicate with each other within valid communication ranges. In such a network, packets must be routed in multi-hops to save the limited radio power, channels and energy. If routing protocols for a MANET adopt periodical broadcasting to find/maintain/update transmission routes, the involved large quantity of packets may produce remarkable extra cost and degrade network performance. *Spontaneous* and *unreliable* are two distinct features for the MANET broadcast – because (1) without previous infrastructures or synchronous mechanisms, nodes never know when they will receive information broadcast by other nodes and (2) without any *acknowledge* mechanisms, a packet sender may lose trace of the packet when a receiver node gets disconnected actively or passively.

- ***The broadcast storm problem***

Flooding – any node receiving a packet will instantly re-broadcast it – is the basic way of direct broadcasting. If flooding becomes the broadcast mode for a MANET, it may generate huge rebroadcast cost and intensive channel competition, especially in large-sized networks. Among schemes to solve the broadcast storm problem, the probabilistic scheme reduces superfluous broadcasting by the set probability, the distance-based scheme uses the distance between two nodes as the threshold to decide whether to rebroadcast a received message, and the location-based scheme allows each node to control the broadcasting range by using GPS.

- ***Multi-Point Relaying (MPR)*** **[1]**

MPR is an efficient design among existing broadcast mechanisms. The main concept is to select a least set from 1-hop neighbors able to contain 2-hop neighbors, and add the set in the broadcast packet. To calculate the smallest set is a NP-complete problem, so MPR makes use of the greedy algorithm. If N(x) = the 1-hop neighbor set, N2(x) = the 2-hop neighbor set and MPR(x) = the selected MPR set, the algorithm can be calculated as follows:

1. Start with an empty set MPR(x)
2. First select any one-hop neighbor node in N(x) which is the only neighbor of some node in N2(x) as the multipoint relay. Add these one-hop neighbor nodes to the multipoint relay set MPR(x) and remove nodes from N2(x) which are now covered by nodes in MPR(x).
3. While there still exist some node in N2(x):

(a) For each node in N(x) which is not in MPR(x), compute the number of nodes in N2(x) which the node can cover

(b) Add a node of N(x) to MPR(x) which covers the maximum number of nodes in N2(x). Remove nodes from N2(x) which are now covered by nodes in MPR(x).

The MPR set can be thus obtained. In Fig. 1, S is the source node; A, C, D and E are 1-hop neighbors; B, F and G are 2-hop neighbors. S initially puts C, D into MPR(S) and removes all 2-hop neighbors they can cover (i.e., B, F and G here). When broadcasting a message, S adds C and D into the packet. After A, C, D and E receive the packet, only C and D need to rebroadcast it, but when rebroadcasting completes, all nodes in the network will receive the packet. In this case, MPR takes only 3 broadcastings to finish a job that flooding will need 6 broadcastings.

III. THE PROPOSED ROUTING PROTOCOL

3.1 Maintaining Neighbor Tables

Nodes will *periodically* broadcast the following Hello Messages to each other to build the 2-hop neighbor tables.

(1) *All Neighbors*: including all current 1-hop neighbors (based on it, a node will build its 2-hop neighbor table)

(2) *Neighbors Deleted*: giving the disconnected neighbors detected by the sender

(3) *Neighbors Unchanged*: indicating the current neighbor table for the sender remains unchanged (so the neighbor count is set as 0 and the neighbor IP address is null).

A Hello message also carries the sequence number of the sender so that a receiver can update its route table and save a new round of route searching when routing a packet to the sender.

The neighbor table in our protocol actually uses the 1-hop tuple and 2-hop tuple to indicate the neighbor relationship. Fig. 2 shows the neighbor table of node A and the Hello messages of nodes B, E, D, and G. Assume that each node receives at least one Hello message from a neighbor node, and there is no topological change. Now, when node A receives a Hello message, it adds the source node into its 1-hop tuple and builds the 2-hop tuple with marked neighbors in the Hello message. It will not add nodes in the 1-hop tuple into the 2-hop tuple. For example,

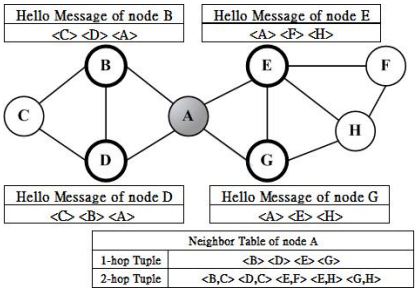

Fig. 2. An example of a 2-hop neighbor table.

in the Hello message of node B, as D is the 1-hop tuple for A, A will record only <B, C>.

3.2 Route Searching and Replying

Our RREQ is basically similar to that of AODV, except that ours encloses the previous two-hop IP address and the forward node IP addresses. When a source node issues a new RREQ, it will record NULL in the previous two-hop IP address and add its ID into the originator and the previous hop IP addresses, calculate the nodes which need to rebroadcast with MPR and the 2-hop neighbor table, and add these nodes into the forward node IP addresses. Any node receiving the RREQ will check if this sender is in its routing table: If not, update by the information and build a reverse route. After updating route information, a node checks the ID: If having received the same RREQ or its own ID is not in the forward node IP addresses, drop the packet; otherwise, record the sender ID into the previous two-hop IP address, generate new forward node IP addresses and rebroadcast the RREQ until the destination node receives it or until TTL$\leqq$0 in the RREQ.

After the destination receives the RREQ, RREP will be sent back to the source by unicast. Our RREP resembles that of AODV except an additional field – the previous two-hop IP address. That is, the destination will record NULL in the previous two-hop IP address and add its ID into the destination and the previous hop IP addresses before forwarding the RREP packet.

3.3 Route Maintenance and Repair

Each node will periodically maintain the already-built routes. Besides the original items of AODV, we have two more items – *last_time_used* and *largest_hops_forwards*. Nodes learn about route damage by two ways: the return report of link layers and the periodical update of neighbor tables. Our routing protocol deals with the link layer issue only – because the return report of the link layer carries more instant and urgent message: An invalid node appears right here in this very spot of this transmission route.

In our protocol, when a node receives ACK from a link layer and realizes the current packet transmission fails, it will check the ID of the next hop, delete it from its 1-hop and 2-hop neighbor tables, and check if the packet is a data packet. If not, drop it; if yes, return it to the route layer and start to repair the route. Conditions for repairing a route:

(1) The current route does not expire or is not under repair.
(2) The next 2-hop neighbor (N) must be valid and at least one of the 1-hop neighbors connecting to N must be valid.
(3) The next-hop node is not the destination node.
(4) The value of last_time_used must be less than 1.5 times packet delivery period earlier than the current time.

If a damaged route fits all the conditions, start the 2-hop repair process. Otherwise, check the largest_hops_forwards: If the value is bigger than the hop counts to the destination, employ the AODV repair mechanism; if smaller, invalidate the route, generate a RERR packet which records all invalid destinations and broadcast the packet (same as AODV).

To conduct a 2-hop repair, we (1) get a 1-hop neighbor which has the longest expiration time and connects to the 2-hop neighbors (based on the next 2-hop neighbor tables), (2) generate a RPRQ carrying the destination ID and the node's sequence number, and (3) send the packet to the next 2-hop node via the picked 1-hop neighbor. Receiving such a RPRQ, a node will check if the destination is itself. If not, add its ID into the previous hop IP address and broadcast the packet. If yes, generate a RPRP, fill in route repair information and send the packet to the RPRQ originator via the next-hop nodes. When facing interrupted transmission or invalid route repair information (in the RPRQ), a node will return a RPF (repair failure) – not a RPRP – to the source.

Receiving a RPRP, a node will check if the destination ID matches its own. If not, update route information based on the route repair information in the packet, add its ID into the previous hop IP address and broadcast the packet. If yes, the node recorded in the previous hop IP address of this RPRP becomes the new next-hop node, update this route following the route repair information in the RPRP and generate a RTCH (route change) packet. Add the failing next-hop node, the new next-hop node and the updated route information into the packet. Broadcast it to notify neighbors that this route has been fixed. After neighbors receive the RPRP, the 2-hop repair operation is completed.

If a node sends out a RPRQ but receives no RPRP or RPF in a period time, it will look into the hop counts: If the value of largest_hops_forwards exceeds the hop counts between itself and the destination, repair the route as AODV; otherwise, generate a RERR and broadcast it.

A complete 2-hop repairing process of our protocol is given in Fig. 3. As we can see from the original network state, there is a connection between nodes S and D (a), and this connection breaks when node E moves (b). Node B detects this disconnection and starts the 2-hop repair mechanism. It first searches the 2-hop neighbor table to locate the next 2-hop node G and sends a RPRQ to G by F (c). After receiving the RPRQ, G returns a RPRP to B by F (F then records the new route to D). When B receives the RPRP, it updates F as the new next 1-hop neighbor to D, and broadcasts a RTCH to notify A of this change. Node A then picks F (instead of E) as its next 2-hop neighbor to D (d), and completes the 2-hop route repair (e).

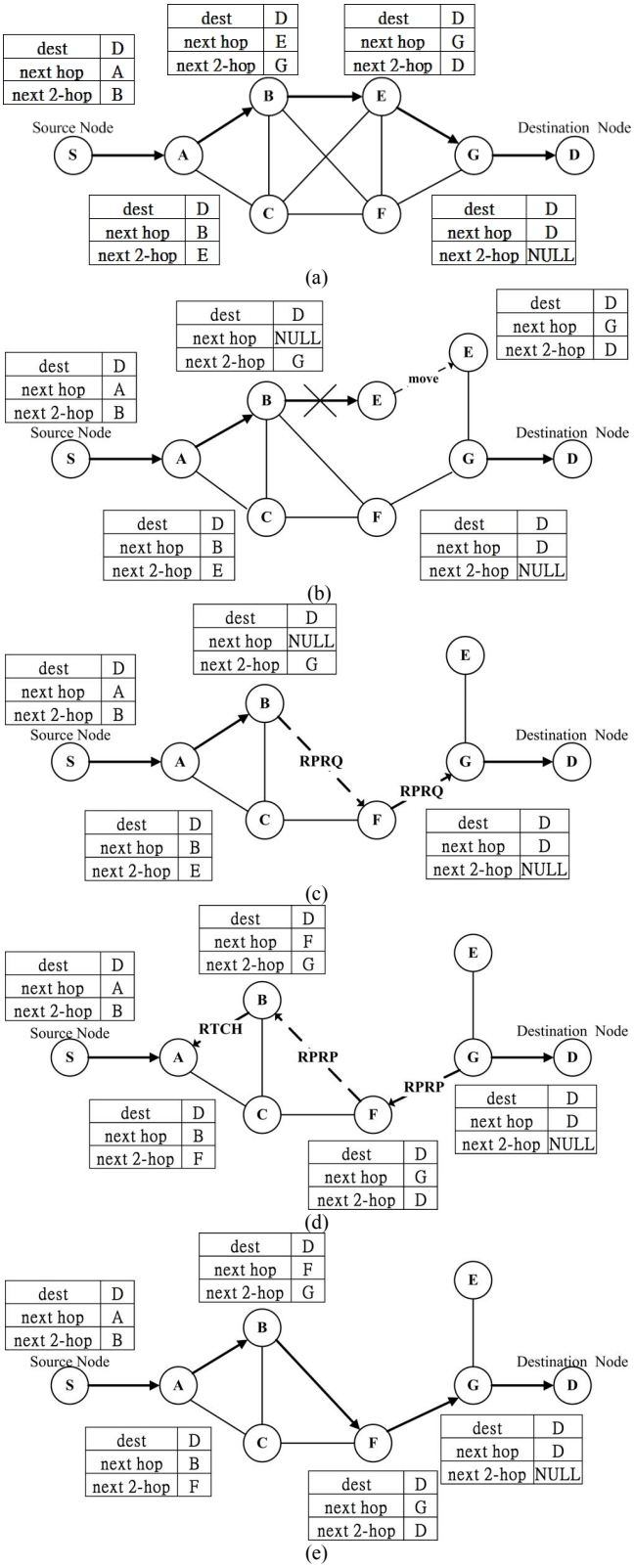

Fig. 3. An Example of 2-hop repair.

3.4 The Advantages

- ***Rapid damage detection and instant repair***

The repair example in Fig. 3 illustrates how to recover a damaged routing path using the 2-hop neighbor tables built by MPR broadcasting. As packet transmission is done by unicast, the ACK mechanism will enable a node that sends an unsuccessful RPRQ to learn quickly about the abortive repair attempt. If transmission is done by broadcast, an abortive repair attempt will take some time (the set broadcast cycle) to surface. Besides, when a large number of nodes detect route disconnection and simultaneously start the repair process, the overflow of packet broadcasting may intensively influence network performance.

- ***Reducing packet amounts***

As our protocol sets the destination of a RPRQ to be the next 2-hop neighbor in the route request, a RPRQ will receive only one RPRP. If we send a RPRQ as AODV-ABR [13], we may get numerous RPRP packets (from all neighbor nodes) with the requested route information or repair failure information – due to node dislocation or loss of route information. (When node dislocation happens, we can still proceed with the repair process because our RPRP carries route information.)

- ***Enhancing repair chances***

Allowing a node to update the next 2-hop node by RTCH messages creates better route repair chances. For instance, OPTAODV [14] fixes a damaged route fast but when a new pre-hop node (which knows nothing about the original pre-2-hop node) disconnects from its upstream node, it is unlikely to restore the route. Neither can OPTAODV repair a reverse route built by RPRQ because it is based on the pre-2-hop information built by RREP.

IV. SIMULATION RESULTS

Simulation is carried out – using Network Simulator-2 version 2.33 [20] and parameters in TABLE I – to evaluate the performance of routing protocols, including AODV, AOMDV, MPRDV, MMDV and our protocol. Performance measures of interest include *packet delivery ratios*, *average hop counts*, *delay per hop* and *control overhead*.

TABLE I. Parameters used in the simulation.

Parameters	Value
MAC Protocol	802.11b
Transmitter Range	250 m
Bandwidth	11Mbps
Simulation Time	100
Number of Nodes	100
Scenario Size	300 x 3000 m^2
Traffic Type	Constant Bit Rate
Packet Size	512 bytes
Rate	1 Packets/s
Movement Model	Random Waypoint
Pause Time	10 s
Route Buffer Size	64 Packets
Interface Queue Size	100 Packets

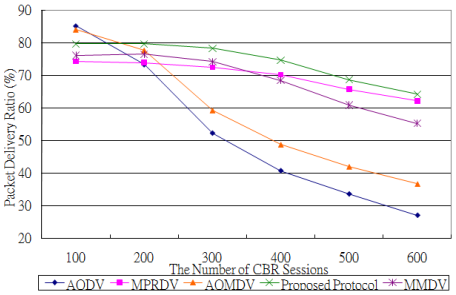

Fig. 4. *PDR* vs. session counts.

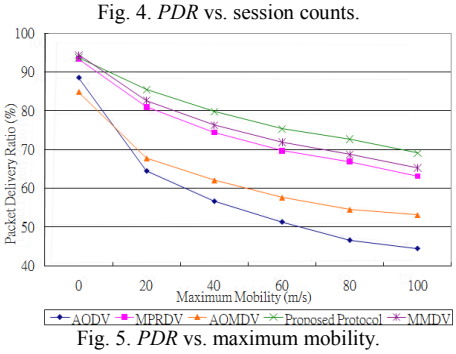

Fig. 5. *PDR* vs. maximum mobility.

● ***Packet delivery ratios (PDR)***

PDR = the total CBR packets received by all destinations ÷ the total CBR packets delivered from all sources (i.e., the ratio that a packet is successfully sent from the source to its destination). Figs.4 and 5 depict *PDR* of each protocol for different CBR session counts and maximum mobility. In Fig. 4, when CBR session counts increase, *PDR* decreases for protocols not incorporated with MPR because the rapidly increased RREQ amount congests the network. Our protocol produces higher *PDR* than MPRDV and MMDV because our 2-hop repair mechanism can repair all current routes. In Fig.5, all protocols yield nearly the same *PDR* at node mobility = 0. When node mobility increases, the advantage of employing MPR appears: Significantly reduced control packet amounts and route searching time – along with shorter new routes – brings higher *PDR*. Our protocol performs the best due to its ability to repair most damaged routes at small cost. In fact, our 2-hop neighbor tables can fix route damages even if all original nodes of a route run out of the communication range of the to-be-repaired node.

● ***Average hop counts (AHC)***

AHC = total hop counts travelled by all packets that reach destinations successfully ÷ the number of these packets.

Figs.6 and 7 give *AHC* at different session counts and mobility. At session count = 100, protocols using MPR take fewer *AHC* than those not using it – because original routes built by MPR broadcast are usually shorter than those built by flooding. When session counts increase, *AHC* decreases for AODV and AOMDV because packet congestion results in packet dropping in longer routes. By contrast, *AHC* decreases in a lesser way for protocols using MPR broadcast as they suffer less from congestion and packet

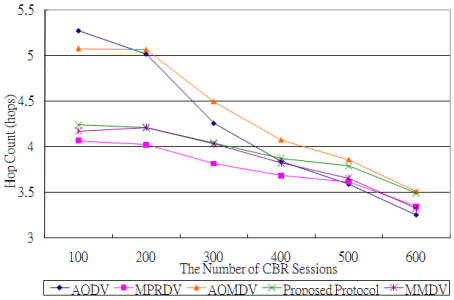

Fig. 6. *AHC* vs. session counts.

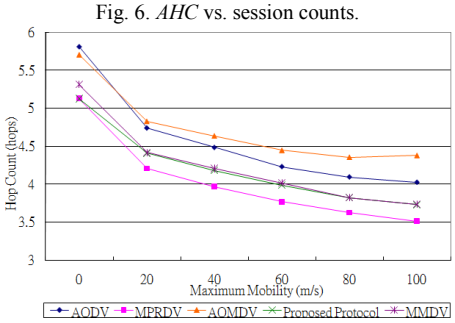

Fig. 7. *AHC* vs. maximum mobility.

dropping. In Fig.7, protocols using MPR broadcast yield quite close results and so are protocols without MPR. When mobility grows, AOMDV, MMDV and our protocol depict higher *AHC* due to higher packet arrival rates, whereas AODV and MPRDV keep lower *AHC* because they are more likely to rebuild routes which get close to the shortest paths.

● ***Delay per hop (DPH)***

DPH = the average end-to-end delay of a successfully delivered packet ÷ the average number of hop counts the packet has travelled. Packet delay evaluation usually refers to the average time for completing a successful packet transmission. This is nevertheless an unfair measure for protocols with good route repair ability and high *PDR* – because protocols able to fix routes rapidly usually take longer to transmit the *rearranged* packets (which tend to have larger hop counts). To attain better evaluation, we mind also the total hop counts, i.e., we take into account both the end-to-end delay and *DPH*.

Fig.8 gives the *end-to-end delay* at varied session counts. It shows that AODV and AOMDV yield shorter delay time at smaller session counts, but when the counts grow, delay

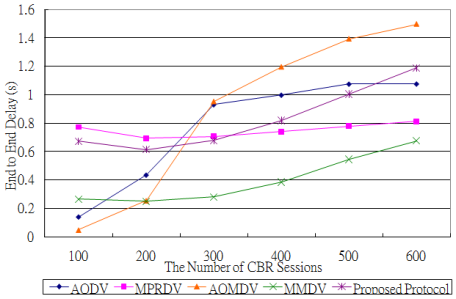

Fig. 8. The end-to-end delay vs. session counts.

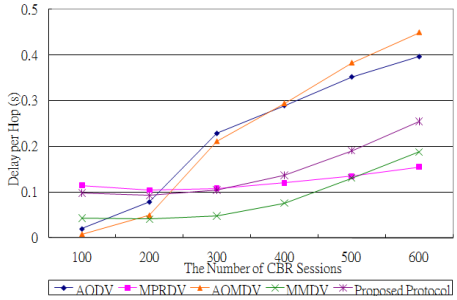

Fig. 9. *DPH* vs. session counts.

rises fast because they do not use MPR broadcast to reduce RREQ amounts. AOMDV yields distinctly longer delay at larger session counts as it needs extra packets to build multiple paths (due to obvious congestion). The same trend can be found for MMDV and MPRDV. Our protocol has shorter delay than MPRDV at small session counts, thanks to its efficient repair design which shortens the repair-waiting duration. For us, packet arrival rates and packet delay both grow with session counts – a normal result from large numbers of packets and network congestion.

Note that, between 500 and 600 sessions, delay time decreases (instead of increasing) for AODV. Fig.9 which gives *DPH* at different session counts offers a reasonable explanation. The result depicts a similar trend as Fig.8 for each protocol except AODV. *DPH* for AODV keeps rising when session counts grow because the excessive control packets AODV needs will lead to congestion and also packet dropping. As packets with longer transmission routes are more likely to suffer route damage and hence need repair, they will produce longer *DPH* than packets with shorter paths. The value of *DPH* indeed duly reflects packet delay caused by network congestion and route repair.

Fig.10 gives *DPH* at different mobility. When mobility = 0, all protocols hold close *DPH* and those without MPR are

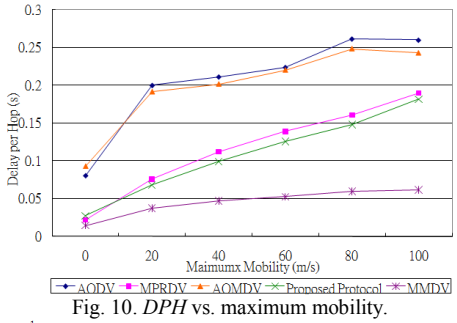

Fig. 10. *DPH* vs. maximum mobility.

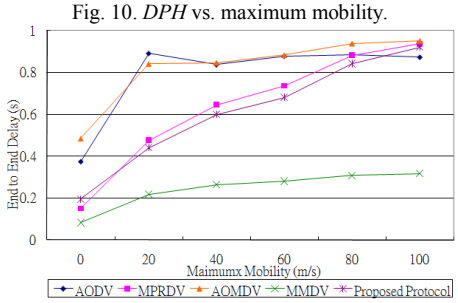

Fig. 11. The end-to-end delay vs. maximum mobility.

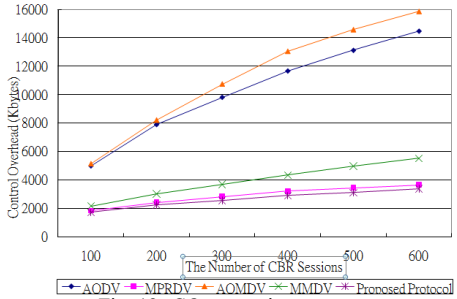

Fig. 12. *CO* vs. session counts.

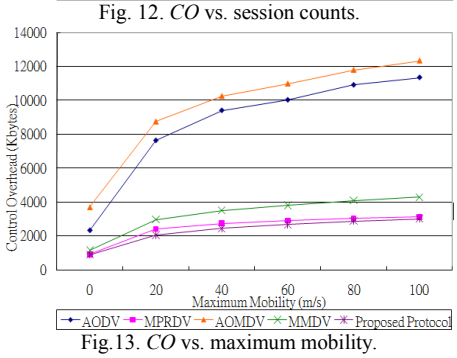

Fig.13. *CO* vs. maximum mobility.

with higher values because they need more broadcast packets. When mobility increases, the benefits of multipath designs start to show, especially for MMDV which has the least *DPH*. Our protocol also performs well. When mobility grows, we maintain higher packet arrival rates with shorter *DPH* than MPRDV – showing our 2-hop repair mechanism works better than the original repair mechanism in AODV. The *end-to-end delay* for different mobility in Fig.11 shows that at mobility = 100, AOMDV has longer delay than AODV but similar delay as MPRDV and our protocol – revealing the end-to-end delay is not a proper delay indicator and meanwhile justifying our adoption of *DPH*.

- ***Control overhead* (*CO*)**

CO = the total bandwidth consumed by control packets. The amount of control packets is the key indicator of extra overload. To attain fair evaluation for protocols without MPR, we consider the bandwidth consumption (instead of broadcast times) for total control packets in broadcast and unicast transmissions. Fig.12, which gives the results at varied session counts, shows that protocols with MPR yield much less *CO* than those without and the gap grows even larger at bigger session counts. For two protocols using the same broadcasting, the one with multipath consumes more bandwidth than the one without – because in building main routes, a multipath protocol also builds alternative routes. Our protocol needs less *CO* than MPRDV because we repair most damaged routes by fewer and shorter control packets sent by unicast, while MPRDV needs to broadcast a large number of RREQ to repair/rebuild routes.

Fig.13 gives *CO* at varied mobility, with bandwidth = 11Mbps. When mobility = 0, AOMDV needs more *CO* than AODV because even at mobility = 0, its link layer may mistake packet collision as transmission failure and decide

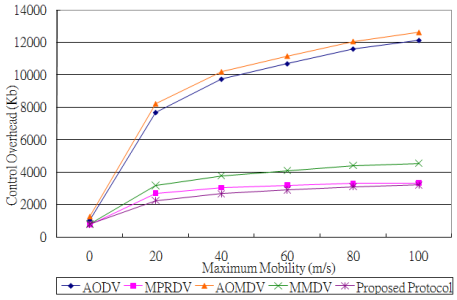

Fig. 14. *CO* vs. maximum mobility with 54Mbps.

to send RERR. Fig.14 displays *CO* at varied mobility, with bandwidth = 54Mbps. Here, the overhead difference for AOMDV and AODV lessens (in Fig.13, AOMDV faces worse congestion and more RERR). For MMDV and MPRDV, the overhead difference remains the same – because, with MPR, MMDV can efficiently reduce network congestion. When mobility increases, our cost comes closer to that of MPRDV. This is expectable as increased mobility will generate more broken routes and our protocol tends to repair all emerging broken routes, hence consuming more bandwidth. For MPRDV, there are more chances to rebuild a new route – with fixed cost – when facing route damages, which makes its *CO* less affected by mobility change.

V. CONCLUSIONS

This paper presents a desirable new routing protocol to attain efficient transmission for a highly active MANET. Based on MPR broadcast and a 2-hop route repair mechanism, our protocol can reduce the bandwidth cost due to route searching and perform efficient local repair by 2-hop neighbor tables. As route repair is carried out by unicast transmission, we can repair routes rapidly and efficiently – to handle the frequent recurrence of route damages in high-mobility networks. Simulation results show that when compared with other routing protocols such as AODV, AOMDV, MPRDV and MMDV, ours performs constantly better in packet delivery ratios, average hop counts, delay per hop and control overhead.

ACKNOWLEDGMENT

This work was supported in part by the National Science Council, Taiwan, R. O. C., under Grant No. NSC 100-2221-E-032-063.

REFERENCES

[1] L. Liang, Y. A. Sekercioglu, and N. Mani, "A survey of multipoint relay based broadcast schemes in wireless ad hoc networks," IEEE Communications Surveys and Tutorials, vol. 8, no. 1-4, pp. 30-46, Dec. 2006.

[2] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector (AODV) routing," Internet Draft, Mobile Ad Hoc Networking Working Group, Mar. 2001.

[3] G. Allard, P. Jacquet, and L. Viennot, "Ad hoc routing protocols with multipoint relaying," 5eme Rencontres Francophones sur les aspects Algorithmiques des Telecommunications, 2003.

[4] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing for ad hoc networks," Proc. 2001 International Conference on Network Protocols, Nov. 2001, pp. 14-23.

[5] A. Mtibaa and F. Kamoun, "MMDV: multipath and MPR based AODV routing protocol," Proc. IFIP 5th Annual Mediterranean Ad Hoc Networking Workshop, 2006, pp. 137-144.

[6] S. Mueller, R.P. Tsang, and D. Ghosal, "Multipath routing in mobile ad hoc networks: issues and challenges," Proc. 11th International Symposium on Modeling, Analysis and Simulation of Computer and Telecomm. Systems Tutorials, 2003, pp. 209-234.

[7] S. J. Lee and M. Gerla, "AODV-BR: Backup routing in ad hoc networks," Proc. 2000 IEEE Wireless Communications and Networking Conference, 2000, pp.1311-1316.

[8] J. Cai and W. Wu, "Degraded link-disjoint multipath routing in ad hoc networks," Proc. 4th International Symposium on Wireless Pervasive Computing, 2009, pp. 1-5.

[9] B. Xue, P. Y. Ren, and S. C. Yan, "Link optimization ad hoc on-demand multipath distance vector routing for mobile ad-hoc networks," Proc. 5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009, pp .1-6.

[10] J. J. Galvez, P. M. Ruiz, and A. Skarmeta, "Achieving spatial disjointness in multipath routing without location information," Proc. 2009 Wireless Communications and Networking Conference, 2009, pp. 5-8.

[11] S. Wang, Q. Li, Y. Jiang, and H. Xiong, "Stable on-demand multipath routing for mobile ad hoc networks," Proc. 2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications, 2009, pp. 318-321.

[12] G. Liu, K. J. Wong, B. S. Lee, B. C. Seet, C. H. Foh, and L. J. Zhu, "PATCH: a novel local recovery mechanism for mobile ad hoc networks," Proc. 2003 IEEE Vehicular Technology Conference, Oct. 2003, vol. 5, pp. 2995-2999.

[13] W. K. Lai, S. Y. Hsiao, and Y. C. Lin, "Adaptive Backup Routing for Ad Hoc Networks," Computer Communications, vol. 30, issue. 2, pp. 453-464, Jan. 2007.

[14] W. Ge and P. W. Li, "(OPTAODV) An optimized AODV protocol for ad hoc network," Proc. 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008, pp. 1-4.

[15] M. Pan, S. Y. Chuang, and S. D. Wang, "Local repair mechanisms for on-demand routing in mobile ad hoc networks," Proc. 11th Pacific Rim International Symposium on Dependable Computing, December 2005, pp. 1-8.

[16] J. Singh, P. Singh, and S. Rani, "Enhanced local repair AODV (ELRAODV)," Proc. 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, Dec. 2009, pp. 787-791.

[17] J. Sirilar and K. Rojviboonchai, "OHO: overhearing on-demand route repair mechanism for mobile ad hoc networks," Proc. 2010 International Conference on Electrical Engineering/Electronics Computer Telecommunications and Information Technology, May 2010, pp. 66-70.

[18] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu, "The broadcast storm problem in a mobile ad hoc network," Proc. 1999 International Conference on Mobile Computing and Networking, 1999, pp. 151-162.

[19] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2002, pp, 194-205.

[20] "The VINT project: the network simulator - ns-2," Available: http://www.isi.edu/nsnam/ns.