2005 Asia-Pacific Conference on Communications, Perth, Western Australia, 3 - 5 October 2005.

An Optimized Ad-hoc On-demand Multipath Distance Vector(AOMDV) Routing Protocol

YuHua Yuan, HuiMin Chen, and Min Jia
School of Communication and Information Engineering
Shanghai University, China
Email: hmchen@staff.shu.edu.cn

***Abstract*-To avoid frequent route discovery, various multipath routing protocol has been proposed based on the existing single path routing protocol in ad hoc networks. Ad hoc On-demand Multipath Distance Vector (AOMDV) is one of extentions to the well-studied Ad hoc On Distance Vector (AODV). In this paper an Optimized AOMDV (OAOMDV) is presented to solve the "route cutoff" problem in AOMDV. The proposed protocol adds a new scheme into AOMDV and simulation results show the performance improvement.**

Ⅰ. INTRODUCTION

Ad hoc network is a collection of wireless mobile nodes without any fixed base station infrastructure and centralized management. Each node acting as both a host and a router moves arbitrarily and communicates with others via multiple wireless links. Therefore, the network topology changes greatly.

In such a dynamic network, it is a challenging issue to be able to get route in time. The routing protocols proposed so far can be classified into two categories: proactive routing protocol and reactive routing protocol [1]. Reactive routing protocol, which initiates route computation only on demand, performs better than proactive routing protocol, which always maintains route to destination by periodically updating, due to its lower control overhead. AODV [2] is a well-known on-demand protocol. Although it outperforms many other routing protocols, it is a single path routing, which needs new route discovery whenever a path breaks. To overcome such inefficiency, several studies [3], [4], [5], [6], [7] have been presented to compute multiple paths. They provide alternative paths to send data packets without executing a new route discovery, if the primary path breaks.

In this paper, we propose an optimized routing protocol based on AOMDV, called OAOMDV, for mobile Ad hoc network. The aim is to solve the "route cutoff" problem in AOMDV to find more reverse paths, so that the route discovery frequency can be decreased and the network performance will be improved.

The rest of the paper is organized as follows: Section Ⅱ overviews related work. Section Ⅲ describes an OAOMDV routing protocol in detail. Performance evaluation via simulation is presented in Section Ⅳ and the conclusion is drew in Section Ⅴ.

Ⅱ. RELATED WORK

To decrease route discovery frequency in single path routing, some multipath routing protocols have been proposed to extend AODV [3], [4], [5]. However, [3] does not perform well by increasing the number of communication sessions. [4] selects node-disjoint paths, and it only performs well with high density of mobile nodes. [5] tries to find link-disjoint paths. But it cannot always find all of existing reverse paths.

AOMDV[5] extends the prominent AODV to discover multiple link-disjoint paths between the source and the destination in every route discovery. It uses the routing information already available in the AODV protocol as much as possible. It makes use of AODV control packets with a few extra fields in the packet header such as advertised hop count and route list which contains multiple paths. The main problem, which is called "route cutoff" in AOMDV, is that when there are one or more common intermediate nodes in a pair of link-disjoint paths, it cannot find both of the reverse paths. To reduce route discovery latency, it is necessary to find out all of the existing link-disjoint reverse paths.

Ⅲ AN OAOMDV ROUTING PROTOCOL

In this section we present the operation details of the proposed OAOMDV routing protocol. It has modified AOMDV to find a pair of link-disjoint paths without "route cutoff" problem. In fact, if there are two link-disjoint paths with one or more common intermediate nodes between the source and destination, two reverse paths and two forward paths ought to be formed. For example in Figure 1, two forward paths (S-A-C-E-D and S-B-C-F-D), and two reverse paths (D-E-C-A-S and D-F-C-B-S) are possible. However, AOMDV detects the two forward paths and only one reverse path (D-E-C-A-S). The other reverse path (D-F-C-B-S) is cut off. To avoid this kind of "route cutoff", we propose a scheme, in which a routing packet RREP_ACK (Route Reply Acknowledgement) is added to be transmitted along the forward path (S-B-C-F-D). Once an intermediate node receives the packet, a reverse path to the source will be set up. So, when destination D receives the packet, the omitted reverse path (D-F-C-B-S) will be achieved.

A. Setting up reverse paths

When a traffic source needs a route to a destination and no route is available, the source initiates a route discovery process by generating a routing packet RREQ (Route Request, see TABLE Ⅰ) and broadcasting it. When any neighbor of the source receives the RREQ, first its address will be copied into the “Last hop” field of RREQ, which is used to check for link-disjoint reverse paths. Then it will update some information from RREQ to its route table entry (see TABLE Ⅱ, TABLE Ⅲ) to form a reverse path. Finally the node will do some modifications to the RREQ such as increasing “Hop count” and broadcast it.

An intermediate node only accepts RREQ with different “Last hop”, and the “Hop count” of the RREQ is not larger than the “Advertised hop count” in corresponding route table entry, which is the longest available path for the destination at the time of first RREQ for a particular destination sequence number. Then it repeats the last two steps like the neighbor of the source. So that at intermediate nodes multiple reverse paths to the same destination will be formed and listed in its “Route list”. Note that the intermediate nodes only broadcast the first received RREQ.

The destination will accept all RREQs, but only set up reverse path with the RREQ, which has different “Last hop”. This process is the same as AOMDV.

TABLE Ⅰ

THE FORMAT OF RREQ

Type	Reversed	Last hop	Hop count
RREQ ID			
Destination IP Address			
Destination Sequence Number			
Originator IP Address			
Originator Sequence Number			

TABLE Ⅱ

ROUTE TABLE ENTRY STRUCTURE

Destination	Sequence number	Advertised hop count	Route list

TABLE Ⅲ

ROUTE LIST STRUCTURE

hop_count$_1$	next_hop$_1$	last_hop$_1$	expiration_timeout$_1$
hop_count$_2$	next_hop$_2$	last_hop$_2$	expiration_timeout$_2$

In Figure 1, if needed, source S will broadcast an RREQ in its radio range. When nodes A and B receive the RREQ, they first add their own addresses into RREQs as “Last hop” respectively. In this paper, the RREQ via node A is labeled as RREQ (A), and the RREQ via node B is labeled as RREQ (B). Then they update some information in their own route table entry to form reverse paths (A-S and B-S). Finally node A broadcasts RREQ (A) and node B broadcasts RREQ (B). After node C receives RREQ (A) and RREQ (B), it will form two link-disjoint reverse paths (C-A-S) and (C-B-S). Then it will broadcast the first received RREQ (A) to others (suppose that RREQ (A) is received first), if it does not have any route available for destination. Otherwise, C will reply an RREP back to source. When nodes E and F receive RREP (A), they will respectively form reverse paths (E-C-A-S and F-C-A-S). At last, destination D successively receives RREQ (A) from E and F. If D first receives RREQ (A) from E, a reverse path (D-E-C-A-S) will be set up. When it receives the second RREQ (A) from F, which contains the same “Last hop”, it will not set up the non-disjoint reverse path (D-F-C-A-S), and vice versa.

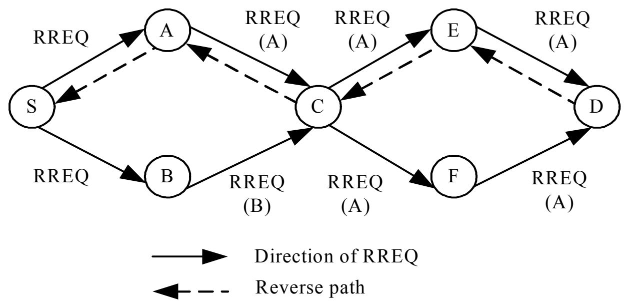

Figure 1 Setting up reverse path process

B. Setting up forward paths

Once destination receives a RREQ, it will reply to it with a routing packet RREP (Route Reply, see TABLE Ⅳ), even if the RREQ has the same “Last hop” as previous RREQs. Before destination sends back the RREP to its neighbor, from which the RREQ is received, the “Last hop” in the RREQ must be copied into the RREP’s “ACK” field to deal with “route cutoff” problem. This is different from AOMDV and the “ACK” will be looked on as a criterion in our scheme.

The RREP received by the neighbor of the destination will first add its address into the “Last hop” of this RREP to identify multiple link-disjoint forward paths to destination. Then it will form a forward path to destination and transmit the RREP to “next_hop” in the node’s route table entry, in which “Destination” is the source initiating the RREQ. Every “next_hop” receiving the RREP will repeat the last step like the destination’s neighbor. When destination receives RREP, it only sets up forward path with the RREP, which has different “Last hop” from others.

In Figure 2, after receiving RREQ (A) from E first, destination D replies to E with an RREP, which has copied the “Last hop” of RREQ (A) into its “ACK” field (ACK=A). When E receives the RREP, it will add its address into RREP as the “Last hop” to identify a link-disjoint forward path. In this paper, the RREP via node E is labeled as RREP (E).

Then, it forms a forward path (E-D) and transmits the RREP (E) to the “next_hop” in its route table entry, the “Destination” of which is source S. As a reverse path (D-E-C-A-S) via E has already been set up, the “$next\_hop_1$” can be founded to be C. So node E will transmit RREP (E) to node C. C will form a forward path (C-E-D) and transmit it to its next hop ($next\_hop_1$=A). Every node on the reverse path receiving the RREP (E) will form a forward path. At last, when source S receives, it will form a forward path (S-A-C-E-D). The source node can immediately send data packets along the forward route without waiting for the arrival of other RREPs.

The destination D will also receive RREQ (A) from node F a little later, and then it will reply to F with a RREP. The “ACK” field of the RREP has already been filled in with the “Last hop” of RREQ (A) (ACK=A). Node F has a route to source S and the “$next\_hop_1$” is C. When C receives the RREP (F), it forms a forward path (C-F-D) and then look for next hop to source S. Node C has two link-disjoint reverse paths (C-A-S and C-B-S), one of which has already been used for transmitting former RREP (E), so only the other path can be used here. The “$next\_hop_2$” in route table entry of node C is node B. Then C will transmit the RREP (F) to B. At last another forward path (S-B-C-F-D) is set up, as the “Last hop” of RREP (F) is different from the previous RREP (E). Note that the two forward paths are link-disjoint.

TABLE Ⅳ

THE FORMAT OF RREP

Type	ACK	Last hop	Hop count
Destination IP Address			
Destination Sequence Number			
Originator IP Address			
Lifetime			

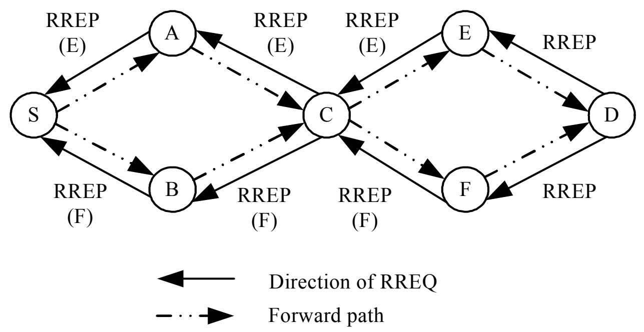

Figure 2 Setting up forward path process

C. Setting up omitted reverse path

Now we can get two forward paths (S-A-C-E-D and S-B-C-F-D) and one reverse path (D-E-C-A-S). The other reverse path (D-F-C-B-S) is omitted. To find out this omitted path, we propose a new scheme, in which a routing packet RREP_ACK (see TABLE Ⅴ) is presented.

When the source receives a RREP, it will determine whether the destination has omitted a reverse path. The source will compare the “ACK” field in RREP with its neighbor from which it receives the RREP. If they are the same node, it means that no reverse path has been omitted. If they are different nodes, it means that one link-disjoint reverse path has been omitted. If so, the source will transmit a control message called RREP_ACK along the link-disjoint forward path set up by this RREP to form a new reverse path. It is different from the RREP_ACK in AODV, which is a one-hop packet and only used to confirm reception of RREP. In this paper, An RREP_ACK play an important role in the proposed routing scheme. It contains the last hop of the omitted reverse path (“Last rphop” in TABLE Ⅴ) and the last hop of forward path (“Last fphop” in TABLE Ⅴ). Both of them are gained from the RREP. The last hop of reverse path is used to make up the omitted link-disjoint reverse path. The last hop of forward path is used to find a link-disjoint forward path, along which the RREP_ACK is forwarded hop by hop and a new reverse path will be set up.

TABLE Ⅴ

THE FORMAT OF RREP_ACK

Type	Last rphop	Last fphop	Destination IP address	Originator IP Address	Originator Sequence Number

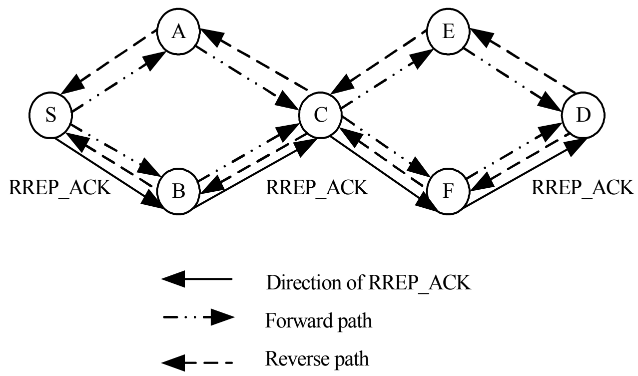

Figure 3 Setting up omitted reverse path process

In Figure 3, when source S receives first RREP (E) from its neighbor A, it will compare the previous hop of the RREP (E) with the “ACK” in RREP (E). The comparison concludes that they are the same node A. It shows that the reverse path (D-E-C-A-S) along the forward path (S-A-C-E-D) has already been set up. When source S receives second RREP (F), it will compare the previous hop of the RREP (F) with the “ACK” in RREP (F). The comparison concludes that they are different nodes. The former is node B, and the later is node A. It shows that a reverse path along the forward path (S-B-C-F-D) set up by this RREP (F) is omitted. So source S will generate an RREP_ACK to destination D along the forward path to form the reverse path. In RREP_ACK, the "last rphop" is node B, and the "last fphop" is node F. When node B receives the RREP_ACK, a reverse path (B-S) will be set up. When node C receives the RREP_ACK, a reverse path (C-B-S) will be set up. At last a reverse path (D-F-C-B-S) will be set up by destination D. Here, the RREP_ACK is not only used to set up an omitted reverse path, but also used to confirm the connection between the source and destination.

Ⅳ SIMULATION RESULT

As AOMDV has been proved to outperform AODV in [5], here we only evaluate the performance of AOMDV and OAOMDV protocol using Network Simulator 2 (ns 2) and compare them. Simulation environment is as follows:

1) *Propagation: TwoRayGround*
2) *Radio range of a node: 250 m*
3) *Channel capacity:* 2 Mb/sec
4) *Medium Access Control (MAC) protocol:* IEEE802.11 Distributed Coordination Function (DCF)
5) *Traffic pattern*: 50 CBR/UDP (we suppose bidirectional communication here)
6) *Size of data packet:* 512 bytes
7) *Data rate:* 1 packet/sec
8) *Simulation area:* 1000 m × 1000 m
9) *Number of nodes:* 50
10) *M*aximum speed: 5m/s, 10m/s, 15m/s, 20m/s
11) *Number of simulations: 50*
12) *Simulation time: 500 seconds*

We have evaluated four important performance metrics: packet loss percentage, average end-to-end delay of data packets, route discovery frequency and routing overhead.

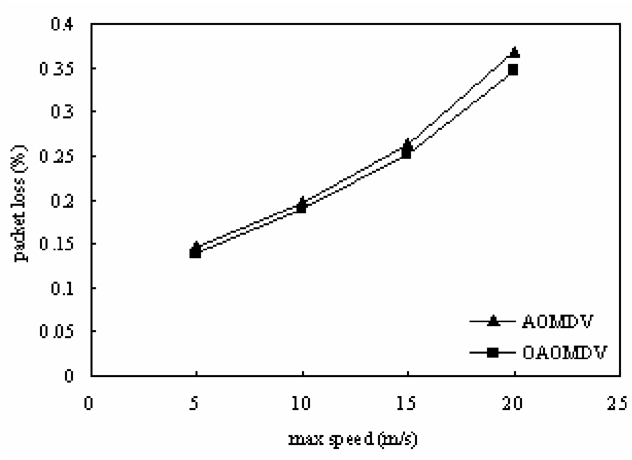

Figure 4 Packet loss

Figure 4 reveals the packet loss percentage of two routing schemes in the network either at the source or at intermediate nodes. AOMDV shows a little poor performance. It is because AOMDV loses the link-disjoint reverse paths with one or more common nodes, so that the available routes in AOMDV are fewer than OAOMDV. More data packets in AOMDV cannot get available routes, and then they will be buffered and wait for new routes. Some overtime packets will be dropped.

Figure 5 presents the route discovery frequency. It means the aggregate number of route requests generated by all sources per second. As AOMDV loses reverse paths during route discovery process, the number of available paths is smaller than OAOMDV. Of course, it is more frequent for AOMDV to initiate route discovery. With the mobility speed increases, much more available routes will be broken, and the number of source-initiated route discoveries will increase rapidly.

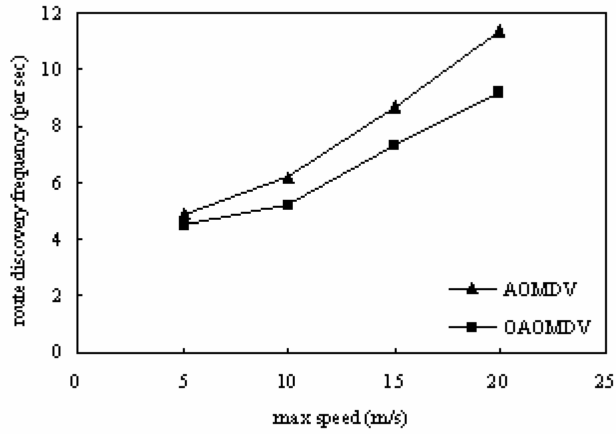

Figure 5 Route discovery frequency

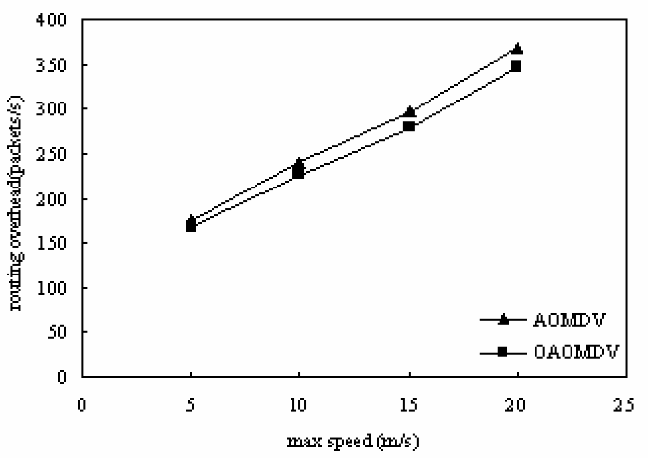

Figure 6 Routing overhead

Figure 6 reports the routing overhead of two routing protocols. It is the total number of routing packets "transmitted" per second. Although a new routing packet (RREP-ACK) is added in OAOMDV protocol, the overhead is still lower than AOMDV. The reasons are: firstly, less route discoveries are initiated in OAOMDV, which will lead to the flooding of RREQs; secondly, the small packet RREP_ACK is only unicasted to destination, the number of which can be ignored, when it is compared with the number of RREQs.

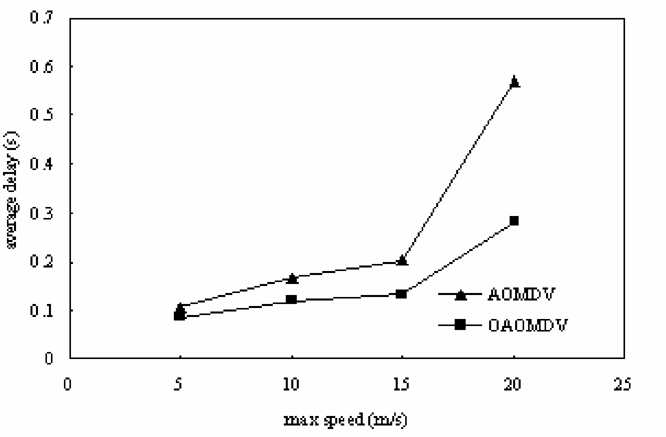

Figure 7 Average end-to-end delay

Figure 7 shows the average end-to-end delay of data packets. This includes all possible delays caused by buffering during route discovery, queuing delay at the interface, retransmission delays at the MAC, propagation and transfer times. With more available routing in OAOMDV, much more data packets will be delivered to destinations without waiting for route discovery latency. So the average end-to-end delay is reduced in OAOMDV, especially when the mobility speed increase.

Ⅴ. CONCLUSION

This paper presents an optimized AOMDV that can solve the “route cutoff” problem in AOMDV by using a new control packet RREP_ACK. This control packet has been defined in AODV, but it is often ignored, and we generally do not apply it. Here we change its primary meaning, and make different use of it in the OAOMDV protocol. Although the proposed routing scheme OAOMDV increases an additional routing packet, simulations show that the routing overhead is decreased. The other performance such as packet loss, route discovery frequency, and average end-to-end delay also has been improved more or less. In future work, the additional packet can be extended for its utility in other fields such as QoS.

ACKNOLEDGEMENT

This work was funded by Shanghai Science and Technology Development Foundation under contract 045115012.

REFERENCES

[1] Abolhasan, M., Wysocki, T., and Dutkiewicz, E., “A review of routing protocols for mobile ad hoc networks,” Ad Hoc Networks 2, pp. 1-22 (2004).

[2] Perkins, C., Belding-Royer, E., and Das, S., “Ad hoc On-Demand Distance Vector (AODV) Routing,” rfc3561.txt (2003).

[3] Lee, S. J. and Gerla, M., “AODV-BR: Backup Routing in Ad hoc Networks,” Proc. of IEEE Wireless Communications and Networking Conference, pp. 1311-1316 (2000).

[4] Li, X. F. and Cuthbert, L., “On-demand Node-disjoint Multipath Routing in Wireless Ad hoc Networks,” Proc. of the 29th Annual IEEE International Conference on Local Computer Networks (LCN’04).

[5] Marina, M. K. and Das, S. R., “On-demand Multipath Distance Vector Routing for Ad Hoc Networks,” Proc. of 9th IEEE Int. Conf. On Network Protocols, pp.14-23 (2001).

[6] Jiang, M. H. and Jan, R. H., “An Efficient Multiple Paths Routing Protocol for Ad-hoc Networks,” Proc. of the 15th International Conference on Information Networking, pp. 544-549 (2001).

[7] Higaki, H. and Umeshima, S., “Multiple-Route Ad hoc On-Demand Distance Vector (MRAODV) Routing Protocol,” Proc. of the 18th Internation Parallel and Distributed Processing Symposium (IPDPS’ 04).