Recovering and optimization routing reply for AODV in mobile ad-hoc networks

S. Subburam and Sheik Abdul Khader

Abstract —The topology of a mobile ad hoc network (MANET) may change unexpectedly and rapidly due to the mobility of its nodes, and thus, setting up a highly reliable route is very challenging. Most protocols of ad hoc networks are composed of two main mechanisms, "Route Discovery" and "Route Maintenance", that work together to allow the nodes to discover and maintain routes to arbitrary destinations. It is well known that broadcasting, a mechanism for route discovery and route maintenance in Ad hoc On-demand Distance Vector (AODV) routing protocols wherein a source node broadcasts a route request to all other nodes in the network, has a limitation of a high number of unsuccessful packet deliveries. The loss of route reply packets will cause serious impairment to the performance of on-demand routing protocols as well as increasing the cost. The AODV–Recovering and Optimization Routing Reply (RORR) mechanism proposed in this paper is expected to eleminate the wastage of control packets while finding the route to destination using the concept of 'trustworthy distance'. In addition, if a link breakage occurs while route reply packets are being sent, this mechanism will identify a new route with the help of a 'recover node'. We performed extensive simulation studies to evaluate the performance of AODV-RORR and found that RORR improved the performance of AODV significantly in a wide range of system parameter values and in all critical metrics, including packet delivery ratio, control overhead and end-to-end delay.

Keywords—Ad hoc Network, AODV, AODV-RORR, trustworthy distance, resolver, route reply

I. INTRODUCTION

A mobile ad hoc network (MANET) is a self-configuring network of mobile devices connected by numerous wireless links [1]. Every device in a MANET is also a router because it is required to forward the traffic unrelated to its own use. Each MANET device is free to move in any arbitrary direction, and thus each device will potentially change its connectivity to other devices on a regular basis. A major challenge in building a MANET is for each device to continuously maintain the information required to properly route the traffic.

Put it simply, a MANET consists of mobile platforms (e.g., a router with multiple hosts and wireless communication devices), herein referred to simply as "nodes", that are free to move about arbitrarily. These nodes may be located on an airplane, ship, truck, car and perhaps even on small devices that people carry with them, and there may be multiple hosts per router. A MANET may operate in isolation or may have gateways to and interface with a fixed network. In the latter, it is typically visualized to operate as a "stub" network connected to a fixed internetwork.

In the nodes are present numerous wireless transmitters and receivers that use antennas, which may be omnidirectional (broadcast), highly directional (point-to-point), possibly steerable or some combination thereof. At a given point in time, depending on the position of the nodes, their transmitter and receiver coverage patterns, transmission power levels and co-channel interference levels, a wireless connectivity in the form of a random, multihop graph or "ad hoc" network exists between the nodes. This ad hoc topology may change as the nodes move or adjust their transmission and reception parameters.

In many MANET applications [2], broadcasting is an essential operation for discovering neighbors. Several routing protocols, such as Ad hoc On-demand Distance Vector Routing (AODV) [1], have been proposed for route discovery. Routing protocols for MANETs are largely of three types, namely Proactive (table-driven), Reactive (on-demand) and Hybrid. Proactive protocols are consistent and update routing information to all the nodes in a network, e.g., DSDV [3] and OLSR. Reactive protocols identify the route only when called for, e.g., AODV [4],[5] and DSR; hybrid protocols combine the features of proactive and on-demand protocols, e.g., ZRP. Quality of service in wireless ad hoc networks is imperative to the success of network-centric warfare as envisioned in future military operations [6].

AODV, an on-demand routing protocol, comprises of two mechanisms: route discovery and route maintenance. When a new route is discovered, it is cached in the memory of the sender node for a certain time. Thus, when a node attempts to transmit data, it first checks the routing table in the memory, and if a route was not found, it then initiates the route discovery process by broadcasting RREQ packets. The RREQ packets are sent to all the neighbouring nodes within the range of the sender. As soon as a RREQ packet is received, the receiving node will check its routing table for the route. If a route was identified, the receiving node will reply to the sender with the route information, or else the receiving node will rebroadcast the RREQ packet. The rebroadcasting will continue till a route is found or until the expiry of RREQ packets. Thus, it is difficult to restrict or control the process of rebroadcasting.

The RREQ message is usually propagated throughout the entire network. When the RREQ from node S ultimately reaches destination D, then a RREP message is sent by D to S. Other intermediate nodes that have a route to D in their routing tables may also send a RREP to S.

A RREP is transmitted using unicast while RREQ using broadcast. When S receives the RREP, it can use the new route to send data packets to D. Route maintenance deals with

S. Subburam is Research Scholar, Satyabama University, Chennai, India. Email:subburam@bsauniv.ac.in

P. Sheik Abdul Khader is with Department of Computer Applications, BSA University, Chennai, India. Email:hodca@bsauniv.ac.in

routing information at nodes, typically involving three possible operations: handling route errors, deleting stale route entries, and learning new routes from the traffic.

In this paper, we propose the technique of recovering and optimization routing reply (RORR) for on-demand routing protocols. Although recovering is well known in dynamic source routing (DSR) [2],[3] its subject is data packets rather than route reply packets. In RORR, when a node cannot deliver a RREP to the intended hop node (we call it recover node), it attempts to recover the RREP through two possible methods. First, if the routing protocol has multi-path capability (e.g., DSR), the recover looks up its own route cache for an alternate path to relay the RREP. Second, if no alternate path was found, usually because of lack of capability of the routing protocol, the recover runs a route discovery to find a new path to the source node. The RORR route discovery typically covers only one-hop neighbors of the recover, which means RORR may save the routing overhead significantly.

Here we present an implementation of RORR in AODV [4] routing protocol. Extensive simulations revealed that RORR improves the performance of AODV significantly in a wide range of system parameter values and in all critical metrics, including packet delivery ratio, control overhead and end-to-end delay. The remainder of the paper is organized as follows. In Section 2, we present the problem definition, and a review of related work is presented in Section 3. The implementation of AODV-RORR is presented in Section 3. Section 4 describes the simulation model and presents the simulation results. Section 5 concludes the paper.

II. PROBLEM DEFINITION

Suppose in an existing AODV a source node has no routing information when attempting to send data to a destination, the source node broadcasts RREQ messages to the neighbor nodes to identify a route from source to destination. In this route discovery procedure, each node will receive a RREQ from its neighbor node as long as it is in the transmission range of its neighbor. However, if two adjacent nodes are located in the border of each other's transmission range and the moving direction is opposite, the creation of a reverse path will be hampered, leading to unsuccessful RREQ transmission. This problem can lead to flooding of control packets in the network.

Moreover, the loss of RREQ packets can cause serious impairment to the performance of on-demand routing protocols, including increasing the costs. Among existing on-demand routing protocols, no protocol seems to offer solution to this problem to the best of our knowledge.

III. RELATED WORK

In the flooding method [7],[8] a source node of a MANET disseminates a message to all its neighbors; each of the neighbors that receive the RREQ will check its own routing table and, if no routing information is available, the message will rebroadcasted at once to all its neighbors. The process goes on until all nodes in the network receive the message.

Although this method is trustworthy for a MANET with low density nodes and high mobility, it is very harmful and unproductive as it causes severe network congestion and quickly exhausts the battery power. Blind flooding ensures coverage and that the broadcast packet is received by every node in the network, provided there is no packet loss caused by collisions in the MAC layer and there is no high-speed movement of nodes during the broadcast process. However, due to the broadcast nature of wireless communications, redundant transmissions in blind flooding may cause the broadcast storm problem [9] due to contention and collision of redundant packets.

In MANETs, a node may move from one location to another or leave the network. As a result, the network topology changes constantly and unpredictably. Therefore, in on-demand routing protocols, RREP packets should be given higher priority that other types of packets.

In recent years, many routing algorithms have been proposed for MANETs based on the coordinates of nodes. To obtain this information, global positioning system (GPS) could be employed [10],[11]. It is possible to calculate the stability of a routing path using the current and future position of nodes. Therefore, the best path is mathematically determined, which is invariably the shortest path. Clearly, if a routing process is developed without any consideration of the movement of nodes and the stability of routing path, the links may be easily broken. There are many routing protocols to deal with network's stability [12],[13]. We know that a route breakage leads to wastage of resources. Therefore, it is important to find a route with longer life time as possible [14]. If a routing protocol can enhance stability, it leads to lower overhead and higher efficiency. A reliable multi-path QoS routing protocol with a slot assignment scheme was proposed [15]. In this protocol, two parameters—the route life time between two connected mobile nodes and the number of hops—are used to select a routing path with low latency and high stability.

IV. MODIFICATION OF ROUTE DISCOVERY (REBROADCAST) IN AODV

Figure 1 describes the route discovery phase in AODV. Here, if two adjacent nodes are located in the border of each other's transmission range and the moving direction is opposite, the creation of a reverse path will be hampered. To overcome this problem, we propose an algorithm based on GPS, which considers the position, speed and direction of the present node. In the proposed protocol, a concept of trustworthy distance, *r*, is defined, which is used to decide whether the node can receive RREQ from its neighbor and whether the link state between the two nodes is reliable. As said before, the mobility of nodes often causes route failures. Hence the reliable distance, *r*, should be changed automatically with the speed and direction of nodes. Now we present the details of our scheme. For simplicity, it is assumed that all nodes in the network have the same transmission range and altitude. To record the present node's information from GPS, the format of RREQ packet should be modified (see Table 1).

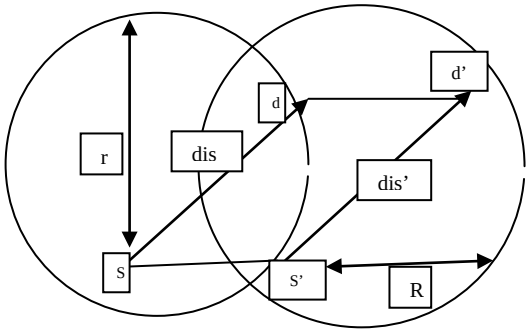

Figure 1: Route discovery phase in AODV

TABLE I

NEW STRUCTURE OF ROUTE REQUEST

original RREQ
present position (x,y)
present direction (cos α, sin α)
present speed v
present time t

Present position (*x, y*), present direction (cos α, sin α) and present speed *v* indicate the present node's information, which the receiving nodes use to decide whether the link is stable and the RREQ should be broadcasted again. Present time *t* denotes the time when the RREQ is sent.

When a source node *S* needs a route to a destination, it initiates a route discovery process by flooding RREQ, including its own GPS information. Then an intermediate node D receives the first RREQ from its neighbor *S* and determines the node *S*'s position at time *t1* when the RREQ is received using the information in RREQ. We assume that speed and direction of two nodes are invariable between the time *t0* when the RREQ is sent and *t1*. Eqns. (1,2) and (3,4) are applied to find the coordinates of sending node and receiving node at time t_1 and t_0:

For finding the coordinates of sending node at t_1,

$$x_{st1} = x_{st0} + (v \times \Delta t \times \cos \alpha) \quad (1)$$

$$y_{st1} = y_{st0} + (v \times \Delta t \times \sin \alpha) \quad (2)$$

For finding the coordinates of receiving node at t_0,

$$x_{rt0} = x_{rt1} - (v \times \Delta t \times \cos \alpha) \quad (3)$$

$$y_{rt0} = y_{rt1} - (v \times \Delta t \times \sin \alpha) \quad (4)$$

where Δt is the difference between time *t1* and *t0*.

Similarly, node D utilizes its own GPS information to get its initial position at *t0*. With these, the initial distance d_{is} at *t0* and final distance d_{is}' at *t1* between *S* and *D* will be found.

Distance between the nodes d_{is} at t_0,

$$d_{is} = \sqrt{(x_{rt0} - x_{st0})^2 + (y_{rt0} - y_{st0})^2} \quad (5)$$

Distance between the nodes d_{is}' at t_1,

$$d_{is}' = \sqrt{(x_{rt1} - x_{st1})^2 + (y_{rt1} - y_{st1})^2} \quad (6)$$

Using Eqns. (5,6), we compare the distances d_{is} and d_{is}' to find whether d_{is} is greater than d_{is}' or not.

If initial distance d_{is} is more than trustworthy distance *r* and final distance d_{is}' exceeds initial distance d_{is}, it is determined that the two nodes are moving in opposite directions. Then node *D* decides that the link state between them is not trustworthy and the RREQ received by node *D* is discarded.

Using this modified algorithm, the flooding area of RREQ and the control overhead are both restricted. The established routes have a longer valid time than that built by original AODV. Note that *r* is a very important parameter in our new mechanism. If *r* is close to *R*, the new route will nearly be the same as the original. However, if *r* is too small, the route discovery latency and control packet will be increased rapidly for route with too many hop-counts [6]. As said above, *r* should be able to change automatically with the variation of node's speed and direction. When the two neighboring nodes move fast in different directions, *r* will be decreased, making the route more trustworthy.

V. IMPLEMENTATION OF RORR IN AODV

We now present the implementation of RORR in AODV. AODV is based on traditional distance vector routing scheme, where routing decision is made on a hop-by-hop basis. Here a route is discovered by adding a backward routing entry pointing to the source at middle nodes when they propagate the RREQ message, and by adding a forward routing entry pointing to the destination at middle nodes when they relay the RREP message to the source. In the process of route discovery, after the destination node receives the RREQ message, it sends a RREP message to the source via intermediate nodes. If an intermediate node cannot deliver the RREP because the intended next hop neighbor is not reachable, the node becomes a recover node and starts the RORR method.

The recover node checks the following three conditions. First, the RREP is not generated by an intermediate node (we mark a RREP generated by an intermediate node as *rrep in* to distinguish it from the RREP generated by the destination node). Second, the RREP has not been recovered before (i.e., it is not marked as *recovered*). Finally, the node is not recovering any other RREPs. If all the conditions are satisfied, the node acts as a recover node and starts the RORR method.

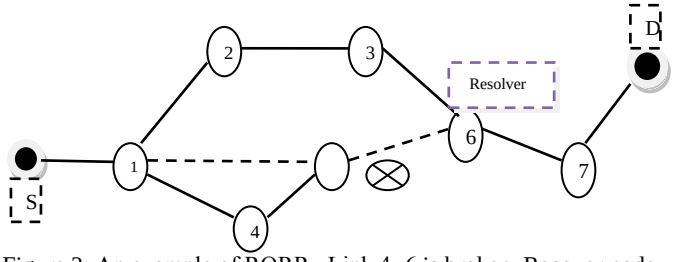

Figure 2: An example of RORR. Link 4−6 is broken. Recover node is 6. The intended RREP return path is D → 7 → 6 → 4 → 1 → S. The actual return path after RORR is D →7 → 6 →3 →2 → 1→ S.

The recover node first saves the RREP message, which will prevent it from recovering other RREPs since a recover node can recover only one RREP at a time. Then the recover node runs the RORR route discovery by broadcasting a $RREQ_{RORR}$ message, which is marked as *recover* to distinguish it from a usual RREQ message. For the $RREQ_{RORR}$, the *initiator* field

is set to the recover; the *target* field is set to the source and the *TTL* field is set to 1.

Upon receiving the $RREQ_{RORR}$, if a neighbor itself is the source, it sends a $RREP_{RORR}$ to the recover directly. Other- wise, it looks up its routing table for a route to the source. One-hop neighbors of the recover should have a route to the source because they recently propagated the original RREQ message and learned a reverse path to the source. If such a route exists, and the route does not use the recover as the next hop to the source (this is to prevent a routing loop), the node sends a $RREP_{RORR}$ to the recover. The recover waits for a short time (e.g., 2 * NODE TRAVERSAL TIME [3]) for $RREP_{RORR}$ from its neighbors. Then it selects the best route according to the standard route update rules in AODV or uses the route with newer sequence number; when sequence numbers are same, a route with fewer hop count is used. Next, the recover recovers the saved RREP by sending it to the source along the path just discovered. The RREP is marked as *recoverd* to prevent other intermediate nodes from recovering it again.

We give an example in Figure 2 to explain how RORR works. Node *S* is discovering a route to node *D*. *D* sends a RREP to *S*, and the initial return path is $D \rightarrow 7 \rightarrow 6 \rightarrow 4 \rightarrow 1 \rightarrow S$. Node 6 cannot send the RREP to node 4 because node 4 will move away. Node 6 becomes the resolver and broadcasts a $RREQ_{RORR}$. Node 3 receives the $RREQ_{RORR}$ and finds a route to *S* from its routing table; so node 3 sends a $RREP_{RORR}$ to node 6. Now node 6 receives the $RREP_{RORR}$ and successfully recovers the original RREP by sending it along the path discovered by RORR. Then the return path after RORR is $D \rightarrow 7 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow S$.

RORR can be applied to virtually all on-demand routing protocols. When a RREP is undeliverable due to link breakage or channel congestion, RORR tries to find an alternative path to relay the undeliverable RREP packet to the source node that initiated the route discovery and is waiting for the RREP. Recovering RREP messages in this manner will prevent a large number of retransmissions of RREQ messages and hence mitigate the congestion in the network. Moreover, RORR will improve the routing performance in terms of packet delivery ratio and end-to-end delay because data packets will wait for less time in buffers and experience less 'overflow' and 'outdated' (packets wait too long) drops.

VI. SIMULATION

We used a comprehensive simulation model based on Ns-2 in our evaluation. The MAC layer protocol used in the simulations was the Distributed Coordination Function (DCF) of IEEE 802.11. DCF uses Request-To-Send (RTS) and Clear-To-Send (CTS) control packets for unicast transmissions. Broadcast packets are sent using the unspotted Carrier Sense Multiple Access protocol with Collision Avoidance (CSMA/CA) [1]. The propagation model used in the simulations was the two-ray model. Wave LAN was modeled as shared media radio with a nominal bit rate of 2 Mb/sec, and the radio range was 250 m.

A.The Traffic and Mobility Models

The trace was constant bit rate (CBR). The source and the destination of each CBR flow were randomly selected but not identical (sources and destinations of different flows might coincide). Each source transmitted 512-byte data packets at a certain rate (packets/second). The mobility model was random waypoint, where the speed of a node was randomly chosen from 0 m/s to a given maximum value, and a node stayed for a pause time after reaching a waypoint. Terrain size was used for simulations: 1400 m^2 field with 100 nodes.

B. Performance Metrics

Three important performance metrics are as follows:

Packet sending rate at source: The ratio of data packets delivered to the destinations to that generated by the CBR sources; also, a related metric, received throughput (kilobits/second) at the destination, was evaluated in some cases. Range of load (packets/second) was 4–16.

Node moving speed: The speed at node movement takes place. The range of max speed (m/s) is 10–40.

Node waiting time: The amount of time a node pauses. The range of node pause time (s) is 0–50.

C. Performance Results

Varying CBR Load :Figure 3(a–c) shows the performance of AODV-RORR and AODV when the CBR packet sending rate was varied. In this set of simulations, the number of CBR flows was kept as 20. The CBR sources send data packets to destinations at different rates, from 4 packets/s to 20 packets/s. When data packet sending rate was greater than 4 per second, the difference of packet delivery ratio between AODV and AODV-RORR was about 20% (e.g., improved from 10% to 20% (Figure 3a)), control packets were saved by about 25% (Figure 3b), and end-to-end delay was reduced by more than 25% (Figure 3c). As a result, RORR was able to recover more RREP packets and improve the performance of AODV more significantly. In Figure 3c (and in Figure 4b,c), AODV performance curve is not stable, i.e., it goes up and down. After applying RORR, the AODV-RORR curve becomes much more stable. This result favors our analysis of the importance of RREP packets in on-demand routing protocols.

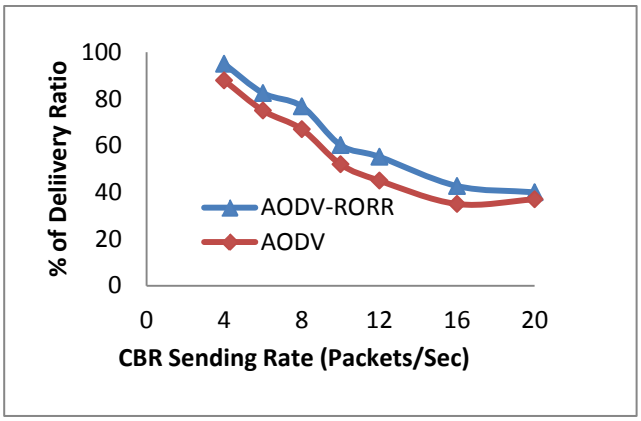

Figure 3(a)

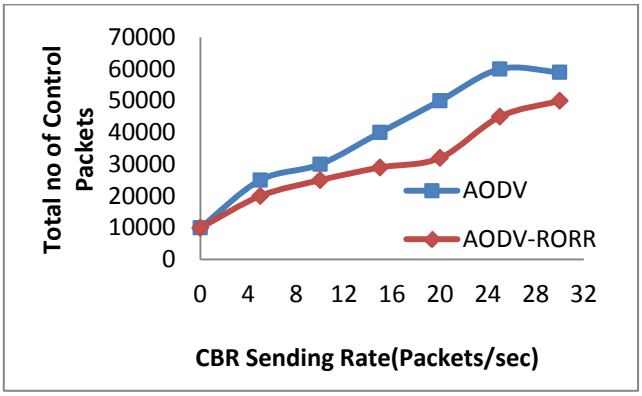

Figure 3(b)

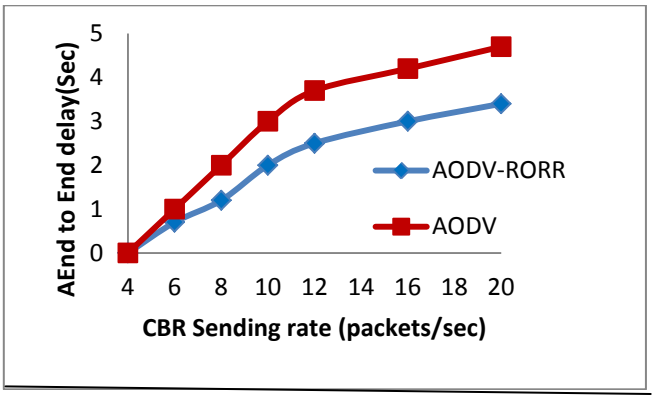

Figure 3(c)

Figure 3: Performance when CBR load changes. Network area 1400 × 1400 m^2, 100 nodes, 20 flows, maximum speed 20 m/s, and pause time 300 s.

Varying Pause Time: Figure 4(a–c) shows the performance of AODV-RORR and AODV when node pause time was varied. In this set of simulations, the number of flows was kept as 25 and maximum node speed was 20 m/s. From the results, we observe that RORR improves the routing performance noticeably; the difference of packet delivery ratio between AODV and AODV-RORR was around 15% (Figure 4a), control packets were saved by more than 30% (Figure 4b), and end-to-end delay was reduced by about 30% (Figure 4c).

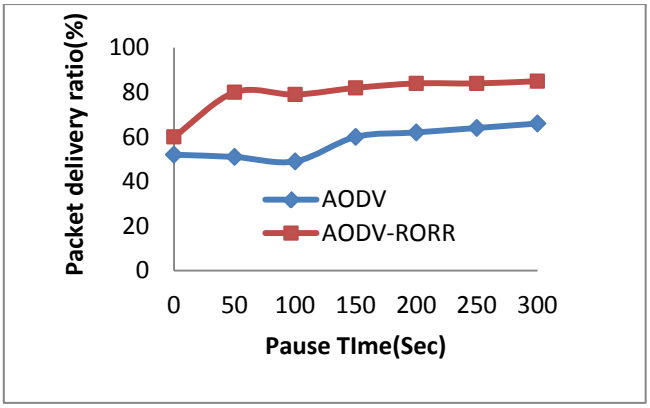

Figure 4(a)

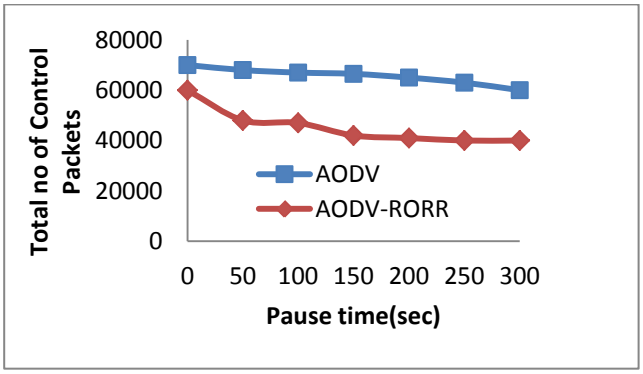

Figure 4(b)

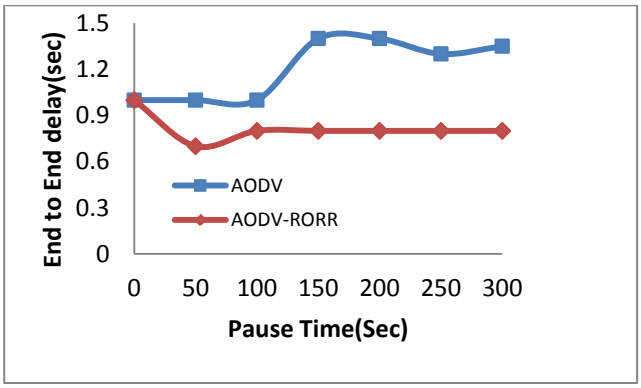

Figure 4(c)

Figure 4: Performance when pause time changes. Network area 1400 × 1400 m^2, 100 nodes, 25 flows and maximum speed 20 m/s.

Varying Maximum Node Speed: Figure 5(a–c) shows the performance of AODV-RORR and AODV when the maximum speed of nodes was varied. In this set of simulations, the number of flows was kept as 25. We used zero pause time to prevent it from affecting the node speed. From the results, we observe that RORR improves all three performance metrics noticeably; the difference of packet delivery ratio between AODV and AODV- RORR was around 20% (Figure 5a), control packets were saved by more than 30% (Figure 5b); and end-to-end delay was reduced by about 30% (Figure 5c). However, since the topology changes faster as the maximum speed increases, the routes established by the recovered RREPs may break soon, and hence the performance improvement was not as high as in previous simulations.

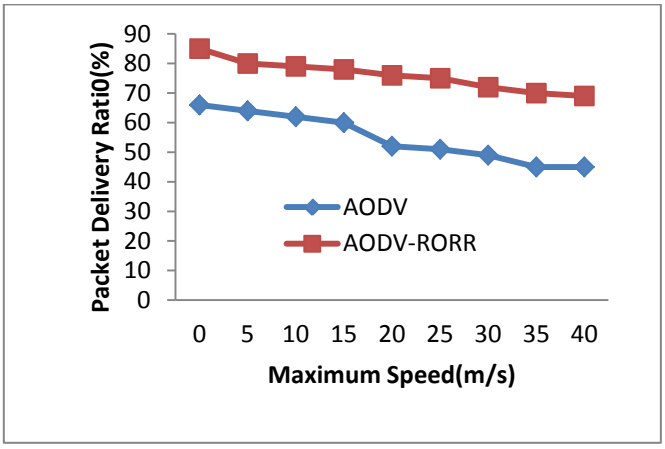

Figure 5(a)

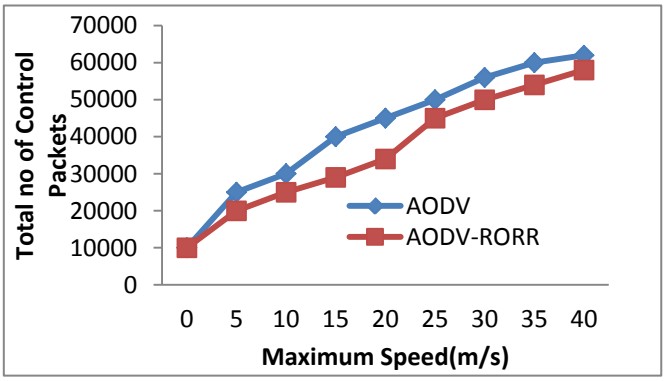

Figure 5(b)

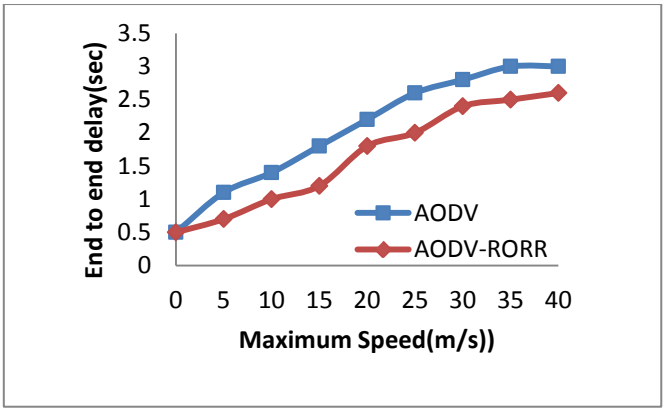

Figure 5 (c)

Figure 5: Performance when maximum node speed changes. Network area 1400 × 1400m^2, 100 nodes, 25 flows and pause time 0.

As the network size grows, the route between a source and a destination becomes longer. A RREP packet has to travel farther to reach the source node. It is more likely that the RREP cannot be relayed to its intended next hop. RORR recovers the undeliverable RREPs and improves the routing performance. However, the improvement may not be dramatic since RORR is not proposed as a radical solution to the scalability problem of AODV routing protocol.

VII. CONCLUSION

From extensive simulation studies, it was understood that RORR significantly improves the performance of AODV in all metrics, namely packet delivery ratio, control overhead and end-to-end delay. To the best of our knowledge, this paper is the first to address an approach to remedy the loss of route reply messages. Our approach may be applied to virtually all on-demand routing protocols, without conflicting the optimizations reviewed in the previous section.

REFERENCES

[1] IEEE 802.11: part 11: Wireless LAN Medium Access control (MAC) and Physical Layer (PHY) specification, Aug.1999.
[2] Satyabrata chakrabarti and Amitabh mishra. Quality of service in Mobile Ad hoc Networks.CRC press LLC.2003
[3] D.B. Johnson, D.A.Maltz, and Y.c.Hu. The dynamic source routing protocol for mobile ad hoc networks (dsr). Internet draft, draft ietfmanet-dsr-09.txt,apr 2003.
[4] C.E.Perkins, E.M. Belding-Royer, and I.D.Chakeres.Ad hoc on demand distance vector (aodv) routing.IETF Internet draft, oct. 2003.
[5] Yih-Chun Hu and David B.Johnson. Implicit Source Routes for On-Demand Ad hoc Network Routing. ACM 2001.
[6] Kui Wu and Jamelle Harms. On-Demand Multipath Routing for Mobile Ad hoc Networks. Proceeding of EPMCC\ACM. Feb 2001.
[7] C. Ho,K. Obraczka, G. Tsudik and K. Viswanath.Flooding for Reliable Multicast in Multi-hop Ad hoc Networks. International Workshop in Discrete Algorithms and Methods for Mobile Computing and Communication, 64–71, 1999.
[8] J. Jetcheva, Y. Hu, D. Maltz and D. Johnson. A Simple Protocol for Multicast and Broadcast in Mobile Ad hoc Networks. Internet Draft, draft- ietf manet-simple-mbcast-01.txt, 2001.
[9] Muneer Bani Yassein , Sanabel Fathi Nimer , Ahmed Y. Al-Dubai A new dynamic counter-based broadcasting scheme for Mobile Ad hoc Networks Simulation Modelling Practice and Theory 19 (2011) pp.553–563.
[10] Boukerche A, Rogers S (2001). GPS query optimization in mobile. In: Proceedings of the sixth IEEE conference on symposium. pp. 198-203.
[11] William S, Mario G (1999). IPv6 flow handoff in ad hoc wireless networks using mobility prediction. In: Proceedings of the 1999 GLOBECOM Conference. pp. 271-275.
[12] Chiu CY, Wu HK, Chen GH (2002). Stability aware cluster routing protocol for mobile ad hoc network. In: Proceedings of the ninth international conference on parallel and distributed systems. Chungli, Taiwan. pp. 471-479.
[13] Kim WI, Kwon DH, Suh YJ (2001). A reliable route selection algorithm using global positioning systems in mobile ad-hoc networks. In: Proceedings of the 2001 IEEE int. conf. commun., pp. 3191-3195.
[14] Sun H, Hughes HD (2003). Adaptive QoS routing based on prediction of local performance in ad hoc networks", In: Proceedings of the 2003 IEEE Conference on Wireless Communications and Networking, pp. 1191-1195.
[15] Wang NC, Lee CY (2009). A reliable QoS aware routing protocol with slot assignment for mobile ad-hoc networks. J. Network Comput. Appl., pp. 1153-1166.