AODV Based Multi-path Local Repairing Scheme for Mobile Ad Hoc Networks
Eman Shaaban, PhD
Computer Systems Dept.
Faculty of computer and information science
Ain-Shams university, Cairo, Egypt
Eman.shaaban@cis.asu.edu.eg

Abstract:- Ad hoc networks have many applications. Multiple ad hoc routing protocols have been proposed, of which on-demand routing protocols are very popular because they are easy to realize and have low routing overhead. The mobility of nodes and instability of the wireless environment may result in link breaks between neighboring nodes, even causes multiple routes to be invalid. Existing local repairing schemes of links breaking for Ad hoc On demand Distance Vector (AODV) routing protocol can recover the disconnected path as needed. This paper proposes an effective path recovery scheme for AODV in which multiple active routes are repaired proactively without waiting for arrival of new packets. Hence incoming data packets for those routes will not be subject to the delay of repairing the route and can be immediately forwarded. The order of locally repairing the routes depends on the total number of packets that traversed the route before broken.

The efficiency of the proposed protocol has been implemented and evaluated using ns2. There has been a noticeable improvement in the packet delivery ratio and also in the reduction of end-to-end delay and routing overheads compared to the original AODV.

Key-words:- Ad hoc networks, routing protocols, AODV, local repair.

1 Introduction
An ad hoc mobile network is a collection of mobile nodes that are dynamically and arbitrarily located in such a manner that the interconnections between nodes are capable of changing on a continual basis. In order to facilitate communication within the network, routing protocols are used to discover routes between nodes. They have to reflect the dynamic topology and provide the best route to the destination. Therefore the reactive route construction is indispensable in this circumstance. However, it is also critical issue to recover the route error caused by nodes’ mobility. To resolve the problem, it is the best way that a source node didn’t detect the route break.

Conventional IP based routing protocols are not appropriate for ad hoc mobile networks because of the temporary nature of the network links and additional constraints on mobile nodes i.e. limited bandwidth and power. Routing protocols for such environments must be able to keep up with the high degree of node mobility that often changes the network topology drastically and unpredictably.

Routing protocols in mobile networks are divided into two basic classes: proactive routing protocols and reactive routing protocols. The proactive routing protocols (e.g. OLSR) are table-driven. They usually use link-state routing algorithms flooding the link information. Link-state algorithms maintain a full or partial copy of the network topology and costs for all known links. The reactive routing protocols create and maintain routes only if these are needed, on demand. They usually use distance-vector routing algorithms that keep only information about next hops to adjacent neighbors and costs for paths to all known destinations. Thus, link-state routing algorithms are more reliable, less bandwidth-intensive, but also more complex and compute- and memory-intensive.

There are many common reactive routing algorithms used in ad hoc networks but AODV and DSR which both are on-demand algorithms, are the most popular ones.

Unfortunately, these protocols do not have any efficient local repair algorithms. In DSR, nodes can record multiple route information to the same destination in route caches. When the route breakage happens, the packet transmissions may not be stopped by aid of multiple route information of intermediate nodes. However, when route break happens and there is no additional route information at the node that detects the route break, the route error message is sent to the source node. Then the source node tries to re-discover a new route by activating route discovery process. This can not be considered as local route repair. In AODV, there exists a local repair process. This process let the node detecting route error find a new route from itself to destination. Its drawback is that the route can be longer than previous one. In addition, the local repair process is activated limitedly when the route error
happens not far from destination node. Thus, the local repair cannot be generally used. As a result, both DSR and AODV rely on route reconstruction from source to destination in most cases.

This paper proposes an effective path recovery scheme for AODV in which multiple active routes are repaired proactively without waiting for arrival of new packets. Hence incoming data packets for those routes will not be subject to the delay of repairing the route and can be immediately forwarded. The proposed AODV modifies the local repair algorithm used in the route maintenance of the AODV routing protocol [6]. The proposed AODV reduces the routing control packets as well as average end to end delay compared to the original AODV, and leads to a noticeable improvement in the packet delivery ratio.

The rest of the paper is organized as follows. Section 1 describes AODV routing algorithm. Section 2 describes path recovery scheme in AODV. Section 3 presents the proposed AODV. Performance environment, simulation and results are shown in section 4. Conclusions and future works are demonstrated in section 5.

1 AODV description
AODV is a widely researched protocol among the research community. Most of research effort has focused on simulations aimed at determining the performance of AODV, and also in comparison to the performance of other ad hoc routing protocols [1], [3]. Currently there exist several AODV implementations that comply with a varying degree to the protocol description defined in [6].

AODV determines a route to a destination only when a node wants to send a packet to that destination. Routes are maintained as long as they are needed by the source. Sequence numbers ensure the freshness of routes and guarantee the loop-free routing.

1.1 Route discovery process
During a route discovery process, the source node broadcasts a route request packet RREQ to its neighbours Fig.1. If any of the neighbors has a route to the destination, it replies to the request with a route reply packet RREP, otherwise, the neighbors rebroadcast the route request packet. Finally, some request packets reach to the destination.

1.2 Routing tables
Each AODV node maintains a routing table, AODV deals with route table management. Route table information must be kept even for short-lived routes, such as are created to temporarily store reverse paths towards nodes originating RREQs. AODV uses the following fields with each route table entry. Each node routing table entry contains the following information [6]:

- Destination IP address
- Destination sequence number
- Next hop
- Number of hops to reach destination (hop count)
- Active neighbors for this route (precursor list)
- Expiration time for this route table entry
- Routing flags
- Network interface

1.3 Maintaining Local Connectivity
Because nodes can move, link breakages can occur. If a node does not receive a Hello message from one of its neighbors for specific amount of time called Hello interval, then the entry for that neighbor in the table will be set as invalid and route error message (RERR) message will be generated to inform other nodes of this link breakage. RERR messages inform all sources using a link when a failure occurs. Each forwarding node should keep track of its continued connectivity to its active next hops, as well as neighbors that have transmitted Hello messages during the last Hello interval. A node can also maintain accurate information about its continued connectivity to these active next hops, using one of any suitable link layer notification, such as those provided by IEEE 802.11. Or if layer-2 notification is not available, passive acknowledgment should be used when the next hop is expected to forward the packet, by listening to the channel for a
transmission attempt made by the next hop to detect transmission within a specified interval (or the next hop is the destination) to determine connectivity. If a link to the next hop cannot be detected by any of these methods, the forwarding node should assume that the link is lost.

### 1.4 Local Repair

When a link break in an active route occurs, the node upstream of that break may choose to repair the link locally. During local repair data packets should be buffered. If, at the end of the discovery period, the repairing node has not received a RREP (or other control message creating or updating the route) for that destination, it transmits a RERR message for that destination. On the other hand, if the node receives one or more RREPs (or other control message creating or updating the route to the desired destination) during the discovery period, it first compares the hop count of the new route with the value in the hop count field of the invalid route table entry for that destination. If the hop count of the newly determined route to the destination is greater than the hop count of the previously known route, the node should issue a RERR message for the destination, A node that receives a RERR message with the 'N' flag set for not deleting the route to that destination. Then the originating node may choose to reinitiate route discovery.

Local repair of link breaks in routes sometimes results in increased path lengths to those destinations. Repairing the link locally is likely to increase the number of data packets that are able to be delivered to the destinations, since data packets will not be dropped as the RERR travels to the originating node. Sending a RERR to the originating node after locally repairing the link break may allow the originator to find a fresh route to the destination that is better, based on current node positions. The TTL of the RREQ should initially be set to the following value:

\[
\text{Max}(\text{MIN\_REPAIR\_TTL}, 0.5*\#\text{hops}) + \text{LOCAL\_ADD\_TTL}
\]

Where MIN\_REPAIR\_TTL is the last known hop count from the upstream node of the failure to the destination. \#hops is the number of hops from the upstream node of the failure to the source of the currently undeliverable packet. LOCAL\_ADD\_TTL is a constant value.

### 1.5 Local repair of link break for multiple destinations

AODV When a link breaks along an active route, there are often multiple destinations that become unreachable. The node that is upstream of the lost link tries an immediate local repair for only the one destination towards which the data packet was traveling. Other routes using the same link must be marked as invalid, but the node handling the local repair may flag each such newly lost route as locally repairable. Before the timeout occurs, these other routes will be repaired as needed when packets arrive for the other destinations.

Fig.2 represents an ad-hoc network with source nodes S1, S2 and destination nodes D1, D2, and D3; and a number of intermediate nodes. There are three active routes S1-D1, S2-D2, and S1-D3. Fig.3 indicates a failure of link B-C. Thus nodes D1, D2, and D3 became unreachable from source nodes. The active routes C-D1, C-D2, and C-D3 in existing AODV local repair implementation are repaired as needed. When a packet arrives at node B towards node D1, the route B-D1 will be locally repaired. Also When a packet arrives at node B towards node D2, the route B-D2 will be locally repaired. And so for destination D3, When a packet arrives at node B towards node D3, the route B-D3 will be locally repaired.

### 2 Related works

To solve the inefficiency problem of route repair in reactive routing protocols, there were several studies to locally recover the disconnected route. LRR scheme [7] assumes that the link error is caused by the relative movement of only one node on the route. A “neighbor node” is defined as a node which is on the route from the source to the destination and is in the immediate vicinity of the moved node. In other words, a neighbor node is a former one of moved node on the route. The aim of this scheme is to patch the route between the two nodes of the broken path through some other link or node. The zone in which the route-repair packet propagates is defined as the “request zone.”

AODV-BR [8] is a modified protocol from AODV. The basic route discovery process has not been changed. A source node sends a RREQ packet to a destination node, and the destination reply to RREQ as a RREP packet. The different thing is that every node in the network operates in promiscuous mode. When a RREP packet comes back to the source node, a neighbor node which is not part of the route overhears a RREP packet,
then it records the node which transmitted a RREP packet as the next hop to the destination in its alternate route table. Likewise this, all nodes existing beside of the route updates their alternate route table after overhearing RREP packet. After these operations, the members of the route and neighbor nodes organize a mesh structure. The original route from the source to the destination is called as primary route, and the rest of the routes are called alternate route. Within these mesh structures; a data packet is delivered via an alternate route when the primary route is disconnected. ALRP [4] is a new ant agent based Local repair routing protocol. In the ALRP specification, when a link break in an active route occur, the node upstream of the break creates a Route Error(RERR) message listing all the destinations which have become unreachable due to the break. It then sends this message to its upstream neighbors, if, instead of sending an error message to the source node, the upstream node attempts to repair the broken link itself, fewer data packets may be lost and the link can be repaired without the source node (and other upstream nodes) being disturbed.

An enhancement over the basic AODV routing protocol which makes the protocol more robust against link breakages was proposed in [1]. If a source node moves, it is able to reinitiate the Route Discovery Protocol (RDP) to find a new route to the destination using path updating. For intermediate node link break a Local Repair Procedure is used to update the path.

A new local repair scheme using promiscuous mode was proposed in [5]. That scheme is mainly composed of two parts: adaptive promiscuous mode and quick local repair scheme. Adaptive promiscuous mode is to repeat the switching processes between promiscuous mode and nonpromiscuous mode to overcome energy limit caused by using promiscuous mode in overall time and quick local repair scheme is to fast perform the local re-route discovery process with the information of the active connection in the local area acquired by promiscuous mode.

Fig. 2. Ad-hoc network with multiple destinations

Fig. 3. Broken Link B-C
3 Proposed scheme
This paper proposes when a node's link to multiple destinations fails, the node may start the following:
- When a new packet arrives to one of these destinations, the flags of all unreachable destinations are set to being in-repaired. The broken route to that destination is first locally repaired reactively.
- The other broken active routes are repaired proactively without waiting for new packets to arrive. Hence incoming data packets for those routes will not be subject to the delay of repairing the route and can be immediately forwarded. The order of locally repairing the routes depends on the total number of packets that traversed the route before broken. The proposed AODV starts repairing the route with the highest packet count, then the next highest count and so on.
- To be able to repair multiple links at the same time, the overhead of broadcasting RREQ packets in local repair processes must be reduced by limiting TTL value to 2 instead of TTL value in equation\#1. This limitation allows to activate local repairing process regardless whether the broken link is closer to the destination or the source.
- The node invalidates the unreachable destinations in its routing table and creates a list of unreachable destinations (RERR message) to inform other nodes of these link breakages and start the process of optimal or short path to the unreachable destinations.

However, the existing Routing table implementation RFC 3561 [6] does not support the proposed scheme. Therefore, a new field packet_count is added to the routing table entry. Packet_count for a destination is the count of data packets traversing the node to reach that destination during the lifetime of the route.

3.1 The scenario for applying the proposed AODV
Applying the proposed AODV to ad-hoc network in Fig.2 as follows:
- each time the packet travels along active routes S1-D1,S2-D2, and S1-D3, all nodes along active routes increment its packet_count field for that destination in the routing table. i.e. if a packet is travelling from S1 to D1, then intermediate nodes A, B, C, T which forward the packet will increment its packet-count for destination D1 in the routing tables of these nodes. Also if a packet is travelling from S1 to D3, then intermediate nodes A, B, C, M, N which forward the packet will increment its packet-count for destination D3 in the routing tables of these nodes, and so on for D2 if a packet is travelling from S2 to D2, then intermediate nodes K, B, C, L which forward the packet will increment its packet-count for destination D2 in the routing tables of these nodes.
- When the link B-C fails, if a data packet arrives at B to reach destination D1, then node B will establish repairing locally route B-D1. Then B checks its routing table for packet-count value for D2 and D3 without waiting to receive data reaching destination D2 and D3. If D2' packet-count larger than D3' packet-count, then B proactively will locally repair route B-D2 followed by route B-D3. Otherwise If D3' packet-count larger than D2' packet-count, then B proactively will locally repair route B-D3 followed by route B-D2.
- The same scenario can be repeated if link B-C has more than three destination, node B will start repairing locally the first destination route on demand, and other destination routes will be repaired on order of their packet-count. The route with the highest packet-count, then repairing the next highest packet-count destination route, and so on till all routes are repaired.

4 Performance evaluation
Performance is evaluated using simulation. Simulation is performed using ns-2 an open source discrete event simulator targeted at networking research. To evaluate the proposed AODV, the proposed AODV based on modifying ns2 version is implemented with the default parameter specified in RFC [6]. Then simulating the original ns2 AODV version with the existing local repair scheme, and the proposed one under the same simulation environment.

The output trace file is analyzed for three important performance metrics: Average End-to-End Delay of packets that includes all possible delays caused by buffering during route discovery phase, queuing at the interface queue, retransmission at the MAC layer, propagation and transfer delays. Packet Delivery Ratio is the percentage of data packets delivered to that of number of data packets sent in simulation time. Control Overhead is the number of routing control packets in simulation time.

4.1 Simulation setup.
The simulation modeled a network of 50 mobile nodes placed randomly within a 300 x 800 meter area. A rectangular shape area is used to create more connection breaks during simulation. The physical radio characteristics of each mobile node’s network...
interface were chosen to approximate Lucent WaveLAN with a channel capacity of 2Mbps and a normal radio range of 250 meters. The two-ray ground reflection model is used for propagation. Random waypoint model used for scenarios generation, and randomness in the selection of packet sources and receivers. There are 5 speeds: 0m/s, 5m/s, 10m/s, 20m/s, 40m/s. The pause time is 0. The simulation time is 1000s. Constant Bit Rate (CBR) traffic is used to simulate UDP performance. There are 20 traffic instances with a rate of 4 packet/s. The packet size is 512 bytes. The collected data is averaged over 5 runs.

4.2 Routing packet overhead

The routing message overhead resulted from both original and modified AODV routing protocols has been presented in Fig.4. Proposed AODV gives lower routing message overhead by an average 8% less than original AODV routing message overhead. This is due to the reduced overhead of local repairing of multiple links in proposed AODV by limiting TTL value in RREQ broadcasting to 2. Reducing routing control packets leads to saving more bandwidth for data packets transfer. However reduced routing control packets is highly affected by mobility. Speed less than 5m/s means low number of broken links, and low number of proactively repaired routes, so the proposed AODV has no improving effect on reducing routing control packets at these low speeds. Speed more than 20m/s means high number of broken links, so repairing high number routes proactively. However these repaired routes soon will break again and required to be re-repaired to adapt with these high speeds. So the proposed AODV has an increased number of routing control packets compared to original AODV at these high speeds.

4.3 Packet delivery ratio.

The number of packets dropped or left wait for a route affect the packet delivery ratio as the increase in the number of packets dropped or left wait for a route reduce the packet delivery ratio. The number of packets dropped or left wait for a route affected by the success of local repair in repairing a failed route, where the number of packets dropped or left wait reduced as the percentage of success local repair attempts increased.

The results shown in Fig.5. demonstrate that the proposed AODV acquires a better packet delivery ratio than original AODV for UDP traffic since proposed AODV is based on early detection and repairing of some destination paths in advance and avoiding of packet buffering delay. Also newly arrived packets will be forwarded on a newly repaired route instead of being dropped when the link breaks. Moreover reducing control packets and saving channel bandwidth will lead to less packet dropping and improvement in packet delivery ratio. The proposed AODV has packet delivery ratio more higher than original AODV routing protocol by an average 1%. However this improvement is affected by high mobility due to increased number of broken links.

![Fig.4. routing packets overhead versus speed](image1)

![Fig.5. packets delivery fraction versus speed](image2)
4.4 Average end to end delay.

The results in Fig.6. show that the proposed AODV demonstrates lower average end to end delay than original AODV routing protocol by an average 2%, where the trials of locally repairing multiple broken links proactively reduce routing message overhead and free bandwidth channels in turn and this led to transfer data packets faster. Even though extra time is consumed by each node to increment its packet_count field in proposed AODV, there is much more time consumed by repairing some destination paths proactively in advance.

5 Conclusion

This paper proposes an effective path recovery scheme for AODV in which multiple active routes are repaired proactively without waiting for arrival of new packets. Hence incoming data packets for those routes will not be subject to the delay of repairing the route and can be immediately forwarded. The order of locally repairing the routes depends on the total number of packets that traversed the route before broken.

The simulation results show that proposed AODV achieves a significant improvement in the packet delivery ratio of UDP traffic and reduction of end-to-end delay and routing overheads compared to the original AODV.

The effect of the proposed AODV on energy consuming and different network loads in comparison with AODV routing protocol is recommended for further studies as well as the effect of the proposed protocol on TCP traffic.

REFERENCES: