Improving AODV with Preemptive Local Route Repair

Sérgio Crisóstomo   Susana Sargento
Pedro Brandão   Rui Prior

Technical Report Series: DCC-2003-07

Departamento de Ciência de Computadores – Faculdade de Ciências

&

Laboratório de Inteligência Artificial e Ciência de Computadores

Universidade do Porto

Rua do Campo Alegre, 823 4150 Porto, Portugal

Tel: +351+2+6078830 – Fax: +351+2+6003654

http://www.ncc.up.pt/fcup/DCC/Pubs/treports.html

Improving AODV with Preemptive Local Route Repair

Sérgio Crisóstomo, Susana Sargento, Pedro Brandão, Rui Prior
Department of Computer Science & LIACC
Faculty of Sciences, University of Porto
Rua do Campo Alegre, 823, 4150-180 Porto, Portugal
{slc, ssargento, pbrandao, rprior}@dcc.fc.up.pt

December 2003

1 Introduction

The mobile telecommunication networks are evolving in a large scale. With an increasing desire to access information on the Internet and to access any information anywhere, current trends push to the integration/merger of the Internet and mobile networks. A particularly hot area of Internet mobility is the ad hoc networking. These networks are temporarily formed, without any infrastructure, with nodes dynamically joining and leaving. In ad hoc networks, the nodes are usually host terminals, which also need to perform routing functions. Since all nodes are mobile, routes are always changing due to the movement of sender, receiver or intermediate nodes. To cope with the dynamic nature of the topology of ad hoc networks, several routing protocols have been proposed by the IETF. One of the most promising protocols is AODV (Ad Hoc On-Demand Distance Vector) [4]. This protocol is reactive, meaning that a search for a route from the source to the destination is performed only when this route is required, and it does not require nodes to maintain routes to destinations that are not in active communication. AODV also provides mechanisms to locally recover from link breakage and changes in network topology. However route repair is only performed when a route actually breaks and it is required to send a packet through that disrupted path. The support of link failure detection in AODV can be based on layer 2 or layer 3 mechanisms, acting as triggers to reactively repair routes. This route repair can be initiated locally in the node located immediately upstream of the broken link (local repair) or by the source after receiving a notification (Routing Error - RERR) of the route failure. Route failures have a significant negative impact in the service experienced by flows crossing ad hoc networks. The time elapsed between the link break detection and the establishment of a new route can be quite high thus introducing significant delay and possibly some packet loss. Losses may occur before the link break is detected or when buffering packets that are waiting for a new path is no longer possible.

There are already some proposals to use mobility prediction and to introduce preemptive route repair in ad hoc networks. Here we list only some more important approaches. In [5], the mobility information is placed in routing packets and piggybacked in data packets, to determine the time a route is about to break. For this purpose, an on-demand unicast routing protocol was proposed but it does not consider local route repair. [2] uses the received transmission power to estimate when a link will break. This mechanism is applied to DSR (Dynamic Source Routing) [3]; AODV is also considered but only superficially. In this proposal route repair is from the responsability of source nodes after receiving a warning about the link break imminence. In [1] it is presented the inclusion of mobility information in DSR route discovery with the aim of optimizing the duration of paths, and preemptive local repair is performed to nodes downstream that link. Note that nodes in DSR have a full knowledge of the path, which implies that the described mechanism cannot be applied to AODV.

In this paper we present an extension to the AODV protocol, denoted Preemptive Local Route Repair (PLRR), that aims to avoid route failures by preemptively local repairing routes when a link break is about to occur. This protocol extension resorts to AODV layer 3 connectivity information with new mobility extensions. Our proposal is to enhance node's information concerning link stability to its neighbors resorting to HELLO messages. These messages are appended with a mobility extension containing the node's position, motion vector and an associated timestamp. This mobility information will be used to predict the instant a link between two neighbors will break. Although we propose that all broadcast AODV messages contain the mobility information, HELLO, Route Request (RREQ) and Route Error (RERR), we will refer to all of them as 'HELLO messages' in order to simplify the text throughout the paper. Note that our proposal does not need to take into account the sender and destination location information, as other location-aided routing protocols. In this proposal, location and mobility information needs to be propagated only between neighbors. This proposal aims at improving the AODV Quality of Service capabilities by minimizing route failures. It eliminates 'communication stalls' in ongoing sessions, packet losses and probably the delay. Moreover, with less packet losses and stalls, TCP sessions will also benefit. The main drawback of the proposed extension is the possible establishment of non-optimal routes in terms of hop count metric. In the final paper, the performance of the proposed extension to AODV will be evaluated based on simulation results.

In the next sections, the location prediction process is addressed and the preemptive route repair procedure is presented. Finally, the most important conclusions from this work and some topics for future work are presented.

2 Diffusion of Connectivity Information

Nodes have to trigger the preemptive local route repair (PLRR) procedure when they predict that a connection is about to break. In order to perform this evaluation, nodes need to know the position and mobility pattern of their neighbors. As already stated, the location and motion pattern information of each neighbor is mainly received via extended HELLO messages. These messages are only generated from nodes belonging to active routes. Notice that this is sufficient in our procedure, since a node only requires to know this information from a neighbor sharing a session. HELLO messages are periodically broadcasted to neighbors, only if during the last HELLO time interval there was no broadcast of a RREQ or RERR message [4]. To provide the neighbor nodes with predictive information on the links' duration, we defined a new extension to be appended to HELLOs and to AODV control messages, which we denote as mobility extension. Its format is depicted in figure 1.

0			31
Type	Length	Timestamp	
Longitude			
Latitude			
Velocity		Angle	
Position Error	Reserved		

Type: 2	Longitude: in hundredths of second
Length: 18	Velocity: in cm/s, range: [0:2^16−1]
Timestamp: in ms; range: [0, 9999]	Angle: in units of 2PI/10000 rad, range: [0,9999]
Longitude: in hundredths of second	Position Error: maximum position error in meters, range: [0, 255]

Figure 1: Mobility extension

In the mobility extension, latitude and longitude are defined in hundredth of seconds, with reference to the equator and to the Greenwich meridian, respectively. Position error is used to carry the node's position accuracy. The elements inserted in the mobility extension field assume that each node has a GPS (Global Positioning System) like receiver which acts as a time reference for all nodes, thus synchronizing their internal clocks. This assumption is performed in most location-based mechanisms proposed in the literature. Detailed explanation on the mobility extension fields will be provided in the final paper.

Using the information of the mobility extension sent from a neighbor N_j, node N_i updates its mobility information about N_j, and determines the (predicted) amount of time they will stay in range (Link Expiration Time - LET), using the following proposed expression:

$$LET_{i,j} = \frac{-(a \times b + c \times d) + \sqrt{(a^2 + c^2) \times (r - errpos_i - errpos_j)^2 - (a \times d - b \times c)^2}}{a^2 + c^2}$$

where r is the minimum transmission range of the two nodes, x and y correspond to the position of the node converted from the latitude and longitude parameters, $a = v_i \times \cos\theta_i - v_j \times \cos\theta_j$, $b = x_i - x_j$, $c = v_i \times \sin\theta_i - v_j \times \sin\theta_j$ and $d = y_i - y_j$. v and θ correspond to the absolute value of the velocity and the motion direction respectively. The indices i/j correspond to the nodes N_i/N_j.

This expression is an extension of the one presented in [5], with the inclusion of the maximum position error ($errpos$) of neighbors N_i and N_j to account for position inaccuracies. It assumes a free space propagation model where the received signal power is a function only of the distance to the transmitter. Nodes transmission ranges can be assumed to be the same. Otherwise, they can be carried in the reserved field of the mobility extension, assuming the above mentioned propagation model.

Every time a node N_i receives mobility information about a neighbor N_j, $LET_{i,j}$ is recomputed, and the instant to initiate a PLRR procedure is (re)determined. This instant must be at maximum $LET_{i,j}$-PLRR_DISCOVERY_TIME in the future, where PLRR_DISCOVERY_TIME is the necessary time to accomplish a PLRR procedure.

3 Preemptive Local Route Repair Procedure

When a link between two nodes is about to break, a PLRR procedure shall be initiated for every affected destination. As mentioned before, the main goal of this procedure is to find a new sub-path towards the destination avoiding that link (besides any other unstable links).

Our procedure for preemptively repairing the route is based on the AODV Route Discovery process enhanced with modified RREQ and RREP messages, RREQp (RREQ preemptive) and RREPp (RREP preemptive), and with specific processing rules. The search for a new sub-path will be localized to the vicinity of the unstable link.

Consider that the link connecting N_i to N_j is about to break, and N_i must find a new sub-path towards the destination. In order to design the protocol, several aspects were taken into account:

1. Every route entry having N_j as next hop in N_i routing table must be changed with a better (more stable) next hop;
2. Packet routing shall not be affected (i.e. interrupted) during an ongoing PLRR;
3. In a PLRR discovery process it is very likely to find a new sub-path that joins the previous path in a node downstream;
4. There is also the possibility of finding a sub-path joining other path towards the destination;
5. Sequence numbers are used to infer the freshness of a route to a destination: route entries with larger destination sequence numbers correspond to fresher routes. The destination sequence number of the RREQp must be the same as the one assigned in the previous route entry for the destination. This is to increase the chance of finding a sub-path that joins the previous path;

6. To increase the probability of finding a new sub-path, a hop count increase in a candidate new path may be tolerated; however, in order to avoid routing loops, the following rules must be followed:

   (a) If the destination sequence number of the sub-path is the same as the one assigned in the previous route entry for the destination, only a tolerance of 2 in the hop count is allowed. This process will be detailed later in this section;

   (b) Besides that, previous hops of the PLRR source node N_i (nodes that have N_i as the next hop towards a destination whose route is being repaired) shall ignore and discard any RREQp they may receive from N_i.

7. Any node that receives a RREQp from a neighbor node must discard it if the LET between them is less then twice the PLRR_DISCOVERY_TIME. This condition guarantees that links expiring before the PLRR conclusion cannot be included in the candidate sub-path, and also that a new sub-path has the property of being PLRR capable, which means that the new sub-path expire time is sufficient enough to conclude a new PLRR if necessary.

Consider D as the set of all affected destinations due to a link break. Every route entry of a destination in set D must be marked as being in a PLRR procedure, denoting that these routes are in a process of PLRR but are still *valid*. At this point, a backoff timer is set to each of the destinations, and retries of the PLRR to a given destination are prevented until the timer expires or is removed by a successful PLRR. Note that PLRR retries immediately after a failed one are very unlikely to have success.

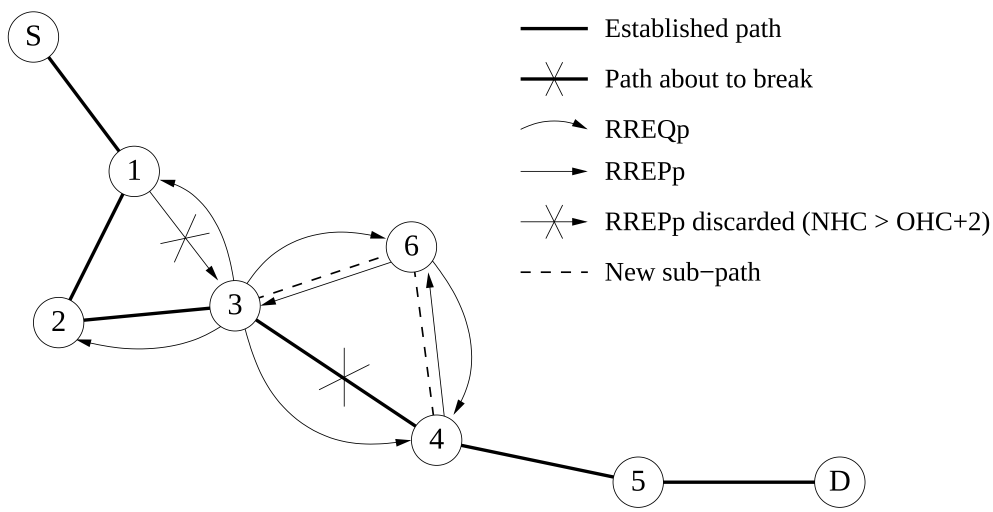

Figure 2: PLRR ilustration

A simplified PLRR procedure is illustrated in figure 2. Considering that the link between nodes 3 and 4 is about to break, a RREQp is generated in node 3 to all its neighbor nodes, with the last known destination sequence number. This RREQp shall have a new flag set (thus called RREQp), denoting a PLRR, and a mobility extension must be appended to the RREQp. The RREQp must be broadcasted with a small time to live (PLRR_TTL) to limit the flooding of RREQp to the nodes in the vicinity of the unstable link.

Alternatively, a node can generate only one RREQp for all destinations in set D with an appended extension called PLRR_DEST_AGR (besides mobility extension), which is presented in figure 3. It contains all the remaining destinations in set D. This aggregation prevents the flooding of a large number of messages, thus minimizing the control message overhead.

0 20 31

Type	Length	NoD	Reserved
Additional Destination IP Address [1]			
Additional Destination Sequence Number [1]			
Additional Destination IP Address [2]			
Additional Destination Sequence Number [2]			
. . .			

Type: 3 NoD: Number of Destinations (max. 30)

Length: variable

Figure 3: PLRR_DEST_AGR extension

When a node receives a RREQp it must process the appended mobility extension to update the mobility information of the neighbor node (the previous hop of the RREQp) and predict the link expire time. Our approach considers that the node receiving the RREQp is the one that evaluates if it can be included in the new sub-path towards the destination. Another approach could be having the sender of the RREQp message filtering the nodes to which it would send the RREQp, taking into account their location and mobility information. Our approach has several advantages: (1) the RREQ in AODV is already a broadcast message, and therefore it is easier to keep its extension, RREQp, with the same characteristics; (2) the HELLO messages in AODV are only sent from nodes with active routes, and a filtered RREQp would only be sent to a subset of the neighbor nodes, precluding the possibility of using other nodes (possibly more stable); (3) the location and mobility information in our approach is up to date, since the node sending the RREQp appends its current mobility information, and the node receiving it also uses its current information.

Upon predicting on the link expire time, the node checks if this value is smaller than twice the PLRR_DISCOVERY_TIME. If it is, the message must be silently discarded. Otherwise, the node checks in its routing table if the next hop to the destination is equal to the RREQp source. If so, the message is discarded. In our example, this processing takes place in node 2, when it receives a RREQp from node 3. If the message is not discarded, normal AODV treatment shall be given to it. In this case, if the node is not the destination, does not have a route to the destination, or if it has a route to the destination with a sequence number lower than the one of the RREQp, the message is forwarded (this is the case of a RREQp processing in node 6). Otherwise, a RREPp is sent to the node that initiated the route repair. If a new RREQp is to be forwarded, the mobility extension refering to the previous hop must be discarded, and a new one refering to the current node added.

If the RREQp contained an aggregate of destinations (a PLRR_DEST_AGR extension),the processing described above must be done to every listed destination in the extension. Then, a new aggregate of destinations may have to be recomputed (listing the destinations to which a RREPp was not sent) and added to the RREQp to be forwarded.

The RREPp is sent when a stable sub-path was achieved. This message may contain a field with the minimum LET of the new sub-path, in order to allow to choose the sub-path with increased stability. Its value should be updated in a hop by hop basis troughout the path back to the source.

A PLRR_DEST_AGR extension may also be appended to the RREPp with any additional requested destinations (that were in the PLRR_DEST_AGR of the RREQp) to which the node has routes with larger or the same sequence number as the one of the RREQp.

The processing of RREPp is equivalent to the one performed by actual AODV, except when the node is the source of the RREQp. The PLRR source can receive several RREPp. The routing entry of the corresponding destination can be updated on the arrival of each RREPp, as long as, (1) the sequence number of the RREPp is larger than the one in the routing entry, (2) the sequence number is equal, but the RREPp is the first one arriving and the RREPp hop count (NHC) is lower or equal to the one in the routing entry (OHC) plus 2 (NHC can be larger than OHC), or (3) the sequence number is equal and the RREPp hop count is lower than the one in the routing entry. The limitation on the hop count, as already stated, is imposed to prevent routing loops. Figure 2 includes such an example where a RREPp message is discarded because the number of hops is 3 units larger than the one of the previous route. In the figure, node 1 receives a RREQp from node 3, since this node is in the same radio range. If its determined LET is large enough, this node can send a RREPp, because it has no idea that it is the previous hop of node 2 (which is the previous hop of node 3). If this reply is accepted, there will be a routing loop. Since the hop count is 3 units larger than the one of the previous route, this reply will not be accepted. As it is not possible to detect these routing loops, one way of preventing them is limiting the hop count of the new route.

When several RREPp arrive with the same sequence number and the same hop count, the sub-path can be chosen based on its stability, that is, based on the minimum LET of the new sub-path contained in the RREPp message.

When a routing entry is updated due to reception of a RREPp, the associated backoff timer is canceled. If the new established sub-path has a larger hop count than the replaced one, a RERR with a 'N' flag set shall be sent to all the precursors of the affected destination in order to notify the sources about the new hop count, as described in [4]. Upon receiving this message, a source may choose to initiate a route discovery process, if the path hop count increase becomes too large. Alternatively, for long lived route entries, a node can periodically send RREQs with a 'D' flag set, indicating that only the destination may answer.

Notice that the PLRR procedure only needs the neighbors location and movement information, which can be achieved only with HELLO messages, precluding the resort to other location-aided network wide mechanisms. The use of mobility information can also be extended to route establishment in the current AODV in order to improve the results of the discovery process. Useless routes can be avoided if nodes only process RREQs coming through stable links, using the stability concept defined in this paper.

4 Conclusions and Future Work

In this paper we presented a proposal to extend AODV with preemptive local routing repair. This extension uses location and mobility information of the neighbors propagated through HELLO messages, to predict the breakage of a link, and actively repair it before it breaks. Extensions to AODV messages and their processing were also proposed in order to optimize the preemptive repair process. Future plans include investigation of triangulation algorithms in order to obtain relative locations and movements.

In the final paper we will present a complete description of the AODV PLRR procedure, and some performance results concerning QoS metrics, control message overhead, and the impact of AODV PLRR in the number of hops in paths. Three simulated models will be presented in distinct mobility scenarios: AODV, AODV with local repair and AODV PLRR.

References

[1] A.J. Cahill, P.L. De Leon, and C.J. Sreenan. Link cache extensions for predictive routing and repair in ad hoc wireless networks. In *Proceedings of the 4th IEEE Conference on Mobile and Wireless Communication Networks (MWCN)*, pages 43–52, September 2002.

[2] Tom Goff, Nael B. Abu-Ghazaleh, Dhananjay S. Phatak, and Ridvan Kahvecioglu. Preemptive routing in ad hoc networks. In *Proc. 7th Int. Conference on Mobile Computing and Networking*, pages 43–52, 2001.

[3] David B. Johnson, David A. Maltz, and Yih-Chun Hu. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). INTERNET-DRAFT <draft-ietf-manet-dsr-09.txt>, Internet Engineering Task Force, April 2003.

[4] C. Perkins and E. Belding-Royer. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, Internet Engineering Task Force, July 2003.

[5] W. Su, S. Lee, and M. Gerla. Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management, Wiley & Sons*, 2000.