2004 International Workshop on Wireless Ad-Hoc Networks

Improving AODV with Preemptive Local Route Repair

Sérgio Crisóstomo[1], Susana Sargento[1,2], Pedro Brandão[1], Rui Prior[1]
[1]DCC-FC & LIACC, University of Porto, Portugal
[2]Instituto de Telecomunicações / University of Aveiro, Portugal
{slc, ssargento, pbrandao, rprior}@ncc.up.pt

***Abstract*—This paper proposes 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 neighbours 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 neighbours will break. 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 neighbours. This proposal aims at improving the AODV Quality of Service capabilities by minimizing route failures.**

***Index Terms*—Ad-hoc routing optimization, AODV, mobility information, preemptive local repair.**

I. 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. 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) [3]. This protocol is reactive, meaning that the search for a route from a 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 link actually breaks and it is required to send a packet through the 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. When local repair is used, the time for the route repair is usually lower. However, losses and delays still exist, since the path is already broken during the route repair procedure. Hence, one way of solving the route breakage problem is to predict the time instant the link will break and repair the route before it actually breaks.

There are already some proposals to use mobility prediction and to introduce preemptive route repair in ad hoc networks [1], [2], [4], [5]. However, the work being carried out concerning mobility prediction in AODV is in its early stage. This protocol is reactive and the only information available concerning a route to a destination is the next hop and the hop count. The scarce available information in a node about the other nodes when using AODV, makes even more difficult to apply mobility prediction. However, since this is one of the winning ad hoc routing protocols in IETF, it is urgent to have a proposal for preemptive local route repair in AODV, decreasing the dependence between its performance (packet losses and delays) and the node's mobility.

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. Our proposal is to enhance nodes information concerning link stability to its neighbours resorting to HELLO messages. These messages are appended with a new 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 neighbours will break. When a node determines that a link to a neighbour node is about to break, using the information on the location and mobility patterns of both nodes, it triggers the local repair procedure to find a new route to the destination, or to a node with a stable route to the destination, before the previous link actually breaks. This procedure will decrease the packet loss probability, since the possibility of having a failed route before a new stable route is discovered is mitigated. Therefore, with preemptive local repair, the Quality of Service (QoS) capabilities of the ad hoc network can be improved by minimizing route failures. The communication stalls in ongoing sessions are highly decreased, as well as packet losses and delays. 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; however this can be minimized in the protocol. This paper is organized as follows. In section II, some proposals for mobility prediction and preemptive route repair are addressed. In section III, the location prediction process is described and the extensions to the AODV message formats are presented. Next, section IV presents the procedure for preemptive route repair, the messages exchange and its processing. Finally, section V addresses the most important conclusions from this work and some topics for future research.

II. Related Work

In this section we list only some important approaches to give an idea of the research conducted in this area and to contrast these approaches with our proposal.

In [4] a predictive location-based QoS routing protocol, applied to a proactive distance vector protocol, is proposed. In this protocol, the information on geographical location and resource availability is distributed along the ad hoc network. There are two types of updates for this information: an update generated periodically, and an update generated when there is a considerable change in the node's velocity or motion direction. This information will help on the prediction of the node's movement, and then will help the routing protocol to adapt the routes considering this mobility. However, this protocol considers that updates from a node in the network are propagated to other nodes by broadcast flooding. In contrast, our approach considers the propagation of mobility information only to the neighbours, through HELLO messages, not flooding the network, which significantly decreases the routing overhead.

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, and re-routing is performed before the link breakage. These mechanisms are applied to a new network-wide ad-hoc on-demand routing protocol called FORP (Flow Oriented Routing Protocol). In this protocol the destination selects a route based on the greatest estimated path duration. FORP does not consider local route repair. It has an overhead due to the network-wide flood based route repair (initiated by destination) and to the mobility information piggybacked in data packets.

In [2] received transmission power is used to estimate when a link is about to break. An algorithm is proposed for on-demand routing. In the imminence of a link break, a warning is sent to the source(s) triggering the initiation of a network-wide route discovery. The direct appliance of this protocol is to DSR, where nodes have information on the overall routing path from the source to the destination. Some extensions to the AODV protocol are only considered in a very superficial way.

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. An appliance of this proposal to the AODV protocol is not possible, since the nature of these two protocols is very different: in DSR nodes have a full knowledge of the path, whereas in AODV nodes only know the number of hops and the next hop to a given destination.

Resuming, there is no proposal addressing preemptive local route repair in AODV. As was referred, since AODV is one of the winning protocols in IETF, it really needs to be improved with this capability, so it can be used in ad hoc networks supporting real time services with higher QoS constraints.

III. 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 neighbours. As already stated, the location and motion pattern information of each neighbour 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 knowing this information from a neighbour sharing an active path. HELLO messages are periodically broadcasted to neighbours, only if during the last HELLO time interval there was no broadcast of a RREQ or RERR message [3]. To provide the neighbour nodes with predictive information on the links' duration, we defined a new extension to be appended to HELLOs and to AODV control messages[1], 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

Length: 18

Timestamp: in ms; range: [0, 9999]

Longitude: in hundredths of second

Longitude: in hundredths of second

Velocity: in cm/s, range: [0:2^16−1]

Angle: in units of 2PI/10000 rad, range: [0,9999]

Position Error: maximum position error in meters, range: [0, 255]

Fig. 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 position accuracy of the node. 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.

[1]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.

Using the information of the mobility extension sent from a neighbour N_j, node N_i updates its mobility information about N_j, and determines the (predicted) amount of time they will stay in range, denoted Link Expiration Time (LET), using the proposed expression (1), which is an extension of the LET expression presented in [5]:

$$LET_{i,j} = \frac{-(a\times b + c\times d) + \sqrt{(a^2+c^2)\times(r - errpos_i - errpos_j)^2 - (a\times d - b\times c)^2}}{a^2+c^2} \quad (1)$$

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.

Expression (1) extends the one presented in [5] in the inclusion of the maximum position error (*errpos*) of neighbours 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 neighbour 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. PLRR_DISCOVERY_TIME will be equally initialized (will be the same) for all nodes in the ad hoc network.

IV. Preemptive Local Route Repair Procedure

When receiving a HELLO message from node N_j, node N_i determines $LET_{i,j}$. Then, the node decides if the link between itself and N_j is about to break, and if so, it initiates a PLRR procedure. We will consider that a link between node N_i and N_j will break in $LET_{i,j}$ seconds. The PLRR procedure must be initiated PLRR_DISCOVERY_TIME before the expected break so to guarantee that the PLRR procedure has enough time to be accomplished.

When a link 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), respectively, 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 (old route expiring after successful 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 or other path towards the destination;
4) 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. Although in reactive route repair procedures of normal AODV, the destination sequence number in route request messages is increased by 1 (to avoid routing loops), in PLRR procedures, it must be the same as the one assigned in the corresponding destination route entry of the node initiating the PLRR. This increases the chance of finding a sub-path that joins the previous path in the node's vicinity. The problem of possibly inducing routing loops with this approach is handled as stated next:
    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 (HC) 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.
5) Any node that receives a RREQp from a neighbour 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.

Figure 2 illustrates the PLRR procedure. In the following sub-sections we will describe the PLRR operation, highlighting its specific characteristics. The example in the figure will be used to illustrate the PLRR process.

A. Generating RREQp

Consider *Dset* as the set of all affected destinations due to a predicted link break. Every route entry of a destination in *Dset* must be marked as being in a PLRR procedure, denoting that these routes are in a process of PLRR but are still *valid*.

To illustrate the PLRR procedure consider the example of figure 2. Considering that the link between nodes 3 and 4 is about to break, for each of the affected destinations, a RREQp is generated and broadcasted by node 3, with the last known

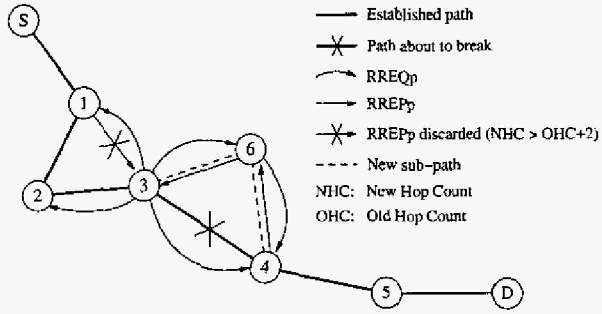

Fig. 2. PLRR illustration

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 *Dset* with an appended extension called PLRR_DEST_AGR (besides mobility extension), which is presented in figure 3. It contains all the remaining destinations in *Dset*, its correspondent sequence number and request identification (Request ID). The request identification value is required to identify the broadcast request for each destination and to guarantee that it will only be processed once by a given node. Beyond other trivial elements, this message also contains the number of carried destinations in the route request process. 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]			
Request ID [1]			
Additional Destination IP Address [2]			
Additional Destination Sequence Number [2]			
Request ID [2]			
...			

Type: 3 NoD: Number of Destinations (max. 21)

Length: variable

Fig. 3. PLRR_DEST_AGR extension

B. Processing and Forwarding RREQp

When a node receives a RREQp it must process the appended mobility extension to update the mobility information of the neighbour 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 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 neighbour nodes, precluding the possibility of using other nodes (possibly more stable); (2) 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.

The RREQp processing is depicted in algorithm 1. When receiving a RREQp, 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. Otherwise, it determines the link expire time and checks if this value is smaller than twice the PLRR_DISCOVERY_TIME. If it is, the message must be silently discarded. 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, in our example). Otherwise, a RREPp is sent to the node that initiated the route repair. If a new RREQp is to be forwarded, the mobility extension referring to the previous hop must be discarded, and a new one referring to the current node added.

Algorithm 1 Processing RREQp

1) Node calculates $LET_{i,j}$
2) **If** *valid route entry* **and** *next_hop = RREQp source*
    a) Discard RREQp
3) **Else if** $LET_{i,j}$ < *2 x PLRR_DISCOVERY_TIME*
    a) Discard RREQp
4) **Else if** *node* $\neq$ *destination* **and** (*no valid route entry* **or** *route entry sequence number < RREQp sequence number*)
    a) Forward RREQp
5) **Else**
    a) Generate RREPp and send towards RREQp source

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.

C. Generating RREPp

The RREPp is sent when a stable sub-path is 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 shall be updated in a hop by hop basis throughout 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.

In the case of no stable sub-path found, no route RREPp is generated.

D. Receiving and Forwarding Route Replies

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 algorithm used for route entry updates when a RREPp arrives at the source of the RREQp is depicted in algorithm 2. 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 defined in the RREPp message.

Algorithm 2 Processing RREPp

1) RREQp source receives RREPp
2) **If** *RREPp sequence number > route entry sequence number*
   a) Update route entry
3) **Else if** *RREPp sequence number = route entry sequence number* **and** *RREPp HC $\leq$ route entry HC + 2* **and** first RREPp received
   a) Update route entry
4) **Else if** *sequence number = route entry sequence number* **and** (*RREPp HC < route entry HC* **or** (*RREPp HC = route entry HC* **and** *LET is larger*))
   a) Update route entry

Now, a new sub-path is established, and the packets start using the new sub-path. The process is performed in such a way that there is a high probability of the establishement of this new sub-path before the previous one breaks. This way, losses due to nodes mobility will be minimized. If at some time instant, a link of the new sub-path reaches the horizon limit, that is, if it is about to break, another PLRR procedure will be applied for that link.

E. Optimization to the PLRR procedure

As was already stated, PLRR can have the side effect of increasing the number of hops in the path. Two alternatives are being considered to solve this problem. First, an increase in route hop counts can be reported to the corresponding sources by means of RERR with the N flag set, whenever the route hop count exceeds some threshold times the initial hop count to a particular destination. Upon receiving this message, a route discovery process should be initiated by interested sources. Second, for long lived route entries, a node should periodically send RREQs with a D flag set, indicating that only the destination may answer.

Notice that the PLRR procedure only needs the neighbours 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 the establishment of routes in the actual AODV (beyond local route repair) in order to improve the results of the discovery process. Useless routes can be avoided if nodes only process RREQs coming through stable links, with the stability concept defined in this paper.

V. 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 neighbours 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.

As future work, we are implementing the PLRR algorithm in the ns-2 simulator, and we plan to address PLRR 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 studied in distinct mobility scenarios: AODV, AODV with local repair and AODV extended with PLRR.

References

[1] A. Cahill, P. D. Leon, and C. 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)*, September 2002, pp. 43–52.

[2] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu, "Preemptive routing in ad hoc networks," in *Proc. 7th Int. Conference on Mobile Computing and Networking*, 2001, pp. 43–52.

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

[4] S. H. Shah and K. Nahrstedt., "Predictive location-based QoS routing in mobile ad hoc networks," in *in Proc. of IEEE International Conference on Communications (ICC 2002), New York, NY, April 28th - May 2nd, 2002.*

[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.