PATCH: *A Novel Local Recovery Mechanism for Mobile Ad-hoc Networks*

Genping Liu[1], Kai Juan Wong[1], Bu Sung Lee[1], Boon Chong Seet[2], Chuan Heng Foh[1], Lijuan Zhu[1],
[1]School of Computer Engineering
[2]Network Technology Research Centre
Nanyang Technological University Singapore 639798

***Abstract*--On-demand routing protocol is an important category of the current ad-hoc routing protocols, in which a route between a communicating node pair is discovered only on demand. However, due to the dynamic and mobile nature of the nodes, intermediate nodes in the route tend to lose connection with each other during the communication process. When this occurs, an end-to-end route discovery is typically performed to establish a new connection for the communication. Such route repair mechanism causes high control overhead and long packet delay. In this paper, we propose a Proximity Approach To Connection Healing (PATCH) local recovery mechanism, which aims to reduce the control overhead and achieve fast recovery when route breakage happens. It is shown that PATCH is simple, robust and effective. We present simulation results to illustrate the performance benefits of using PATCH mechanism.**

***Keywords*--Routing Protocols, Local recovery, Mobile Ad-hoc Network (MANET)**

I. INTRODUCTION

A Mobile Ad-hoc network (MANET) is an autonomous system that is made up of mobile nodes communicating through wireless links in an environment without the need for any static infrastructure. Nodes in this network are self-organizing and rely on each other to relay messages between nodes. As the nodes are free to move around randomly, the network topology changes dynamically. Thus the routing protocol must be adaptive and able to maintain routes in spite of the changing network connectivity. Such networks are very useful in military and other tactical applications such as emergency rescue or exploration missions, where cellular infrastructure is unavailable. Commercial applications are also likely where there is a need for ubiquitous communication services without the presence or use of a fixed infrastructure.

Routing is the most challenging problem in ad hoc networking. Much work has been done in this area and many protocols have been proposed. Of particular interest is the class of on-demand, source-initiated protocols, which set up and maintain routes from a source to a destination on an “as needed” basis. Dynamic Source Routing (DSR) [1] and Ad-hoc On-demand Distance Vector Protocol (AODV) [2] are two of the most popular on-demand routing protocol.

On-demand routing protocol relies on global flooding of route request packet for the route discovery. Once the source performs the route discovery successfully, data are sent via the discovered route. However, due to the dynamic and mobile nature of the nodes, intermediate nodes in the route tend to lose connection with each other during the communication process, that is, the route is likely to be broken. When this occurs, different protocols may take different actions to re-establish a route in order to maintain the communication channel between the source and destination nodes.

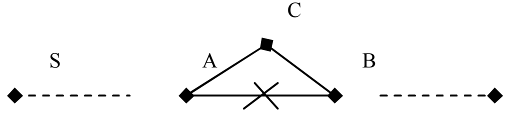


Figure 1. A typical routing scenario in MANET

For example, with reference to Figure 1, a route is discovered and established between the source node *S* and the destination node *D*. Nodes *A* and *B* are intermediate nodes in the established route. In the situation that the link between *A* and its next node *B* break off, the route between *S* and *D* is said to be broken. When DSR is used, *A* sends back an ERROR message to the data source *S* after which *S* initiates another round of route discovery process in an attempt to find a new route to *D*. This mechanism, used by DSR, is also know as end-to-end route recovery and is not scalable. That is to say, in larger maps, this mechanism may cause long delay, high overhead, and low delivery ratio. The ns [8] implementation of AODV, on the other hand, provides an option to overcome route breakages with a mechanism known as local recovery. Local recovery attempts to repair the broken route from intermediate node *A* instead of having the source node *S* performing the entire route discovery process again. In AODV, the optional local recovery mechanism is implemented as such: if intermediate node *A* finds that the link to next hop *B* breaks off, and if it finds that it is nearer to *D* than source *S* to itself, it will start a local recovery process which is the same as the normal route discovery with the only difference that *A* initiates the route discovery process instead of *S*. However, such local recovery still relies on the large area flooding, which consumes considerable amount of overheads.

The motivation behind the on-demand protocols is that the control overhead is typically less as only the actively used route is maintained. However, in a large or highly mobile network, where frequent route-broken happens, frequent route recovery relying on the global flooding will render long recovery delay, and high control overhead. Thus there is a need for better route repair mechanisms that should provide fast recovery, less control overhead. There is a number of works currently done on local route recovery, which include WAR (Witness-Aided Routing Protocol) [3], ABR (Associatively-Based Routing Protocol) [4, 5] and RDMAR (Relative Distance Microdiscovery Ad-hoc Routing Protocol) [6]. The Proximity Approach To Connection Healing (PATCH) mechanism introduced in the paper attempts to overcome some of their shortfalls while keeping complexity of the algorithm to a minimal. Complexity is an important attribute because nodes in MANET are generally small portable devices with little processing power, thus, the local recovery scheme must also cater to this property.

The rest of this paper is organized as follows. In Section 2, several local route repair mechanisms in WAR, ABR, and RDMAR are briefly reviewed. In Section 3, the proposed local recovery mechanism is described in detail. In Section 4, the performance of the PATCH mechanism is evaluated via simulation. Finally, some important conclusions are drawn in Section 5.

II. EXISTING ROUTE REPAIR MECHANISMS

WAR [3] is designed to use witness hosts to overcome the transient unidirectional links. Upon link breakage happens, it perform local recovery by broadcasting of the data packets with a predefined hop limits. This way it provide fast local recovery for data packet, but it then induce high control overhead as the data packet is broadcasted as a recovery packet. ABR [4, 5] employs an associability based routing scheme to select the routes likely to be long-lived. However, if link breakage occurs, two cases could happen. If *A* is located at the first half of the route (i.e, it is nearer to the source than to the destination), then a route error is reported to the source, and the source will initiate an end-to-end route discovery to recover the route. Otherwise, *A* will broadcast a route request with a hop limit equal to the remaining number of hops that it was supposed to travel before the route is broken. Only the destination is able to reply to the route request. If this succeeds, this route is remedied and no route error will be reported. Otherwise, a route error will be reported to the host preceding *A*, which will in turn repeat trying the above two cases again. This is recursively repeated until either the broken route is remedied or one host at the first half of the original route is reached. As can be seen, this approach may take more bandwidth and longer delay if the above recursion keeps on failing. RDMAR [6, 7] employs a similar approach of local repair as ABR. However, the region of the localized route repair is estimated from the history distance between the current node *A* and destination *D* using a location prediction model.

III. PROXIMITY APPROACH TO CONNECTION HEALING ALGORITHM

With reference to Figure 1, if the direct link (*A->B*) breaks off, there should exist, in most cases, some indirect route from *A* to the original next node *B* through some neighbor node *C*. In these situations, if a request packet is sent out to find the original next hop or other node which is at the further part of the original route with limited time-to-live (eg. 2 hops), the possibility of repairing the current route should be high and the overhead should be much lower than using end-to-end global recovery. With the above assumptions, a new algorithm for the proposed PATCH mechanism is described as follows:

When an intermediate node *A* discovers that the link to the next hop has broken, it would:

a. Save the data packet into local repair data buffer, set a timeout timer with a timeout of 0.1s (can be tuned)

b. Send out a two-hop wide local recovery request, containing a sequence of nodes including all the further nodes on the original route from *A* till the destination node *D*. Any node, on receiving the local recovery request, will check whether it is on the node list, and if it is, would send back a local recovery reply

c. On receiving local recovery request from an intermediate node, *A* salvages the data packet at the local repair buffer and transmits it using the repaired route. At the same time, it would send the repaired route and broken link information back to the source *S*. The source would use the repaired route for future communication.

d. If no reply received after the local repair timer expires, the data packet is dropped, and an error message is send back to the source from *A*, the source may then initiate a new round of end-to-end route discovery

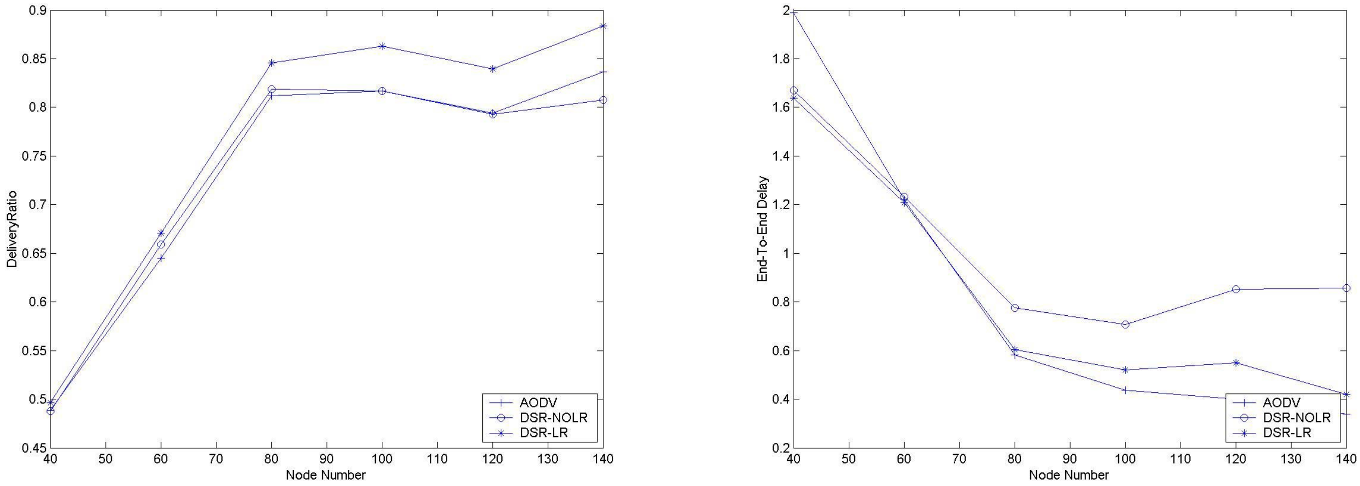
In this algorithm, the propagation limit of the local recovery request can be manipulated. The larger the propagation limit, the higher the control overhead, but the higher possibility the local recovery would succeed. PATCH tries to repair the broken route locally with minimum control overhead. As the next hop resides most likely 2 hops away at the time the breakage happens, the route would be repaired quickly. Thus the data packet can be salvaged faster, and the communication goes smoothly with less interruption. In this way, the fast recovery is achieved with low control overhead.

IV. SIMULATIONS

A. *Simulation settings*

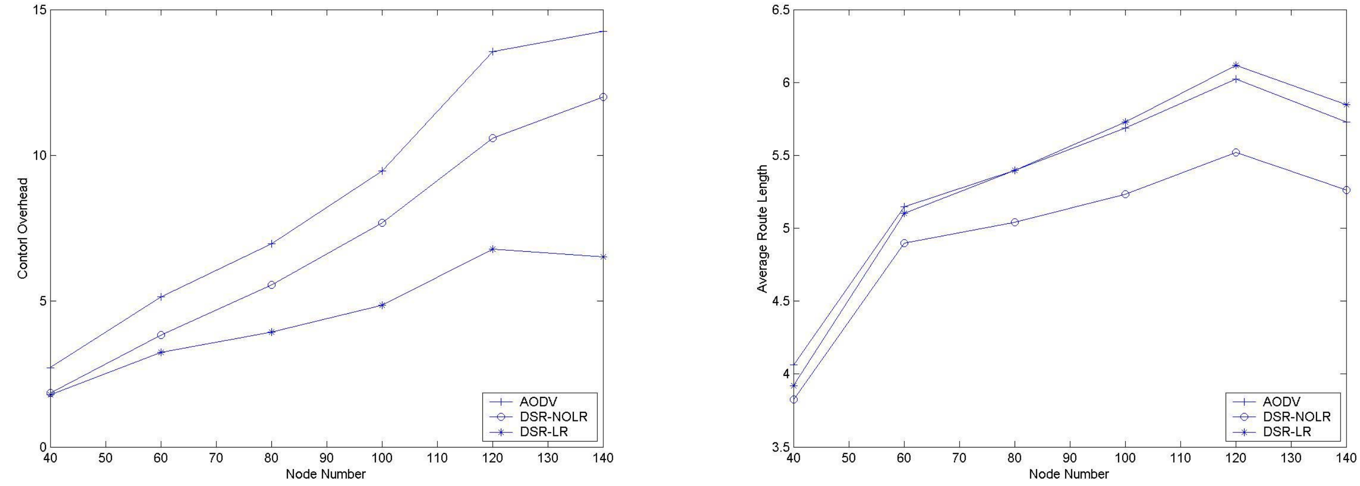
Simulations were carried out using ns [8], a packet level simulator, to evaluate the proposed local recovery mechanism. The original DSR in ns is extended to include PATCH. Simulation is conducted and performance comparison is taken on the original DSR (DSR-NOLR), DSR with the proposed local recovery mechanism (DSR-LR) and AODV. The simulation scenario parameters are presented in table 1.

For our simulations, high mobility is used such that consistent breakages in the routes can be observed. 20 pairs of CBR transmissions are used to simulate the moderate traffic load with which the control overhead is usually high. To emphasize the effectiveness of the proposed mechanism, a long map of 4000x300m^2 was used, such that the average route length is generally long. However, considerable improvements are also seen from simulations run on a broad map of 2000x1600m^2. Lastly, simulations were run across various densities. With increasing density, the average degree (number of neighboring node) of the node keeps increasing, thus the possibility of successful local recovery is also increased. Results of the simulations were then compared based on Data Delivery Ratio, End-to-End Delay, Control Overhead and



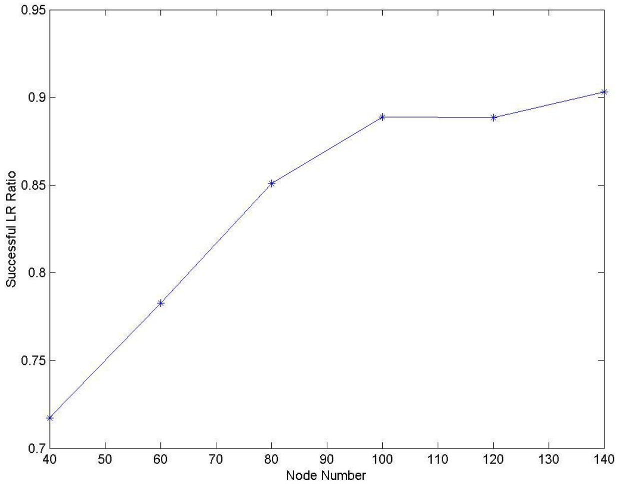
(a) Delivery Ratio

(b) End-To-End Delay



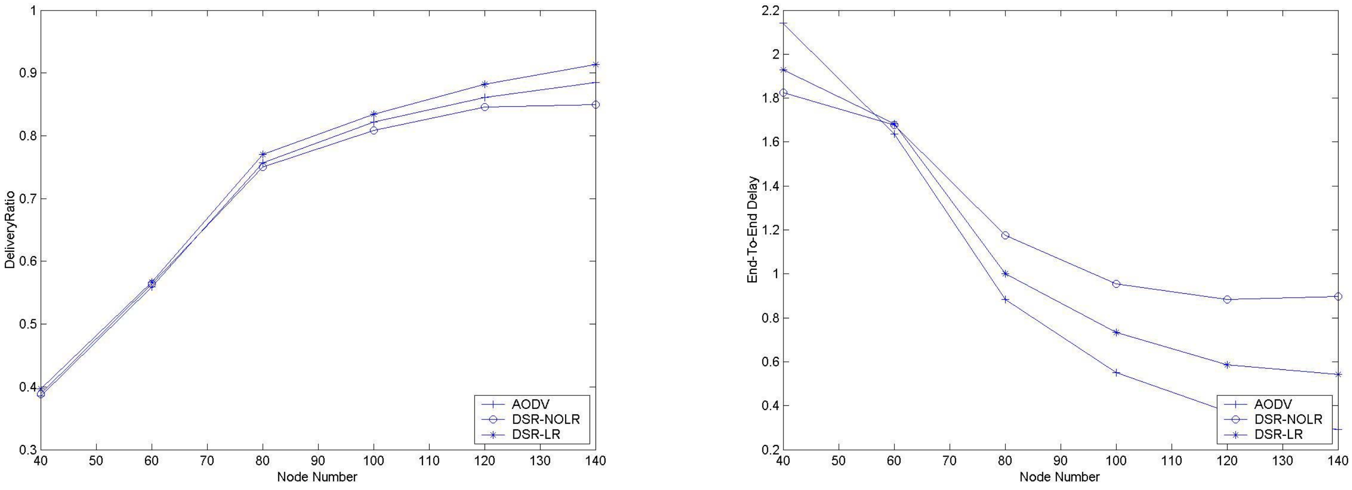
(c) Control Overhead

(d) Average Route Length



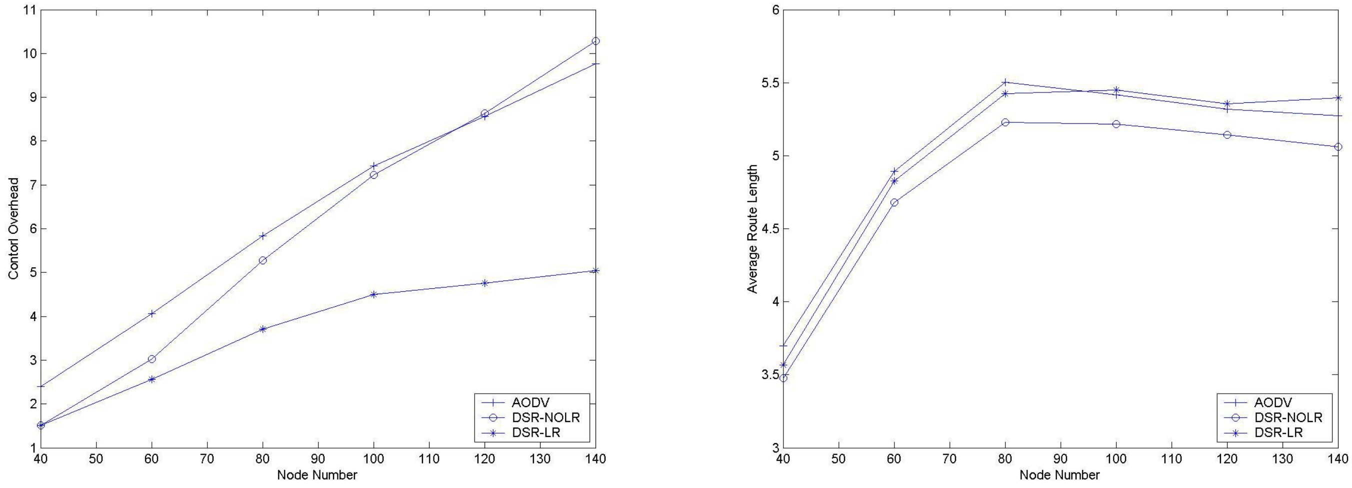
(e) Successful Local Recovery Ratio

Figure 2: Simulation results of a 4000x300m^2 map



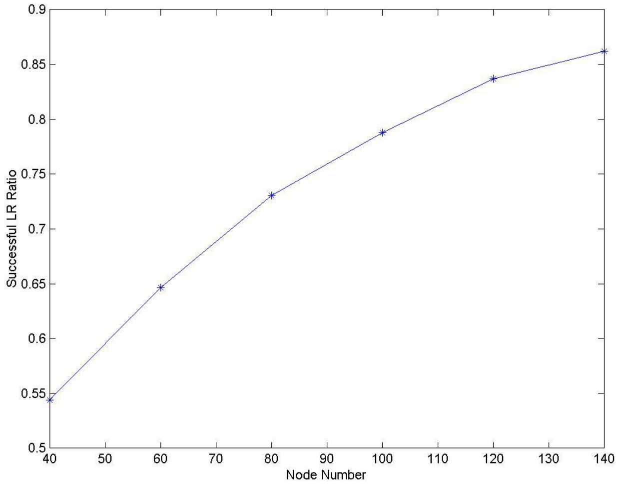
(a) Delivery Ratio

(b) End-To-End Delay



(c) Control Overhead

(d) Average Route Length



(e) Successful Local Recovery Ratio

Figure 3. Simulation results on a 2000x1600m^2 map

Average Route Length. The ratio of successful local recovery for various densities is collected. All the result is averaged over 20 sets of simulation runs.

Table 1: Simulation parameters

Mobility pattern	Random way point
Traffic	20 pairs of CBR transmission
Transmission range	250m
Mobility	pause time 20s, speed 0-20m/s
Map	4000x300m^2, 2000x1600m^2
Node number	40, 60, 80, 100, 120, 140
Simulation time	200s

B. *Simulation result and analysis*

Figure 2 shows the simulation results for the long map of 4000x300m^2. From Figure 2(e), we can see that the average successful local recovery ratio keeps increasing as the density increases. At low density, PATCH does not show much advantage. Especially when node number equals to 40, the connectivity of the whole network is not quite good and partitioning is severe. Most of the transmission is successful only in small partitions with short route length. In such situation, the local recovery covers most portion of the whole partition already, thus we cannot see obvious control packet saving at low density. The data delivery ratio gets only little improvement at low density. However, as the density goes higher, the connectivity of the network becomes higher, transmission with longer route length can be formed. In such a situation, the local recovery starts to show obvious advantage over end-to-end recover scheme, as the local recovery only floods the request in a small region while the end-to-end recovery floods the entire network. Successful local recovery ratio goes higher as the average degree of node goes higher. The overall control overhead saving becomes very obvious, and reaches as high as 50% at high density, compared with both DSR and AODV. Because of source routing, DSR generally shows longer delay than AODV. With PATCH, benefited from both the reduction of the congestion level caused by control packet and the fast recovery, the end-to-end delay decreases as much as 50% of the delay of the normal DSR, and comparable with AODV. When link breakage happens, PATCH helps salvage the data packet locally using the repaired route, thus improves the data delivery ratio as much as 10%. Seen from Figure 2(d), local recovery causes longer route length than normal DSR, but still comparable with AODV.

Figure 3 shows the simulation results for broad map of 2000x1600m^2. As this map get much wider area than the earlier long map, thus for the same node number, the average degree of this scenario is much lower. Seen from Figure 3(e), the successful local recovery ratio is always lower than that in the long map with the same number of nodes. Comparable performance improvement can still be observed in such a map.

Based on the algorithm, PATCH only floods a small region, which is a circle with the radius of 2 hops distance. Thus it outperforms end-to-end recovery in scenarios with big map size and high density, in which it only relies a small percentage of the entire network to achieve the route recovery process.

V. CONCLUSION

We have presented a new scalable local recovery mechanism, called PATCH, for mobile ad-hoc networks. In contrast to the end-to-end recovery mechanism, PATCH relies on broadcast of route request within the region where breakage happens, to repair the broken route quickly, thus providing a way for fast recovery. As PATCH only floods a small region, which is a circle with the radius of 2 hops distance, it will mostly outperform end-to-end recovery in scenarios with big map size and high density, in which it only relies on a small percentage of the whole network to achieve the route recovery process. Hence, it is a scalable solution for the route re-discovery process.

On the whole, vast improvements could be seen with the implementation of the proposed local recovery scheme in DSR. DSR with PATCH achieves higher data delivery ratio all the time, and which can be seen as high as 10% improvement over normal DSR. Control overheads are dramatically reduced to only 30-50% of the normal DSR or AODV. The saving in bandwidth is especially high in the scenario with high density. Furthermore, PATCH also reduces the end-to-end delay significantly, which reaches as high as 50% compared to the normal DSR, and is comparable with AODV. The only shortfall of PATCH is that the result shows a higher average route length than that of the normal DSR but gets comparable result as AODV. Sending gratuitous reply in DSR has eased this problem of increasing route length. By incorporating further “route-shorting” algorithm such as the techniques in [9] into the proposed mechanism, it will further ease the problem of increasing route length and this would be the theme of our future work.

REFERENCE

[1] D.B. Johnson and D. A. Maltz, “Dynamic Source Routing Protocol for Mobile Ad Hoc Networks”, Mobile Computing, T. Imielinski and H. Korth, Eds., Kluwer, 1996, pp. 153-81.

[2] C.E. Perkins and E. M. Royer, “Ad-hoc On-Demand Distance Vectored Routing”, Proc. 2nd IEEE Workshop. Mobile Comp. Sys. and Apps., Feb. 1999, pp. 90-100.

[3] Aron and S. Gupta. “A Witness-Aided Routing Protocol for Mobile Ad-Hoc Networks with Unidirectional Links”, Proc. First lnt'l Conf. on Mobile Data Access (MDA '99), Hong-Kong, Dec. 1999, pp. 24-33.

[4] C-K. Toh, "A novel distributed routing protocol to support ad-hoc mobile computing," IEEE International Phoenix Conf. on Computers and Communications, IPCCC'96.

[5] C. K. Toh, "Long-lived ad hoc routing based on the concept of associativity", Internet draft, IETF, Mar. 1999.

[6] George Aggelou , Rahim Tafazolli, “RDMAR: a bandwidth-efficient routing protocol for mobile ad hoc networks”, Proceedings of the 2nd ACM international workshop on Wireless mobile multimedia, Seattle, Washington, United States, August 20-20, 1999, pp.26-33.

[7] Aggelou G, Tafazolli R. “A Simulation Analysis on reactive Route Repair techniques for QoS sensitive applications in Mobile Ad Hoc networks”, In Proceedings of the First Annual Workshop on Mobile Ad Hoc Networking & Computing (MobiHoc), Boston, USA, August 2000.

[8] Ns-2 simulation, http://www.isi.edu/nsnam/

[9] C. Gui and P. Mohapatra, “A Self-Healing and Optimizing Routing Technique for Ad Hoc Networks”, Computer Science Department, UC Davis, Technical Report CSE-2002-23.