

# Study and Improvement on Optimal Reactive Routing Protocol (ORRP) for Mobile Ad-Hoc Networks

Soma Saha<sup>1</sup>, Tamojay Deb<sup>2</sup>

<sup>1</sup>Women's Polytechnic, Dept. of IT, Hapania, Tripura

<sup>2</sup>Dept. of IT, Dasaratha Deb Memorial College, Khowai, Tripura  
{somasaha84, tamojaydeb}@gmail.com

**Abstract:** In MANET, mobile nodes dynamically form temporary networks without using conventional infrastructure or centralized administration. In this paper, we have improved Optimal Reactive Routing Protocol (ORRP) [2], an existing on-demand route discovery approach that returns the shortest path in between a source-destination pair. ORRP does not use flooding. It finds the optimal route based on a cost vector. However, the protocol ORRP does not mention any effective way to compute this cost vector. This paper is a significant and two-fold extension of ORRP. We have worked on some of the basic incompleteness of ORRP. The most significant contribution is in incorporating a periodic HELLO message exchange for sensing neighborhood as well as for determination of cost vector.

**Keywords:** MANET, Reactive, Optimal, Secure, Beaconing, Reactive

## 1. Introduction

A Mobile Ad-hoc network (MANET) is a kind of wireless network where the participating nodes dynamically and arbitrarily forms a network. In such a network, each mobile node operates not only as a host but also as a router[1]. MANET is an infrastructure-less network where, there are no routers, servers, access points or cables. Participating nodes can move freely and in arbitrary ways, so it may change its location from time to time. In order to enable communication between any two nodes, a routing protocol is employed. The duty of the routing protocol in MANET is to discover the topology to ensure that each node has the recent image of the network topology to construct routes for communication. Currently, two complementary classes of routing protocols exist in the MANET world. Reactive protocols (such as AODV and DSR) acquire routes on demand, while the proactive protocols (such as OLSR, OSPF, DSDV) ensure that topological information is maintained through periodic message exchange. In both the cases it is necessary for one

mobile node to enlist atleast its neighboring nodes in forwarding a packet to its destination due to the limited transmission range of wireless network interfaces. When a node in a MANET wants to communicate with another node, which is out of the transmission range of the first node then the intermediate nodes act as router to forward the packet.

The main idea behind this paper is to study the ORRP protocol and to find out the short comings in the existing protocol. Then I have introduced a modified version of the original Optimal Reactive Routing Protocol (ORRP) [2]. In this protocol, I have added the concept of periodic HELLO message exchanging for neighbor sensing and Cost vector initialization. I am aiming to make a comparison study based on performance between the existing ORRP and the extended version proposed in this paper.

The rest of the paper is organized as follows: Section 2 presents a very brief review on reactive routing protocols for mobile ad hoc networks. In fact, as we would be working on Optimal Reactive Routing Protocol only, we need to study the existing ORRP in details. This is done in section 3. The newer version is built upon pointing the limitations and incompleteness of ORRP. Section 4 describes the proposed improved version with illustrative example to explain the operation of the improved protocol. In order to identify the improved version from the original protocol, we refer the newer one as ORRP-1 in rest of the paper. In section 5, we present a comparison study between ORRP and ORRP-1. Section 6 concludes the paper.

## 2. Related Works

A number of researches are done on routing protocols in MANET. I briefly outline the most relevant characteristics of them. Reactive protocols are on demand protocols that discover the route once needed (eg AODV [3]). The reactive protocols display

considerable bandwidth and overhead advantages over proactive protocols. AODV routing protocol offers quick adaptation to dynamic link conditions, low processing, low memory overheads, and low network utilization [4]. But it can't ensure loop free optimal route between two communicating nodes. Upon based on AODV several other routing protocols has been introduced. Some previous works [4] [5] [6] [7] are made on comparative study among those. Some new types of routing protocols also introduced to withstand the limitations of MANET i.e., limited bandwidth and a high degree of mobility. This work is based on ORRP [2] that loop free and optimal path every time.

### 3. Review on the ORRP

Optimal Reactive Routing Protocol (ORRP) is a reactive protocol that finds a loop-free, optimal path between the end nodes. In this paper I tried to expose the short comings in the existing ORRP[2] and to introduce the necessary modification. I refer the newer version of ORRP as ORRP-1 in this paper. ORRP like other reactive routing protocol is a source initiated routing algorithm. It assumes that at any given instance, any node in the network maintains a list of its neighbors and also stores the cost vectors to reach the neighboring nodes from the node. Any change in the topology including deletion of a host or a link must be communicated to the neighboring nodes.

ORRP assumes symmetric links between neighboring nodes. The working of ORRP is based on two basic operations: Procedure Update( ) and Procedure FindRoute( Ni,Nj). Procedure Update is responsible for neighbor sensing. If any node fails then the Update( ) procedure deletes the entry for that node in the list of its neighbors. Again if any node joins the network the same procedure is used for entering the proper information in appropriate place. On the other hand Procedure FindRoute(Ni,Nj) finds the shortest route between source node Ni and destination node Nj. The procedure makes use of Dijkstra's shortest path algorithm for finding path. ORRP does not compute all possible routes between a node and the remaining nodes like the proactive protocol, but it computes the shortest path from the information maintained by the participating nodes when any node wants to communicate with other node that keeps the routing overhead low.

#### 3.1 Shortcomings of ORRP

ORRP gives innovative idea being a reactive routing protocol but need some further modification before implementation. ORRP assumes each participating nodes maintains a list of its neighbors and the corresponding cost vector for each entry. But it remains

silent regarding the value assignment for the cost vectors. ORRP does not introduce any mechanism to store the intermediate routing information. ORRP introduced two procedures. Procedure FindRoute(Ni,Nj) is responsible for finding the optimal path . Procedure Update( ) maintains the information regarding random topology changes. ORRP tells about cost vector in both the procedures but it does not define any procedure to assign the cost vector. Moreover ORRP needs neighbors information of each node so that the Dijkstra's algorithm can be executed but it fails to introduce any such mechanism for neighbor sensing. ORRP-1 is an effort to overcome all those incompleteness present in ORRP so that it can be implemented.

### 4. The Proposed ORRP-1

The basic idea behind ORRP remains unchanged in ORRP-1. But it includes few extensions to eliminate some of the deficiencies relate to the cost assignment and neighbor sensing. ORRP-1 makes use of periodic HELLO message exchange for implementing those.

#### 4.1 Periodic beaconing in ORRP

In this approach, each node periodically broadcasts a HELLO message to its neighbors, so that each node has the local knowledge of all its neighbors and cost vector assigned to each link. Basically, neighbor sensing is the process through which a node detects changes to its neighborhood. The neighbor sensing mechanism in ORRP is designed to operate independently in the following way: each node periodically emits a HELLO-message every HELLO\_INTERVAL seconds, containing the node's own address as well as the list of neighbors known to the node, including the timestamp. Upon receiving HELLO-messages, a node can thus gather information describing its neighborhood. Each node maintains an information set, describing the neighbors. Such information is considered valid for a limited period of time, and must be refreshed at least periodically to remain valid. Expired information is purged from the neighbor sets.

A node may sometimes broadcast a triggered HELLO message in response to some event that needs quick action. When a node joins an existing network it will broadcast HELLO message within its radio transmission range. The proposed protocol assumes symmetric link between neighboring nodes. The HELLO message will identify the symmetric link and will assign the cost. We propose some additional data structures towards this. The cost of link will be determined as a function of time stamp assigned by both the adjacent nodes of a symmetric link. The neighbor set of a node  $N_i$  may be defined as follows:  $NS_i = \{N_j\}$ , where  $N_j$ : Node

adjacent to  $N_i$ , for  $I=1, \dots, m$ . The following expression represents the contents of a HELLO message from  $N_i$ :

**HELLO ( $N_i, ST_i, NS_i$ )**

IP address of $N_i$	Length	$ST_i$
Neighbor <sub>1</sub> IP address		
Neighbor <sub>2</sub> IP address		

**Figure 1: HELLO message**

**Field description**

Length: The number of neighbors listed

$ST_i$ : Time stamp of broadcasting the HELLO message by  $N_i$ .

Neighbor IP Address: The address of a neighbor. The IP addresses of the neighbors are taken from the RECORD table.

Each entry in the RECORD Table is associated with a timer. A table entry will be removed if a HELLO message from the entry's node is not received for a period of  $(HELLO\_LOSS) * HELLO\_INTERVAL$ , allowing HELLO\_LOSS consecutive HELLO messages to be lost from that node. If a node don't get any HELLO message from its neighbors listed in its RECORD table for more than  $(HELLO\_LOSS) * HELLO\_INTERVAL$  time it will discard that node from its RECORD table.

**HELLO message processing**

Upon receiving a HELLO message by  $N_j$  from  $N_i$ , the node  $N_j$  should update the neighbor entry corresponding to the sender node address. At first it will take the time stamp  $RT_j$ .

1. If the sender IP address does not exist in the RECORD table the receiving node it adds one entry for the sender in its RECORD table.

1.1 Then the receiver will generate another Control message to  $N_j$  and will send it back to  $N_i$  at time  $ST_i$ , where  $ST_j$  is the Time stamp of sending reply HELLO message by  $N_j$  upon receiving a HELLO message from  $N_i$ .

IP address of $N_i$	IP address of $N_j$	
$ST_i$	$RT_j$	$ST_j$

**Figure 2: Control message**

1.1.1. Node  $N_i$  will receive the control message at time  $RT_i$ . Node  $N_i$  will check whether  $N_j$  is present in its RECORD table or not. If  $N_j$  is not present it will add  $N_j$  to its record table and entry the cost associated with the link in between  $N_i$  and  $N_j$ . The cost vector  $C_{ij}$  can be

calculated using the formula:

$$C_{ij} = ((RT_j - ST_i) + (RT_i - ST_j)) / 2$$

1.1.2. If  $N_j$  is present in the RECORD table of  $N_i$ , then the cost vector will be calculated with the same formula and the entry will be updated with the newly derived  $C_{ij}$ .

2. If  $N_i$  is present in the RECORD table of  $N_j$  then also  $N_j$  will send back a reply - control message to  $N_i$

2.1 Upon receiving the control message by  $N_i$  from  $N_j$  the same function like step 1.1 will be carried out.

Time stamp of sending Hello message	Adjacent Nodes $\{N_j\}$ for $J \in [1..m]$	Cost $C[J]$
-------------------------------------	---	-------------

**Figure 3: RECORD table for  $N_i$**

Depending upon the Time stamp the most recent cost vector can be identified. The protocol, as evident here, does not require maintaining any other data structure like sequence numbering etc. We can determine the most recent cost vector and update the older one with newer one from the time stamp itself.

Due to the random mobility of the network, a node may get out of the transmission range of the other nodes within the network. Then it will not receive any HELLO message. As a result the link between that node and its adjacent nodes will break down. When any node or a link fails then it will send a failure message to its entire neighbor set. Upon receiving that failure message by the adjacent nodes those nodes will delete the entry for that node from its RECORD list.

The HELLO message is sent by a node to its entire neighbor at a fixed interval of time. But when the topology of the network changes or any link between two nodes breaks then the HELLO message is generated immediately to find the current state of the network.

**4.2 Procedure FindRoute( $N_s, N_d$ )**

ORRP-1 employees another procedure named **FindRoute ( $N_s, N_d$ )** [where  $N_s$ =source node and  $N_d$ =destination node] to identify the route between source and destination. EORRP ensures optimal shortest path discovery procedure as it employees Dijkstra algorithm.

For finding the routing route the source node  $N_s$  sends a message containing some special fields to its adjacent neighbor having the least weighted/cost link with the source from the RECORD table. The format of the message FindRoute( $N_s, N_d$ ) is:

1	2	3	4	5	6	7
FRP_seq	Ns	Nd	Nj	Nprev	Cprev_s	T

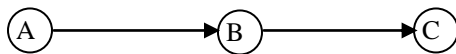
Figure 4: FindRoute() message format

**FRP\_seq::** The source node assigns a FRP\_seq i.e. FRP sequence no which remains fixed through out the route discovery process . FRP\_seq will be used during the propagation of routing route by the destination to the source.

**Ns::** Ns is the source node. But the value for Ns changes every time when a HELLO message is generated by the intermediate node during the route discovery process having same FRP\_seq no.

**Nd::** Nd is the destination node. It remains same until reached to the original destination.

**Nj:** Nj is the adjacent node of the source node having the least link cost value which is the next node in the source to destination route.



**Nprev:** Nprev is the source node of the current source node. As for example, A is the Nprev for node C

**Cprev\_s:** Cprev\_s is the cost associate with the source node and the source node of the current source node. Here cost between A and B is Cprev\_s.

**Field 7:=T:** The seventh field contains a token, that remains unaltered during the route finding procedure.

### 4.3 Algorithm for Finding Route

Step 1: The initial source node  $N_S$  assign value for FRP\_seq and select the adjacent neighbor having least link cost value and sends a FindRoute() message to that node.  $N_S$  puts NULL value in field 5 & 6 of the FindRoute() message.

Step 2: Upon receiving that message by the adjacent node  $N_J$  it will check whether it is the destination node or not.  
If ( $N_J = N_D$ ) goto step 6 Else goto step 3

Step 3: If ( $N_{PREV} = \text{NULL}$ ) then  
FindRoute() message is updated.  
 $N_{PREV} \leftarrow N_S$   
 $N_S \leftarrow N_J$   
FRP-Seq and  $N_D$  remains unaltered  
 $N_J$  is selected from the list of adjacent

nodes in RECORD table having least link cost value of newly selected source. The updated FindRoute() message is sent to the next adjacent node  $N_J$ .

Goto step 2

Else  
goto step 4

Step 4: Upon receiving the message by  $N_J$  it makes a search in its RECORD table for finding the cost associate with  $N_S$  i.e.  $C_{S,J}$  . And also in RECORD table of  $N_J$  a search will made to find whether  $N_{PREV}$  is present in the RECORD table of  $N_J$ .

If Nprev is present in the RECORD table of  $N_J$  then

goto step 4.1

Else  
goto step 4.2

Step 4.1: Consider the cost between Nprev and  $N_J$  from the RECORD table of  $N_J$  as  $C_{PREV,J}$ , Then perform the following:

If ( $C_{PREV,J} \leq (C_{S,J} + C_{PREV,S})$ ) then

Store FRP\_seq no and Nprev in the routing table.

Else  
Store FRP\_seq no and the IP address of  $N_S$ .

Step 4.2: Store FRP\_seq no and the IP address of  $N_S$ .

Step 5: If ( $N_J \neq N_D$ ) then

Update the content of the FindRoute message.

Keep values for field 1 and 3 unaltered.

$N_S \leftarrow N_J$

New  $N_J$  will be selected from the RECORD table of older  $N_J$

If ( $C_{PREV,J} \leq (C_{S,J} + C_{PREV,S})$ ) then

Nprev will remain unchanged.

Else

$N_{PREV} \leftarrow N_S$

$C_{PREV,S}$  will be cost between new  $N_S$  and new  $N_{PREV}$  from the RECORD table of new  $N_S$ .

Step 6: When ( $N_J = N_D$ ) then

The destination node  $N_D$  stores the content of the seventh field i.e. T.

The acknowledgement will be backtracked to the source node from the destination node. From the cache of  $N_D$  the previous node can be obtained. The acknowledgement will be sent to the previous node.

Upon receiving ACK message by Nprev it will compare the FRP\_seq no of ACK message with the FRP\_ack stored in its

cache.

If both are same, that message will be delivered to its previous node stored in its cache.

Thus, the ACK message will reach the original source node and the route is discovered.

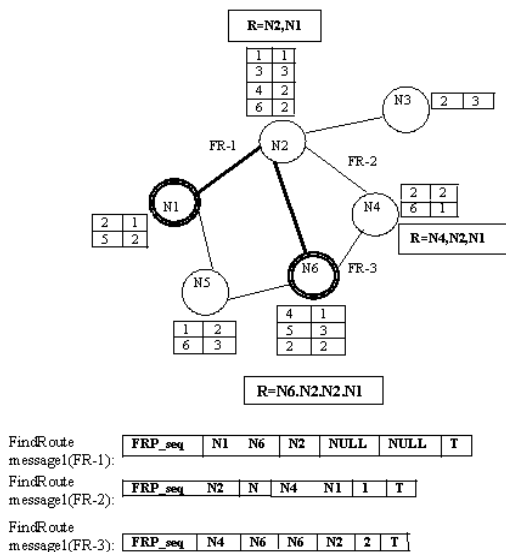


Figure 5. A configurable topology

### 4.3 Illustrative Example for ORRP-1

Figure 5 shows an ad hoc network with 6 host nodes. The links between each node are considered symmetric. The record table in the figure only considers two fields adjacent node and the cost determined by the HELLO message and control message exchange. N1 and N6 are source and destination respectively. N1 initiates the route finding procedures and the optimal route determined at the end of the procedure is N6, N2, N1.

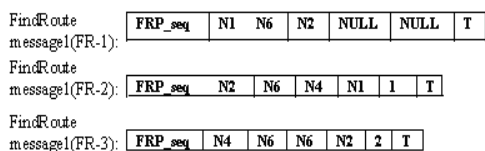
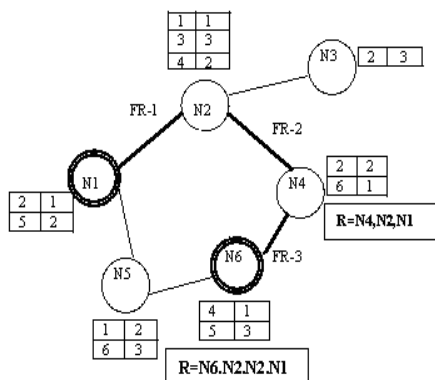


Figure 6. The revised topology

Due to mobility when node N6 comes within the transmission range of node N2 it will be sensed by N2 due to periodic HELLO message exchange and the record table will be updated accordingly. Figure 6 shows the revised topology and new route determined by FindRoute() procedure.

Likewise when any node leaves or enter the network it will be sensed by HELLO the other nodes and action will be taken accordingly.

## 5. Comparative Performance Study

In proactive routing protocols, the nodes keep updating their routing tables, by sending periodical messages. These tables require frequent updates for keeping the updated information regarding the network topology due to the mobility of participating nodes in MANET. A huge amount of bandwidth is wasted for periodic update of routing tables because the routing information are flooded in the whole network. These protocols require a huge amount of memory in order to maintain the global topology information by each node and require complex processing.

Reactive (On Demand) routing protocols, where routes are created only when needed. A route is established when a node wants to communicate with another one, which needs to broadcast HELLO messages that also consumes a considerable amount of bandwidth.

However, even though all possible routes between each and every nodes in the network are not determined before hand the protocols are comparatively simple then the proactive one. Reactive routing protocols can determine a path every time it is executed but can't ensure the optimal path every time.

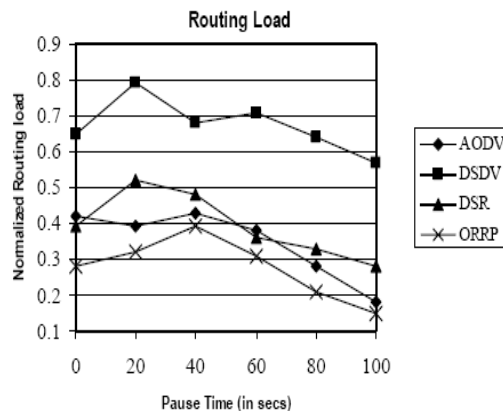


Figure 7: Study of comparative routing load

On the other hand ORRP being a reactive routing

protocol assure optimal path. ORRP uses Dijkstra's shortest path finding algorithm which always returns the shortest path within lower time bound. Only two procedures are involved in ORRP that are capable of finding the shortest path and updating of the network topology. From the implementation point of view ORRP requires some modifications that are incorporated in ORRP-1 presented in this paper.

The result of the simulation based on the routing load as the performance metric shows that the routing load for ORRP is lower than other three protocols AODV, DSR, DSDV[2]. The routing load has been measured in terms of the average number of control packets transmitted per data packet delivered at the destination. Routing load = packets sent / packets received.

ORRP-1 has all the advantages of ORRP. It also offers some more advantages. All the nodes simply keep information of its 1-hop neighbors and the cost vector of the symmetric link to each of them. To maintain this information the same HELLO message is used that are used for neighbor sensing. It involves less control message exchange. ORRP-1 does not broadcast the route request control message during route discovery, which saves a considerable amount of the bandwidth.

## 6. Conclusion

In this paper, the ORRP have been critically studied. Although the ORRP protocol has many advantages, major limitations have been identified towards implementing the protocol. In this paper, solutions are proposed to overcome this. We have introduced periodic HELLO message exchange. This provides an effective means to compute the cost vectors besides sensing the neighborhood. ORRP-1 is made of using the best characteristics of both reactive and proactive routing protocol. Like any other reactive routing protocols it determines the route between source and destination pair only on demand. The processing overheads in ORRP-1 are also reduced by efficient use of HELLO messaging and keeping the protocol to be executed on-demand. However it ensures the shortest path like most of the proactive routing protocols.

## References

- [1] H Yang, H Y. Luo, F Ye, S W. Lu, and L Zhang, "Security in mobile ad hoc networks: Challenges and solutions" (2004). IEEE Wireless Communications. 11 (1), pp. 38-47, 2004.
- [2] N. Chaki, R. Chaki; "ORRP: Optimal Reactive Routing Protocol for Mobile Ad-Hoc Networks", Proceedings of the IEEE Int'l Conf. on Computer

- Information Systems and Industrial Management Applications (CISIM 2007), pp. 185-190, 2007.
- [3] C.E. Perkins, E. Belding Royer, and S.R. Das, "Ad hoc On demand distance vector (AODV) routing", IETF RFC 3561, July 2003.
- [4] Mohammed F. Al-Hunaity, Prince Abdullah Bin Ghazi, "A Comparative Study between Various Protocols of MANET Networks". American Journal of Applied Sciences 4 (9):pp. 663-665, 2007.
- [5] E.M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Networks," IEEE Personal Communications, vol. 6, no. 2, Apr., 1999, pp. 46-55.
- [6] Clausen, Thomas Heide ; Jacquet, P.; Viennot, L. "Comparative Study of Routing Protocols for Mobile Ad-hoc NETWORKS ".IFIP Med Hoc Net 2002
- [7] Das, S.R. Perkins, C.E. Royer, E.M, "Performance comparison of two on-demand routing protocols for adhoc networks", INFOCOM 2000. 19<sup>th</sup> Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol.: 1, pp. 3-12, 2000

## Author Biographies



**Soma Saha** had born at Agartala, Tripura on 2<sup>nd</sup> May, 1984.

She did his BE in Computer Sc & Engg from N.I.T., Agartala, Tripura, India in 2006 and her Masters in Computer Sc & Engg. from Tripura University, Tripura, India in 2008. Her major fields of study are in Mobile Computing, Mobile ADHOC networks.



**Tamojay Deb** had born at Agartala, Tripura on 12<sup>th</sup> September, 1978.

He did his BE in Computer Sc & Engg from G.B.P.E.C., H. N. B. Garhwal University, Uttarakhand, India in 2002 and his Masters in Computer Sc & Engg. from Jadavpur University, Kolkata, India in 2009. His major fields of study are in Mobile ADHOC networks and Soft-computing (Image Processing).