International Journal of Computer Applications (0975 – 8887)
Volume 48– No.15, June 2012

ARRP: Azimuth Restricted Routing Protocol for Ad Hoc Networks

Reno Robert R
Department of Cyber Security
Amrita School of Engineering
Coimbatore, India

Ramachandran S
Sri Sairam Engineering College
Anna University
Chennai, India

Rajan P.S
Sri Sairam Engineering College
Anna University
Chennai, India

ABSTRACT

Ad Hoc Networks consists of set of mobile nodes forming spontaneous network. There is a need for an efficient routing protocol for Ad Hoc networks over the years. Bandwidth utilization and node power consumption are two important factors in wireless transmissions. This paper presents simulation model of new routing protocol ARRP (Azimuth Restricted Routing Protocol). ARRP uses position information of nodes and azimuth angle to define a range to restrict the broadcast of RREQ packets rather than flooding the entire network. This will reduce routing overhead in route discovery phase, saving wireless bandwidth and node power. Simulation of this protocol is done using Network Simulator 2.

General Terms

Wireless; Mobile; Routing; Range; Protocol; Position

Keywords

Ad hoc network; Azimuth; Bandwidth; Network Simulator

1. INTRODUCTION

Ad hoc networks are self-organized, dynamically changing multi-hop networks in which the nodes can acts as both host and routers. They never require a central administrator or existing infrastructure to route packets. Problems in ad hoc networks are the scarcity of bandwidth, power constraints of nodes, routing overhead and dynamic topology caused by the mobility of nodes. In order to overcome these problems, there is a need to design a routing protocol which consumes minimum bandwidth and also energy efficient.

Traditional routing in Ad Hoc is done using proactive, reactive protocols or hybrid protocols. Proactive protocols maintain information of other nodes using periodic control messages. Though this technique is not scalable, it enables to find various path through which data can be passed. On the other hand Reactive protocols like AODV update the network information only when they have data to send.

In this paper we propose a routing protocol that takes advantage of position information of nodes. Azimuth Restricted Routing Protocol computes the azimuth angle between the source and destination node position. Based on this azimuth angle, a flooding range is calculated. Intermediate nodes takes forwarding decision based on this flooding angle. Each node adds its location information into the RREQ and RREP it generates. RREQ additionally carry the computed range based on the destination position.

Whenever a node receives a control packet it updates the routing table, adding the position information. When the source has the position information of destination it range based flooding else it is broadcasted over the entire network. This protocol reduces routing overhead and bandwidth consumption significantly.

Rest of the paper is organized as follows: Section 2 describes the related work. Summary of azimuth angle is in section 3. Description of simulation platform is in section 4 of the paper. The main idea of the paper is proposed in section 5. Section 6 explains in detail about the simulation of the protocol. Simulation results and future work to be done is described in section 7 and 8 respectively. The paper concludes in section 9.

2. RELATED WORK

Routing based on Geographic location requires an efficient location service in order to find the exact location of nodes which are dynamic. e.g. Global Positioning System or a distributed database recording the location of every destination node. Many papers have suggested the concept of using GPS to determine the location of a particular node. A brief outline of work done by other researchers is given below.

Angular Routing Protocol [2] ARP is a position-based routing protocol that uses an improved geographic forwarding to route packets to the destination as and when possible. If the geographic forwarding fails, it uses an angle-based forwarding.

Simulation of Position-based Routing Protocol in Wireless Mobile Ad Hoc Network [4]: This paper proposes restriction of route request packet to quadrants. Q-DIR transmits route request to all nodes that are in the same quadrant as the source and destination instead of broadcasting in all direction.

Performance Comparison of a Position-Based Routing Protocol for VANET [6]: Here each node knows its geographical location. Its uses two methods: Directed flooding and next hop routing. In Next hop routing, packet is destined to specific node while in Directed flooding packet is destined to a zone. Packet delivery ration is increased in directed flooding.

Geo-LANMAR [11] combines IP addressing with geo coordinates for efficient packet for efficient packet delivery. It combines the benefit of LANMAR protocol in terms of group motion support with the scalability of routing.

Enhancements in AODV Routing using Mobility Aware Agents introduce the concept of agents installed on nodes which send their location information to other nodes. The selection of best path is done by intermediate nodes.

3. AZIMUTH ANGLE

Azimuth angle is the angle between two points (source and destination) on earth with reference to geographic north measured clockwise from source. Spherical trigonometric calculations can be used to compute the azimuth angle given the source and destination positions are known. GPS devices are used to find the position information with high degree of accuracy with assistance from other technologies. The values of azimuth ranges from 0o to 360o degrees with North represented as 0o degree. Figure. 1 shows the representation of azimuth angle with point A as source and C as destination point.

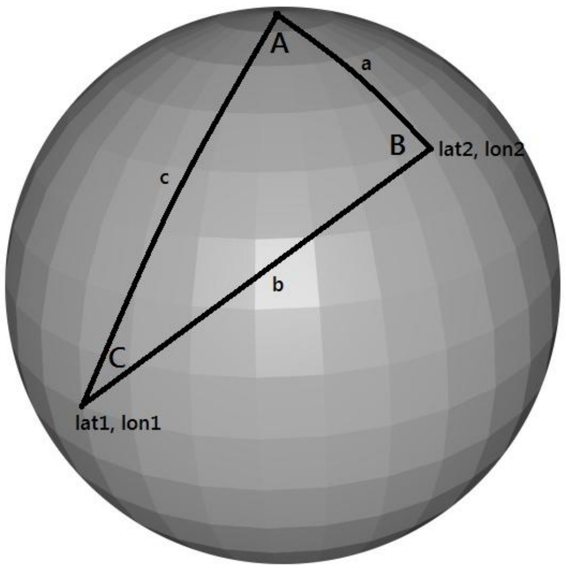


Figure 1: Representation of Azimuth angle

4. SIMULATION PLATFORM

Network Simulator 2 is an open source simulator used for simulating wired and wireless networks. It supports wide range of protocols which can be easily modified and tested. NS2 is written using OTcl and C++ languages. Here, OTcl acts as front end which can be used for configuration, setup or to run simulation with existing NS2 modules. C++ acts as back end running the accrual simulation. TclCL or Tcl with classes acts as a OTcl/C++ interface. The modification or changes to the process of protocol or packets should be done in the C++ compiled hierarchy. In NS2, the timing of various events during simulation is maintained in a Scheduler. It dispatches one event after another with respect to the scheduled time. It has single unique ID for each event. When a new event is added, Scheduler creates a new unique ID.

Once simulation is complete, there are many ways to collect the output or trace data. The results of the simulation are stored in a file for analysis. This is the trace file of a simulation. OTcl consists of number of classes and helper procedures which are used for trace file support.

NS-2 can be installed in both Windows (Cygwin) and UNIX platforms. In this simulation Backtrack Linux version 3 is used. Ns- allinone-2.34 is installed and patched with dymoum-0.3. DYMOUM is an implementation of the DYMO (Dynamic MANET On-demand) routing protocol both for Linux kernels and the ns2 network simulator. The basic operations of the DYMO protocol are route discovery and route maintenance. Routing Messages are used to disseminate routing information. There are two DYMO message types that are considered to be routing messages, RREQ and RREP. RREQ is the route request message, whereas a RREP is the generated in response to RREQ. RERR message is used to disseminate the information that a route is not available for one or more particular addresses.

5. METHOD DESCRIPTION

This protocol works independent of localization techniques which is used to determine the position of other nodes using remote positioning system. The main idea behind this protocol is to compute an expected area based on azimuth angle and restricting the broadcast within that expected area. This is done by calculating a range, based on azimuth angle. Intermediate nodes makes forwarding decisions based on this range.

During the route discovery the node first checks its routing table for any information about the destination. If no information is available the range is set as (0,360). This means that all nodes that receive this packet falls within the range and forward the packet. When the destination node receives the packet it generates the RREP packet with its position information inserted into it. On receiving this RREP packet, the source node adds the destination position information into the routing table. Next time when the node generates a RREQ packet to that destination, it uses this position information to calculate the azimuth with respect to its own position. Based on this calculated azimuth, a range is computed (azimuth-20, azimuth+20). This range information along with position information is added into the RREQ and broadcasted. When an intermediate node receives this RREQ it computes the azimuth angle using the source position information of the RREQ packet and its current position. This computed azimuth angle is then compared with the range in the RREQ packet. If the computed azimuth falls within the range, the RREQ is forwarded. Otherwise the packet is dropped. This restricts the flooding only to the range. Graphical representation of this working is shown in Figure. 2.

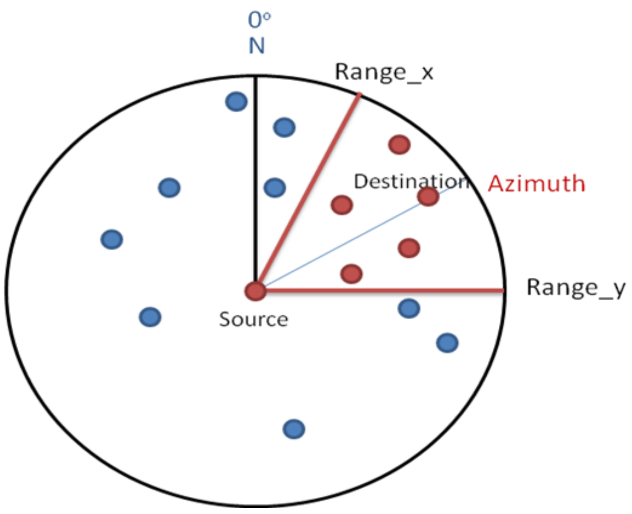


Figure 2: Range representation using azimuth angle

Since Ad-hoc networks are mobile there can be link breakages and the destination node may move out of the range, computed using azimuth angle. In this case when the RREQ is reissued the range is incremented for each RREQ tries, minimizing link breakage and the need for a complete flooding the entire network.

5.1 Position Information

Position information of nodes can be defined using OTcl command and configuration interface as given below

```
$node_(0) set X_ 30.0
$node_(0) set Y_ 40.0
$node_(0) set Z_ 0.0
```

After defining the position information of N number of nodes in OTcl configuration it can be extracted in C++ hierarchy using the below code.

```
Node*node=Node::get_node_by_address(re_node_addr.s_addr);
((MobileNode *)node)->getLoc(&x,&y,&z);
```

The position information (x, y coordinates) of nodes is taken as GPS location information. Since NS-2 supports only positive node positioning, the position of nodes is chosen such that longitude position ranges from 0o to +180o and latitude ranges from 0o to +90o.

5.2 Modified Packets

Few changes are made in the DYMO RE messages by modifying the *dymo_re.h* file. Four variables *srcx, srcy, range_x, range_y* are declared so that the packets can accommodate the latitude and longitude position information and upper and lower limits of range. Table I and Table II represent additional details in the RREQ and RREP packets respectively.

Table 1. Additional Details in RREQ

Current Position of Source (srcx)	Current Position of Source (srcy)
Range_x	Range_y

Only the position information of destination is inserted into the RREP packets.

Table 2. Additional Details in RREP

Current Position of Destination (x)	Current Position of Destination (y)

5.3 Routing Table

In addition to other details of nodes, the position information of nodes is also maintained in the routing table. For this we add two variables in the rt_entry structure. The rtable_insert () and rtable_update () functions are also modified. Now whenever a RE packet is received, the position information is extracted from pos_x, pos_y of received packet and the routing table is updated.

6. DEVELOPMENT AND WORKING

During the route discovery process the routing table is checked for any information. NS_CLASS route_discovery () function performs the initial operation of checking for pending RREQ and routing information. Here we initialize two variables which hold the position information of the destination nodes. When there is no entry in the routing table, these variables are initialized to NULL. This information along with number of route request tries are passed to re_send_rreq () function and then to the re_create_rreq () function. In the re_create_rreq () azimuth and range is calculated and added to the RREQ packets. First the destination position of nodes is checked. If they are NULL (No entry in routing table) range is set to (0,360).

```
If (dstx = NULL && dsty = NULL)
{
        range_x=0;
        range_y=360;
}
```

If destination position is present in the routing table then the source node extracts its location information from the OTcl configuration and computes the azimuth angle based on the following algorithm.

```
If (dstx != NULL && dsty != NULL)
{
srcxmin = srcx-floor (srcx);
srcymin = srcy-floor (scry);
dstxmin = dstx-floor (dstx);
dstymin = dsty-floor (dsty);
if (srcy< 0)
homelongitude = srcy-srcymin/60;
else
homelongitude = srcy+srcymin/60;
if (srcx < 0)
homelatitude = srcx-srcxmin/60;
else
homelatitude = srcx+srcxmin/60;
if (dsty < 0)
distlongitude = dsty-dstymin/60;
else
distlongitude = dsty+dstymin/60;
if (dstx < 0)
distlatitude = dstx-dstxmin/60;
else
distlatitude = dstx+dstxmin/60;
dl = 720.0-distlongitude;
hl = 720.0-homelongitude;
dong = (int)(dl-hl);
if (abs(dong)>180)
{
if ((distlongitude<0) && (homelongitude>0))
distlongitude=distlongitude+360;
if ((homelongitude<0) && (distlongitude>0))
distlongitude=distlongitude-360;
dl = 720.0-distlongitude;
hl = 720.0-homelongitude;
dong = (int)(dl-hl);
}
R=acos(cos(dong/RAD2DEG)*cos(distlatitude/RAD2DEG)*
cos(homelatitude/RAD2DEG)+sin(homelatitude/RAD2DEG)
*sin(distlatitude/RAD2DEG));
S = (sin(distlatitude/RAD2DEG)-
cos(r)*sin(homelatitude/RAD2DEG))/(sin(r)*cos(homelatitud
e/RAD2DEG));
}
```

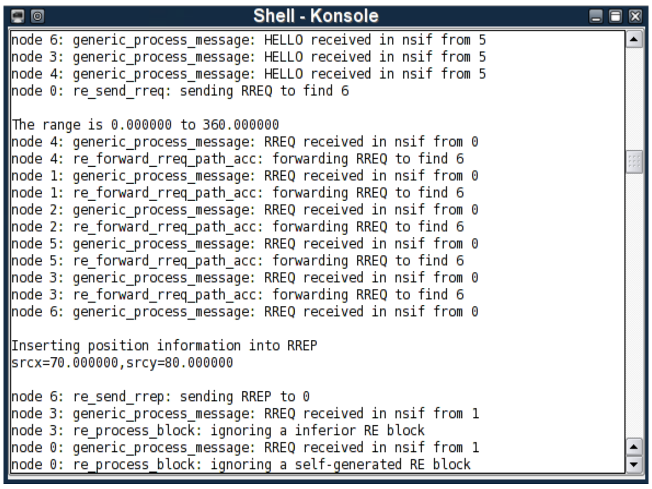
For S values greater than 1, S is positive of 1. For S values less than -1, S is negative of 1. RAD2DEG is a factor for converting radians to degrees. Finally the azimuth and the range is calculated as

```
T = acos(S) *RAD2DEG;
if (hl<dl)
        azimuth = 360.0-T;
else
        azimuth = T;
```

```
fazimuth = ceil(azimuth);
range_x=fazimuth-20;
range_y=fazimuth+20;
if(range_x<0)
        range_x=range_x+360;
if(range_y>360)
        range_y=range_y-360;
```

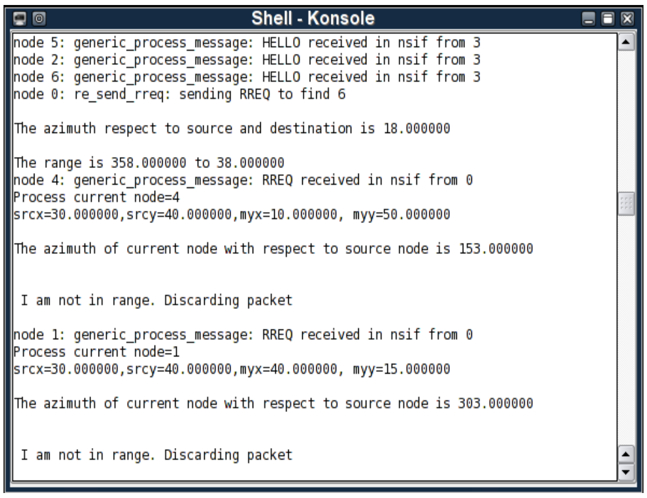
Once the range is calculated, both the source node position and range is inserted into the RREQ packet and transmitted. These details are used by intermediate node to take forwarding decisions.

To demonstrate the working we simulate a network of 7 nodes and placed in such a way that few nodes are outside the range and few remains within it. Node 0 is taken as source node and node 6 is the destination. Azimuth angle between the source and destination position is 180. Based on this azimuth, a range of 3580 to 380 is taken as shown in Figure .4.Intermediate nodes computes the azimuth angle when they receive this RREQ and takes forwarding decisions. Nodes 1, 2, 3 and 4 drops the request as they are out of range whereas Node 5 forwards the request as its azimuth falls within the range. Finally destination Node 6 replies with a RREP packet, which is shown in Figure. 5.



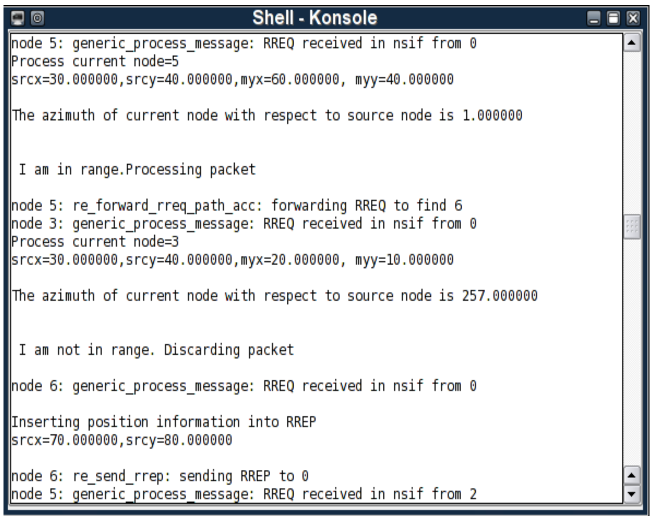
```
node 6: generic_process_message: HELLO received in nsif from 5
node 3: generic_process_message: HELLO received in nsif from 5
node 4: generic_process_message: HELLO received in nsif from 5
node 0: re_send_rreq: sending RREQ to find 6

The range is 0.000000 to 360.000000
node 4: generic_process_message: RREQ received in nsif from 0
node 4: re_forward_rreq_path_acc: forwarding RREQ to find 6
node 1: generic_process_message: RREQ received in nsif from 0
node 1: re_forward_rreq_path_acc: forwarding RREQ to find 6
node 2: generic_process_message: RREQ received in nsif from 0
node 2: re_forward_rreq_path_acc: forwarding RREQ to find 6
node 5: generic_process_message: RREQ received in nsif from 0
node 5: re_forward_rreq_path_acc: forwarding RREQ to find 6
node 3: generic_process_message: RREQ received in nsif from 0
node 3: re_forward_rreq_path_acc: forwarding RREQ to find 6
node 6: generic_process_message: RREQ received in nsif from 0

Inserting position information into RREP
srcx=70.000000,srcy=80.000000

node 6: re_send_rrep: sending RREP to 0
node 3: generic_process_message: RREQ received in nsif from 1
node 3: re_process_block: ignoring a inferior RE block
node 0: generic_process_message: RREQ received in nsif from 1
node 0: re_process_block: ignoring a self-generated RE block
```

Figure 3: Flooding with default range and RREP from destination with its position information



```
node 5: generic_process_message: HELLO received in nsif from 3
node 2: generic_process_message: HELLO received in nsif from 3
node 6: generic_process_message: HELLO received in nsif from 3
node 0: re_send_rreq: sending RREQ to find 6

The azimuth respect to source and destination is 18.000000

The range is 358.000000 to 38.000000
node 4: generic_process_message: RREQ received in nsif from 0
Process current node=4
srcx=30.000000,srcy=40.000000,myx=10.000000, myy=50.000000

The azimuth of current node with respect to source node is 153.000000


 I am not in range. Discarding packet

node 1: generic_process_message: RREQ received in nsif from 0
Process current node=1
srcx=30.000000,srcy=40.000000,myx=40.000000, myy=15.000000

The azimuth of current node with respect to source node is 303.000000


 I am not in range. Discarding packet
```

Figure 4: Range calculation based on destination position and decision by intermediate nodes



```
node 5: generic_process_message: RREQ received in nsif from 0
Process current node=5
srcx=30.000000,srcy=40.000000,myx=60.000000, myy=40.000000

The azimuth of current node with respect to source node is 1.000000


 I am in range.Processing packet

node 5: re_forward_rreq_path_acc: forwarding RREQ to find 6
node 3: generic_process_message: RREQ received in nsif from 0
Process current node=3
srcx=30.000000,srcy=40.000000,myx=20.000000, myy=10.000000

The azimuth of current node with respect to source node is 257.000000


 I am not in range. Discarding packet

node 6: generic_process_message: RREQ received in nsif from 0

Inserting position information into RREP
srcx=70.000000,srcy=80.000000

node 6: re_send_rrep: sending RREP to 0
node 5: generic_process_message: RREQ received in nsif from 2
```

Figure 5: Forwarding decision by intermediate node and reply from destination

6.1 Forwarding Decision

When a RE message is received by a node and processed by NS_CLASS re_process () function, it is first checked whether the receiving node is the target node. If the receiving node is the target node RREQ must not be retransmitted. When the receiving node is not the target node the forwarding decision is made. The receiving node extracts the range_x and range_y and checks it. If the range is found to be (0,360), the RREQ is simply rebroadcasted after updating the routing table. For range other than (0,360) the receiving node computes the azimuth angle using the position information of source (srcx, srcy) and its own position as destination.

```
if (re->range_x!=0 && re->range_y!=360)
        return myazimuth;

if (re->range_y>re->range_x)
{
 if (myazimuth>=re->ran_x &&
                myazimuth<=re->range_y)
        Forward;
  else
        Drop;
}

else if (re->range_x>re->range_y)
{
if ((myazimuth>re->range_x && myazimuth<360) ||
(myazimuth<re->range_y && myazimuth>=0))
        Forward;
  else
        Drop;
}
```

The computed azimuth is compared with the range in the RREQ packet (range_x, range_y). If the computed azimuth falls within the range, the request is forwarded else the packet is dropped.

6.2 Reissue of request packet

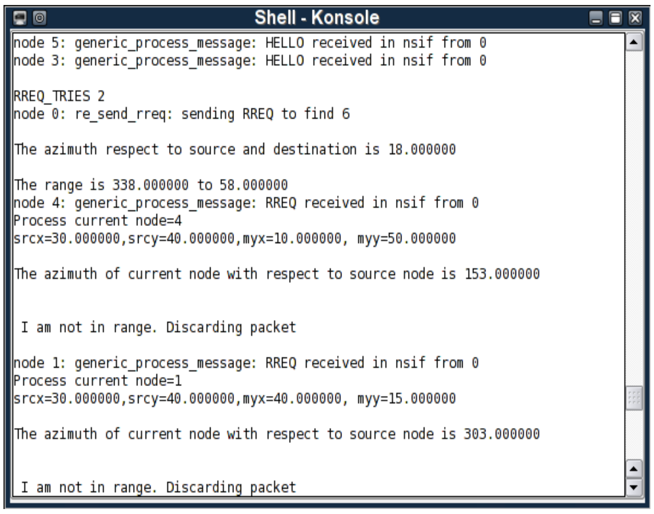
Since nodes in ad-hoc networks are mobile, the possibilities of link breakages are very high. When source node sends a RREQ packet based on the last known position of destination and the destination has moved out of range, then the RREQ packet never reaches the destination. In this case RREQ has to be rebroadcasted. During the reissue of RREQ we are incrementing the range by some value based on the RREQ_TRIES. To simulate this, the reissue_rreq_ is set to TRUE in the OTcl script. The NS_CLASS route_discovery_timeout () function in dymo_timeout.c increments the value the number of tries with every RREQ reissue. This value is passed to the re_create_rreq (). When the range is computed, we check the RREQ reissue for the number of tries and increment the range value on the following condition

```
If (tries > 0 && tries <= RREQ_TRIES)
            Increment Range;
```

RREQ_TRIES is set to a maximum of 3 and depending on the nth try, the N values are defined

```
            range_x=range_x-N;
            range_y=range_y+N;
```

This causes the range to expand with every reissue of RREQ packet. Figure. 6 shows the computation of range during the second retry. Here range is computed as (azimuth-40,azimuth+40) while normal range calculation is (azimuth-20,azimuth+20).



```
node 5: generic_process_message: HELLO received in nsif from 0
node 3: generic_process_message: HELLO received in nsif from 0

RREQ_TRIES 2
node 0: re_send_rreq: sending RREQ to find 6

The azimuth respect to source and destination is 18.000000

The range is 338.000000 to 58.000000
node 4: generic_process_message: RREQ received in nsif from 0
Process current node=4
srcx=30.000000,srcy=40.000000,myx=10.000000, myy=50.000000

The azimuth of current node with respect to source node is 153.000000


 I am not in range. Discarding packet

node 1: generic_process_message: RREQ received in nsif from 0
Process current node=1
srcx=30.000000,srcy=40.000000,myx=40.000000, myy=15.000000

The azimuth of current node with respect to source node is 303.000000


 I am not in range. Discarding packet
```

Figure 6: Range is incremented from its default value during the reissue of RREQ packet by N value.

7. SIMULATION RESULT

Our simulation is done on small area 90 x 180 as these position parameters are used as GPS degree inputs (-90 to +90 for latitude and -180 to +180 for longitude). The transmission power and threshold power are adjusted to setup a multi hop network within this area. 35 stationary nodes were used. Simulation time was 100 seconds with 28 connections across different pairs of source and destination. As the Figure. 7 shows, ARRP considerably reduced the number of control packets used during the route discovery process when compared with DYMO implementation.

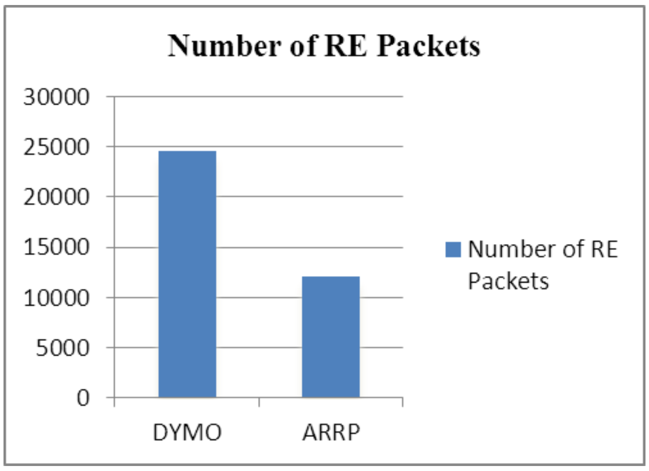


Figure 7: Comparison of DYMO and ARRP

8. FUTURE WORK

This protocol works well in dense networks where many mobile nodes are present and traffic rates are high. However further enhancements can be done to optimize communication using MAC layer support and localization algorithms. Manipulation of calculated range using speed of the moving node is also considered.

9. CONCLUSION

The paper proposes a routing technique, Azimuth Restricted Routing Protocol (ARRP) using position information nodes and azimuth angle. The idea behind this protocol is to reduce the search region and control the number of flooding packets used during route discovery to save bandwidth and node power. Additional information is added to RREQ and RREP packets and the routing table is modified to support the routing needs. Simulations carried out showed significant reduction in number of control packets generated during route discovery over conditions of dense networks and traffic loads. Further research is carried out to improve the efficiency of the protocol.

10. ACKNOWLEDGMENT

We would like to thank Assoc. Prof. Dr. Liza Abdul Latiff, Universiti Teknologi Malaysia International Campus for her highly appreciated help during this project and timely answers for all the questions that popped up.

11. REFERENCES

[1] Reno Robert .R “Enhanced AODV for Directional flooding using Coordinate system” IEEE International Conference ICNIT, pp. 329-332, 2010

[2] Venkata C. Giruka, Mukesh Singhal “Angular Routing Protocol for Mobile Ad-hoc Networks” IEEE International Conference on Distributed Computing Systems Workshops, 2005

[3] Y. Ko and N. H. Vaidya, “Flooding-based geocasting protocols for mobile ad hoc networks,” Mobile Networks and Applications, 7(6), Dec. 2002, pp. 471-480.

[4] L.A. Latiff, N. Fisal, S. S. Ariffin “Simulation of Position-based Routing Protocol in Wireless Mobile Ad Hoc Network” Third Asia International Conference on Modelling & Simulation, 2009

[5] A. Ali, L.A. Latiff, Chia-Ching Ooi, and N.Fisal, "Location-based Geocasting and Forwarding (LGF) Strategy in Mobile Ad Hoc Network (MANET)" ICT 2005, Cape Town,South Africa, May 2005.

[6] Takano,A. Okada, H. Mase, K. "Performance Comparison of Position-Based Routing Protocol for VANET", 2007 IEEE

[7] C.E. Perkins and E.M. Royer "Ad hoc On- Demand Distance Vector Routing," Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, Feb. 1999, pp. 90-100.

[8] Won-Ik Kim, Dong-Hee Kwon, Young-Joo Suh. "A reliable Route selection algorithm using Global Positioning Systems in Mobile ad-hoc networks" IEEE International Conference ICC, June 2001, pp.3191-3195

[9] Y.-B. Ko and N.H. Vaidya, "Using location information to Improve routing in ad hoc networks", Technical report 97- 013, Texas A&M University (1997)

[10] Polar coordinate systems, http://www.kartografie.nl/ geometrics/index.html

[11] B. Zhou, Y. Lee, M. Gerla, and F. de Rango, "Geo-LANMAR: a scalable routing protocol for ad hoc networks with group motion: Research Articles", Wireless Communications & Mobile Computing,6(7), Nov. 2006, pp. 989-1002

[12] D. Espes, Z. Mammeri. "Adaptive expanding search methods to improve AODV Protocol," IST Mobile and Wireless Communications Summit, July 2005.

[13] M.Mauve, J.Widmer and H. Hartenstein, " A Survey on Position based Routing in Mobile Ad-Hoc Networks" IEEE Networks, 5(6):30-39, November/December 2001.

[14] J.Li, J. Jannotti, D.S.J. De Couto, D.R. Karger, R. Morris, " A Scalable Location Service for Geographic AdHoc Routing", Mobicom 2000.