IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 10, Issue 5, Ver. II (Sep - Oct .2015), PP 50-57
www.iosrjournals.org

Mobile AntNet Routing for Mobile Ad hoc Network

Javeed Basha Haroon[1], Abdulkadir Ahmed[2], Olalekan Ogunbiyi[3]
[1]*(Electronic Engineering Group, School of Engineering and Applied Secince, Aston University, UK)*
[2,3]*(Department of Electrical and Computer Engineering, Kwara State University, Nigeria)*

Abstract: *Routing in a Mobile Ad hoc Network (MANET) still pose great challenges which need to be addressed. Whenever a routing algorithm is developed, it should address the need of improving the quality of service (QoS) requirements, these requirements ranges from throughput, less overhead, packet loss, security etc. This paper focus on designing a nature inspired routing algorithm called the Mobile AntNet for Mobile Ad hoc network. Much of the previous implementations of this algorithm have been revolving around the fixed networks and improving the QoS for fixed networks. The AntNet algorithm was implemented in mobile ad hoc network using NS-2.34 simulator, the implemented algorithm was compared with other traditional routing algorithms in terms of node transmission range, node mobility, number of nodes and node failure. Multiple simulations were performed to obtain a statistical output by varying the different simulation parameters and graphs were plotted against three important performance metrics: throughput, routing overhead and end-to-end delay. Simulation result shows the performance of the designed algorithm in terms of throughput, packet loss and node failure.*
Keywords: *AntNet, Packet, Protocols, Routing, Throughput.*

I. Introduction

Ever since the development of the various mobile computing devices such as the laptops, handheld portable devices, personal digital assistants etc., there has been a significant change in the computing and communication word. We are now in the age of ubiquitous computing, where computing engages many computational devices and systems simultaneously; the nature of these computing devices has led to the significant growth in the use of wireless technologies such as the Wireless Local Area Network (LAN), Personal Area network (PAN) and Body Area Network (BAN).

One of the advantages of the ubiquitous computing is the ability of the technology to adapt itself to the required user mode, without the need of the user manually modifying their behaviour and knowledge. This philosophy is actually on the basis of the ambient intelligence concept [1]. The basic objective of the ambient intelligence concept is the integration of digital device and networks into everyday environment. The ambient intelligence relies heavily on the wireless and mobile communication paradigm, in which the user's mobile devices form the basic network and they actively interact together in order to provide the functionality which is generally provided by the wired network infrastructure. Hence, such systems are called as the mobile ad hoc network (MANETs or infrastructure-less wireless network).

A Mobile Ad Hoc network is a collection of self-configuring wireless mobile nodes; these nodes can organize themselves into arbitrary and temporary network topologies, hence allowing the users to access the internetwork without the need of any pre-existing infrastructure and without any interruption [2], [3], [4].
Moving on to the smaller scale ad-hoc networks, we have the LAN, PAN and the BAN wireless network, more specifically the market version of these networks have already started to appear. "These technologies constitute the building blocks to construct the small multi-hop ad-hoc networks that extend the range of the ad-hoc networks over a few radio hops" [1], [5]. The MANETs are classified into four different categories, the overall classification of the mobile ad-hoc networks are: Wireless Body Area Networks (WBANs), Wireless Personal Area Network (WPAN), Combination of WBAN and WPAN, Wireless Local Area Networks (WLANs) [4], [5], [6].

In this research, we designed a routing algorithm for mobile and ad-hoc networks called Mobile AntNet; this protocol is actually derived from the original AntNet routing algorithm designed by Dr. Marco Dorigo for fixed networks. The primary objective of this research was to investigate the nature inspired routing algorithms (NIRA), preferably the Genetic algorithm, Beehive algorithm and the AntNet algorithm, and to investigate the feasibility in implementing AntNet algorithm in the mobile ad hoc network using a simulation tool. The secondary objective includes the design and implementation of the AntNet algorithm for the mobile ad hoc network, rather than the current available one for fixed network. We also analyzed the performance evaluation of the designed algorithm in terms of throughput, packet loss and node failure [4], [7].

The basic approach undertaken was to first evaluate the necessary simulation tools available in the market and preferably use the one with essential back-up and documentation available and also the one with open source attributes, which can be modified as per the project requirement. Under this process, we decided to use Network Simulator 2 as our simulation software, although much complex when compared to the other simulators such as the QualNet and GTNet. The second step was to analyse the various reactive, proactive and hybrid routing protocols involved, including the NIRA such as the AntNet, Beehive and Genetic algorithm. The final step was designing and implementing the Mobile AntNet algorithm, using the basic approach taken by the original AntNet algorithm and evaluate its performance in terms of throughput, packet loss and node failure [8].

II. Ant Net Algorithm Modelling for Ad Hoc Networks

The AntNet is a routing algorithm for adaptive routing in IP networks. Basically it is design around the Ant Colony Optimization (ACO) technique, where the shortest path determination technique that is observed in ant colonies is defined for a Nature inspired metaheuristic optimization. AntNet algorithm was developed by two notable Italian mathematicians namely, Dr. Gianni Di Caro and Dr. Marco Dorigo. The algorithm is based on the principles of the ACO (Ant Colony Optimization) routing algorithm, although much of the algorithm has been modified in order to implement AntNet algorithm successfully in packet-switched networks, but the core concepts of the ACO algorithm are retained. Hence, AntNet algorithm follows the basic concept of stigmergy andpheromones.

Fig. 1 represents the basic structure adopted by the ants in searching/foraging for the food. At first, a single ant/forager ant starts it journey to hunt for the food, as it starts it journey; it leaves a trial of chemical substance called the pheromone to notify other ants about the path it took to reach its destination, slowly other ants begin their journey towards the destination, the ants follow their trial of pheromone laid by the forager ant to reach the destination. At the beginning, a few ants take a different path towards the destination, but later on, the ants start following the path which has the highest pheromone substance present in them, evidently the path with the highest pheromone substance is also the shortest path available to reach the destination. This form of communication using the chemical substance called the pheromone is called as stigmergy [7], [8]

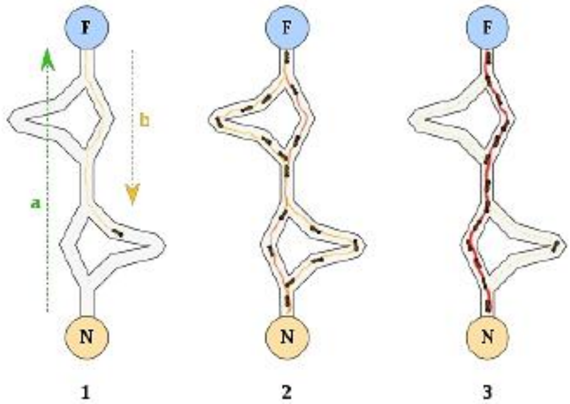

Fig 1: Ant's searching for food [9]

The AntNet algorithm is a dynamic distributed hybrid routing algorithm that consist both the instances of reactive and pro-active routing protocols. That is, the route setup process is a reactive process and the route maintenance, route improvement process is based on the proactive mechanisms. AntNet algorithm uses core ACO mechanism, and hence it is able to retain the working principles of the ants and hence the name AntNet. The AntNet algorithm also consist elements from the Distance-Vector routing namely the route sampling strategy used in here is of the D-V routing. In spite of being a hybrid routing algorithm, the routing information is not updated until the currently available / known routes are exhausted, hence the algorithm mostly function as a pro-active routing algorithm. The basic working principle of the AntNet algorithm in the Mobile Ad-Hoc networks are briefly summarized in points given below, Figure 2, represents the working functionality of the Mobile AntNetalgorithm (the implementation of the AntNet algorithm in ad-hoc networks).

Pheromone Table: In the AntNet algorithm, each node i, maintains one pheromone table $P_i$which is basically a 2-D matrix. Entry to the pheromone table is made as Pijd, such that the table has an entry/information about the route from node ito the destination d over neighbourj. Hence, Pijdindicates the relative goodness of going over node j, when travelling form node ito destination d, as well as the statistics information about the path. The node also maintains a neighbour table which keeps the track of nodes it has a wireless link.

During the initial start-up phase, the source node checks its pheromone table for any available path to the destination, if a path is available then it begins its transmission, if not, it generates a forward ant packet (FA) and broadcast it to the nodes within its transmitting range, just like the RREQ (Route Request) packet in the AODV protocol [7], [8], [9].

Each of the intermediate node upon receiving the FA packet, will check its pheromone table for the requested path to the destination, if available it will kill the existing FA packet and will generate a backward ant packet (BA). The BA packet is sent back to the sender by unicast method.
If the intermediate node does not have the requested path information in its pheromone table, then it broadcasts the FA packet to its neighbouring node within its transmitting range.

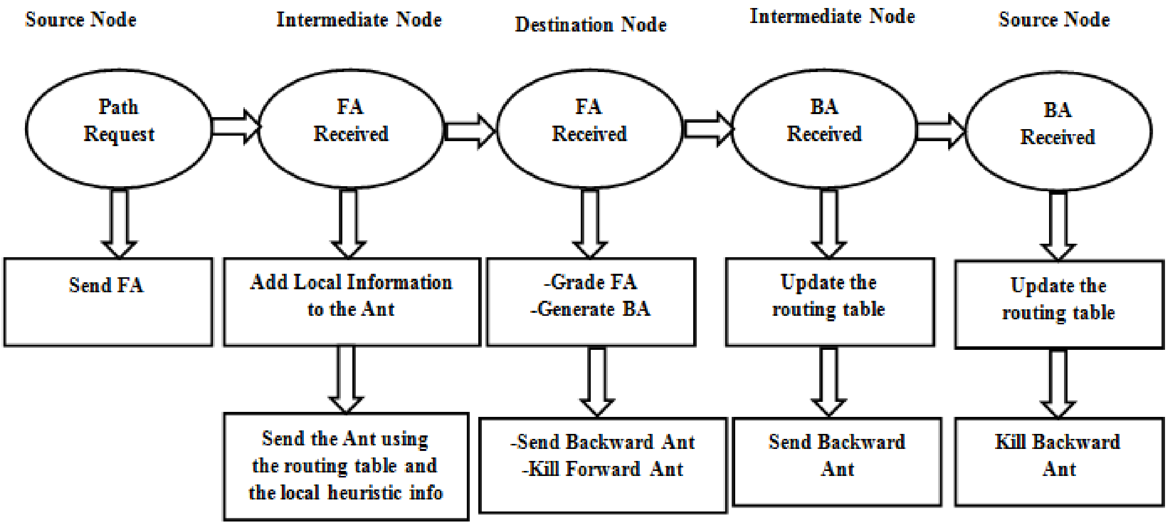

Fig 2:High Level Flow-Chart describing the AntNet functionality

Each time the FA packet collects the full array of local routing information of each node as it travels, upon reaching the destination, the BA packet is generated and the FA packet is killed.
The other FA packets sent from the different intermediate nodes are then destroyed, this is because the AntNet algorithm considers the first FA packet to have the best optimal path and hence the subsequent copies are destroyed.

The BA packet is sent back to the sender along the exact path as it was transmitted, but this time the BA packet is sent through unicast method. The BA packet on its way back to the sender collects quality information about each of the existing links in the journey.

At each of the intermediate node and the source node, it updates the routing tables based on the information collected, by this way a first route between the source and the destination is generated and hence completes the reactive route setup process.

III. Implementation of the AntNet Algorithm in NS-2.34

The AntNet algorithm we have implemented in the mobile ad-hoc networking mode follows the same basic principle of the algorithm designed initially by Lavina Jain for wired networks simulation in NS-2.33 [9]. However, we have modified the existing AntNet algorithm to such an extent that it can now perform perfectly under wireless network conditions, hence, the name Mobile AntNet. The basic files of the previous algorithm has been retained, however, additional C++ programming files have been added so that the packets can be transmitted via broadcast and unicast methods. The NS-2.34 library files have also been modified to an extent such that it can cope well with the front end programming files (TCL). Extra precautions have been taken to avoid the segmentation error while executing the simulation scripts.

The files modified in NS-2.34, are: Makefile.in, ns-library.tcl, ns-packet.tcl,packet.h,packet.cc,cmu-trace.h, cmu-trace.cc and priqueue.cc. Tools such as gdb, dmallocand tcl-debug have been used to debug errors in C++ and TCL programming languages. A total of 45 TCL scripts have been generated to perform the overall testing of the AntNet algorithm with other traditional algorithms such as the AOMDV, DSR and DSDV.

IV. Simulation

This section analyses the various results obtained during the simulation implementation of the AntNet algorithm and aims to provide a conclusion for each result comparing with three different wireless ad-hoc routing protocols. The protocols are compared with three different measured performance criteria such as: Throughput, Packet Delay and Routing Overhead. The above mentioned performance metrics are compared by modifying different simulation specific criteria such as: Number of Nodes Impact, Node Mobility Impact, Network Failure Adaptability and Transmission Range Impact. The different protocols chosen to be compared with AntNetalgorithm are as follows; AOMDV (Ad-hoc On-demand Multipath Distance Vector), DSDV (Destination-Sequenced Distance Vector) and DSR (Dynamic Source Routing).

4.1 The Network Simulator NS-2.34

The Network Simulator 2.34 is an open-source software tool that predicts behaviour of a network without the use of an actual network. This software tool is quite famous and is constantly used in the field of research and areas where testing/predicting behaviour of the network is a necessity. For instance, a researcher/engineer can predict the actual damage caused in the network when a DoS (Denial of Service) attack takes place, hence, network simulators are quite useful in allowing network architects/engineers to test new network protocols as well as test a new network design/topology. The NS-2.34 is the latest version in the series of NS design tools, the very latest one being the NS-3, which is still under development. In order to overcome the delay in releasing the NS-3 simulator, an interim simulator called as NS-2.35 has been released officially, the NS-2.35, contains some of the advanced traits of NS-3 simulator and has additional protocols present in it. The most vital part being that NS-2.35 has managed to fix some critical bugs present in the previous versions of NS. NS-2.34 was built in C++ and provides a simulation interface through OTcl; an object-oriented version of TCL, hence, the user specifies a network topology by writing TCL scripts and then calls the C++ program to simulate the script. We have implemented our algorithm in NS-2.34 simulator, although much complex when compared with other open-source simulators, NS has the best technical support in our opinion. With many researchers working on NS, the ability to solve a bug or fix a problem in the simulator is very high, also, NS has inbuilt debugging tools available in it, these being gdband tcl-debug.

We have successfully implemented the AntNet algorithm in the network simulator and have modified it to be able to work in Mobile Ad-hoc networks; we then compare simulation results with other traditional routing algorithms. Our contribution to the project is as follows; Implementation of AntNet algorithm in NS-2.34, Creation of TCL scripts to determine the performance metrics and generation of scenario files for mobile node movement.

4.2 Performance Metrics

To evaluate the performance of our algorithm and compare it with other traditional routing algorithms, we consider three main performance metrics as stated below: Throughput (%), packet delay (End to End Delay) and routing overhead. The three criteria are chosen in order to determine the impact caused by the algorithm in the network under different varying simulation parameters. All the three performance metrics have direct relationship with the performance of the network.

Throughput: It is one of the most important criteria which determines the overall amount of data packets transmitted and received in the network, mathematically throughput can be written as;

$$T = \lambda / \mu \ (\%) \qquad (1)$$

Where, λ – Packets received, μ – Packets Departed

Packet Delay (End-End Delay): End -to-End delay is the average of the time taken by the transmitted packets to reach the destination across the network.

Routing Overhead: Routing Overhead are the router (RTR) packets generated in the network to establish a new path and maintain the link connectivity in the network. In the case of AntNet algorithm, the FA (Forward Ant) and the BA (Backward Ant) are the basic routing overhead packets generated in the network.

4.3 Simulation Parameters

The simulation parameters used are specified below; all the routing algorithms were simulated under the exact simulation parameters multiple times to obtain a perfect output. Different algorithms are examined along with the AntNet algorithm, in few simulations the algorithms such as the DSDV and the DSR are not used, the reason being that the DSDV and DSR algorithms provided segmentation faults during simulation, the fault occurrence is due to NS-2.34 file named as ns-packet.tcl, modifying this file may result in other latest algorithms not being able to perform well. The bugs present in this file, have been fixed in the latest version of NS-2.35.

Simulation Parameters

Algorithms Examined:	AOMDV, AntNET, DSDV and DSR
Channel Used:	Wireless Channel
Network Interface:	Wireless Physical
MAC Type:	IEEE 802.11
Queue Type:	Drop-Tail or Priority Queue
Link Layer Type:	Uses ARP to resolve IP addresses to MAC address
Antenna Type:	Omni Antenna
Default Wireless Physical Settings: 914MHz Lucent WaveLAN DSSS	

Queue Length:	50 Packets
Number of Nodes:	10, 20, 30, 40 and 50
Maximum Area:	1000 X 1000 meters
Simulation Time:	Maximum of 20s
Pause Time:	5s
Node Mobility Speed:	20, 40, 60, 80, and 100 meters/s
Node Transmitting Range:	150, 200, 250, 300 and 350 meters
Packet Size:	512 Kb/s
Propagation Type:	Two Ray Ground
Node Movement Model:	Random Way Point

V. Results

The various results obtained during the simulation implementation of the AntNet algorithm are presented below. The protocols are compared with three different measured performance criteria such as: Throughput, Packet Delay and Routing Overhead.

5.1 Throughput

Throughput refers to overall data packets transmitted in the network; this parameter has been used in multiple simulations by modifying the number of nodes, node mobility speed, node failure and transmission range of the nodes.

Fig. 3represents the throughput value with reference to the number of nodes; each of the algorithms is simulated multiple times by increasing the number of nodes to the maximum of 50 nodes. The AntNet algorithm fares slightly better compared to DSDV algorithm, the AOMDV has the best throughput value (100%) when compared with other wireless routing algorithms, the reason that the algorithms such as the AOMDV and DSR performed well is due to the nodes having multiple paths towards the destination.

Fig. 4represents the throughput value with reference to node mobility speed; the speed range used in the simulation is a maximum of 100meters/s for each node, the minimum being 20meters/s. As the speed increases the AntNet algorithm fails to deliver the data packets to the destination, the throughput value for the DSR and the DSDV algorithm reaches to zero as the speed increases, AntNet algorithm fares slightly better when compared to DSDV algorithm. The AOMDV can perform well during high node mobility.

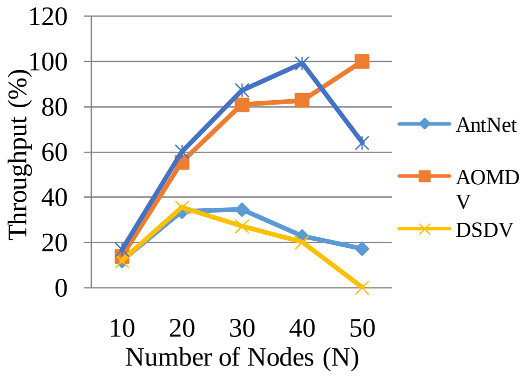

Fig 3: Nodes Vs Throughput

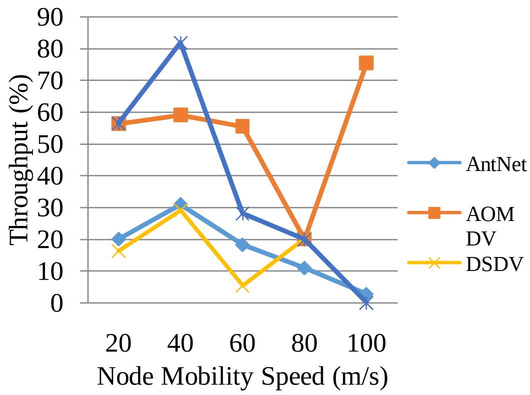

Fig 4: Node Mobility Vs Throughput

Fig. 5 represents the throughput value with reference to node failure, the maximum number of nodes present in the network is 50 nodes and the maximum failed nodes in the network are around 30 nodes. The AntNet and the DSDV algorithm provide a constant throughput at all times, since each considers only a single path to the destination and finds route only when needed. The AOMDV and DSR start to drop packets as the number of failed nodes increases steadily; almost all the algorithms provide an overall throughput rate of 70% at the time of maximum node failure, since these three algorithms have multiple paths towards the destination, the throughput decreases steadily as the node failure increases.

Fig. 6 represents the throughput value with reference to the transmission range of each node, the maximum number of nodes used in the network was 50 nodes, and the total area was around 1 million square meters. The Transmitting range of each node is increased from 150m to a maximum of 400m.

The throughput value of each algorithm increases as the transmission range increases, the AOMDV and DSR reaches the throughput value of 100% during the maximum transmission range. The AntNet algorithm reaches to the throughput value of around 82% during the maximum transmission range.

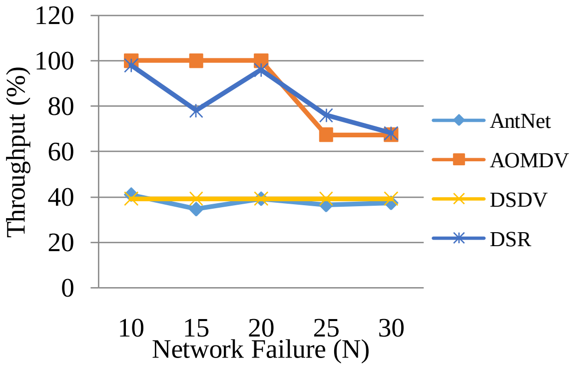

Fig5: Network Failure Vs Throughput

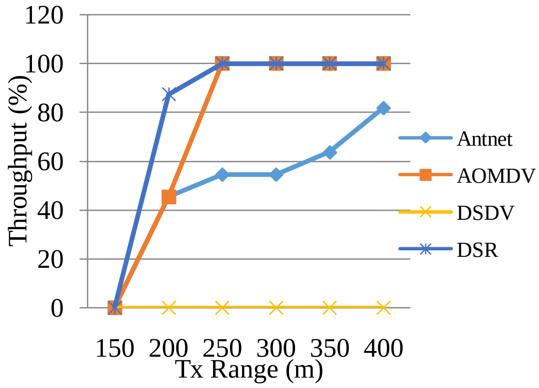

Fig 6: Transmission Range Vs Throughput

5.2 End-to-End Delay

End-to-End Delay is the time taken by the packet to travel from source to the destination; the parameter is simulated multiple times in order to achieve a perfect output, the graph is plotted against node mobility speed and number of nodes

Fig. 7 represents end-to-end delay plotted against the increasing number of nodes, with the maximum number of nodes to be simulated being 50 nodes. The pause time used in this simulation is 10s, as the graph represents AntNet has the most highest packet delay comparing to other traditional routing algorithms, as the number of nodes increases the delay for the packet to reach the destination also increases, although this is common for all the routing algorithms, AntNet suffers the most with the delay reaching up to 0.12ms, this is common in the AntNet algorithm because the algorithm works in the pro-active way and hence the ant agents take up the same queue as the data packets meant for transmission hence, as the number of nodes increases the ant agents take up much of the space in the queue meant for data packets making larger delays.

Fig. 8 represents end-to-end delay plotted against the increasing node mobility; with the maximum node mobility to be simulated being 100m/s and the maximum nodes used in the simulation is 50 nodes.The mobility of the node also has large impact on the packet delay, except the algorithms that use multiple paths such as the AOMDV, the other algorithms fared well. The AntNet algorithm was able to achieve 0.0ms delay when the node mobility increased to around 100m/s.

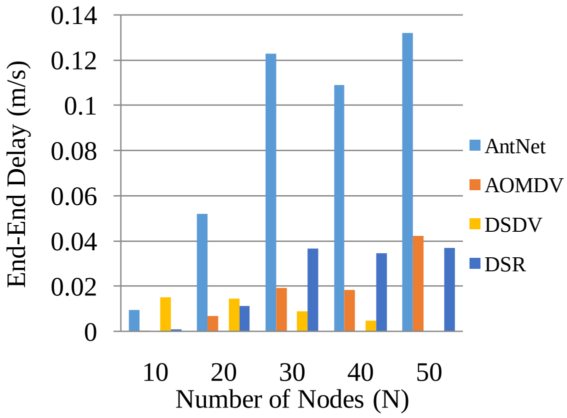

Fig 7: Nodes Vs End-End Delay

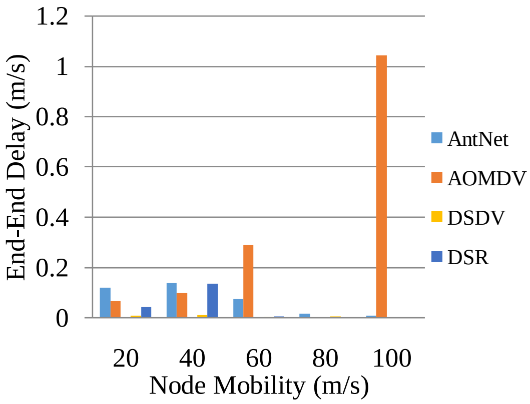

Fig 8: Mobility Vs End-to-End Delay

5.3 Routing Packets Overhead (RTR)

Routing packet overhead refers to the amount of routing packets generated in the network, in any wireless network the routing information from the sender to the destination is found by flooding the network with control packets, which is responsible for finding the destination requested by the sender. The basic types of routing packets generated in the AntNet algorithm are the Forward Ants (FA) and the Backward Ants (BA).

Fig. 9 and 10 represents routing overhead packets in comparison with the node mobility and increasing number of nodes. In Fig. 9, the AntNet algorithm maintains a steady rate in the routing packets, the maximum number of routing packets generated is around 100 packets, DSDV and AntNet has the highest number of routing packets and AntNet algorithm maintains a steady overhead packets, the number of routing packets generated by the DSDV algorithm is around 25 packets.

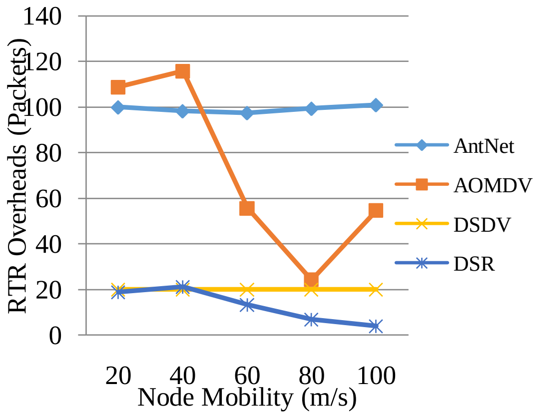

Fig 9: Mobility Vs RTR Overheads

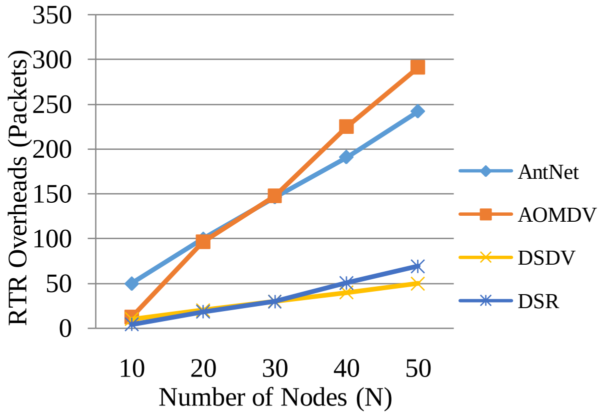

Fig 10: Nodes Vs RTR Overheads

The DSR algorithm has the lowest routing packets generated, the reason behind this is that the DSR algorithm is a reactive protocol and there is no need for it to flood the network to update its routing table.

In Fig. 10, the routing packets generated by the AntNet algorithm increases steadily, this is same for all the routing algorithms. Hence, as the number of nodes increases the routing overhead packets generated to maintain the routing table also increases steadily.

In Fig. 11, almost all algorithms generate constant routing overheads as the transmission range of each node increases; there is no much fluctuation in the amount of routing overheads generated. The maximum number of nodes used in the simulation is 50 nodes and the simulation area is around 1 million square meters. Considering the performance of the individual algorithms, DSR has the least number of routing overhead packets generated, while the AOMDV has the highest amount of routing packets generated. AntNet algorithm performs at an average rate.

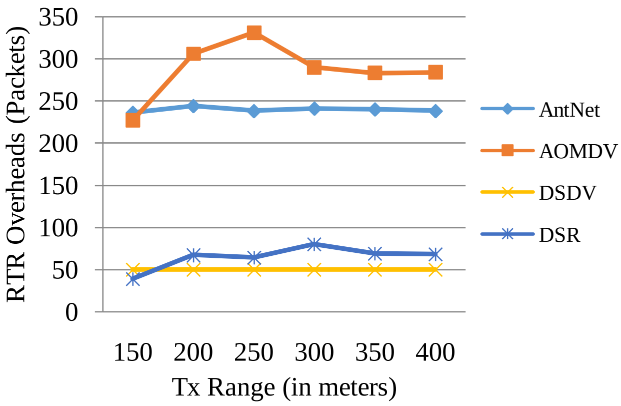

Fig 11: Transmission Range Vs RTR Overhead

VI. Conclusion

In this research, the AntNet algorithm was implemented in Mobile Ad hoc Networks (MANETs) using NS-2.34 simulator, the implemented algorithm was compared with other traditional routing algorithms in terms of node transmission range, node mobility, number of nodes and node failure. Multiple simulation were performed to obtain a statistical output by varying the different simulation parameters and graphs were plotted against three important performance metrics such as the Throughput, Routing Overhead Packets and End-to-End Delay. It was discovered that the AntNet algorithm performs averagely in all the cases of simulation. Its can also be concluded that the algorithm cannot perform well in higher ranges of node mobility. Furthermore, a large number of routing overheads are created by the AntNet algorithm significantly reducing the performance of the algorithm, moreover, since the control packets occupy the same queue as of the data packets, it can be noted that the probability of dropping the control and data packets are very high. This factor also increases the average time for the packets to reach the destination and hence a very high packet delay is present. Although, the AntNet algorithm is a hybrid algorithm, it uses proactive routing mechanisms to find and update its routing table, that is, the algorithm tries to find the information about the existing paths while the communication session is going on, hence producing large overheads. The routing tables used by the AntNet algorithms are not accurate enough as it selects only the path which has the highest pheromone value, if there is congestion present in the chosen path then the chances of reduced performance are very high. Finally, the use of proactive routing algorithms in the ad-hoc wireless networks is not recommendable, as the simulation results suggest. The best way to obtain the maximum performance in the ad-hoc networks is the use of reactive algorithms, instead of the proactive one.

References

[1] R. Bruno, M. Conti, and E. Gregori: Body, Personal and Local Ad Hoc Wireless Networks, The Handbook of AdHoc Wireless Networks, pp. 13-34, 2003

[2] K. Van Dam, S. Pitchers, and M. Barnard.: Body Area Networks: Towards a Wearable Future, Proc. Wireless World Research Forum, Munich, March 2001.

[3] K. Pahlavan and P. Krishnamurthy: Principles of Wireless Networks - A Unified Approach Prentice Hall, 2002.

[4] M. Conti and S. Giordano: Mobile Ad Hoc Networking, Cluster Computing Journal, Vol. 5, No. 2, April 2002

[5] T. G. Zimmerman: Personal Area Networks: Near-field intra body communication, IBM Systems Journal, 1996.

[6] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang: A wireless hierarchical routing protocol with group mobility. In Proceedings of IEEE WCNC'99, Sep 1999

[7] Di Caro G. A., Dorigo M.: AntNet: Distributed Stigmergetic Control for Communications Networks, Journal of Artificial Intelligence Research (JAIR), Vol. 9, Pg. 317-365, 1998.

[8] Lavina Jain: Implementation of AntNET algorithm in Wired Networks using NS-2.33", http://antnet.wordpress.com/.

[9] Ant Colony Optimization, Stigmergy Example, picture obtained from Wikipedia, http://de.wikipedia.org/wiki/Ameisenalgorithmus.

[10] M. Gerla: FSR: Fisheye State Routing Protocol for ad hoc networks, http://tools.ietf.org/html/draft-ietf-manet-fsr-03.

[11] J. J. Garcia-Luna-Aceves: STAR-Source Tree Routing in Wireless Networks, Presented in the International conference on Network Protocols, 1994.

[12] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum and L. Viennot: OLSR: Optimized link state routing protocol for ad hoc networks, IEEE INMIC, 2001.

[13] C. Chiang: CGSR: Routing in clustered mobile multi-hop ad hoc networks, IEEE SICON, pp. 197-211, April 1997.

[14] C. Toh: A novel distributed routing protocol to support ad hoc mobile computing, IEEE 15th Annual Phoenix Conference, pp.480-486, 1996.

[15] C. Perkins, S. Das and E. Royer: AODV: Ad Hoc on Demand Distance Vector Routing Protocol, Internet Draft -13, http://tools.ietf.org/search/draft-ietf-manet-aodv-13.

[16] Johnson, Maltz and Hu: DSR: The Dynamic Source Routing Protocol for Mobile Adhoc networks", 16/April/2003.

[17] N. Nikaein, H. Laboid and C. Bonnet: Distributed dynamic routing algorithm (DDR) for mobile ad hoc networks", MobiHOC 2000: Workshop on Mobile Ad Hoc Networking and Computing, 2000.

[18] C.E. Perkins, Bhagwat, "Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers", in: ACM SIGCOMM-94 Conference on Communications Architectures, London, UK, 1994.

[19] Xingwei Wang, Guang Ling and Min Huang: A BeeHive Algorithm Based QoS Unicast Routing Scheme with ABC Supported, Springer Link, APPT 2007, LNCS 4847, pp. 450-459, 2007.

[20] Li Fan and Li Taoshen, "Implementation and Performance Analyses of AnycastQoS Routing Algorithm Based on Genetic Algorithm in NS2", IEEE Computer Society, 978-0-7695-3634-7/09.

[21] S. Murthy J.J. Garcia-Luna-Aceves: A routing protocol for packet radio networks, First Annual ACM International Conference on Mobile Computing and Networking, Berkeley, CA, 1995, pp. 86–95.