Performance Enhancement of DYMO Routing Protocol with Ant Colony Optimization

Anuj K. Gupta
PTU, Punjab.
anujgupta@rimt.ac.in

Harsh Sadawarti
RIMT Group, Punjab.
harshsada@yahoo.com

Anil K. Verma
Thapar University, Punjab.
akverma@thapar.edu

***Abstract*—In this paper, an enhancement of the existing DYMO protocol has been proposed by applying ACO technique. Ant agents have been applied instead of TCP agents while generating traffic. The proposed routing protocol has been implemented using ant based routing. Network simulator ns2 version 2.34 has been used to simulate, verify and validate the results. The proposed protocol combines route maintenance feature of DYMO and shortest best path feature of ant routing technique. Shortest routes to the destination have been computed using pheromone trail values generated by ant agents and stored in pheromone routing tables. Node connectivity has been increased by using ant agents with DYMO mechanism.**

***Index Terms*—MANETs, DYMO, ACO, E-DYMO**

I. Introduction

A rapid growth and research has been seen in the field of Mobile Ad Hoc Networks (MANETs) due to their dynamic nature and infrastructure-less end-to-end communication. A Mobile Ad Hoc Network (MANET) is a collection of autonomous mobile nodes which dynamically form ad hoc network without the use of any infrastructure or centralized administration. It is sometimes also called as a multi-hop wireless network [1]. Routing in MANETs is a challenging task and has gained a remarkable attention from researchers worldwide. This has led to the development of different routing protocols with each protocol providing an improved version in respect to various strategies for a particular network scenario. None of the existing protocols is the best that justifies the characteristics and is suitable to perform an efficient routing. Researchers strive to uncover the efficiency of existing routing protocols by enhancing their performance. Lot of techniques exist which can be applied on the existing protocols to make them more efficient and less vulnerable to outer attacks. In the presented work Ant Colony Optimization (ACO) has been implemented to enhance the performance of a routing protocol. The proposed technique uses Artificial Intelligence on all the nodes in an ad hoc network to compute the most efficient and shortest route. The proposed protocol named E-DYMO (Enhanced DYMO) works better than the original one.

Manuscript received August 13, 2013; revised January 4, 2014.

II. DYMO Routing Protocol

Dynamic Manet On demand routing protocol (DYMO) has been proposed by Perkins and Chakeres [2] as advancement to the existing AODV protocol developed by Perkins (2003). DYMO is a purely reactive protocol in which routes are computed on demand i.e. as and when required. It is a descendant of AODV protocol, and can act both as proactively as well as reactively. It extends AODV with source route path accumulation feature of DSR. DYMO is a combination of AODV with DSR i.e. it is based on AODV structure but works on the mechanism of DSR. The DYMO protocol has two basic processes that are route discovery and route management. In route discovery process, the source node initiates broadcasting of Route Request (RREQ) packet all through the network to locate the sink node. When the sink node receives a RREQ it replies back with a Route Reply (RREP) message which is unicast towards the

doi: 10.12720/ijeee.2.3.188-194

source. When the source receives RREP, a bidirectional route is established between the nodes. Also in case of any link break or node failure a Route Error (RERR) packet is sent to the source node to indicate the broken route. Once the source node receives an RERR it re-initiates the route discovery process. Sequence numbers are used to ensure loop freedom. In addition, DYMO routing protocol can easily adapt to changes of topology and establish a unicast route between sink and source nodes [3].

As DYMO routing protocol is successor to the popular Ad hoc On-Demand Distance Vector routing protocol, it shares many of its benefits. DYMO protocol has the similar basic functions and operations to AODV. As a reactive protocol, DYMO does not explicitly store the network topology. Instead, nodes compute a unicast route towards the desired destination only when needed. As a result, little routing information is exchanged, which reduces network traffic overhead and thus saves bandwidth and power. DYMO is applicable to memory constrained devices. AODV supports unicast, multicast and broadcast. The DYMO only supports unicast routing established between the on-demand nodes in the network. DYMO routing protocol with excellent performance is simple, compact, easy to implement and highly scalable characteristics, and is a very promising protocol. Table I shows the comparison of some of the existing reactive routing protocols [4].

TABLE I. COMPARISON OF REACTIVE ROUTING PROTOCOLS

Parameters	AODV	DSR	TORA	DYMO
Route Creation	By source	By source	Locally	By source
Periodic updation	No	No	No	No
Performance Metrics	Speed	Shortness	Speed	Shortness
Routing overhead	High	High	High	Medium
Caching overhead	Low	High	Medium	Medium
Throughput	High	Low	Low	High
Multipath	No	Yes	Yes	No
Route updation	Non-periodic	Non-periodic	High routing overhead	Non-periodic
Congestion control	Yes	Yes	No	Yes

Route Discovery

The DYMO route discovery is very similar to that of AODV except for the path accumulation feature. Fig. 1 shows the DYMO route discovery process. If a source has no route entry to a destination, it broadcasts a RREQ message to its immediate neighbours. If a neighbour has an entry to the destination, it replies with an RREP message else it broadcasts the RREQ message. While broadcasting the RREQ message, the intermediate node will attach its address to the message. Every intermediate node that disseminates the RREQ message makes a note of the backward path. With respect to Fig. 3.6, source node 1 wants to communicate with destination node 10. It generates a RREQ packet which contains its own address, sequence number, hop count, destination address, and broadcasts it on the network. Each intermediate node having a valid path to the destination keeps on adding its address and sequence number to the RREQ packet as shown with nodes 2 and 6, till destination is reached. The source node waits for a RREP message. The Destination replies with RREQ message. A similar path accumulation process takes place along the backward path. This makes sure that the forward path is built and every intermediate node knows a route to every other node along the path. If source does not receive RREP within a specified TTL value, RREQ may be resent.

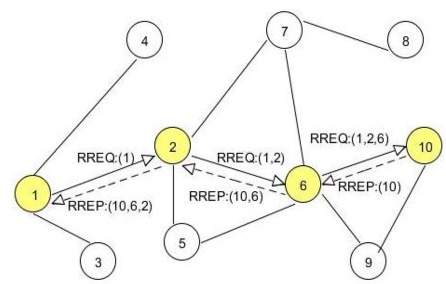

Figure 1. DYMO route discovery

Each node maintains a unique sequence number in order to avoid loops in the route and also to discard the stale packets if any. Every time a RREQ is sent, the router updates its sequence number. If the incoming packet has a same or inferior sequence number, the information is discarded. Messages with superior sequence numbers are updated in the routing table. If the sequence number associated with the incoming route is the same as the node sequence number then a loop is possible. In such case, the incoming packet is discarded. One of the special features of DYMO is that it is energy efficient. If a node is low on energy, it has the option to not participate in the route discovery process. In such a case, the node will not forward any of the incoming RREQ messages. It however will analyze the incoming RREP messages and update its routing tables for future use [5].

Route Maintenance

During the routing operations each node has to continuously monitor the status of links and maintain the latest updates within the routing tables. The route maintenance process is actually accomplished with the help of RERR messages. The RERR message must be generated by a node if and when a link to any other node breaks. The generating node multicasts the RERR message to only those nodes which are concerned with the link failure. Upon reception of a RERR message, the routing table is updated and the entry with the broken link is deleted. If any of the nodes face a packet to the same destination after deletion of the route entry, route discovery process needs to be initiated again [6].

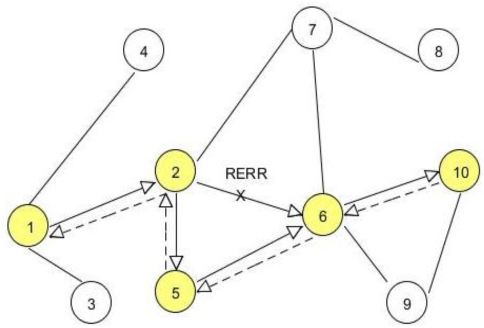

Figure 2. DYMO Route **Maintenance**

As seen from the Fig. 2, node 2 has received a packet that needs to go to node 6, but the route from node 2 to node 6 is found broken. In this case a RERR message is generated by node 2 and forward towards the source node 1. All the intermediate nodes on the path instantly update their routing table entries with the new updated information regarding link failure and new route changes. Now the packets will be forwarded from node 2 towards node 5 and then to node 6 and lastly to node 10 so as to reach the destination [7].

III. Ant Colony Optimization

Ant Colony Optimization (ACO) has been proposed by Dorigo *et al.* [8] and is basically inspired from the behaviour and action of the natural ants. The ants navigate their designated selection of paths while depositing a certain amount of substance called pheromone on the ground, marking a trail. The idea behind this technique is that the more ants follow a particular trail, the more attractive is that trail for being followed by other ants. They therefore dynamically find a path on the fly, using the explained notion of stigmergy to communicate indirectly amongst them. Stigmergy word is derived from stigma i.e. sting and ergon i.e. work, which means simulation by work. An ant chooses a trail depending on the amount of pheromone deposited on the ground. Each ant compares the amounts of trails, for the selected destination on each link, toward the neighbouring nodes. The larger the concentration of pheromone in a particular trail, the greater is the probability of trail being selected by an ant. The concentration of the pheromone on the links formed evaporates with time at a certain rate. Each node in the network has a routing table which helps it determine where to send the next packet or ant. These routing tables have the neighbours of the node as rows, and all of the other nodes in the network as columns. ACO algorithms have been employed to solve numerous problems in ad hoc networks. Dorigo was the first to propose ant algorithms as a multi-agent approach to difficult combinatorial optimizations problems such as the travelling salesman problem, graph colouring, quadratic assignment problem and routing in communication networks and so on [9].

An important behaviour of ant colonies is their foraging behaviour i.e. how ants find the shortest paths between food sources and their nest. While searching for food, ants deposit on ground an amount $\Delta\tau$ of special substance called pheromone at each visited node. The pheromone trail helps ants to find their way back to the food source. The ants which traverse through the shortest path reinforce the path with more amount of pheromone that helps other ants follow that path. However the deposition of amount of pheromone diverges from the observed behaviour of real ants.

Role of ACO in MANETs

Ant Colony Optimization technique is very much suitable for mobile ad hoc networks because of following characteristics:

- No single point of failure as the algorithm is fully distributive in nature
- Simple routing operations
- Ant agents fully cooperate and obey the routing rules
- No need of defining path recovery procedures as it is self organizing and fault tolerant
- Very much adaptable to the network traffic
- Compliant to the dynamically changing topology of ad hoc networks

IV. Implementation of Proposed Protocol

An enhancement of the existing DYMO protocol has been proposed and named it as E-DYMO (Enhanced DYMO). Ant agents have been applied instead of TCP agents while generating traffic. Ant Colony Optimization (ACO) has been implemented for computing shorter paths to the destination node based upon the pheromone trail values calculated for every valid path [10]. This value is used by a node to calculate probability of each next node which is to be next hop in order to reach the destination. For each destination-neighbour pair (d, n), the pheromone routing table T_k stores pheromone trail value P_{nd} which helps in choosing n as next hop to destination d. The proposed work is an implementation of this collective behaviour that has been applied to ad hoc network routing protocol. Since a routing protocol completely relies on a certain set of governing rules that help in monitoring the whole communication process within an ad hoc network, hence an important role has been played by the mobile nodes in managing these processes and network resources.

Ant agents

The primary step in implementation of ACO on an ad hoc routing protocol is to propose the ant based network model. This model defines the ant agent structure with some default values. These agents have been applied on the existing DYMO routing protocol during route discovery process to compute all the available paths from source to destination with their respective pheromone levels [11].

Network model

The network model consists of packet class and structure. It consists of information related to data packets and two special types of packets called artificial ants. One is named as Front Ant (FANT) and other one is named as Rear Ant (RANT). The role of FANTs is to search or find out new paths from its source and have been used in creating pheromone trails along the route from source to specific destination node in the network whereas the RANTs contain information related to destination node.

Pheromone routing table

Every intermediate node including the end nodes in the network maintain information in the form of a routing table, but contains one special field called pheromone trail value. These tables have been named as pheromone routing tables. Each node stores a pheromone routing table T_k organized as in distance-vector algorithms for storing probability trail values [12]. For each destination-neighbour pair (d, n), T_k stores a probability value P_{nd} which helps in choosing n as next-hop node to destination d, such that

$$\sum_{n \in N_k} P_{nd} = 1\,, d \in [1, N], N_k = \{neighbors(k)\}$$

The pheromone trail values at any edge (i, j) are updated by all the ants that have completed the path length as follows:

$$\tau_{ij} = (1 - \lambda).\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

where m is the number of ants that have completed the path.

$\lambda \epsilon (0,1)$ is the evaporation constant that determines the evaporation rate of the pheromone.

$\Delta\tau_{ij}^{k}$ is the quantity of pheromone deposited by ant k on edge (i, j).

A pheromone routing table is maintained at each node in the network. It represents a valid path from a node to every destination with the information of immediate next node. Also it contains respective pheromone values at each step. A pheromone routing table is made up of three distinct fields as shown below:

Dest_id	**Next_id**	**Pheromone trail value**

- **Dest_id**: It signifies the address of destination node.
- **Next_id**: It signifies the address of immediate next node to reach destination node.
- **Pheromone trail value**: This value is used by each node to compute the probability of each next node which is to be the next hop in order to reach the destination node. It always lies between 0 and 1.
- **Neighbour List**: It stores all the information related to neighbours in the form of list.

V. Procedure

Ant routing was basically proposed for fixed wired networks, which derives features from parallel replicated Monte Carlo systems, previous work on artificial ant colonies techniques and telephone network routing. An effort has been made to implement ant routing in ad hoc network. The entire network has been represented as a graph *G (n, l)* consisting of *n* nodes and *l* links. All the links in the network are considered bidirectional. Each node is considered both as a host and router. Every node in the network maintains a buffer composed of a single queue in which pheromone values of neighbouring nodes are maintained. This buffer is then consulted to form a routing table. Data packets do not maintain any routing information but use the information stored at routing tables for travelling from the source to the destination node. FANTs and RANTs act as mobile agents that are used to update the pheromone routing tables and distribute information about the traffic load in the network.

When a node receives a packet from any of its neighbours, the packet is first stored in its buffer. A routing table organized as a matrix with probabilistic pheromone entries is formed at each node. Each row in the routing table corresponds to one source, next node and pheromone value at that particular node while traversing the shortest path to the destination from current node. The pheromone routing table stores a probability value expressing the probability of choosing n as the next node while moving towards the destination node.

VI. Simulation Parameters

A rectangular grid of 1200 m x 300 m has been chosen for the use of longer routes between nodes. The mobility model uses the random waypoint model [13] in the rectangular field. The simulations have been run multiple times for variable pause times i.e. 0, 50, 100, 200, 300, 400 and 500 seconds. After pausing for a specific pause time the mobile node again selects a new destination and proceeds at a speed distributed uniformly between 0 and a maximum speed. The simulations are based on Continuous Bit Rate (CBR) sources which have been chosen to test the routing protocols. The source and destination nodes have been chosen randomly with uniform probabilities. Different parameters have been considered for simulations performed as shown in Table II below.

TABLE II. SIMULATION PARAMETERS

Parameters	Value
Simulation area	1200 m x 300 m
Transmission range	250 m
Simulation time	600 sec
Maximum velocity	15 m/s
Traffic type	CBR
Radio propagation model	Two-ray ground
Packet size	512 bytes
Mobility model	Random waypoint
Link bandwidth	2 Mbps
Packet rate	2 pkts/sec
Pause time	0 to 500 (interval of 100 sec)
Number of nodes	20, 30, 40, 50, 60

All the simulations have been performed with network size of 20, 30, 40, 50 and 60 nodes. The pause time has been varied with an interval of 100 sec up to 500 sec. Histograms have been plotted between varying network size and performance metrics for different pause times. From the available set of performance metrics four have been considered in this study and simulations have been performed on all the protocols to calculate the effective values for metrics in each case [14], [15].

- **Packet Delivery FRACTION (PDF)**: It is the ratio of the amount of data packets delivered to the destination and total number of data packets sent by source.
- **Average End-to-End Delay (AEED)**: The interval time between sending by the source node and receiving by the destination node, which includes the processing time and queuing time.
- **Routing Overhead (RO)**: The total number of routing packets transmitted during simulation. Routing Overhead is important as it measures the scalability of a protocol, the degree to which it will function in congested or low bandwidth environments.
- **Throughput (TP):** It is the average number of messages successfully delivered per unit time i.e. average number of bits delivered per second. Also refers to the amount of data transfer from source mode to destination in a specified amount of time.

VII. RESULTS

Extensive simulations have been carried out to evaluate and validate the proposed E-DYMO routing protocol with the conventional ant based routing technique. Network simulator ns2.34 has been used to simulate the protocol which can easily model and simulate a multi hop wireless ad hoc network. It is a discrete event simulator. The radio propagation model used for the simulation is two-ray ground reflection. For simulating the proposed protocol the number of ants was kept equal to the number of nodes. Similar movement and traffic scenarios have been used for all the simulations performed. The results have been computed and compared with existing version of DYMO protocol. Fig. 3 to Fig. 6 show the graphical representation of the simulation values for all the four performance metrics with network size.

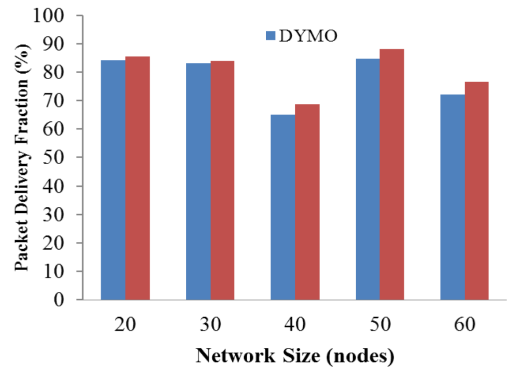

Figure 3. PDF with varying network size

A greater value of PDF for the proposed protocol has been observed from the Fig. 2. This shows that in E-DYMO, the link failures could be handled and dropping of packets might be reduced because the proposed protocol uses link failure detection from the existing DYMO protocol and shortest alternative path feature from ACO. As the pause time increases packet delivery fraction of E-DYMO increases in comparison to DYMO with a constant factor. Hence E-DYMO yields highest PDF as compare to DYMO.

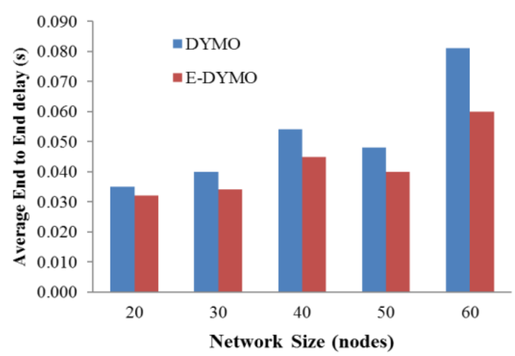

Figure 4. AEED with varying network size

As seen from the Fig. 4, the end-to-end delay is less for E-DYMO than DYMO. This is because of the ant agents that have been used to find optimal shortest route on route discovery. On link failures route discovery is not started from initial because of the presence of pheromone routing table which contain next alternative path to destination hence reduces the delay for sending of data packets. It has been observed during simulation that with varying pause time the delay values have shown a gradual increase in DYMO than in proposed E-DYMO.

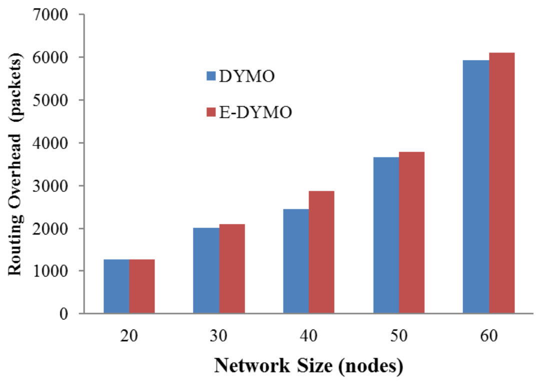

Figure 5. RO with varying network size

As seen from Fig. 5, the proposed E-DYMO protocol performs better in terms of all the metrics considered but incurs a little high overhead that the existing DYMO protocol which is within tolerable limits. This is because of the extra traffic generated by introducing special ant packets. However, these ants (Front and Rear) have been proved to be beneficial in computing multiple paths based on the pheromone values from source to destination node for each route.

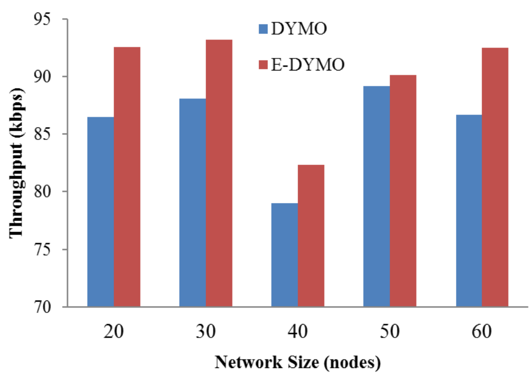

Figure 6. TP with varying network size

It has been observed from Fig. 5.6 that throughput of E-DYMO is high as compared to the existing DYMO protocol. This reflects a better performance for the proposed protocol.

VIII. Discussions

A network environment has been chosen that is rectangular in shape to make use of long distance routes between mobile nodes. ACO technique has been applied and simulated over the existing DYMO protocol. The lengths of the routing paths and the amount of pheromone trail values available on the connections between the nodes have been considered as design parameters to construct a pheromone routing table. It has been observed from the computations and histograms that the proposed E-DYMO protocol gives a better throughput value than the existing DYMO protocol w.r.t. different network size and pause time. Thus it can be said that E-DYMO performs better and the packet delivery rate of successfully reached packets is more than the existing version. E-DYMO incurs a high throughput which reflects that the proposed protocol has a very high success rate of packet transmission.

IX. Conclusion

The proposed E-DYMO routing protocol has been verified and validated through effective simulations performed on ns2 by varying the number of nodes and pause time. A total of six simulation rounds have been executed under each scenario of 20, 30, 40, 50 and 60 nodes to calculate the accurate value of every metric. In all the four metrics considered for simulation E-DYMO protocol has proven to be giving better values than the existing protocol. It has been observed that the ACO technique applied on the existing DYMO protocol has improved its performance and efficiency in terms of packet delivery ratio, average end to end delay and throughput but has a little high routing overhead which is within tolerable limits. In a nutshell it is stated that E-DYMO is a better and enhanced version of DYMO.

References

[1] A. Boukerche, B. Turgut, N. Aydin, M. Ahmad, L. Bölöni, and D. Turgut, "Routing protocols in ad hoc networks: A survey," *Elsevier Computer Networks*, vol. 55, pp. 3032–3080, 2011.

[2] I. Chakeres and C. Perkins. (2009). Dynamic MANET On-demand (DYMO) Routing draft-ietf-manetdymo-17. Internet Engineering Task Force. [Online]. Available: http://tools.ietf.org/html/draft-ietf-manet-dymo-17 as draft

[3] A. Gupta, H. Sadawarti, and A. Verma, "Implementation of DYMO routing protocol," *International Journal of Information Technology, Modeling and Computing*, vol. 1, no. 2, pp. 49-57, 2013.

[4] A. Gupta, H. Sadawarti, and A. Verma, "Review of various routing protocols for MANETs," *International Journal of Information & Electronics Engineering*, vol. 1, no. 3, pp. 251-259, 2011.

[5] P. Hofner, R. Glabbeek, W. Tan, M. Portmann, A. McIver, and A. Fehnker, "A rigorous analysis of AODV and its variants," in *Proc.Modelling, Analysis and Simulation of Wireless and Mobile Systems*, ACM press, 2012.

[6] C. Sommer and F. Dressler, "The DYMO routing protocol in VANET scenarios," in *Proc. 66th IEEE vehicular technology conference (VTC2007-Fall), IEEE*, Baltimore, 2007, pp. 16–20.

[7] R. E. Thorup, "Implementing and evaluating the DYMO," M.S Thesis, 2007.

[8] M. Dorigo and C. Blum, “Ant colony optimization theory: A survey,” *Theoretical Computer Science*, vol. 344, no. 2–3, pp. 243–278, 2005.

[9] X. Yang, L. Layuan, and C. Chuanhui, “Application research based ant colony optimization for MANET,” in *Proc. IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, Wuhan, China, 2006, pp. 1-4.

[10] J. Martins, S. Correia, Junior, J., “Ant-DYMO: A bio-inspired algorithm for MANETs,” in *Proc. 17th International Conference on Telecommunications*, Qatar, Apr 4-7, 2010, pp. 748–754.

[11] A. Gupta, H. Sadawarti, and A. Verma, "Analysis of various swarm based and Ant based algorithms," in *Proc. International conference on Advances in Computing and Artificial Intelligence ACM Chapter*, Chitkara University, Punjab, 21-22 July 2011, pp. 39-43.

[12] A. Gupta, H. Sadawarti, and A. Verma, “Performance analysis of MANET routing protocols in different mobility models,” *International Journal of Information Technology and Computer Science*, vol. 5, no. 6, pp. 73-82, 2013.

[13] P. Ruiz and F. Ros, “DYMOUM–A DYMO implementation for real world and simulation”, 2005.

[14] A. Gupta, H. Sadawarti, and A. Verma, “Computation of pheromone values in the AntNet algorithm,” *International Journal of Computer Network and Information Security*, vol. 4, no. 9, pp. 47-54, 2012.

[15] Network Simulator (ns2). [Online]. Available: http://www.isi.edu/nsnam/ns

Mr. Anuj K. Gupta is a PhD research fellow in Punjab Technical University, Punjab, India. His research area is Mobile ad hoc networks, wireless networks & Data Communication. He has a teaching experience of more 10 years. He is currently working as Head of CSE Dept. at RIMT group, Punjab.

Dr. Harsh Sadawarti is Director of RIMT Technical Campus, Punjab. He has a vast teaching and research experience of more than 20 years in the field of computer science. His areas of research are ad hoc networks, parallel computing & Distributed systems.

Dr. Anil K. Verma is faculty in Computer Sci. &Engg. Dept. at Thapar University, Patiala, Punjab. He has a vast teaching & research experience of more than 20 years. His areas of research are mobile ad hoc networks, wireless sensor networks & Network Security.

