

DESIGN AND DEVELOPMENT OF ACO ROUTING PROTOCOL FOR MANETs

**A Thesis submitted in fulfillment of the requirement
for the award of the degree of**

**DOCTOR OF PHILOSOPHY
IN
COMPUTER SCIENCE AND ENGINEERING**

Submitted by
Gurpreet Singh
(Registration No. 951003003)

Under the guidance of

Dr. Neeraj Kumar
Associate Professor
Computer Science & Engineering
Thapar University, Patiala (Punjab), India.

Dr. Anil K. Verma
Associate Professor
Computer Science & Engineering
Thapar University, Patiala (Punjab), India.



**COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY, PATIALA – 147004 (PUNJAB – INDIA)**

JULY 2014

CERTIFICATE

I, Gurpreet Singh, Regn. No. 951003003, hereby declares that the thesis entitled "Design and Development of ACO Routing Table Protocol for MANETs" submitted to the Department of Computer Science & Engineering at Thapar University, Patiala, Punjab, India is an authenticated record of my own work for the award of the degree of "Doctor of Philosophy" under the supervision of Dr. Neeraj Kumar and Dr. Anil Kumar Verma. This report has not been submitted to any other institution for the award of any other degree.

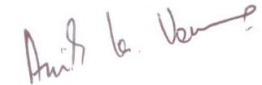

Gurpreet Singh
951003003

Place: Patiala
Date: 07/07/2014

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Approved by:


Dr. Neeraj Kumar,
(Supervisor)
Associate Professor,
Department of CSE,
Thapar University, Patiala.


Dr. Anil Kumar Verma,
(Supervisor)
Associate Professor,
Department of CSE,
Thapar University, Patiala.

ABSTRACT

Due to their wide applicability in wide range of application, Mobile Ad-hoc Networks (MANETs) in recent years have gained a lot of attention. MANETs can operate without any existing fixed infrastructure and have rapid changes in the network topology. Preparing and maintaining the routing table in communication networks, especially in MANETs, is a challenging task. Nodes dynamics in MANETs make the management of routing tables a complex job to be carried out. There exist many solutions for the routing problem in MANETs; each one having its advantages and limitations. These existing solutions need to be improved to achieve better performance with respect to various parameters. In fact, there is a new generation of bio-inspired routing protocols that have the potential to provide better performance than traditional protocols with the help of moving agents. Agents collaborate with each other to update the routing table quickly with less delay.

Ant Colony Optimization (ACO) is one of the most interesting bio inspired meta-heuristics technique. It simulates ant behavior to solve complex combinatorial problems. ACO algorithms include the method of optimization and reinforcement learning. ACO algorithms are decentralized, robust and resilient. These features make ants a good choice for taking routing decisions; where not only shorter paths need to be found, but also there is a need to achieve higher performance and greater reliability over the traditional protocols. It has been observed that ant algorithms influence the performance of the network due to their self organization characteristics.

The main contribution of this thesis is to design and develop new ant colony algorithms. The thesis contribution includes the blueprint and formulation of new ant-based algorithms for routing in MANETs. The presented framework has implemented using the novel ant routing algorithms in a simulated network. The experimental results show that ACO algorithms are well suited to the environment of MANETs. The proposed and implemented algorithms weave ACO properties. ACO algorithms were based on agent systems and worked with a group of ants that allow a high adaptation to the dynamic topology of MANETs. In contrast to other approaches for the formation of routing

information, ACO algorithms resulted in passing of local information at a fast pace by transmitting collected information to neighbors in the network. Each node maintains a routing table with entries for all of its neighbors which are called as the pheromone table. The decision rule, to select the next node, was built on the concentration of the pheromone in pheromone table. Further to realize ACO, a modified data structure was proposed that includes the new packet format for different types of ant, memory structure, traffic format and pheromone table. The work represented in thesis compared the developed protocol with traditional and other available ant based routing algorithms of MANETs. The performance of the proposed algorithms was found better than other state of the art algorithms.

Both the algorithms (ANTALG, OANTALG) are tested using simulation in NS-2.33 by creating random waypoint model. Various performance evaluation metrics such as throughput, packet loss, delivery ration, end to end delay, number of packets sent, and jitter were selected for valuation purpose. The performance of both the algorithms is compared with other state of art algorithms such as AntNet, AODV, DSR, ZRP and HOPNET. The result obtained with respect to above metrics prove the superior performance of the proposed algorithms. Specially, average throughput has been minimally increased by 2% to 37%; average jitter has been reduced by minimally 25%. Lesser number of packets were dropped which lead to better delivery ratio. At least 1.5% more number of packets was sent during communication and path length has also been reduced between 2.15% to 62% approximately.

Keywords: MANET, ACO, ANTALG, OANTALG, Routing Table, Ant Routing

ACKNOWLEDGEMENTS

First of all and the greatest importance, I would be thankful to GOD for all his blessings without which nothing of my work would have been carried out.

I am highly indebted to Prof. (Dr.) Prakash Gopalan, Director, Thapar University, Patiala and Prof. (Dr.) K. K. Raina, Deputy Director, Thapar University for providing me the opportunity to pursue my Ph.D. and course work.

This thesis would not be possible without the continuous guidance from my supervisor Dr. Anil Kumar Verma, Associate Professor, Computer Science & Engineering Department, Thapar University, Patiala. The extensive discussions with him have helped me to stay on the right track towards finishing this work. He was always patience and helpful whenever his guidance and assistance was needed. He also helped me by sparing his valuable time for reviewing my experimental setup, publications, reports and presentation of my work from time to time. He also helped me in shaping my initial work of thesis during synopsis submission and I would like to acknowledge him with deferential thanks.

I owe countless and immense gratitude to my supervisor Dr. Neeraj Kumar, Associate Professor, Computer Science & Engineering Department, Thapar University, Patiala for his day and night advice, guidance and support throughout the entire work. As my supervisor, he provided me with the encouragement and freedom to pursue my own ideas. He taught me how to do the research. In other words, he taught me how to make gold out of stones. He always remained tranquil while outlaying his edifying time on me. I am also grateful to him for spending several hours reviewing the document for accuracy and providing valuable feedback during the several iterations that this document went through. Therefore, I want to show my deep appreciation to him.

I would also like to convey my thanks to Dr. Deepak Garg, Head, Computer Science & Engineering Department for providing me the continuous feedback throughout my Ph.D. work. I am thankful to my doctoral committee members Dr. V. P. Singh Kaushal

and Dr. Rinkle Aggarwal for their constructive comments and ensuring the progress of my research work in the right direction. Their constant opinions and ideas help me to reach at this stage of my course. The help rendered by them is greatly appreciated. My deep regards to Dr. P.K Bajpai, Dean of Research and Sponsored Projects for his enormous help in completion of my course work.

I will go amiss if I forget to thank Dr. Maninder Singh, Dr. Inderveer Channa, Dr. Kulbir Singh, and Dr. Parteek Bhatia for their invaluable suggestions during various progress monitoring presentations. I wish to thank all the faculty and staff members of Computer Science and Engineering Department of Thapar University, Patiala for their co-operation and support.

Lastly, I would like to pay my gratitude to my grandmother, parents, wife, children, brother, and his family, mother in law and to all friends and colleagues for their love and encouragement all through my life. They have always helped me at every turn of my life when I am in a need.

Words cannot truly express my deepest gratitude and appreciation to all of the above.

I am sorry, if I have forgotten someone and I cannot thank everyone enough for the involvement they have shown and the willingness they have expressed to take on the completion of tasks beyond their comfort zones.

Gurpreet Singh

CONTENTS

Certificate	i
Abstract	ii
Acknowledgement	iv
List of Figures	x
List of Tables	xii
List of Important Abbreviations	xiv
List of Symbols and their meanings used in Equations and Data Structures	xvi
List of Publications	xviii
1. Introduction	1
1.1. Routing in MANETs	6
1.2. Classification of Routing Protocols in MANETs	6
1.2.1. Table-Driven Routing (Global/Proactive protocols)	7
1.2.2. On-demand Routing (Reactive Protocols)	8
1.2.3. Hybrid Protocols	9
1.3. Background	10
1.4. Ant Colony Optimization (ACO)	11
1.5. Main ACO Algorithms	13
1.6. Fitness of the ACO Approach in MANET Environment	14
1.7. Motivation	15
1.8. Problem Formulation	16
1.9. Research Goals / Objectives	17
1.10. Summary of Contributions	17
1.11. Organization of the Thesis	19
2. Literature Survey and Background of Ant Based Routing algorithms for MANETs	21
2.1. Categorization of Various ACO Routing Algorithms	21
2.2. Table Driven / Reactive Ant Based Routing Protocols	23
2.2.1. Ant Based Control (ABC)	23

2.2.2.	AntNet	24
2.2.3.	Accelerated Ants Routing (AAR)	26
2.2.4.	Ant Routing Algorithm for Mobile Ad-hoc Networks(ARAMA)	27
2.2.5.	Mobile Agent Based Routing (MABR)	28
2.2.6.	Probabilistic Ant Routing (PAR)	29
2.2.7.	Automatic Clustering inspired by Ant Dynamics (ACAD)	29
2.2.8.	Position Based Ant Colony Routing Algorithm for MANETs (PBANT)	30
2.2.9.	Modified Termite algorithms (MTA)	31
2.3.	On Demand / Reactive Ant based Routing Protocols	32
2.3.1.	Cooperative Asymmetric Forward (CAF)	32
2.3.2.	Termite	32
2.3.3.	ANSI	33
2.3.4.	The Ant-Colony-Based Routing Algorithm (ARA)	33
2.3.5.	Probabilistic Emergent Routing Algorithm (PERA)	35
2.3.6.	Ant-Based Distributed Route for Ad-hoc Network (ADRA)	35
2.3.7.	Position Based Ant Colony Routing Algorithm for Mobile Ad-hoc Networks (POSANT)	36
2.3.8.	Modified Ant Colony Algorithm (MACA)	38
2.3.9.	Ant Based QoS Routing Protocol for MANET (ARMAN)	38
2.3.10.	Ring Search Based Ant Routing for MANETs (RSAR)	39
2.3.11.	Enhanced Multi Path Dynamic Source Routing (EMP-DSR)	40
2.3.12.	Load Balanced Multi-Route AODV Ant Routing Algorithm (LBMRAA)	40
2.3.13.	Distributed Ant Routing (DAR)	41
2.3.14.	The Simple Ant Routing Algorithm (SARA)	42
2.3.15.	Ant Dynamic Source Routing (ADSR)	43
2.3.16.	Improvised Ant Colony Optimization Algorithm for Mobile Ad-hoc Networks (PACONET)	44
2.3.17.	A Nature Inspired Scalable Routing Protocol for Mobile Ad Hoc Networks (NISR)	45
2.4.	Hybrid Ant Based Routing Protocols	45
2.4.1.	Ant – AODV	45
2.4.2.	AntHocNet	46

2.4.3. Ant Routing Algorithm for Mobile Ad-hoc Networks Based on Adaptive Improvement (ARAAI)	47
2.4.4. Multi Agent Ant Based Routing Algorithm (MAARA)	48
2.4.5. HOPNET	50
2.4.6. Hybrid Routing Algorithm Based on Ant Colony and ZHLS Routing Protocol for MANET (HRAZHLS)	51
2.4.7. Ant based Dynamic MANET On-demand Routing (Ant-DYMO)	51
2.4.8. Ant-Based Routing Algorithm using Enhanced Path Maintenance (EPMAR)	52
2.4.9. Improved Ant Colony Optimization Based Multicasting in MANET (IACO)	53
2.5. Comparison of Reviewed Ant Based MANET Protocols	54
2.6. Research Paradigm / Methodology	57
2.7. Introduction to Network Simulator (NS-2)	58
3. Design of Innovative ACO Approach in Routing	62
3.1. Problem Background	62
3.2. Data Structures Used	64
3.3. ANTALG: Ant Routing Algorithm	66
3.4. Phases in ANTALG	68
3.4.1. Route Setup Phase	68
3.4.2. Route Maintenance Phase	70
3.4.3. Route Recovery Phase	71
3.5. Complexity Analysis	74
3.6. Illustrative Example 1: ANTALG	75
3.7. OANTALG: <u>O</u> rientation Based <u>ANT</u> Routing <u>AL</u> Gorithm	78
3.8. Modifications Done in OANTALG from ANTALG	79
3.9. Pseudo Code for OANTALG	80
3.10. Illustrative Example 2: OANTALG	82
4. Experimental Setup and Comparative Results	86
4.1. Performance Metrics	86
4.1.1. Throughput	86
4.1.2. Packet Drop and Delivery Ratio	87

4.1.3. Data Packets Sent	87
4.1.4. Path Length	87
4.1.5. End-to-End Delay	87
4.1.6. Jitter	88
4.2. Parameter Settings	88
4.3. Assumptions	90
4.4. Experimental Results of First Stage of Simulation Work	91
4.4.1. Routing Table / Optimized Path Calculation	93
4.4.1.1. Case 1 (Shorter Path Length)	93
4.4.1.2. Case 2 (Better Concentrated Pheromone Path)	96
4.4.2. Jitter	102
4.4.3. End to End Delay	103
4.4.4. Control Packet Overhead	105
4.4.5. Packet Loss	106
4.4.6. Delivery Ratio	107
4.4.7. Conclusion at the End of Stage One	109
4.5. Experimental Results of Second Stage of Simulation Work	109
4.5.1. Throughput	110
4.5.2. Number of Packets Dropped and Delivery Ratio	112
4.5.3. Number of Packets Sent	113
4.5.4. Path Length	114
4.5.5. End to End Delay	115
4.5.6. Jitter	116
4.5.7. Conclusion at the End of Results from Stage Two	117
4.6. Experimental Results of Third Stage of Simulation Work	117
4.6.1. Effect on Throughput	118
4.6.2. Effect on Packet Drop and Packet Delivery Ratio	120
4.6.3. Effect on Data Packets Sent	121
4.6.4. Effect on Path Length	122
4.6.5. Effect on End to End Delay	122
4.6.6. Effect on Jitter	123
4.6.7. Conclusion at the End of Results from Stage Three	124
5. Conclusion & Future Work	127
References	129

LIST OF FIGURES

Table No. Description

1.1	History of Wireless Data Networks	2
1.2 (a)	Mobile Ad-hoc Network	4
1.2 (b)	History of MANETs	4
1.3	Applications of MANETs (Realistic Projects by Various Countries)	5
1.4	Outline of the different flat-routing protocols for MANETs.	7
1.5 (a)	Double Bridge Experiment – Ants on the shorter path reaches to food source earlier.	11
1.5 (b)	Ants on the shorter path return to nest before the ants on longer path1	1
1.6	Applications of Ant Colony optimization	12
1.7	Real World ACO Applications	13
2.1	Categorization of Ant Based Routing Table Protocols for MANETs	22
2.2	General Screenshot of Network Simulator-2	60
3.1	Route Setup Phase	70
3.2	Route Maintenance & Route Recovery Phase	73
3.3	MANET representing set of nodes within the range for communication	75
3.4(a)-(b)	Generalized Network Topology in the proposed scheme	83
4.1	Screenshot of ANTALG in static network	92
4.2	Jitter observed in different executions	102
4.3	Average Jitter	103
4.4	End to End Delay observed from different Experiments	104
4.5	Average End to End Delay from different experiments	104
4.6	Control Packets sent in different experiments	105
4.7	Average Control Packets sent calculated from different experiments	105
4.8	Number of Packets lost during different experiments	106
4.9	Total number of Packets lost in all the experiments	107
4.10	Delivery Ratio in all the experiments	108
4.11	Delivery Ratio from all the experiments	108
4.12	Running screenshot for wireless network running using ANTALG	110

4.13	Throughput (Packets per Second)	110
4.14	Throughput (Bytes per Second)	111
4.15	Average Throughput (Bytes per Second)	111
4.16	Number of Packets lost	112
4.17	Delivery Ratio	113
4.18	Total number of Data Packets Sent	114
4.19	Average Path Length	115
4.20	Average End to End Delay	115
4.21	Average Jitter	116
4.22	Running screenshot for wireless network running using OANTALG	118
4.23	Variation in throughput measured in Bytes/Second after every 20 seconds	119
4.24	Average throughput after complete time	119
4.25	Total Number of Packets Drop	120
4.26	Delivery Ratio	121
4.27	Total TCP Packets sent by each Protocol	121
4.28	Average Path Length (Number of hops between the source and destination)	122
4.29	Total End- to- End Delay between the source and destination	123
4.30	Average End- to- End Delay between all sources and destinations	123
4.31	Total Jitter observed between the source and destination	124
4.32	Average Jitter between all sources and destinations	124

LIST OF TABLES

Figure No.	Description	
2.1	Pheromone table for node N	25
2.2	Brief tabular comparison between the all ants based MANET Protocols	54
3.1	Pheromone table	64
3.2	Ant Memory Structure	65
3.3	Ant Packet Format	65
3.4	Traffic Matrix	66
3.5	Pheromone table of node A	76
3.6	Pheromone table of node G	76
3.7	Pheromone table of node D	76
3.8	Pheromone table of node C	76
3.9	Pheromone table of node F	76
3.10	Pheromone table of node B	77
3.11	Pheromone table of node E	77
3.12	Pheromone table of node D (<i>before updation</i>)	78
3.12(a)	Pheromone table of node D (<i>after updation</i>)	78
4.1	Stages in simulation work	89
4.2	General Parameters Used during simulation in different stages	89
4.3	Routing Table at node 10	93
4.4	Routing Table at node 19	94
4.5	Routing Table at node 16 (AntNet), node 5 (ANTALG)	95
4.6	Routing Table at node 15 (AntNet), node 7 (ANTALG)	96
4.7	Routing Table at node 0	97
4.8	Routing Table at node 2	98
4.9	Routing Table at node 3	98
4.10	Routing Table at node 5	99
4.11	Routing Table at node 7	100
4.12	Routing Table at node 11	100
4.13	Routing Table at node 13	101

4.14	Jitter surveyed in different executions	102
4.15	End to End Delay monitored in different executions	103
4.16	Control Packet overhead monitored in different executions	105
4.17	Total number of packets lost in different executions	106
4.18	Delivery ratio in different experiments	107
4.19	Relative comparison of all algorithms with respect to various parameters	125
4.20	Percentage Improvement of OANTALG over other Algorithms	126

LIST OF IMPORTANT ABBREVIATIONS

Abbreviations /

Acronyms

Meanings

AAR	Accelerated Ants Routing
ABC	Ant Based Control
ACAD	Automatic Clustering Inspired by Ant Dynamics
ACO	Ant Colony Optimization
ACS	Ant Colony System
ADRA	Ant-Based Distributed Route for Ad-hoc Network
ADSR	Ant Dynamic Source Routing
Ant- AODV	Ant based Ad-hoc on Demand Distance Vector Routing
ANSI	Adhoc Networking with Swarm Intelligence
Ant-DYMO	Ant Based Dynamic MANET On-demand Routing
AODV	Ad hoc on Demand Distance Vector Routing Algorithm
ARA	Ant based Routing Algorithm
ARAAI	Ant Routing Algorithm for Mobile A-hoc Networks based on Adaptive Improvement
ARAMA	Ant Routing Algorithm for Mobile Ad-hoc Networks
ARMAN	Ant based QoS Routing Protocol for MANET
ARPA	Advanced Research Project Agency
ARPANET	Advanced Research Project Agency Network
AS	Ant System
BANT	Backward Ant
CAF	Cooperative Asymmetric Forward
CBRP	Cluster based routing protocol for MANETs
CGSR	Cluster Head Gateway Switch Routing Protocol
CLAR	Cluster Oriented Agent Based Routing Protocol
DAR	Distributed Ant Routing
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
FANT	Forward Ant
EMP-DSR	Enhanced Multi Path Dynamic Source Routing

EPMAR	Ant-based Routing Algorithm using Enhanced Path Maintenance
GPS	Global Positioning System
HRAZHLS	Hybrid Routing Algorithm Based on Ant Colony and ZHLS Routing Protocol for MANET (HRAZHLS)
IACO	Improved Ant Colony Optimization based Multicasting in MANET
LBMRAA	Load Balanced Multi-Route AODV Ant Routing Algorithm
MABR	Mobile Agent Based Routing Algorithm
MACA	Modified ant Colony Algorithm
MALBACO	Multi-Agent Load Balanced Ant Colony Optimization Algorithm
MANET	Mobile Ad-hoc Network
MMAS	Max-Min Ant System
MAARAA	Multi Agent Ant Based Routing Algorithm
MTA	Modified Termite Algorithm
NISR	Nature Inspired Scalable Routing Protocol for MANETs
NS-2 / NS-2.33	Network Simulator version 2/2.33
OPS	Optical Packet-Switching
OSI	Open Systems Interconnection
PACONET	ImProved Ant Colony Optimization Algorithm for MANETs
PAR	Probabilistic Ant Routing
PBANT	Position Based Ant Colony Routing Algorithm for MANETs
PERA	Probabilistic Emergent Routing Algorithm
PSO	Particle Swarm Optimization
POSANT	Position based ant colony routing algorithm for MANETs
QoS	Quality of Service
RSAR	Ring search based ant routing
SACO	Station ant colony routing algorithm
SARA	Simple Ant Colony algorithm
SI	Swarm Intelligence
TSP	Travelling Salesman Problem
TTL	Time to Live
WRON	Wavelength Routed Optical Networks
ZRP	Zone Routing Protocol
ZHLS	Zone Based Hybrid Link State Routing Protocol

LIST OF IMPORTANT SYMBOLS AND THEIR MEANING USED IN EQUATIONS AND DATA STRUCTURES

Symbol	Meaning
nn_p	p^{th} Neighbor Node
tn_o	o^{th} Target (Destination Node)
$\Omega(o, p)$	Probability of selecting the p^{th} neighbor node nn_p as next node while moving toward the o^{th} destination node tn_o
Ω_{ijd}	To compute the normalized sum of pheromone
NO_Add	Node Address
TR_Time	Trip Time to visit a node by ant
PKT_Type	Type of packet, i.e., packet to be used by forward ant and backward ant
SRC_Add	Source Node Address
TAR_Add	Target Node Address
PKT_Len	Length of the packet
SQN_NO	Sequence number of the packets
ST_Time	Start time of the packet.
RI_Max_No	Maximum number of nodes present in the route information
MEM_RI	Size of memory required by the route information.
M_TT	Mean of the ant's trip time
V_TT	Variance of traffic trip time
B_TT	Best trip time observed by the ant
ρ / ρ_{high}	Pheromone Value / Highest pheromone value obtained in iteration
ρ_{ijd}	Pheromone of edge $\langle i, j \rangle$ while moving towards destination d
θ	Weigh the importance of the heuristic function w.r.t. pheromone deposited
$ Ni $	Number of Neighbors
Γ	Heuristic Function (controlled by the value of the length of the queue)
l_{ij}	Length of the queue on the link connecting the node i with its neighbor j
$c1 \ \& \ c2$	Real values in the interval (0, 1) so that $c1 + c2=1$
C	Constant which is dependent on to the simulation time

r	Reinforcement factor is a dimensionless constant whose value ranges from (0, 1].
$t_{i \rightarrow d}$	The trip time observed by newly forward ant from node i to destination node d
t_{best_d}	The best time observed by the forward ant moving towards destination d
t_{sup}	Represents the time observed by ant in the observation window.
$T_{i,xy}$	Time needed by the packet while traveling from node x to node y
TT_{xy}	Transmission Time of link between node x and y .
ℓ_x	Length of low priority queue of link x and y .
Cap_{xy}	Capacity of the link x and y .
$SO(packet)$	Size of the packet.
H	Maximum hop count of an ant
N_a	Number of ants generated by a node
P_{max}	Upper bound of the maximum number of paths between source node and destination node.
$BN_{(s, j, d)}$	Best next node to reach node j while moving from source s towards destination d .
R_i^j	Route to reach node j from i
SN	Represent source node in data structure
sel_node	Shows the randomly selected node
CN	Represent current node's identity
NN	Shows selected next node identity
TN	Represent target node's identity
$Cos\alpha / CT$	Represents the cosine value of the axial coordinates of the two orientation vectors v_1 and v_2 . For $Cos\alpha$ the word CT is used in Pseudo code
v_1	Represent orientation vector directed from current node to the next node
v_2	Represent orientation vector directed from current node to target node
(x_n, y_n)	Used to represent the coordinates of the vectors
N	Number of nodes in the network

LIST OF PUBLICATIONS

- I. Gurpreet Singh, Neeraj Kumar, Anil Kumar Verma, “Ant colony algorithms in MANETs: A review”, Journal of Network and Computer Applications, Vol. 35, Issue 6, pp. 1964-72, 2012. **(SCI-Thomson Reuters Rated Journal with Impact Factor 1.467)**

- II. Gurpreet Singh, Neeraj Kumar, Anil Kumar Verma, “OANTALG: An Orientation based ANT Colony ALGORITHM for Mobile Ad-hoc Networks”, Wireless Personal Communication, Volume 77, Issue 3, pages 1859-1884, DOI: 10.1007/511277-014-1613-6. **(SCI-Thomson Reuters Rated Journal with Impact Factor 0.428)**

Chapter 1

Introduction

A computer network, usually called as a network, is a collection of computers and other devices interconnected by channels that facilitate communication between users and allow them to share information and resources. Computer networking or Data communication is the engineering discipline concerned with the communication between computer systems or devices [1] [2]. In the 1960s, the Advanced Research Projects Agency (ARPA) initiated to fund the design of the Advanced Research Projects Agency Network (ARPANET) for the United States Department of Defense. Computer networks can be used for several purposes like facilitating communication, sharing hardware, information preservation, security, sharing of files, data, information, software, etc. [2] [3].

Improvements in the diverse technology networking and communication are flourishing with time. Different kind of communication devices and medium are available for communication and numerous networks mixtures are present and even within these networks different preposition and parametric approaches for implementations are deployed according to various organizational needs [2] [3] [4]. Further, these are driven by the selection of efficient routing protocol to support data transmission successfully. Wireless communication has become very pervasive and ubiquitous. The number of mobile phones and wireless Internet users has increased significantly in recent years. Technological innovations in engineering during the 20th century have brought a deep change in the lifestyle of the public. Wireless communication is the fastest growing sector of the communication industry. It has captured the attention of the large segment of network users [5] [6].

Wireless communication systems have transfigured the way people communicate [8]. The vision of wireless communication in supporting communication of information

exchange between end users is the communication frontier of the next few decades, and a larger part of it already exists in many forms. Wireless networking has already crossed a few decades of its age and many are expected in upcoming years. The Figure 1.1 illustrates the history and development milestones in the wireless data networking [3] [7] [9].

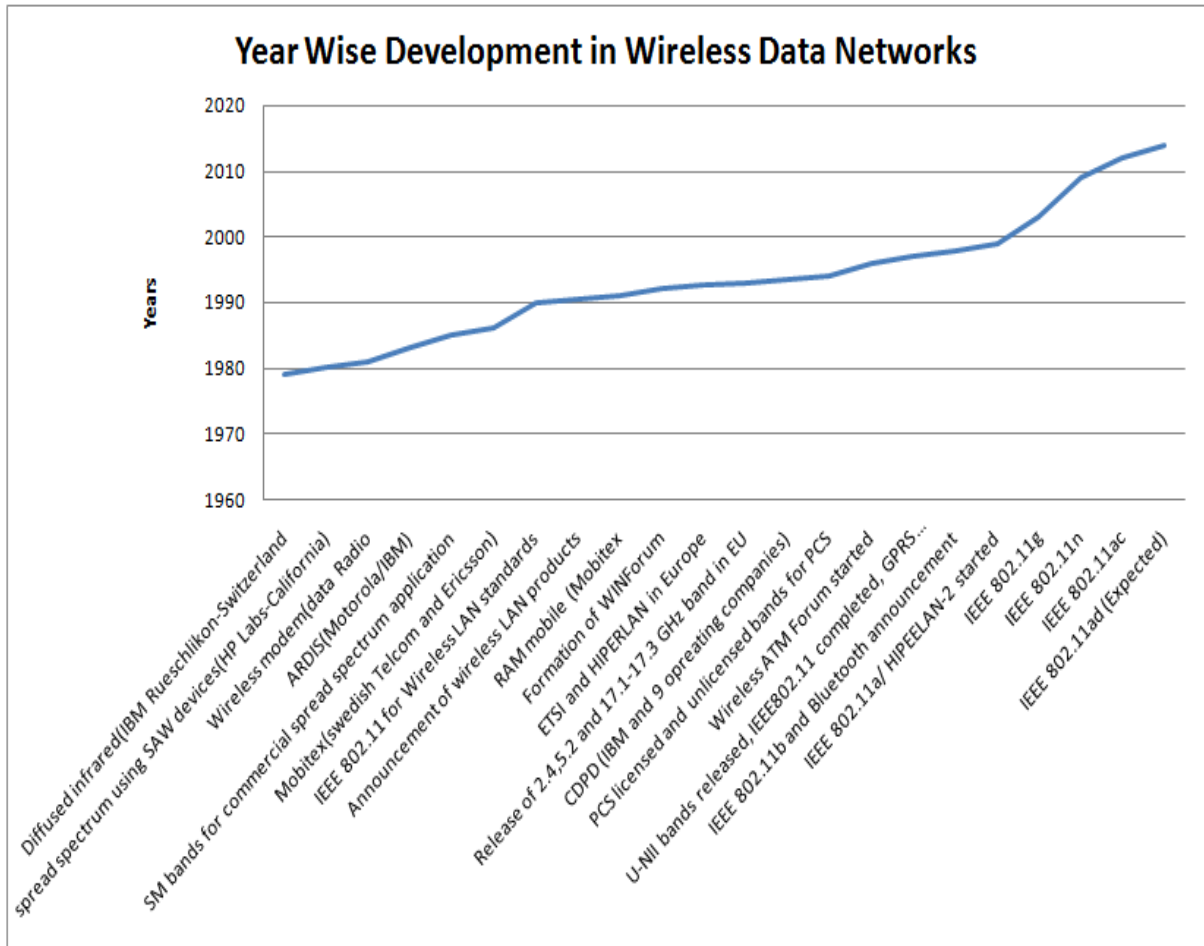


Figure 1.1: History of Wireless Data Networks

Due to its ability to fetch the data anytime, anywhere, anyhow the wireless networks have become popular in the last few years. There exist mainly three types of mobile wireless networks [9] [10]:

- I. Infrastructure networks,
- II. Ad-hoc networks and
- III. Hybrid networks

Infrastructure networks are deployed with the help of both the fixed and wireless nodes. MANET is a collection of wireless mobile nodes which form a temporary network without the use of any centralized administration, access point, and infrastructure [11]. Hybrid networks combine features of both infrastructure and ad-hoc networks. In MANETs, multiple hops are required to establish a data transmission between the nodes due to the limited transmission range. Without any pre installed fixed network infrastructure and with the rapid configuration of wireless connections on-the-fly, network topology in MANETs keeps on changing frequently. Thus, achieving data communication among wireless nodes in MANETs with better QoS parameters has become a challenging task [12] [13] [14]. To accomplish this, one has to focus on the routing protocol. Mobility of the nodes makes the situation even more complicated and designing a routing protocol for MANET has to face the challenge of frequently changing topology, low transmission power and asymmetric links [11] [15] [16]. Some of the characteristics of MANETs are as follows:

- I. As the nodes move freely in all directions, at an unknown speed. So, dynamic topologies are created.
- II. Routing is a multi-hop fashion process. Every node in the network can act as a router, and may deliver packets through an established route.
- III. Bandwidth is one of the major constraints of MANET due to the wireless nature of the communication media. Because wireless media can be easily influenced by multipath interference, noise, fading and also have limited availability of spectrum.
- IV. The operations in MANET are mostly power-restricted due to battery life.
- V. Because of decentralization, a MANET is robust and has limited physical security. MANETs are vulnerable to multiple types of attacks.
- VI. Scalability is also a main limitation of MANET due to their resource poorness nature.

A typical MANET made by portable devices can be represented as illustrated in Figure 1.2(a). Historically, the whole life cycle of MANETs could be categorized as follows in various technological terms. The chronological sequence of milestones in the history of MANETs evolution is shown in Figure 1.2(b) [1] [2] [3].

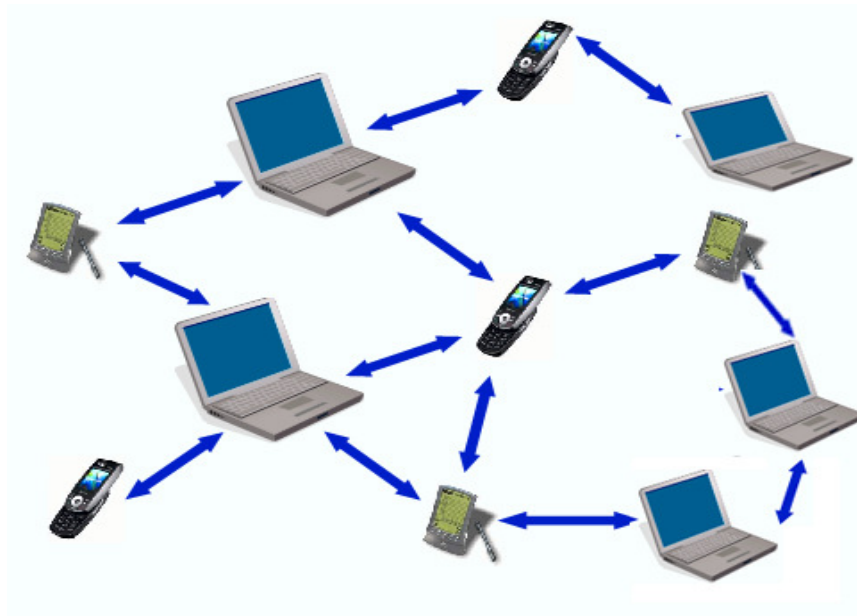


Figure 1.2 (a): Mobile Ad-hoc Network



Figure 1.2 (b): History of MANETs

Much of the progress into the performance has already been made in data transfer rates of wireless networks. This trend is shaping a future in which computers will become increasingly smaller and inexpensive, and therefore abundant. With today's growing use of technology, MANETs are used in a number of environments for various

reasons [1] [3]. Figure 1.3 represents the few areas of applications of MANETs started as realistic projects by different research organizations or countries.

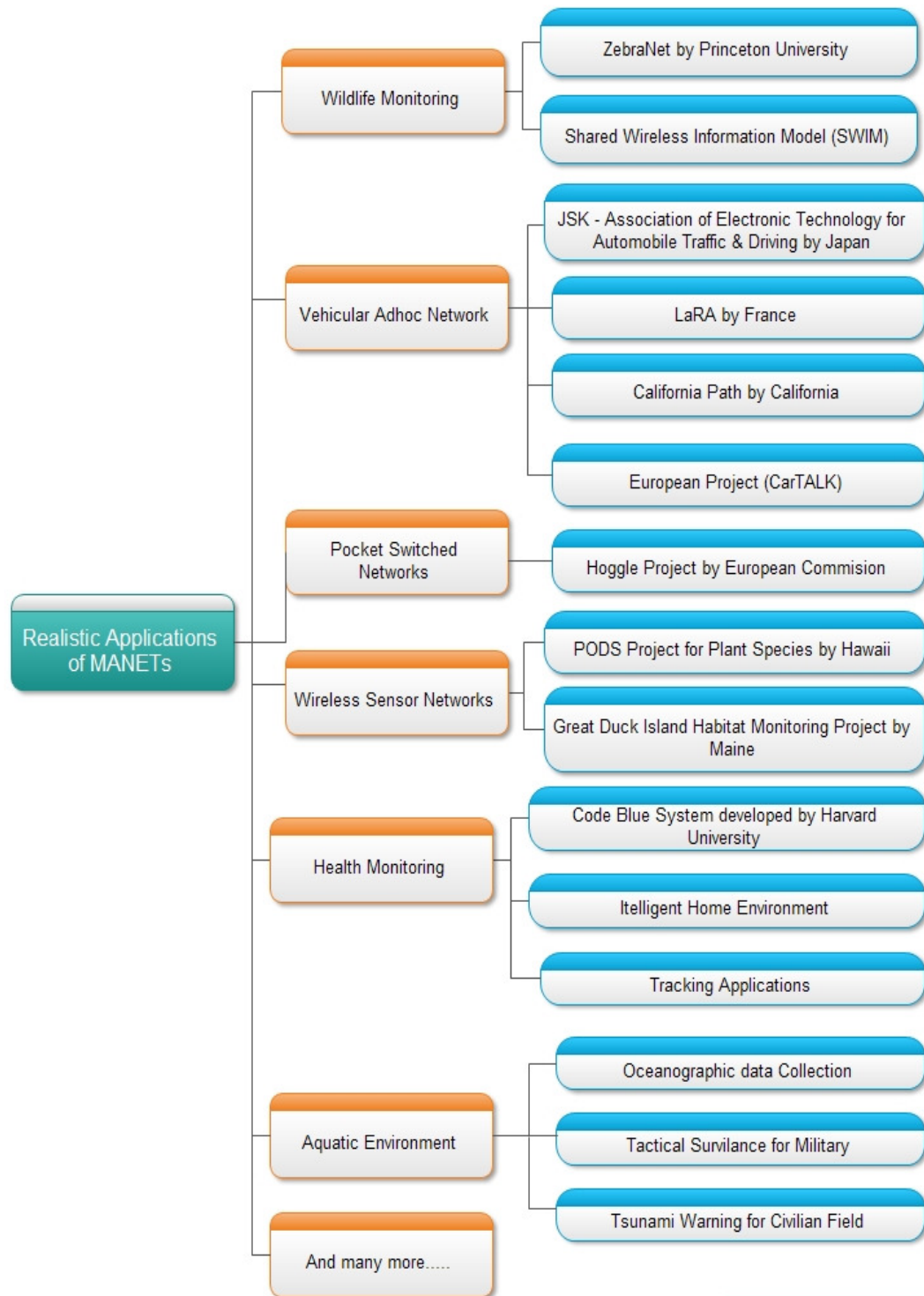


Figure 1.3: Applications of MANETs (Realistic Projects by Various Countries)

1.1 Routing in MANETs

A routing protocol is the mechanism by which user traffic is directed and transported through the network from the source node to the destination node. Objectives include maximizing network performance from the application point of view - application requirements- while minimizing the cost of the network itself in accordance with its capacity [17] [18]. The evaluation parameters of such applications include hop count, delay, throughput, loss rate, stability, jitter and cost. The definitions of these parameters are covered in the chapter comprising of the comparative results during the evaluation of new protocols.

The core basic routing functionalities for MANETs are path generation, path selection, data forwarding and path maintenance. Due to its characteristics, other desirable features of Ad-hoc routing protocols include: fast route establishment, multiple route selection, energy/bandwidth efficiency and fast adaptability to link changes. Almost all routing systems respond in some way to the changes in network and user traffic state [19] [20]. However, routing systems vary widely in the types of state changes to which they respond and the speed of their response.

1.2 Classification of Routing Protocols in MANETs

The routing protocols can be mainly categorized as: flat routing, hierarchical routing and location aware routing on network structure. In flat-based routing, all nodes play the same role. In hierarchical –based routing, however, nodes will play a different role in the network. In location aware-based routing, node positions are exploited to route data in the network. According to the routing strategy, the routing protocols can be categorized as table-driven and source initiated, while depending on the network structure these are classified as flat routing, hierarchical routing and location aware routing [21]. Both the table-driven and source initiated protocols come under the flat routing. As the emphasis of the research work being presented is on flat routing, the thesis work further presents the different categorization in flat routing and its strategies.

There are three schemes in flat routing, namely, table-driven (or proactive) routing protocols, on-demand (or reactive) routing protocols, and hybrid routing protocols. One thing is general for both protocol classes is that every node participating in routing plays an equal role [21] [22]. Figure 1.4 represents this scenario of flat routing with some famous protocols under each category.

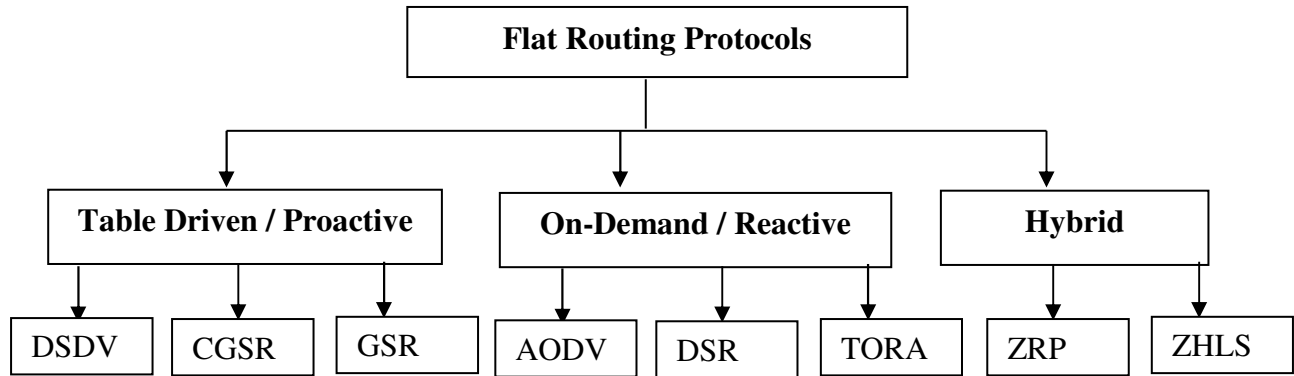


Figure 1.4: Outline of the different flat-routing protocols for MANETs [3] [5] [10] [11]

1.2.1 Table-Driven Routing (Global/Proactive protocols)

In the proactive routing protocols [19] [20] [21], the routes to all the destinations (or parts of the network) are determined at the startup, and maintained by using a periodic route update process. In proactive routing protocol, each node maintains the information about the other nodes in the form of Tables. That is why they are called as Table-driven protocols. The various proactive routing protocols differ in the way in which they update the routing information in the Tables. Routing information is exchanged by nodes by using the broadcast [19]. Through a regular exchange of network topology packets between the nodes of the network, at every single node an absolute picture of the network is maintained. There is hence minimal delay in determining the route to be taken. This is especially important for time-critical traffic. When the routing information becomes worthless quickly, there are many short-lived routes that are being determined and not used before they turn invalid. Therefore, another drawback resulting from the increased mobility is the amount of traffic overhead generated when evaluating these unnecessary routes. This is especially altered when the network size increases. Thus, proactive MANET protocols work best in networks that have low node mobility or where the nodes transmit data frequently [20] [21]. This type

of protocols maintains fresh lists of destinations and their routes by periodically distributing routing Tables throughout the network. The main limitations of such algorithms are:

- I. The amount of respective data used for maintenance.
- II. Slow response on restructuring of topology and failures.

Examples of Proactive MANETs Protocols are:

- I. Destination-Sequenced Distance Vector (DSDV)
- II. Cluster-head Gateway Switch Routing Protocol (CGSR), etc...

1.2.2 On-demand Routing (Reactive Protocols)

In reactive protocols [20] [21] [22] [23], routes are determined when they are required by the source using a route discovery process. These protocols were designed to reduce the overhead encountered in proactive protocols by maintaining information for active routes only. Route discovery usually occurs by flooding route request packets through the network. When the request reaches a node with a route to the destination (or the destination itself) a route reply is sent back to the source node by using link reversal [20]. Reactive protocols can be classified into two categories; Source routing, and Hop-by-Hop routing.

In Source routed on-demand protocols, each data packets carry the complete source to destination address. Therefore, each intermediate node forwards these packets according to the information kept in the header of each packet. This means that the intermediate nodes do not need to maintain up-to-date routing information for each active route in order to forward the packet towards the destination [19]. Furthermore, nodes do not need to maintain neighbor's connectivity through periodic beaconing messages. In hop-by-hop routing (also known as point-to-point routing), each data packet only carries the destination address and the next hop address. Therefore, each intermediate node in the path to the destination uses its routing table to forward each data packet towards the destination. The advantage of this strategy is that routes are adaptable to the dynamically changing environment of MANETs, since each node can update its routing table when they receive fresh topology information and henceforward

the data packets over fresher and better routes. The disadvantage of this strategy is that each intermediate node must store and maintain routing information for each active route and each node may require being aware of their surrounding neighbors through the use of beaconing messages [22] [23]. The main limitations of such algorithms are:

- I. Larger time consumption in route request process during route discovery phase.
- II. Network blockage can happen due to excessive flooding.

The different types of On Demand protocols are:

- I. Ad hoc On Demand Distance Vector (AODV)
- II. Dynamic Source Routing Protocol (DSR) etc.

1.2.3 Hybrid Protocols

Hybrid routing protocols [21] [22] [23] combine the basic property of the two classes of flat routing protocol into one. That is, they are both reactive and proactive in nature. These protocols are designed to increase scalability by allowing nodes with close proximity to work together to form some sort of a backbone to reduce the route discovery overheads. This is mostly achieved by proactively maintaining routes to nearby nodes and determining routes to far away nodes using a route discovery strategy. Most hybrid protocols proposed are zone-based, which means that the network is partitioned or seen as a number of zones by each node [22]. Other nodes group into trees or clusters. The routing is initially established with some proactively prospected routes and then serves the demand from additionally activated nodes through reactive flooding. The choice for one or the other method requires predetermination for typical cases [21] [23]. The main limitations of hybrid algorithms are:

- I. Overhead of control packets is more than overhead in reactive protocols.
- II. Advantage depends on the number of other nodes activated.
- III. Demand in response to traffic depends on the gradient of traffic volume.

The different types of On Demand driven protocols are:

- I. Zone Based Routing Protocol (ZRP)
- II. Zone Based Hybrid Link State Routing Protocol (ZHLS) etc.

1.3 Background

Nature has been a source of inspiration for developing a variety of problems involving meta-heuristics. These meta-heuristic techniques are not bound to solve a single problem, but represent a general framework for solving many types of optimization problems. The optimization problems are often intractable problems and present considerable challenges to researchers and practitioners. Even slight increases in solution quality of such problems can translate to increased company profit, lower consumer prices and improved services [24]. As a result of this, numerous optimization techniques have been studied, developed and refined. The traditional operations research techniques of divide and conquer; cutting planes and dynamic programming has been widely used [25]. But these can be computationally expensive for large and complex problems. As a result, newer meta-heuristic search algorithms such as simulated annealing, tabu search, fuzzy interpretation [27] and genetic algorithms have been applied as they generally find good solutions in a moderate amount of computational time [26].

Swarm Intelligence (SI) is relatively a new field in terms of its application to combinatorial optimization problems and was firstly used to describe the coordination of robots [25]. SI involves systems that are typically made up of a population of simple agents or bodies interacting locally with one another within the present environment. The agents follow very simple rules during the process of communication [28] [29]. There does not exist any centralized fixed infrastructure to control the individual agents about their behavior and interactions between agents. After that, two new interpretations of SI have been used for deriving methods to solve optimization problems. The first is Particle Swarm Optimization (PSO), which is an evolutionary computation technique inspired by the social behavior of bird flocks or fishes. PSO works by maintaining the swarm of particles which roam in the search space prejudiced by the developments discovered by the other particles. Here, population of solutions evolves by moving through the solution space, with good solutions acting as attractors. In 1992, the other concept of field is the used involving ant algorithms [30] [31] [32]. Since that time, it has been applied to both theoretical and as well as practical optimization problems with enormous triumph. The performance shown by ant algorithms and their possibility of

adaptation encourage the study of this new field very useful. The nature of ant algorithms is iterative and they use probabilistic meta-heuristic for finding solutions to combinatorial optimization problems. Their behavior is built on the foraging mechanism shown by real ants while attempting to find a shorter path from their nest to the food source.

1.4 Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) [32] is a meta-heuristic approach for solving optimization problems. The inspiring source of ACO is the foraging behavior of real ants that consents to find shortest paths between food sources and the nest. In fact, while walking from food sources to the nest and vice versa, ants release a chemical substance (the pheromone) on the ground, and the direction chosen by the following ants is the path marked by a stronger pheromone concentration [32]. Figures 1.5 (a) & (b) illustrate how, after some time, the ants on the shorter path reach the food source sooner as compare to ants on the long path. Ants on reaching the destination; start a new route backward towards the source nest by following the same path and biases the path by depositing more pheromone on the shortest path. As time progresses, the pheromone on non-optimal paths evaporate while the pheromone on near-optimal paths is reinforced. The basic principles driving this system can be applied to many combinatorial optimization problems like routing in data networks [33].

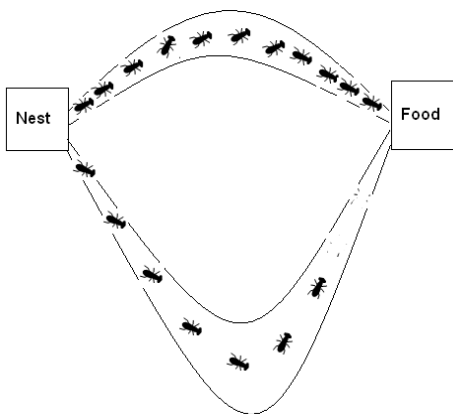


Figure 1.5: (a)

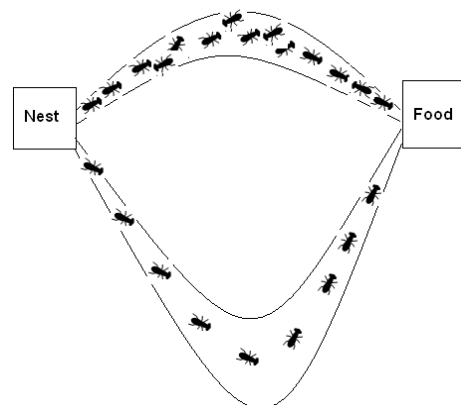


Figure 1.5: (b)

Figure 1.5. (a) Double Bridge Experiment – Ants on the shorter path reach to food source earlier. **(b)** Ants on the shorter path return to nest before the ants on longer path

All the elements which play an essential role in the ant colony foraging behavior can be understood, thoroughly reverse-engineered and can be put together to work for solving problems of combinatorial optimization [33]. The ACO's synthesis was also motivated by the usual good performance shown by the algorithms (e.g., for several important combinatorial problems like the quadratic assignment, vehicle routing and job shop scheduling, ACO implementations have outperformed state-of-the-art algorithms) [34]. So, it can be said that the ACO is modeled on the foraging behavior of ants. The seminal work by various scientists showed that this behavior could be used to solve discrete optimization problems. The ACO algorithm has been successfully used in the routing problem generally by using two sets of homogeneous mobile agents, called FANTs and BANTs [42] [43] [44]. Both types of agents have the same structure, but they can “sense” different inputs and produce different independent outputs [35] [36]. There are many available numerous successful implementations of the ACO meta-heuristic applied to a variety of different combinatorial optimization problems. Figure 1.6 represents the various fields of applications of ACO.



Figure 1.6: Applications of Ant Colony Optimization

Ant colony algorithms are gaining popularity among researchers and scientists since last decade [37] [38] [45]. Many real time applications have been developed based on ACO. The following Figure 1.7 represents some of the real time applications of ant algorithms.

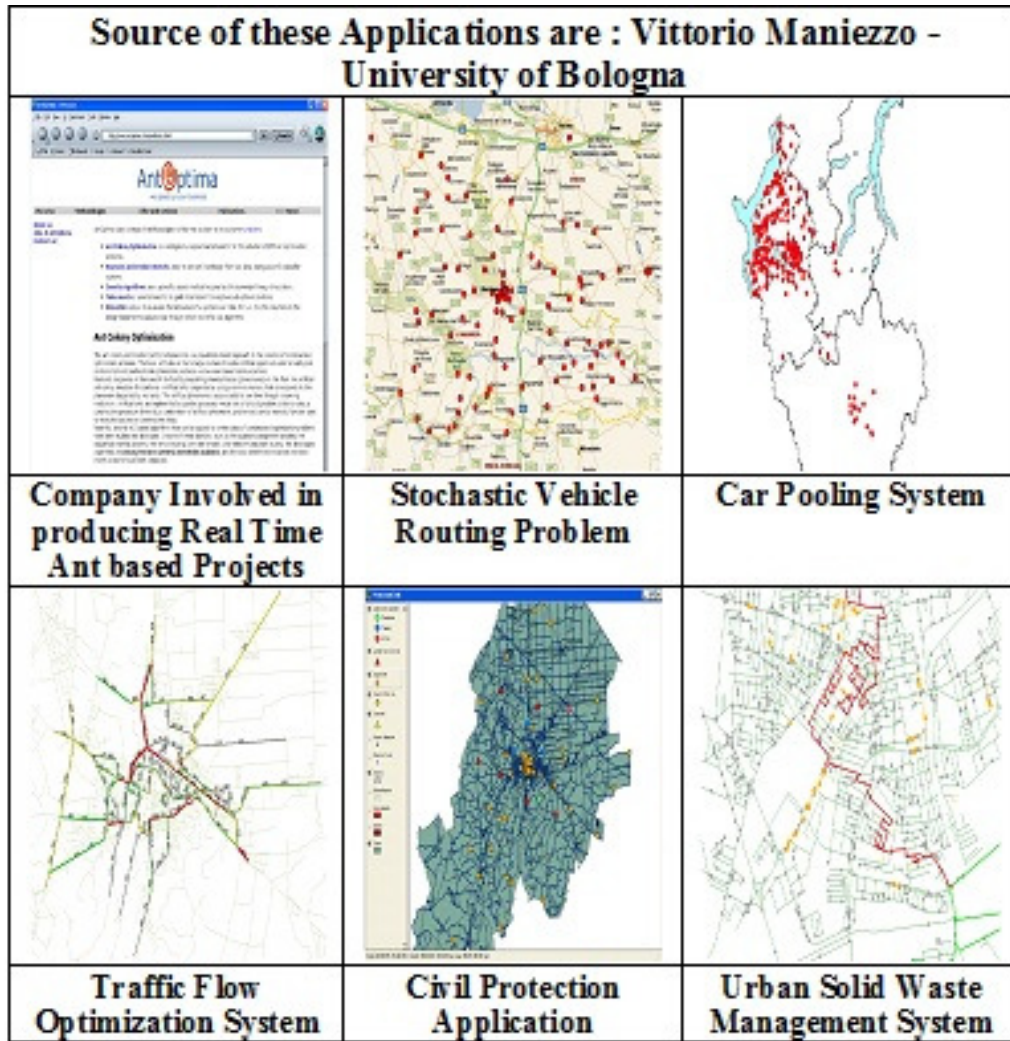


Figure 1.7: Real World ACO Applications

1.5 Main ACO Algorithms

Several special cases of the ACO meta-heuristic have been proposed in the literature [39] [40] [41] [42]. Here, a few of the famous algorithms of the ACO are presented briefly, in the historical order in which they were introduced [32] [37] [38].

I. Ant system (AS),

- II. Ant colony system (ACS),
- III. MAX-MIN ant system (MMAS).

AS was the first ACO algorithm to be proposed in its category. The main characteristic of the AS is that the updation of pheromone values is done by all the ants, which have completed their tour [32] [38].

The first improvement to AS was proposed as an ACS [32] [37], which was introduced by Dorigo and Gambardella. The main difference between ACS and AS is the decision rule used by the ants during the construction process. Ants in ACS use the so-called pseudorandom proportional rule: the probability for an ant to move from the one city to another depends on a random variable uniformly distributed over $[0,1]$.

MMAS [37] [38] is another improvement, proposed by Stutzle and Hoos (2000), over the original AS idea and to ACS also. MMAS differs from AS and ACS in that

- (i) Only the best ant adds pheromone trails, and
- (ii) The minimum and maximum values of the pheromone are explicitly limited

1.6 Fitness of the ACO Approach in MANET Environment

ACO approach is suitable for solving the routing problem in a MANET environment due to the following contributions of ants [46] [47] [48].

- I. ACO meta-heuristic updates the pheromone and does the calculations related to probabilities. These calculations are performed only at nodes without the use of any central computational mechanism, but depend upon only the local information collected by ants. This suits with the characteristics of MANETs because of their spatially distribution and without any central or global information support.
- II. In contrast to other routing approaches, the ant algorithm is based mostly on local information gathered by ants. They need not exchange any routing tables or other information blocks between nodes of the network.

- III. Ant's decisions are stochastic in nature, i.e., their value depends only upon the previous available value. They continuously explore the search space for better solutions. Ant selects the next node based on the probability calculated based upon pheromone value in much the similar way the next node is selected in networks or MANETs using routing Table.
- IV. As the topology of MANETs change continuously alike the amount of pheromone deposited changes and ultimately decision made is changed endlessly.
- V. Dynamic change in topology is responsible for the poor performance of many classical routing algorithms in MANETs. The ant algorithm is based on autonomous agent systems, imitating individual ants. This allows a high adaptation to the current topology of the network.
- VI. ACO meta-heuristic can support multi objective decision support selection system similar to the way the node is selected based on multi values like battery life, position or congestion levels etc.

1.7 Motivation

With the development in the computing environment, the applications and services of the MANETs have also increased. MANETs are vulnerable due to various physical characteristics of environments and devices itself. After being provided evidence from many research proposals, it can be considered that the ants have the capability to find the shortest path between the food source and their nest. This is mainly due to a chemical substance called pheromone that ants lay down while walking from their source toward the final destination. The ants in search of food choose a path with a higher probability, which is a function of the pheromone. So, the large amount of pheromone can attract more number of ants leading to positive feedback on favorable paths and negative on the other paths. Algorithms influenced by ants have shown an effective mechanism that seeks to incorporate a memory in the learning process which increases the stability of the scheme, which in turn helps in increasing the learning efficiency of the network. Proposed routing algorithms in this thesis are based on orientation and distributed learning mechanism, which are influenced by the protocols such as AntNet [61], PERA [94], MACA [99], ADAA [100] and AntHocNet [121] that

provide formative of forwarding of message packets (ant packets) from the source to the destination node in the direction of the target node. In the proposed algorithm, all the links in the network are assumed to be bi-directional in nature, and all the nodes participate in the process of finding the path to the target node.

1.8 Problem Formulation

One of the major issues that affect the performance of MANET was the way routing has been implemented in a network. Routing algorithms used in conventional wired networks cannot be used in MANETs due to their inability to fine tune to the dynamic topology nature of the mobile environment. Connections between hosts and nodes in MANET were established in a temporary manner. Nodes move in an arbitrarily manner with changing speeds, often resulting in connectivity problems. The high mobility and the arbitrary movement of nodes in MANET cause links between different hosts to break frequently.

To realize the real world applications of MANETs, one need efficient routing algorithms which can adjust themselves with the changing topology and can successfully deliver the packets to the final destination while maintaining high performance [5] [6].

Ant algorithms have shown good performance with respect to changing environmental conditions in such problem domains as defined above, and hence can also prove to be good for the MANET. The study of ACO can be applied to solve the MANET routing problem [24] [25].

Ant colony algorithms are a subset of SI and have the ability to solve complex problems by cooperation. The interesting point is that the ants did not need any direct communication for the solution process, but they communicate with a process called as stigmergy [48]. The notion of stigmergy means the indirect communication of individuals by modifying their environment. In recent years, many algorithms which were based on ant colony problems were introduced to solve different problems on MANET. To address different routing algorithms in MANETs, several design strategies

for MANETs have been proposed and each is having its own advantages and limitations [12] [23] [33] [49] [71] [141].

The research work in this thesis is focused on the formulation of an ant colony based algorithm by combining reactive and proactive nature of MANETs routing algorithm, because both existing proactive and reactive routing protocols prove to be inefficient under these unknown circumstances of MANETs.

1.9 Research Goals / Objectives

This thesis presents the novel, dynamic and adaptive routing algorithms for MANETs inspired by the behavior exhibit by the ants. ACO algorithms have been thoroughly studied in the past presented in past for solving routing problem in wireless communication networks for formulating various goals. After formulating the problem, the thesis work, hereby presents main as follows:

- I. To study and review various ant colony optimization techniques available with respect to MANETs.
- II. To design and develop efficient ant based routing algorithms for MANETs.
- III. To verify and validate the proposed ant based algorithms.
- IV. To compare and evaluate these with ant based and traditional MANET routing algorithms.

1.10 Summary of the Contributions

- I. To accomplish the first objective, wide range of available literature has been studied. After cavernous mining from the various research articles, it has been found that there is still breathing space where researchers need to concentrate. Few of such gaps have been mentioned in the above Section.

II. From the available space, some of the points have been picked which laid down the foundation for the design of the new algorithm. The novelty factors of the thesis work include as follows:

- i. Random selection of the source and the destination were made from a set of nodes for transmission of ant agents (as packets). This helps in covering the broader view of topology. A large number of the previous existing ant based MANET algorithms considered only a single destination or random selection of the destination nodes, which might cover only a subset of the overall network until the discovery of the route.
- ii. An orientation factor was used to flood the search packets in the right direction. This helped the data packets and control packets to move forward in the right direction towards the destination.
- iii. Ant based algorithms were proposed with the adaptive nature because of new pheromone modification and evaporation Equations by selecting the best parameters in each iteration. Proposed algorithm used the best pheromone value (ρ_{high}) computed in iterations to globally update the pheromone Equations. This concept was motivated by the Travelling Salesman Problem (TSP), which has been solved with the help of ACS, Min-Max and Rank based solutions [32] [37] [38]. Global updating in general was intended to increase the attractiveness of promising route. This is the reason that the proposed scheme is more effective since it avoided long convergence time by directly concentrating on the search in a neighborhood of the best tour found up to the current iteration of the algorithm.

III. To achieve the third objective, the above designed algorithms have been simulated using NS-2.33. The research work has created a new control packet format for the moving agents, which were responsible for the construction of the necessary data structures in the form of pheromone table and the memory used.

- IV. Fourth objective was achieved by evaluating the proposed algorithm in various scenarios with respect to other state of the art traditional and ant based algorithms with respect to various parameters. The new algorithms were compared with some of the existing ant based and few traditional algorithms. The results obtained show that the proposed algorithms outperform the other existing algorithms with respect to various selected parameters.

1.11 Organization of the Thesis

- I. The first chapter elaborated the details of the MANETs and the historical background of ACO. It included the challenges in MANETs and the classification of routing protocols. The introduction of ACO gave the background and the formulation of the ant based systems. It helped in understanding the basics of stigmergy, i.e., process of indirect communication between ants through the chemical substance called as pheromone.
- II. The second chapter summarized the various ant based algorithms for MANETs. The rigorous study of many existing ant based MANET algorithms has been done. A comparison table was also drawn to compare their features. This chapter also helped us in understanding the basic gaps between existing protocols. At the end, various objectives have been formulated for the work. The research methodology was also presented in this chapter. This chapter discussed about the research paradigm and the introduction of tools used for experimental and simulation work.
- III. In the third chapter, the new packet formats for the agents is introduced and created various data structures like ants' memory, pheromone table, or traffic matrix, etc. The detail of the new proposed algorithm was also mentioned in the third chapter. The complexity analysis of the algorithms was also computed in this chapter.
- IV. Experimental setup is discussed in the fourth chapter using the network simulator NS-2.33. The scenarios used for the simulation and experimental

purpose were given in this chapter. This chapter also evaluated the performance of proposed algorithms with respect to various parameters by presenting the comparative results of the proposed algorithm. The new algorithms were compared with other existing ant based and traditional algorithms based on various performance metrics like throughput, packet loss, delivery ratio, end to end delay, jitter, and path length.

- V. The fifth chapter concludes with the findings of the proposed algorithm. The results obtained from the previous chapter helped in concluding that the new proposed algorithms perform better than all the other algorithms used for comparison. This validated, and verified the proposed algorithms. Lastly, the directions for the future enhancement of the proposed algorithms have also been stated.

Chapter 2

Literature Survey and Background of Ant based Routing Algorithms for MANETs

2.1 Categorization of Various ACO Routing Algorithms

ACO due its nature can thus be applied to the combinatorial problems such as the routing problem in MANETs. For the application of ACO to network routing, the problem can be reduced to finding the shortest path between the nodes in a network as the source and destination nodes much like in conventional distance-vector or link state routing protocols [48].

Artificial ants are relatively small in size and are personified by special routing packets. They can be piggybacked with network data packets. So, control information can be transmitted in the network easily along with the data packets when compared to traditional routing packets which could only carry Tables with them [49]. Another promising feature of ACO based routing algorithms over traditional routing algorithms is their ability to produce a good suboptimal path based on many performance parameters for QoS like link quality, congestion or other cost metrics [50].

The contents of this chapter have been peer reviewed and accepted for publications in

- I.** G. Singh, N. Kumar, A. K. Verma, "Ant colony algorithms in MANETs: A review", *Journal of Network and Computer Applications*, Vol. 35, Issue 6, pp. 1964-72, 2012. DOI: 10.1016/j.jnca.2012.07.018 (**SCI-Thomson Reuters Rated Journal with Impact Factor 1.467**)
- II.** G. Singh, N. Kumar, A. K. Verma, "OANTALG: An Orientation based ANT Colony ALGORITHM for Mobile Ad-hoc Networks", *Wireless Personal Communication*, Volume 77, Issue 3, pages 1859-1884, .DOI: 10.1007/511277-014-1613-6 (**SCI-Thomson Reuters Rated Journal with Impact Factor 0.428**)

As discussed in chapter 1 that MANETs can be classified into three categories like proactive, reactive and hybrid approach, ACO applications can also be classified into similar categories over MANETs.

- I. Table Driven / Proactive Ant based Routing Protocols
- II. On-Demand / Reactive Ant based Routing Protocols
- III. Hybrid Ant based Routing Protocols

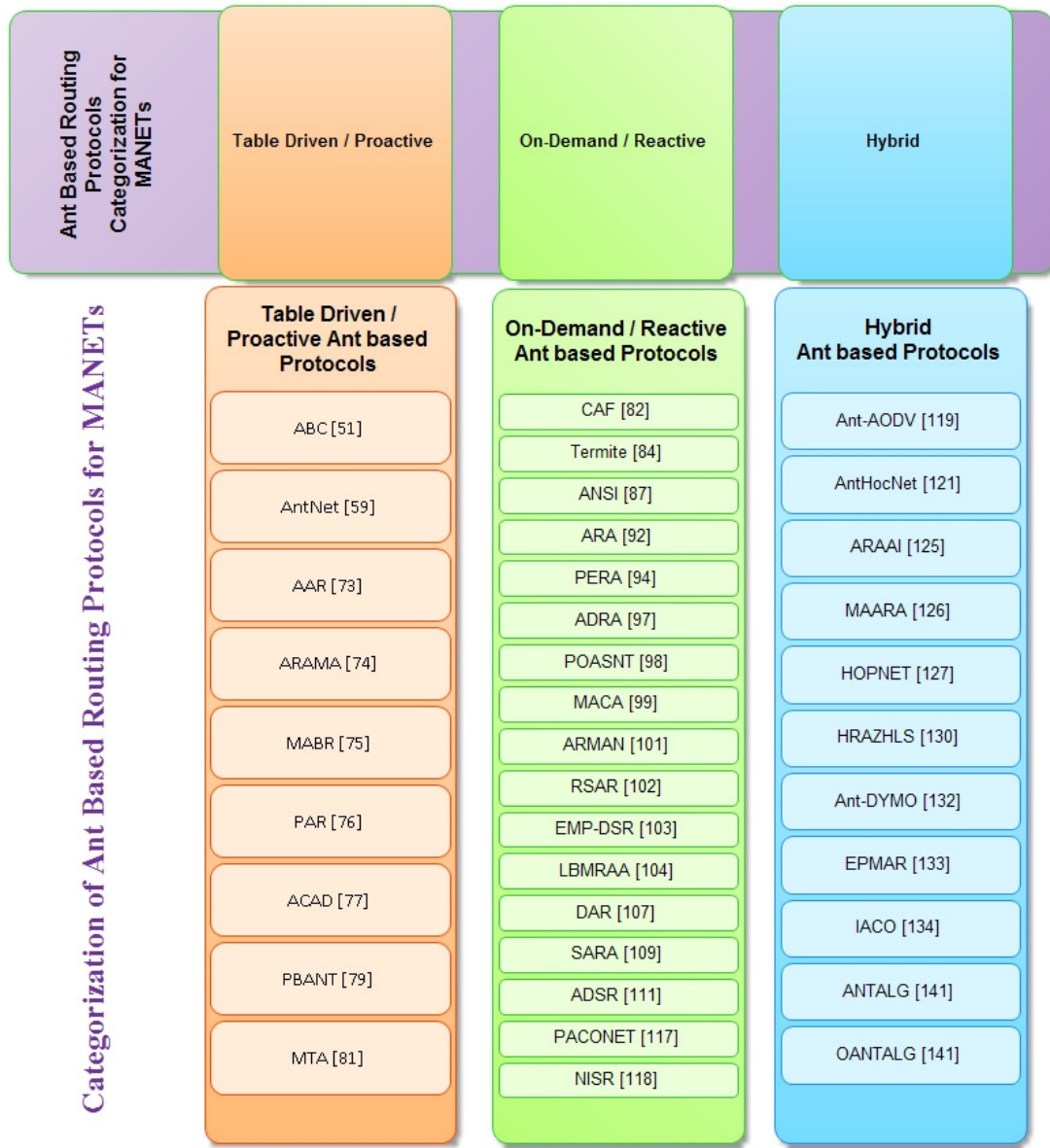


Figure2.1: Categorization of Ant Based Routing Protocols for MANETs

2.2 Table Driven / Proactive Ant Based Routing Protocols

2.2.1 Ant Based Control (ABC)

ABC [51] scheme was proposed for routing in telephone networks. The algorithm is purely proactive in nature and is suitable for MANETs due to their decentralized nature, high robustness to node failures, load balancing and adapt to highly dynamic environments. In the ABC routing scheme, there exist two kinds of routing tasks: exploratory ants, which make probabilistic decisions, and actual calls which made deterministic decisions (that is, choosing the link with the largest pheromone value in the column corresponding to the destination) [51] [52]. Exploratory ants are used for source updates. The source node S issues a number of exploratory ants. Each of these ants goes toward a randomly selected destination D (the ant is deleted when it reaches D). The routing table at each node contains neighbor as rows and all possible destinations as columns, and each entry corresponds to the amount of pheromone on the link towards a particular neighbor for a particular destination. These amounts are normalized in each column (the sum is one), so that they can be used as probabilities for selecting the best link. At each current node C, the entry in the routing table at C corresponding to the source node S is updated. Exploratory ants make the next node choice by generating a random number and using it to select a link based on their probabilities in the routing table [53].

The amount of pheromone left on a trail depends on how well the ant performs. Two new aspects introduced in the ABC algorithm include the aging and delaying of ants [54]. Thus, the authors intend to primarily encourage the ants to find short paths, and to secondly prevent the agents from visiting heavily congested nodes. Aging is used to measure performance. In each hop, the delay depends on the amount of spare capacity of the node, and is added to the age. Both ants and telephone calls travel on the same queue. Calls make a deterministic choice of a link with the highest probability, but do not leave any pheromone. The second objective is achieved by adding an artificial delay to the ants when passing congested nodes. This additional delay is related to the degree of congestion and tries to normalize the situation by the following effects [56]:

- I. Delaying the ant forwarding from a congested node leads to a reduced flow rate of agents to its neighbors, which thereby prevent ants from positively reinforcing the path towards the congested node. This allows other ants arriving from different nodes to find alternative routes around the obstacle and to reinforce them.
- II. The additional delay increases the age of the ants when they arrive at the neighboring node. Therefore, the probability increase value turns out smaller and has less impact on the updated probability value

The ant based algorithm was compared with other algorithms for shortest path like Dijkstra's algorithm used for network management [71]. The ABC algorithm is shown to result in fewer call failures than the other methods, while exhibiting many attractive features of the distributed control [58].

2.2.2 AntNet

The AntNet [59] framework is an ACO-based algorithm used for routing of packets in telecommunication networks. AntNet can be used in Wavelength Routed Optical Networks (WRON) and Optical Packet-Switching (OPS) networks [60] [64] [65].

The original AntNet framework uses the delay introduced by each hop as the metric for routing [59]. This is not applicable in circuit-switching networks, where the main metric is generally the number of hops. The ant used in this algorithm carry in its memory the node identification of each node that it passes by. As stated above, the ants lay down a thin layer of signaling chemical called pheromones while traveling to find food. The other ants detect these pheromones and follow the same path of the chemical mark [61]. The thicker the pheromone trail, the more likely other ants will follow the path. In fact, all of the ants' movements are based on immediate reactions to its surroundings or to its fellow ants. This indirect communication or stigmergy makes ants to represent a highly structured and complex social organization. Emergence of coherent and global patterns from these social interactions has called the attention of researchers from various fields. Scientists from different domain are also exploring the application of ACO based algorithms for routing in computer networks. In order to apply this trail

laying and following behavior to networks, routing, the routing Tables in the network nodes are replaced by tables of probabilities, which are called as pheromone tables, see Table 2.1, as the pheromone strengths are represented by these probabilities [62] [63].

Table 2.1: Pheromone table for node N

Next Node	Destination Node			
	1	2		N
1	P_{11}	P_{12}		P_{1N}
2	P_{21}	P_{22}		P_{2N}
.....				
L	P_{L1}	P_{L2}		P_{LN}

Each row in this table corresponds to a neighbor and each column to a destination. The entries in the table are the probabilities that the next-hop is a specific neighbor. These probabilities are used by ants to allow them to randomly explore the network and possibly find new and better routes. Then, once the routes are discovered, the next-hop probabilities are updated to reflect the new discoveries [64] [70].

The AntNet algorithm can be summarized as follows:

- I. Periodically create new FANTs, from the source (s) to the destination (d).
- II. Select the next link (node j) of a FANT is selected using probability $P'(j, d)$ which is a function of the length of the corresponding output queue and the routing table containing probabilities as shown in Table 2.1. $P(j, d)$ is the probability of selecting node j as the next hop moving towards destination d [65] [66].
- III. When the next node is determined and this node is different from the destination, then FANT checks the buffer to see if this node has been visited before. If this is same as the previous node, it shows that the ant is entering a cycle and, hence, must die. Otherwise ant saves the previously visited node identifier (ID) and time stamp at which the ant was serviced by the current node in a buffer. When the next node selected is the destination, and then the FANT should become a BANT. The

information recorded in the buffer of the FANT should be used to retrace the route followed by the FANT [66] [72].

- IV. The BANT takes the same path as that of its corresponding FANT, but in the reverse direction. At each node k along the path it pops the corresponding stack to know the next node to be hopped. Backward ants do not share the same priority link queues as data packets, but they use higher priority queues, because their task is to quickly propagate to the routing Tables the information collected by the FANTs [66] [67].
- V. When the node is visited by the backward ant the probabilities in the routing table of each node is updated. A reinforcement factor r is used, which is a number between 0 and 1 indicating the quality (length), the congestion, and the underlying network dynamics [66] [68].

2.2.3 Accelerated Ants Routing (AAR)

Matsuo et al. [73] introduced a new ant based algorithm called as AAR. It is a location-based algorithm which makes use of ant agents to disseminate routing information. The ants serve as an efficient form of flooding. It increases the convergence speed in order to obtain good routing path. In AAR, uniform ants are equipped with a stack where the last n visited nodes are stored. This updates the pheromone tables for all the last n intermediate nodes. The AAR has shown its superior performance and faster convergence as compared with AntNet, Q-routing and PQ-routing [71] [72]. It uses ant-like agents which go through the network randomly, without a specific destination, updating pheromone entries pointing to their source.

Experimental results on dynamic network showed that AAR learns the optimum routing in terms of convergence speed and average packet latency. AAR consists of two techniques named as “no return rule” and “N step backward exploration”. The first technique is to eliminate “return rule” during the selection of the next node. This technique leads to the improvement in efficiency of the learning at the initial stage. In reinforcement learning, ineffective rule is difficult to repress. This idea eliminates the detour (two step loop routing) and may not contribute to the acquisition of the reward.

Generally in ant routing, only a single routing entry is updated that implies routing information for only one destination node. In the second technique, if a packet contains own routing history, some routing entry can be updated simultaneously when a single packet has been received [73].

2.2.4 Ant Routing Algorithm for Mobile Ad-hoc Networks (ARAMA)

ARAMA [74] uses the similar types of ants as are used by AntNet. FANTs are used to collect information about the network and BANTs are used to update the routing information in the nodes. ARAMA defines a value called grade, calculated by each BANT and is a function of the path information stored in the FANT [74]. It emphasizes mainly on the fair energy usage across the network by including the local battery/energy information of the node into the ant packets. It also uses negative BANTs and destination trail ants other than the generally used ants. A FANT may be killed on experiencing any unwanted path constraint like looping, reached time-to-live (TTL), etc. Then a BANT with a negative grade can be created and will be sent back to the source of the corresponding FANT. This type of ant is called as a negative BANT. The negative ant reduces the emphasizes of the probability of visiting that path and thus increase the probability of visiting other paths.

The ant routing algorithm class guarantees to find the optimum solution to some extent. But as the network grows, the probability to find the destination and an optimum path becomes smaller, which increases the connection setup time and the initialization/learning time. A new type of ant called as a destination trail ant is used to overcome this problem. The destination node generates destination trail ants with a low rate to insure low overheads. This ant moves randomly in the network to modify the pheromone tables and probability routing Tables favoring the links that lead to its origin node. The effect of this ant to the network pheromone table is called as the destination's trail [141]. So the pheromone values in some scattered nodes are biased toward the destination and toward the best path regions. If few of such ants are sent in the network, the probability of a FANT to intersect with one of the such trails increases. This technique reduces the time needed for connection setup and ensures the delivery of data [74].

2.2.5 Mobile Agent Based Routing (MABR)

Dhillon et al. [75] introduced MABR in which the nodes maintain probabilistic routing tables which are updated periodically by moving agents, also called as ants based on the quality of paths. MABR is also used for large-scale MANETs inspired by social insects and is based on AntNet. The quality of paths may be expressed in terms of metrics such as hop count, end to end delay, packet loss, etc. MABR uses hop-by-hop routing approach [75]. Traffic in the network is continuously monitored by the mobile agents. The paths in MABR are not free from loops and might be trapped in the oscillations. MABR has outperformed in terms of load balancing, i.e., it can distribute traffic along multiple paths, and automatically adapt to a node or link failure in static topologies.

The protocol contains three layers named as Topology Abstraction Protocol, Mobile Ant Based Routing, and Straight Packet Forwarding. The algorithm is compared with the AntNet, Link-State and Distance-Vector algorithms for checking its performance. Each node acts as a part of a specific logical router at every moment of time and, thus, makes use of the corresponding routing Tables. When a node wants to send the data packets to another node, it firstly discovers the destination node's location through any location management scheme and stores destination coordinates in the header fields of the packet.

The source node determines to which zone the destination coordinates belongs and selects any logical link with the probability given in the routing table for that zone. The forwarded packet is tagged with the corresponding logical router address, i.e., with geographical coordinates as the next logical hop for movement. In this way messages are passed between the logical nodes. The advantages of this approach include the ability to respond quickly to local and global changes which are not only included the network topology, but also included in the communication bandwidth, in the transmission delay, etc. However the shortfall with this approach lies in that no random explorations of routes are possible. The overhead of doing proactive routing may far prevail over the benefits of doing so in a network where the topology changes frequently [71] [75].

2.2.6 Probabilistic Ant Routing (PAR)

The PAR [76] algorithm uses two kinds of agents –FANTs and BANTs. The FANT are probabilistic and explore the network to collect the network traffic information. They are routed on normal priority queues, i.e., they use the same queues as normal data packets. When the FANT reach the destination, it is deallocated and the BANT inherits the stack contained in the FANT. The BANT is deterministic and is sent out on a high priority queue. The BANTs retrace the path of the FANT and utilize this information to update the routing Tables and other data structures periodically [76]. These mobile agents are small and light packets containing source IP address, Destination IP address, Packet ID and a dynamically growing stack consisting of Node_ID and the Node_Traversal_Time [72]. Each node maintains a probabilistic routing Table. A node builds a neighbor list by using single hop HELLO message packets. These packets are transmitted at regular intervals.

Each node is assigned a probability of $1/N$ where N is the number of neighbors of a node. This initializes the routing table at every node. The identical probabilities indicate that nothing is known about the state of the network. The FANTs are generated by the source node at regular intervals. The FANT may be unicast or broadcast depending on the availability of a route to the destination at a node. If the route to the destination is available at the node, the ant is unicast [76].

If the route to destination is not available, then the ant is broadcasted. While moving towards the destination, the FANT pushes the Node_ID and the Node_Traversal_Time of every intermediate node; it visited into its stack until it reaches the destination. When the FANT reaches the destination, it passes all its information about the route to the BANT created by the node and dies.

2.2.7 Automatic Clustering Inspired by Ant Dynamics (ACAD)

ACAD [77] is a simple heuristic algorithm that can automatically detect any number of well separated clusters, which may be of any shape e.g., convex and/or non-convex. The algorithm draws inspiration from the dynamics of ants and iteratively partitions the dataset based on its proximity matrix. It is noticeably different from the existing ant

colony based clustering techniques [72]. As ants go along a path, they deposit a chemical substance known as a pheromone to which other ants are attracted. This pheromone decays over time. This is the aspect of ant dynamics that have been used in this technique. A number of “pseudo-ants” are successively initialized in data space to search for clusters. An ant is randomly assigned its initial position in the population. It traverses the dataset based on the proximity matrix of the given population. It deposits “anti-pheromone” as it moves to the next point from its present location [77]. After coming back to its initial position, the movement of the ant is terminated, indicating a cluster. This cluster is saved by the algorithm and the points are subsequently removed from the population.

A new pseudo ant is again initialized and this process continues until there are no more points remaining in the population, indicating that the data has been clustered. The number of clusters and the number of points belonging to each cluster is not known initially. Thus, clusters are evolved dynamically without any prior knowledge about the nature of the dataset. A runtime complexity analysis shows that the complexity of the algorithm is either quadratic or cubic with respect to the size of the dataset.

The proposed technique can encompass both shell data as well as solid clusters. It can detect outliers from the data and is also able to identify the situation when the data do not have any natural clusters at all. It can check whether there is any real clustering tendency in the data under test. Promising results on both real and artificial datasets have shown the effectiveness of the proposed technique [78].

2.2.8 Position Based Ant Colony Routing Algorithm for MANETs (PBANT)

Sujatha et al. [79] introduced a novel ant based algorithm called as PBANT which optimizes the route discovery process by considering the position of the nodes. The position details of the various nodes like source node, its neighbors and the destination can be obtained by positioning instruments such as Global Positioning System (GPS). PBANT is similar to ARA where position details of the nodes are known in advance. It uses the location information of the nodes to expand the routing efficiency and to reduce the overheads involved with the algorithm [79]. As PBANT is reactive, a route is

searched only when there is a need of data communication. It also uses the concept of zoning, so whole network is divided into zones.

In PBANT, each node maintains a table of values containing the pheromone trails assigned to its outgoing links for different destinations. An intermediate node on receiving a packet for a specific destination will check the existence of at least one pheromone trail for that destination. If pheromone trail exists, then a stochastic decision is chosen for the next hop. If no pheromone trail exists, then the pheromone initialization process starts and the process assigns pheromone trails to all the outgoing links. The amount of the pheromone to be deposited on each link depends on the zone of the corresponding neighbor. The value of the pheromone trails for a particular destination will be deleted from the table of a node if that node does not receive any packet pointing to that destination for more than a distinct period of time which is normally defined in the order of seconds. The entries related to that destination will too be deleted after that time in the reverse routing table [79].

2.2.9 Modified Termite algorithms (MTA)

Sharvani et al. [81] proposed the stagnation avoidance by locally repairing the routes taking help of ant colonies and fine tuning of pheromone concentration. Overhead and end to end delay was also reduced. The MTA also developed a technique that addresses the problem of load balancing [80]. MTA is proposed with a prediction mechanism technique which anticipates link breaks, and repairs the up going broken links before they break. This avoids unnecessary warning messages and also reduces control overhead. Segmentation of data packets is provided which needs to be transmitted from the source to the destination, i.e., data transmission is done through an alternative path to the destination which had not been transmitted earlier prior to link failure through the failed route. The MTA enhances the performance of the network in terms of throughput, reduced end to end delay and routing overheads by including QoS, efficient route maintenance, and local repair strategy by prediction of node failures. The algorithm too provides best effort delivery and transmits the data to those paths which can satisfy the required bandwidth [81].

2.3 On Demand / Reactive Ant based Routing Protocols

2.3.1 Cooperative Asymmetric Forward (CAF)

Heusse et al. [82] introduced CAF mechanism that used the FANTs for updating routing information about the path to their source without sending a BANT and without assuming symmetric path costs. It works both for wired as well as wireless networks and finds a path only on demand, so it is a reactive in nature. FANTs are flooded from source to destination. These ants collect information regarding various nodes in network as according to the routing table maintained at each node and also record the launch-time CAF extends the ABC strategy and used average number of packets waiting in the queue and the average packet delay as performance measures [82]. The main advantage is that the overhead created by BANTs is avoided. CAF used a new strategy for step-by-step updating in a cost-asymmetric network. The ant estimate the traveling cost from an intermediate node to all the nodes the ant has visited so far by updating step-by-step the pheromone tables in the direction opposite to the ant movement. If an ant arrives after the arrival of a data packet, then the information carried back by the ant might be obsolete. CAF was evaluated under various static and dynamic conditions by using the average number of packets which are waiting in the queues and the average packet delay as performance measures. It was too found better while calculating the effectiveness of load balancing as compared to more classical approaches. Results of CAF outperformed over Q-routing and earlier version of AntNet under all test situations [83].

2.3.2 Termite

Roth et al. [84] proposed Termite algorithm that adopted the analogy of termites instead of ants. Each network node is an analogy to a termite hill. The more of the termite passes through a node, the more is the collection of pheromone at the node. That makes the node with highest pheromone a preferred next-hop node for other packets. There are five types of packets used in Termite named data, hello, RREQ (route request), RREP (route reply) and seed [84]. Data packets are routed following the pheromones of each outgoing link. Each node will increase the pheromone of the previous node from where

the packet came. There is no flooding involve in Termite. A limited number of route request packets perform a random walk over the entire network in search for the destination. The reply packet is routed by following the pheromone trail left by the route request packet. However, such random walk may sometimes result in the destination not being found or poor route being discovered. Termite depends on a global routing solution as an emergent behavior, the result of the interactions of many packets. Its probabilistic approach is based on deterministic algorithms on graphs and is different from other standard routing practices [88]. Use of the seed packets is the unique feature of this algorithm. Nodes generate seed packets and send them on random walk for the advertisements about the existence of the node and setting of return path to the source node. The packet randomly selects the next node leaving the node on the link which it arrived from [85].

2.3.3 Ad-hoc Networking with Swarm Intelligence (ANSI)

Rajagopalan et al. [87] proposed ANSI which uses a reactive routing approach. The objective of ANSI is to leverage the potential of proactive routing with the flexibility and scalability of a purely reactive routing. Every node maintains and manages the connectivity with neighbors located within a zone radius in a pure proactive approach. The zone radius is dependable on the whole network size and mobility [86]. For a bigger size or a higher mobility network, the zone radius is made to be larger so that the amount of proactive activity can be increased [89]. ANSI evaluates the quality of a route based also on the congestion along the route. ANSI has shown better performance due to the continuously working of the SI mechanisms at each node. This helps in maintaining the routing information and local information more effectively than other traditional reactive MANET routing protocols. The congestion-awareness property of ANSI helps in controlling the high rates of congestion in high traffic scenarios [87].

2.3.4 The Ant-Colony-Based Routing Algorithm (ARA)

In ARA [92] for MANETs, the network agents (ants) are only transmitted on demand and are flooded through the entire network in a similar process as AODV. The routing table entries do not contain probabilities rather than it stores pheromone concentrations, which are transformed into probabilities later on. ARA works in three phases [92]:

- I. **Route Discovery Phase:** In the first one, Route Discovery Phase, new paths are discovered. The creation of new routes requires the use of a FANT (FANT), which establishes the pheromone track to the source node, and a BANT, which establishes the track to the destination node. FANTs are broadcasted by the sender to all its neighbors. Each FANT has a unique sequence number to avoid duplicates. A node receiving a FANT for the first time creates a record (destination address, next hop, pheromone value) in its routing Table. The node interprets the source address of the FANT as destination address, the address of the previous node as next hop, and computes the pheromone value depending on the number of hops the FANT needed to reach the node. Then the node relays the FANT to its neighbors. When the FANT reaches destination, it is processed in a special way. The destination node extracts the information and then destroys the FANT. A BANT is created and sent towards the source node. In that way, the path is established and data packets can be sent [90].
- II. **Route Maintenance:** Once the FANTs and BANTs have established the pheromone tracks for the source and the destination node, subsequent data packets also increase the pheromone value. Data packets are used to maintain the path, so no overhead is introduced. Pheromone values keep on changing. When a node relays a data packet toward destination to a neighbor node, it increases the pheromone value for that entry. The same happens in the opposite direction. The evaporation process is simulated by regular decreasing of the pheromone values [91].
- III. **Route Failure Handling:** ARA recognizes a route failure through a missing acknowledgement on the MAC layer and to deactivate that link by resetting the pheromone concentration to 0. Then, the routing table is checked for different links towards the destination and the packet gets relayed accordingly. If a message has to take a long path towards a target it lasts long until the sender is able to realize that the packet was lost. First, an attempt is made to send the packet over an alternate route; otherwise, it is returned to the previous hop expecting that there exists an alternate route in the network. New routes are never discovered in ARA until a route error message notifies the source to

initiate a new route discovery action. ARA has no updating mechanism to adapt the changes in a dynamic network such as MANET [93].

2.3.5 Probabilistic Emergent Routing Algorithm (PERA)

PERA [94] uses the routing table that stores the probability distribution for the neighboring nodes. This probability associated with a neighbor reflects the relative likelihood of that neighbor forwarding and eventually delivering the packet forwarded for itself to the destination node. The route discovery is done by two mobile agents – FANT and the BANT. When a node does not have a route to the destination, it creates a FANT and sends it periodically on the network as long as the route is required. The FANT uses a structure which stores information comprising the IP address of the source node, IP address of the destination node, a sequence number, a hop count and a dynamically growing stack. The stack contains information about the route traversed and the timestamp when that node was traversed by the FANT [94]. Upon reaching the destination, the destination node creates a BANT, which uses the information in the FANT to traverse back to the source node.

The BANT updates the probability distribution at each node on its way [95]. The drawback of this routing protocol is that at each node, both FANTs and BANTs are broadcast which leads to a huge number of duplicate ants in the network. In this approach, the FANTs are, in practice, identical to route request messages in AODV and DSR. The only difference stays in the fact that multiple routes are configured, but actually only the one with the highest probability is actually used by data, with the other being available for quick recovery from link failures. The performance of the algorithm is comparable to that of AODV [96].

2.3.6 Ant-Based Distributed Route for Ad-hoc Network (ADRA)

ADRA [97] improves the convergence rate of ant-based routing by introducing two different ants: the shortcut reinforce ant named enforce-ant, and the congestion repression ants named anti-ant. When the load on the intermediate node exceeds its predefined congestion threshold, it will send a congestion control anti-ant to its upstream neighbor nodes to modify their probability route Tables [97]. After receiving

enforce-ant or anti-ant, the node updates the routing table to avoid the congested routes and to reinforce the good routes. The ants deposit simulated pheromones as a function of the quality of the link, the distance from the source node, the congestion encountered on their journey, the current pheromones and the velocity while they move across the network between pairs of nodes. The node changes the pheromones by itself according to the link quality and ages the link by pheromone evaporation.

An ant selects the next node according to the distribution of simulated pheromones at each node. To speed up the convergence rate of the congestion problem and the shortcut problem, the algorithm gives the parameters different weight values to update the probability routing Table. The results obtained are compared with the results achieved by using fixed shortest-path routes already proposed for use in ad-hoc networks. Packet loss ratio, control overhead as well as end to end packet delay metrics is used as performance parameters. The ADRA system has shown better performance as fewer call failures than the other methods, while exhibiting many attractive features of distributed control [97].

2.3.7 Position Based Ant Colony Routing Algorithm for Mobile Ad-hoc Networks (POSANT)

Kamali et al. [98] proposed a new ant based routing algorithm named as POSANT. It uses the information about the position of nodes to increase the efficiency of ant routing. In contrast to other position based routing algorithms, POSANT does not fail when the network contains nodes with different transmission ranges. Unlike the single path algorithms, POSANT is a multi-path routing algorithm. Some of the regular position based routing algorithms find a route which is much longer than the shortest path, but POSANT is able to find routes which are close in length to the shortest path [98].

POSANT is a reactive protocol, so the route is only searched when there are data packets are needed to be sent from a source node to a destination node. Sending of data packets will start only after the establishment of a route between source and destination. Before establishment of route, the FANTs and BANTs exchange among various nodes between the end nodes. POSANT while finding a route keep the number of generated

ants as small as possible. Zones are being defined and the information about the position of nodes is used as a heuristic value. The algorithm partitions its neighbors into 3 different zones based on an angle. During the beginning of a route establishment process, some of the ants take non-optimum paths to reach the destination. However, pheromone evaporation also reduces the effect of such non-optimum routes. The adopted stochastic strategy in this algorithm establishes multiple paths between the source and the destination [90].

POSANT finds multi routes to the destination in comparison to other position based routing algorithms which normally find only a single route to the destination. The main advantage of multipath routing is to the reduced chances of congestion in the network. It is always important to start sending data packets rather than control packets at an appropriate time. But starting sending data packets too early may increase the chances of the packets to get lost or they may follow longer routes which increase the congestion in the network.

Nevertheless, on the other hand, sending the data packets late may increase the end to end delay. To estimate the best time to start sending the data packets, POSANT uses the three zones of the sender node for the given destination. The sending node calculates the average and standard deviation of the delays reported by BANTs using the hop count. So each BANT carries with them the length of the path followed from the destination to that current hop [141]. Whenever a BANT is received by the sender, the algorithm update the average and standard deviation reported by packet delays for that corresponding zone using the delay observed by this ant. A Fixed size of the window for each zone is defined that contains recently received BANTs from that zone to reduce the effect of old BANTs. The average and standard deviation of delays are calculated only for the BANTs in that window. The new BANT is put in the window of the corresponding zone, discarding the oldest ant when the window size has been reached. So it is important to select an appropriate window size. If the selection of the window size is too small then the average delay calculated from the window information would be too far from the real average. But if the selected window size is large, existence of old ants would affect the result for a longer time [98] [144].

2.3.8 Modified Ant Colony Algorithm (MACA)

Wang et al. [99] introduced multicast routing for delay variation bound using a Modified Ant Colony Algorithm (MACA). Authors have overcome the problem of slow convergence and speed up the process of finding paths in ant based algorithms for MANETs. The algorithm proposes an ant colony algorithm with orientation factor and applies it to the problem of multicast routing with the constraints of delay variation bound. The orientation factor helps the ants to get relieve from the initial blindness searching paths when the pheromone initialization has not been done to make use of the search results and reducing the misguiding effect of pheromone on irrelevant paths [99]. So this overcome the drawbacks of slow convergence existing in the basic ant colony algorithm, increase the speed of convergence and speed up the finding of feasible solution of the problem. It uses a group to represent the number of target nodes of the multicast group and another group of the number of ants sent out to a given destination node.

Each ant in MACA searches for a path to the destination node [141]. Number of paths from the source node to a destination node is equal to the size of the group of ants and is searched according to the path-search mechanism. These are put in a bi-dimensional array. The size of the first dimension of the array is equal to the size of the group of ants, that of the second dimension is equal to the size of a group of target nodes. Every element in that bi-dimensional array stands for a path and every path must satisfy the delay constraint. The simulation results of execution of MACA show that it is possible to find a feasible solution to the multicast routing problem with delay variation bound. The convergence speed in the MACA is also found to be improved on comparing it with the other conventional ant colony algorithms [100] [141].

2.3.9 Ant Based QoS Routing Protocol for MANET (ARMAN)

Deepalakshmi et al. [101] proposed an ant based QoS routing protocol for MANET (ARMAN). Their main motive was to support multi-media communication. Multiple paths are found from a source node to destination node with varying path preference probability. The multimedia data are sent over the path which has a higher path preference probability that can provide lesser delay, higher bandwidth and shorter path

in terms of number of hops. The paths are reinforced positively making it more desirable for further selection during the time of data transmission. The load over the chosen path may increase, which may result more delay and less available bandwidth. Due to movements the nodes might have been moved which will cause link failures [101]. In such case, the path preference probability will automatically decrease and hence alternate routes can be used which are found during route discovery phase. The alternate routes are also validated even though they are not currently used.

It uses two types of ants. First ant is called as `route_request_ant` which is broadcast on receiving the route request to the destination. At each node, the hop count is incremented and the id of the node is entered into the stack of visited nodes. The new bandwidth will be calculated as the minimum value of the bandwidth in the received `route_request_ant` and the value of bandwidth of all outgoing links. Second ant is `route_reply_ant`, created at destination node after receiving the `route_request_ant` and is unicasted to the source of `route_request_ant`. The intermediate nodes during forwarding `route_reply_ant` do not alter the hop count and starting time. At each intermediate node, the stack is popped to find the next node to which the reply has to be forwarded. This is because the `route_reply_ant` will be forwarded along the path through which the `route_request_ant` was forwarded but in reverse direction. The paths are reinforced positively making them more desirable for further selection during the time of data transmission. The load on the selected paths may increase causing more delay and less available bandwidth because nodes might have moved causing link failures during a session is in progress. In such cases, the path preference probability is decreased and alternate routes can be used which were found during the process of route discovery. Even though the alternate paths are not currently used, but still they are periodically checked for their validity [101].

2.3.10 Ring Search Based Ant Routing for MANETs (RSAR)

Gupta [102] proposed a speedy process of route discovery to reduce the flooding in RSAR and adapted a third party reply model [102]. RSAR is a reactive routing algorithm, which is based on SI and its design, is inspired from the collective behavior of social insects. RSAR has three different phases like route discovery, route maintenance and route failure handling phase.

The packets used in the network are divided into two categories called as data packets and control packets. Control packets consist of a FANT and a BANT that are used to update the routing Tables and to distribute routing information about the traffic load in the network [102]. Whereas data packets represent the information that an end-users exchange with each other. Each FANT carries in its header a stack listing the address of each visited node. Each node that is visited by a FANT appends its own address to the stack maintained by before propagating the ant to other nodes. Nodes are able to distinguish duplicate packets on the basis of stack header and source address of the FANT.

2.3.11 Enhanced Multi Path Dynamic Source Routing (EMP-DSR)

EMP-DSR [103] uses a modified version of AntNet to populate its routing table by using the improved fuzzy cost mechanism. The bandwidth and traffic factors are included in the route discovery process [103]. The system provides user choice based route discovery process, cost based route estimation and global optimized route discovery, thus leading to high reliability in multipath routing. The goodness value obtained by the algorithm plays a significant role towards the changes made in the original AntNet algorithm. The algorithm used end delay as the main factor for calculating goodness of the path. When the FANT starts the trip to the destination node, they pass through the intermediate node where it needs to update its reliability field by multiplying the reliability factor with the link availability of the link where the FANT has just traversed. This process prolongs till the FANT eventually reaches to its destination node. As a result, reliability at the destination will become the overall end-to-end traversed path reliability [103].

2.3.12 Load Balanced Multi-Route AODV Ant Routing Algorithm (LBMRAA)

Elmoniem et al. [104] added two procedures to improve the performance of Ad-hoc On-Demand Distance-Vector (AODV) protocol called as Multi-Route AODV Ant routing (MRAA). A new multi-path AODV protocol is proposed along with ACO. LBMRAA uses six different sub procedures, namely Neighbor Connectivity, Route Establishment Request, Route Establishment Reply, Route Expiry, Connection Loss and Local repair.

Pheromone matrix and routing table are used as the data structure. The pheromone matrix is used to store probabilistic entries, which is the pheromone value of each possible route and is organized similar to vector distance routing table algorithms [104]. The matrix defines the probabilistic routing policy which is to be adopted at each node for each possible destination by selecting them from all the nodes in neighbors. Each cell value of matrix stores a pheromone value for expressing the goodness of edge, under the present network-wide routing policy, of choosing next node for the given destination node [104]. A routing table is organized by populating it with pheromone values as is maintained in distance vector routing algorithms

2.3.13 Distributed Ant Routing (DAR)

DAR [107] creates routes reactively and tries to tackle the problem of low routing signaling load. FANTs collect the information about the identities of the crossed nodes and move towards the destination choosing the next hop on the pheromone basis [105]. The amount of pheromone deposited is kept constant while visiting by BANTs on each cross link. Routing Tables are built stochastically. If the routing entry for the destination is available in the routing Table, then the datagram is forwarded otherwise, the datagram is buffered and FANTs are sent out at a constant rate in order to search the path to the destination. Mostly routing protocol's implementation constitutes the third layer of the OSI, which is the network layer for making decisions. The easiness of the protocol helps in achieving seamless routing in networks comprised of heterogeneous elements. The network can easily be expanded with additional nodes without the need of any complicated update procedures if the routing protocol is simple [106]. That's why DAR was introduced as a powerful mean to enhance communication in meshed regular and irregular satellite constellations. If traditional techniques either fail completely or at least face intractable complexity, then DAR could be an appealing solution in such network scenarios as a heuristic approach.

The routing Tables are stochastically maintained at each node, i.e., the next hop is selected according to weighted probabilities calculated on the basis of the pheromone trails left by ants. On receiving a datagram for a destination the node checks that if the routing entry for the destination is available, then the datagram is forwarded to next node chosen from the routing table [107]. Otherwise, the datagram is buffered at that

node and FANTs are sent out at a constant rate in order to search a path to the destination. Two hop-by-hop routing modes can be implemented by a routing protocol. First is hop-by-bop random routing, i.e., nodes randomly chose a neighbor to deliver datagram packets. Second is hop-by-hop optimal routing, i.e., nodes choose optimal next hop to deliver datagram packets. Results for hop-by-bop random routing were found encouraging in the case of static networks with small topologies. This might not be a suitable for MANETs because of rapid changes in topology. That's why DAR adopts hop-by-hop optimal routing. The FANT is routed at each node by looking into the probabilities for the next hop in the pheromone routing table at the current node. So, the forwarding of the ant is probabilistic in nature and allows exploration of paths available in the network [108]. Datagrams are routed deterministically based on the maximum probability in the pheromone routing table at each intermediate node from the source node to the destination node.

2.3.14 Simple Ant Routing Algorithm (SARA)

Correia et al. [109] proposed a distributed ant routing algorithm for Ad-hoc networks called as SARA, where nodes are frequently joining and leaving the network and minimum signaling overhead is required. SARA found that ant on arriving at a node update the probabilities of that node pheromone table entry corresponding to their source node. The increase in these probabilities is a decreasing function of the age of the ant. The ants get delayed on parts of the system that are heavily used. Some noise can be added to avoid freezing of pheromone trails. SARA presents a low overhead solution, by optimizing the routing process [109]. SARA uses three complementary strategies. First is route discovery that used a new broadcast mechanism, called the Controlled Neighbor Broadcast (CNB) where each node broadcasts a control message as a FANT to its neighbors, but only one of recipients from neighborhood broadcasts this message again. During the route maintenance phase, SARA reduces the overhead by the use of data packets to refresh the paths during the active sessions. The route repair phase is also enhanced by the use of a deep search procedure as a way of restricting the number of nodes employed to recover a route. Yinsheng et al. [110] also proposed a service oriented travel portal based upon previous work of reengineering similar to the learning process of agents.

SARA [109] initiates the discovery of a new path between the source and the destination nodes instead of discovering a new path between end nodes of the broken link. A broader search is executed when an extreme search fails to succeed. SARA uses a controlled neighbor broadcast route discovery process which aimed at reducing the routing overhead. Every node collects routing information, but only one of the receiving nodes forward the routing information further. Only the node is selected which is responsible for forwarding the information [141]. The results of simulation work show that SARA not only reduces the overhead, but also detect early congestion and re-route the traffic to the alternate routes on availability. SARA aims at minimizing complexity in the nodes making it as simple as possible; SARA can be helpful in achieving seamless routing in heterogeneous networks. However, this simplicity comes with a drawback that it does not capture the network dynamics in terms of link characteristics and interference and taking necessary steps accordingly. That is why this algorithm is not suitable for networks with high nodal densities and traffic flows moving to mostly one side of the network (gateways) typically like in wireless mesh networks.

2.3.15 Ant Dynamic Source Routing (ADSR)

ADSR [111] takes into consideration of three QoS parameters, delay, jitter and energy. Dynamic Source Routing (DSR) protocol is an on-demand routing protocol that is based on the idea of source routing [111] [112]. Mobile nodes are required to maintain route caches that contain the source routes, of which the mobile is aware. Entries in the route cache are continually updated as new routes are learnt. The protocol consists of two major phases: route discovery and route maintenance. In route discovery, to send a packet, it checks the route cache to determine whether they're in an entry to the destination or not. If the node does not have such a route, it initiates route discovery by broadcasting a route request packet [113].

A route reply is generated when the route request either reaches the destination itself, or reaches an intermediate node which contains in its route cache an unexpired route to the destination. By the time, the packet reaches either the destination or such an intermediate node; it contains a route record yielding the sequence of hops taken [116]. If the node generating the route reply is the destination, it places the route record contained in the route request into the route reply [115]. If the responding node is an

intermediate node, it will append its cached route to the route record and then generate the route reply. Route maintenance is accomplished through the use of route error packets and acknowledgments. Acknowledgments are used to verify the correct operation of the route links. In Ant DSR (ADSR) the FANT and BANT packets are added in the route request and route reply of DSR respectively. FANT and BANT packets are used in the route discovery process [111] [114].

2.3.16 Improvised Ant Colony Optimization Algorithm for Mobile Ad-hoc Networks (PACONET)

PACONET [117] uses two kinds of mobile agents, FANT and BANT. The FANTs are transmitted in a controlled broadcast manner to determine new routes. The BANTs establish the path based on the information gathered by FANTs. These mobile agents leave a certain amount of pheromone at the time they depart from the node [141].

The data packets are routed stochastically on the path that has the highest pheromone value along the path to the destination. FANTs travel along the path of unvisited nodes and take the path which has the highest level of pheromone concentration if all the nodes in the routing table have been visited. Each node maintains a routing table in which the rows represent the neighboring nodes of the current node and the columns signify all the nodes in the network. The (row, column) position in the table holds two values.

- I. A binary value to indicate if the node has been visited or not
- II. The Pheromone concentration value of the node pair

The FANT uses the value of the pheromone to decide its next hop only when all the values in the column have been visited, thus ensuring that all the possible routes through that node has been traversed. This ensures that the chosen path is the best one among the available options [117].

2.3.17 A Nature Inspired Scalable Routing Protocol for Mobile Ad-hoc Networks (NISR)

TORA is the base protocol for NISR [118] protocol, which is a reactive multipath routing protocol that discovers all paths between source and destination and finally selects the shortest one with fewer hops. NISR protocol is enhanced by inspiration from bee and ant colonies. Ants and Bees help each other to find food sources, update the quality of paths to these food sources continually, and determine pheromone of paths [118] [72].

Bees are responsible to discover new food sources and update information about the quality of food sources during the time whereas ants determine pheromone of paths. Scout bees travel randomly from one path to another to search for flower patch. When scout bees return to the hive, it performs dance to announce three pieces of information about flower patch: direction, distance from hive and the quality of the food source. By using these information colony sends its bees to food sources. When bees arrive at food source, an ant, and a bee sent to the hive, where bee informs new quality of flowers and ant update pheromone of the path. Some scout bees keep on searching and give announcement about new information on the nature, i.e., if new paths exist; these paths are announced by scout bees [118].

2.4 Hybrid Ant based Routing Protocols

2.4.1 Ant Based Ad-hoc on Demand Distance Vector Routing (Ant – AODV)

This algorithm overcomes the inherent shortcomings of the AODV. It is hybrid in nature to some extent. The Ant-AODV [119] combines ACO proactive characteristics with reactive properties of AODV to improve the performance of routing search delay and to adapt to the constantly changing of the network's topology. This algorithm decreases the end to end delay and latency in route discovery and increases the connectivity between the nodes. Ant-AODV supports on-demand route discovery to the destination node if they do not have recent route entries [72] [120]. The use of ants increases the network connectivity (the number of destinations for which a node has un-

expired routes). This in turn reduces the amount of route discoveries even if a node launches a RREQ (for a destination, it does not have a fresh enough route) as the probability of receiving quick replies from the nearby nodes for the Route Request (RREQ) is very high due to increased connectivity.

As the ants constantly find better routes, the nodes send data packets in shorter routes. This decreases the end to end delay in the network [71] [142]. The Ant – AODV technique uses Route Error (RERR) packets to inform link failures in the network similar to that in AODV. Ant-AODV is compared to the AODV and results of simulation indicate that the end to end delay and packet delivery fraction of Ant-AODV is comparable to AODV. Ant-AODV has a slightly higher normalized routing overhead than AODV due to the continuous proactive movement of FANTs [119].

2.4.2 AntHocNet

AntHocNet [121] is an ACO routing algorithm for MANETs. AntHocNet can be considered a hybrid algorithm, since it contains both reactive and proactive elements. It is reactive in the sense that it only gathers routing information about destinations that are involved in communication sessions [121] [72]. It is proactive in the sense that it tries to maintain, improve and extend routes while the communication session is going on. This hybrid architecture improves the efficiency by focusing efforts on ongoing sessions. Another factor to help the efficiency is found in the organization of the proactive route maintenance and improvement process, which combines ant-based path sampling with other forms of information gathering. According to the common practice in ACO algorithms, the paths are set up in the form of pheromone tables indicating their respective quality. After the route setup, data packets are routed stochastically over the different paths following these pheromone tables. While the data session is going on, the paths are monitored, maintained and improved proactively using different agents, called proactive FANTs. The algorithm reacts to link failures with either a local route repair or by warning preceding nodes on the paths [71] [124].

At a node, when a packet is received by the network layer from the higher layers, the node checks to see if the routing information is available for destination d over any of its neighbors. If found, it forwards the packet over that node, if not, it

broadcasts a FANT to find a path to the destination. When a FANT is received by an intermediate node, the node checks if it has routing information for the destination over its neighbors and if found, it unicasts the ant over that neighbor, else, it broadcasts the FANT. Loops are prevented by a sequence id mechanism and endless flooding is restricted by enforcing the maximum number of hops for the ant. Once the ant reaches the destination it becomes a BANT and it follows the same path it came from, back to the source. Any duplicate FANTs reaching the destination are discarded. Each node in the backward path updates its routing table with the information of the BANT. When a node receives a BANT it updates its pheromone information [122].

AntHocNet uses a proactive method to update the quality of the route. Nodes periodically broadcast information about the best pheromone values. The neighboring nodes on receiving the information then make changes to the existing pheromone values of the routing table entries to every destination over the broadcasting node. This diffusion process is slow and could result in new paths being discovered to the destination. However, these paths are not reliable and are thus not used directly in packet forwarding, and are marked as virtual pheromones to be explored later during another route discovery phase. Neighborhood discovery is done through the periodic broadcast of Hello messages [123].

A link error may be detected when a Hello message is not received from a neighbor for a timeout period, or if a packet transmission failed on a link. In the case of packet sending failure, AntHocNet checks for alternative routes. If alternative routes are not found, then a route repair process is initiated. The node also broadcasts a link failure notification to inform its neighbors about the change in routing information [142].

2.4.3 Ant Routing Algorithm for Mobile Ad-hoc Networks Based on Adaptive Improvement (ARAAI)

The proactive routing is not appropriate for dynamic topology. Also reactive / on-demand routing schemes lead to greater communication delay. Both of these need large uncontrolled overheads to solve the routing problem. ARAAI [125] will combine the advantages of proactive routing and reactive routing, providing multi-path, and offering adaptive control. The algorithm uses two tables; one is a routing table that can be

represented as (initial node, last node, and heuristic value). “Initial node” records the leaving initial place of ants, and “last node” records the address of the previous node. The heuristic value is local node energy information collection. The second routing table contains the neighbor information and is represented as the connection between local and other nodes. It is defined as: (neighbor, time, and pheromone). Among it, “neighbor” is the neighbors of the current node. “Time” is a monitor that maintains the connections with its neighbors in the way of sending a hello broadcast every specific second to see if the neighbors are still there.

The “pheromone” represents the local link situation and quality, and it’s a reflection of link reliability. When the source node wants to send the data, firstly it sees the routing table for any route to destination. If no route is found, then route discovery process is initiated. Firstly, some numbers of FANTs are dispatched to search the routes from source to destination, and collect paths’ information and intermediate nodes’ local information as they travel. When the FANTs reach at the destination, all of them are recalled and the BANTs are dispatched by destination. In the process initiated by FANTs, current node will select one of the neighbor nodes as next hop randomly. Then the ants will update the node table with last node and pheromone. The FANT chooses a next hop consider both pheromone value and local heuristic value of the link (i,j) between node i and node j [72] [142].

2.4.4 Multi Agent Ant Based Routing Algorithm (MAARA)

MAARA [126] is a hybrid algorithm (both reactive and proactive) which unites ant based routing and multi agent systems. The algorithm is proactive, since the nodes establish path only when there is a need for transmission of data between a source and destination. It is reactive in the sense that the nodes maintain a routing table till the end of the communication session. The routing algorithm consists of 5 phases such as: Route Discovery, Route Updation, Data Routing, Route Maintenance, and Route Failure [126].

Route Discovery: When the source has to send data to the destination, new routes are discovered with the help of FANTs which are spread over the network towards the destination. The FANTs have a unique sequence number that spread over the network to

find the destination. Nodes distinguish the duplicate packets based on the sequence numbers and source address. The first ant that reached the destination becomes the reactive BANT and is returned back towards source which updates the routing table entries in each node. The BANTs insert the destination address, the next hop and the pheromone value in the routing Table. Duplicate ants are identified through the unique sequence number and removed. When the sender receives the backward and from the destination node, the path is established and data packets can be sent.

Route Updation: The forward may broadcast or unicast based on the probability information on available for destination in the neighbouring nodes. A reactive FANT contains the information like source address, destination address, generation number, trip time, list of visited nodes, number of nodes visited and a flag for a reactive BANT generation at the intermediate node. The ants that have the same source address, destination address and generation numbers are called as same generation ants. Using this concept, only one path is set up initially. More number of paths can be added during the course of the communication session with the help of proactive path maintenance mechanism. Each FANT maintains the list of nodes, it visited. The FANT on becoming the BANT updates the pheromone quantity on the reverse path [72] [126].

Data Routing: Data packets are forwarded based on the pheromone values in the routing Table. If the node has multiple paths to the destination, it selects the next hop at random with the probability. The probabilistic routing strategy leads to data load spreading with consequent automatic load balancing. When a path is worse than others, it will be avoided and its congestion will be relieved. Other paths will get more traffic, leading to higher congestion, which in terms increases average end to end delay.

Route Maintenance: A source node sends proactive FANTs periodically according to the data sending rate to maintain the established paths and to find better paths. A proactive FANT can be unicast probabilistically or be broadcasted. A BANT does the same for the direction from the destination back to the source. If an ant got broadcast at any point, it will leave the current known pheromone trails and explore new paths. If it does not find any pheromone pointing towards the destination then it will be broadcast again. The ant will quickly proliferate and flood the network, like a reactive FANT.

Here the number of broadcasts is limited to two, in order to avoid proliferating [71] [126].

Route Failure Handling: Route failures are caused due to mobility of nodes. Link failures are detected when unicast transmission fail or when unexpected periodic pheromone diffusion messages were not received. When a node discovers that a node is missing from the neighbour it removes it from its neighbour list and all the associated entries from its routing Table. All its neighbours receive the message and update their pheromone table using the new values. If a node detects link failure through the failed transmission of a data packet and it has no further paths available for the destination of this packet it starts a local route repair. The node broadcasts a route repair ant that travels to the involved destination like a reactive FANT. It follows available routing information when it can and it broadcasts otherwise. If the local repair fails, the node broadcasts a new link failure notification message to warn its neighbours [126].

2.4.5 HOPNET

HOPNET [127] is a hybrid ACO routing protocol based on ants hopping from one zone to the next. HOPNET is highly scalable for large networks compared to other hybrid protocols [127]. The HOPNET algorithm consists of the local proactive route discovery within a node's neighborhood and reactive communication between the neighborhoods. The network is divided into zones which are the node's local neighborhood. A routing zone consists of the nodes and all other nodes within the specified radius length measured in hops. A node may be within multiple overlapping zones and zones could vary in size. The nodes can be categorized as interior and boundary (or peripheral) nodes with respect to the central node. Each node has two routing Tables: Intrazone Routing Table (IntraRT) and Interzone Routing Table (InterRT). The IntraRT is proactively maintained so that a node can obtain a path to any node within its zone quickly [128].

This is done by periodically sending out FANTs to the sample path within its zone and determines any topology changes. Once a FANT reaches a destination, a corresponding BANT is sent back along the path discovered. The InterRT stores the path to a node beyond its zone [145]. This source routing table is set up on demand as a

route outside a zone is required. The peripheral nodes of the zone are used to find routes between zones. For a small number of nodes, due to the constant movement of border nodes, new routes have to be determined continuously, resulting in more delay than other hybrid routing protocols [129].

2.4.6 Hybrid Routing Algorithm Based on Ant Colony and ZHLS Routing Protocol for MANET (HRAZHLS)

HRAZHLS [130] uses the Zone based Hierarchical Link State (ZHLS) protocol that uses proactive routing scheme within a zone and reactive routing scheme outside the zone, i.e., between the zones [130]. The network is divided into non overlapping zones. The zone size depends on node mobility, network density, transmission power and propagation characteristics. Each node knows its physical location by geo-location techniques such as GPS. The nodes can be categorized as interior and gateway nodes. The algorithm has two routing Tables, Intrazone Routing Table (IntraRT) and Interzone Routing Table (InterRT). IntraRT is a routing table maintained proactively. A node can determine a path to any node within its zone immediately by using IntraRT Table. InterRT is a routing table for storing routes to a destination outside the zone. The gateway nodes of the zone are used to find routes between zones. The ants in this scheme are same as the ants in HOPNET algorithm that classified into 5 types: internal FANT, external FANT, BANT, notification ant and error ant [131]. The internal FANT is the responsible for maintaining the proactive routing table continuously within its zone. The external FANT performs the reactive routing to nodes beyond its zone. When an external FANT is received at the destination, it is converted to a BANT and sent back along the discovered route. If a new route is reactively discovered, then a notification ant will be sent to source node and to all nodes on the route to update their reactive routing Table. The error ant is utilized to warn some changes in the network topology and to restart a new search by the destination if the source still needs a route [72] [130].

2.4.7 Ant Based Dynamic MANET On-demand Routing (Ant-DYMO)

Martius et al. [132] proposed Dynamic MANET On-demand Routing (Ant-DYMO), which is a hybrid protocol that uses an ant-based approach in its proactive phase while

DYMO is the basis for the reactive one. The authors have inculcated some characteristics from the dynamic MANET on-demand routing protocol (DYMO) and ant based MANET protocols in order to propose the new routing algorithm called as Ant-DYMO. It defines two types of artificial ants: Explorer Ants (EANT) which is responsible for creating routes to its source and search ant (ARREQ). Similar to the BANTs of other algorithms, the EANTs carry the information on the destination node and create (or enforce) pheromone trails along the way. Route exploration, route discovery and route maintenance are the major phases of Ant-DYMO [132].

Ant-DYMO is a hybrid and multi-hop algorithm. Information about the neighborhood is collected by nodes by the limited flooding of hello messages. At the stage of route exploration, each node creates its routing probability table on the basis of receiving responses with the help of an ACO approach in the form of pheromone table, which replaces the traditional routing table of past. At the start of the simulations, each node broadcast EANT to its neighbors. A node receiving an EANT for the first time creates an entry in its nodes' pheromone table. After creating the entry, the EANT is then broadcasted to the neighbors of that node. At the stage of route discovery, when a node source node wants to send packets to a destination node which is not present in its routing Table, then it creates an ARREQ with its address and broadcasts it to its neighbors. The route maintenance phase is responsible for improving the quality of the existing routes [132].

2.4.8 Ant-Based Routing Algorithm using Enhanced Path Maintenance (EPMAR)

Woo et al. [133] introduced an ant-based routing algorithm called as EPMAR in 2011 which enhances route selection method and the process upon link failure of EAR. The procedure of route set-up phase of EPMAR is same as that of the EAR [133]. If the source does not have any route available to the destination then it broadcasts a reactive FANT (RFA) to probe paths. If an intermediate node receives a reactive FANT, it checks whether it has already received any other ant with same generation and sequence number. If it happens then it discards the RFA, which it has received. Otherwise, the intermediate node checks whether it has routing entry of the destination in its routing Table. If it does not have such entry, then it saves the values of source address,

destination address, generation number, trip time, and list of visited nodes in its routing Table. After that it updates trip time, a list of visited nodes, and number of visited nodes in the reactive FANT, and broadcasts the ant.

If the intermediate node already has a routing entry of the destination then it generates a reactive BANT. EPMAR alters data transfer phase and route recovery phase of AntHocNet and augments it with a basic EAR algorithm. EPMAR uses the methods of the EAR in route setup and route maintenance phases. EPMAR focuses on the link failure phase much more efficiently than AntHocNet and EAR. EPMAR outperforms AntHocNet and EAR on the basis of varying transmission rates of constant bit rate (CBR). The simulation results show that EPMAR provide a higher packet delivery ratio and less critical link failure than AntHocNet and EAR [133].

2.4.9 Improved Ant Colony Optimization Based Multicasting in MANET (IACO)

Dhull et al. [134] presented an Improved Ant Colony Optimization (IACO) based multicasting in the MANET and treated traffic-engineering multicast problem as a single-purpose problem with several constraints. This study offers a design on Ant Based Multicast Routing (AMR) algorithm for multicast routing in MANETs. Other than the existing constraints like distance, delay and bandwidth, this algorithm calculates one more additional constraint in the cost metric which is the product of average-delay and the maximum depth of the multicast tree [134]. IACO also attempts to minimize the overall cost metric. By reducing the number of group members which are participating in the construction of the multicast tree, by providing robustness to the mobility of nodes and by performing broadcasts in densely clustered local regions, IACO achieves packet delivery statistics that are comparable to that with a pure multicast protocol. This will ultimate reduce the overheads from the network. Ant agent based adaptive, multicast protocol exploits group members' desire to simplify multicast routing and appeal to broadcast operations in appropriate localized regimes has been proposed. This protocol achieves better Packet Delivery Fraction (PDF) with reduced overhead and routing load. Simulation results surpass all the previous algorithms by developing multicast trees of different sizes [135] [142].

2.5 Comparison of reviewed ant based MANET protocols

A lot of research articles or papers have been studied and presented in Sections 2.2, 2.3 and 2.4 representing proactive, reactive and hybrid approach accordingly. The following Table 2.2 briefly compares all those protocols on the basis of their type, key feature and orientation factor presence.

Table 2.2 Brief tabular comparison between the all ant based MANET Protocols

Year	Algorithm	Authors	Type of Algorithm	Goal / Key Feature	Orientation Factor (Yes / No)
1997	ABC	Schoonderwoerd et al.	Proactive	Telephone Networks	No
1998	AntNet	Caro et al.	Proactive	Wired Networks	No
2001	AAR	Matsuo et al.	Proactive	Faster Convergence	No
2003	ARAMA	Hussein et al.	Proactive	Grade (Function of Path)	No
2007	MABR	Dhillon et al.	Proactive	Hop by hop metric and load balancing	No
2012	PAR	Prasad et al.	Proactive	Improved Maintenance	No
2012	ACAD	Chowdhury et al.	Proactive	Handles cluster of any shape like convex or Non Convex	No
2010	PBANT	Sujatha et al.	Proactive	Uses GPS	No
2012	MTA	Sharvani et al.	Proactive	Modified Termite algorithm	No

Year	Algorithm	Authors	Type of Algorithm	Goal / Key Feature	Orientation Factor (Yes / No)
1998	CAF	Heusse et al.	Reactive	Cost-asymmetric networks	No
2003	TERMITE	Roth et al.	Reactive	Fast Convergence	No
2003	ANSI	Rajagopalan et al.	Reactive	Zone creation	No
2002	ARA	Gunes et al.	Reactive	To Reduce Overhead	No
2003	PERA	Baras et al.	Reactive	Multiple path discovery	No
2004	ADRA	Qan et al.	Reactive	Congestion and convergence	No
2008	POSANT	Kamali et al.	Proactive	Multipath with GPS	No
2009	MACA	Wang et al.	Reactive	Multicast	Yes
2009	ARMAN	Deepalakshmi et al.	Reactive	Qos Routing is considered	No
2012	RSAR	Gupta	Reactive	Ring search model	No
2010	EMP-DSR	Sarala et al.	Reactive	Fuzzy cost mechanism with ants	No
2011	LBMRAA	Elmoniem et al.	Reactive	Multi Route AODV using ant with load balancing	No
2008	DAR	Rosati et al.	Reactive	Minimize the complexity of node	No
2010	SARA	Correia et al.	Reactive	Improvement of ARA	No

Year	Algorithm	Authors	Type of Algorithm	Goal / Key Feature	Orientation Factor (Yes / No)
2011	ADSR	Asokan et al.	Reactive	Improvement over DSR	No
2008	PACONET	Osagie et al.	Reactive	Reduces overhead	No
2011	NISR	Gudakahriz et al	Reactive	Use Bee and Ant Colony Collectively	No
2002	Ant-AODV	Marwaha et al.	Hybrid	Reduce delay of starting Phase	No
2004	AntHocNet	Dorigo et al.	Hybrid	Reduce Overhead	No
2005	ARAAI	Yuan et al.	Hybrid	Consider Stagnation problem	No
2010	MAARA	Sivakumar et al.	Hybrid	Better Node connectivity Technique	No
2008	HOPNET	Wang et al.	Hybrid	Extension of ZRP with ACO	No
2010	HRAZHLS	Rafsanjani et al.	Hybrid	Similar to HOPNET but Uses ZHLS as base protocol	No
2010	Ant-DYMO	Martins et al.	Hybrid	Use ants in DYMO environment	No
2011	EPMAR	Woo	Hybrid	Enhanced path Maintenance	No
2013	IACO	Dhull et al.	Hybrid	Multicast	No
2013	ANTALG	Singh et al.	Hybrid	Global pheromone handling	No

2014	OANTALG	Singh et al.	Hybrid	Orientation factor added with ant	Yes
------	---------	--------------	--------	-----------------------------------	------------

2.6 Research Paradigm / Methodology

Based on networking and its related areas, analytical modeling, direct or real experiment and computer simulation are the methodologies used for modeling and performance measurements of routing protocols. Mathematical computation and analysis of a problem are part of analytical modeling. Although analytical analysis form a better ground for formulating new routing protocols. But its main weaknesses recline in operating and controlling of protocols. Furthermore analytical modeling sometimes is not sufficient to represent the dynamic nature of MANETs because of its configuration and reconfiguration occurring in large networks. However, real experiments have an advantage in obtaining comparatively precise results because the research is carried out in a real environment. In real environment the influences and routing behaviors can be noticed based on the surroundings. Although, simple levels of abstraction in simulation cannot provide a solid base validation of routing protocol behavior as compared to real experiment. The complex nature of the MANETs could not be tested completely in real experiments due to the high cost of implementation and the complex nature of MANETs.

A simulation experiment can be characterized as a process of modeling of a system that consists of a scenario similar to the original problem and then conducting simulation experiments with constituted a model using a computer program. The computer program may be executed with the help of a simulation tool or with the help of some 4th generation high level language. This will ultimately help in solving and finding a solution to the simulated problem. The goal of using a simulator is to precisely model and foresee the behavior of the real world problem in a system.

Computer simulation helps to generalize obtained results, validate analytical models, evaluate the performance of existing protocols and also estimate the performance of the new prototype. However, there may exists a probable problem in simulation work where simulators are not generating accurate or representative results.

A broader knowledge about a good simulator is necessary to overcome such type of problem. The creditability of appropriate simulation not only lies in using a good simulator, but also has an advantage that the simulation process passes through validation and verification of the results. If the guidelines given by the simulator are followed appropriately, then it will reduce simulation pitfalls and errors.

Computer simulation methodology is used to carry out the proposed research in the thesis. Generally, the performance of mostly routing protocols is estimated by using the computer simulation techniques which, behave more like real systems and use fewer assumptions unlike analytical methods. The complexity presented by routing algorithms is another driving force for accepting simulation methodology. Also, simulation offers more flexibility in the development of the model, its validation, verification, and performance evaluation. Network Simulator 2 (NS-2) has got the major share among the researchers and scientists (approximately 80%) to investigate the problem of routing and the issues related to them.

The designed protocol in this research has also been successfully implemented and verified using simulation work in network simulator. Due to constant changes in network technologies, an open platform network simulator is always required as the preferred choice of researchers.

2.7 Introduction to Network Simulator (NS-2)

There are many tools available for simulations of network topologies. But the proposed algorithm is simulated using the Network Simulator 2 (NS-2) tool.

NS-2 is an open-source discrete event simulator designed mainly for networking research. NS-2 is an event driven packet level network simulator developed as part of the VINT project (Virtual Internet Testbed). This was a collaboration of many institutes, including UC Berkeley, AT&T, XEROX PARC and ETH. Version 1 of NS was developed in 1995 and with version 2 released in 1996. Version 2 included a scripting language called Object Oriented Tcl (OTcl). It is an open source software package available for both Windows 32 and Linux platforms.

NS-2 has been widely accepted as a reliable simulation tool for computer communication networks both in academia and industry. NS-2 is an object-oriented, discrete event driven network simulator developed at UC Berkely written in C++ and OTCL (a TCL script language with Object-oriented extensions). It implements network protocols such as TCP and UPD, traffic source behavior such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms such as Dijkstra, and more. It also implements multicasting and some of the MAC layer protocols for LAN simulations [147].

NS-2 includes a tool for viewing the simulation results, called NAM. NAM is a Tcl/TK based animation tool for viewing network simulation traces and real world packet trace data. The first step to use NAM is to produce the trace file. The trace file should contain topology information, e.g., nodes, links, as well as packet traces. Usually, the trace file is generated by NS-2. During an NS simulation, users can produce topology configurations, layout the information, and packet traces using tracing events in NS.

When the trace file is generated, it is ready to be animated by NAM. Upon startup, NAM will read the trace file, create topology, pop up a window, do layout if necessary, and then pause at the time of the first packet in the trace file. Through its user interface, NAM provides control over many aspects of animation [147].

NS-2 has many and expanding uses, including:

- I. To evaluate the performance of existing network protocols.
- II. To evaluate new network protocols before use.
- III. To run large scale experiments not possible in real time scenarios.
- IV. To simulate a variety of IP networks, etc.

The following Figure 2.2 represents the general screenshot of network animator under network simulator NS-2

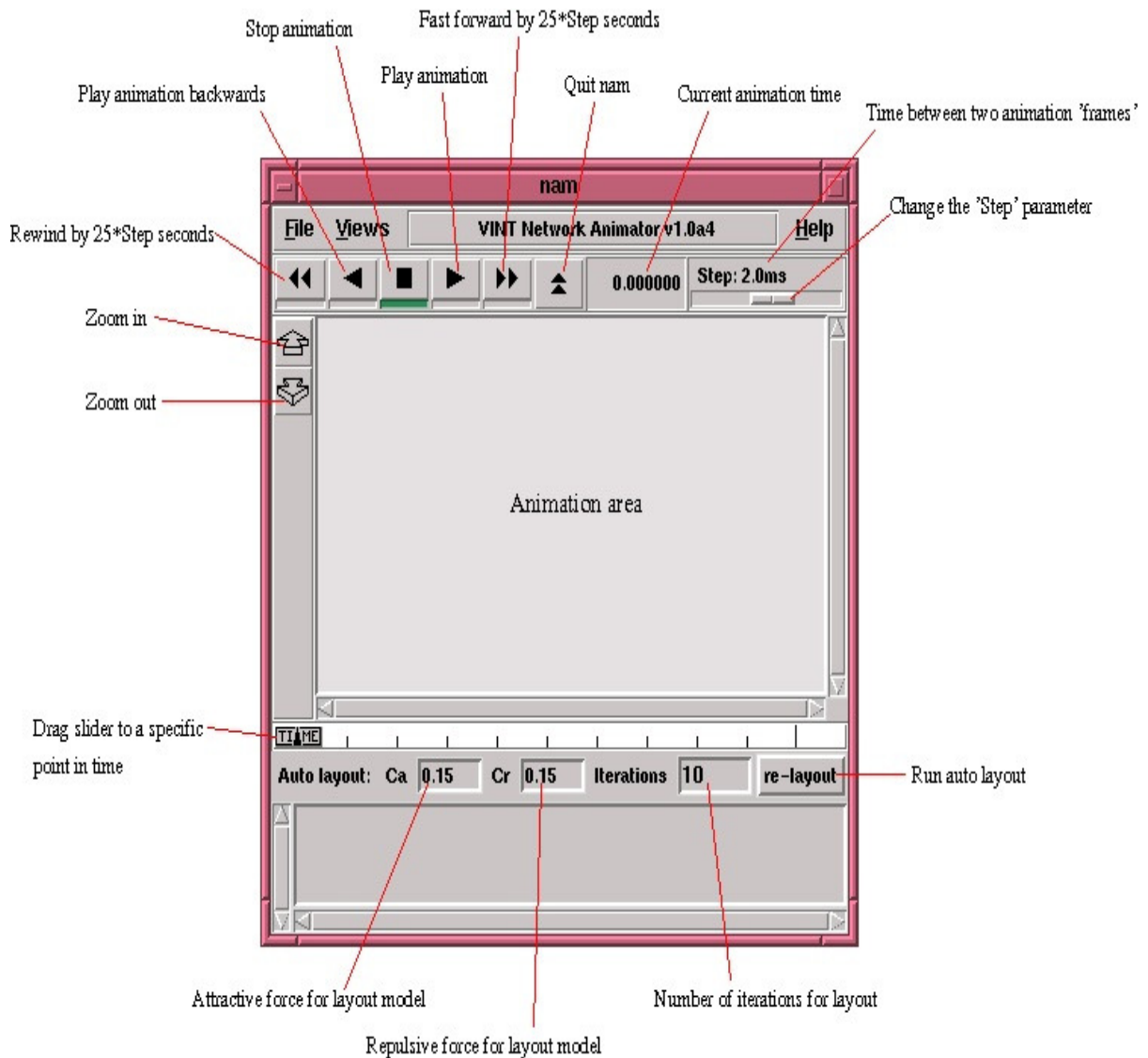


Figure 2.2: General Screenshot of Network Simulator-2

The topology is specified by alternating node objects in NAM. But a layout mechanism is needed in a comprehensive way for displaying the topology. Presently, NAM provides three methods for layouts.

First, user need to specify layout by mentioning the link's orientation. A link orientation is the angle between the edge and a horizontal line, in the interval of 0 degree to 2π degree. During layout, NAM will honor the given link orientation. One can firstly choose a reference node, and then place other nodes using link orientations and link length; small topologies can be generated manually.

Second, one may need to automatically adjust layout during dealing with randomly generated topologies. An automatic graph layout algorithm has been adapted and implemented. The basic idea of the algorithm is to model the graph with balls as nodes connected by links. There are three parameters to tune the layout in an automatic process. First is 'Ca' used for attractive force constant, which controls link's force between balls. Second is 'Cr' used for the repulsive force constant, which controls the repulsive force between balls. Third is number of iterations that control the times to run the automatic layout procedure should run.

Third, there is a $\langle x, y \rangle$ coordinate style layout that was developed for use in displaying a wireless topology where permanent links do not exist. Using this style, nodes events are given x and y coordinate values indicating where those nodes should be placed in a world represented by Cartesian system.

The next chapter represents the detailed design of the proposed algorithm. Section 3.1 briefs the problem background. Section 3.2 shows the various data structures used for the implementation. Sections 3.3 and 3.4 present the proposed algorithms and its various phases respectively. Complexity analysis of the new algorithm is given in Section 3.5 with an illustrative example in Section 3.6. Section 3.8 modifies the algorithm presented in Section 3.3 by adding the orientation factor and used name OANTALG for the new algorithm. Example of orientation factor is given in Section 3.10.

Chapter 3

Design of Innovative ACO Approach in Routing

3.1 Problem Background

With the continuing growth and dynamism of large-scale networks, the need for alternate routing algorithms is becoming increasingly important due to many factors. The appearance of low-bandwidth networks requires routing algorithms that can distribute data traffic across different paths and quickly adapt to changing conditions in the environment and the neighborhood [140] [142].

The Internet as a whole and other autonomous network system in particular are also often fed with heavy traffic load and severe traffic fluctuations. Some standardized routing protocols are widely used in today's networks. Currently, routing protocols are the result of several decades of protocol evolution with a goal to promise some sort of the robustness and self-regulation of networks [139] [141]. Commonly used in packet switched networks, the routing protocols construct tables at each node that specify for each destination the next hop to use for data packet forwarding. A minimal requirement is that the computed routing tables should be free from loops when the network is stable. In dynamic environments, a more rigorous requirement is that the routing tables be loop-free not only when the network is stable but at every instant. The loops, even if temporary, can rapidly degrade performance. The flexibility offered by multi-agent mobile nodes is also as matter of concern in loosely coupled decentralized distributed environment [136] [137].

The contents of this chapter have been peer reviewed and accepted for publications in

- G. Singh, N. Kumar, A. K. Verma, "OANTALG: An Orientation based ANT Colony ALGORITHM for Mobile Ad-hoc Networks", *Wireless Personal Communication. Volume 77, Issue 3, pages 1859-1884, DOI: 10.1007/S11277-014-1613-6 (SCI-Thomson Reuters Rated Journal with Impact Factor 0.428)*

The competence of a routing protocol directly impacts many parameters like throughput, jitter, and end to end delay, etc. The routing protocols which are presently used in networks are more concerned with deriving shortest-cost routes between a source and destination.

The characteristics of the routing problem make it well suited to be solved by moving multi-agent approach (ants) [138] [143]. This processing paradigm is a good match for the distributed nature of the problem which presents a high level of redundancy, fault-tolerance, and can handle multiple objectives and constraints in a flexible way [145] [146]. Ants are moving agents which travel from one node to an adjacent node in the search for viable paths between source and destination nodes. The components used in ACO match well with the network nodes, and, accordingly, routing tables in routing correspond to pheromone tables constructed by ants. Each tabular value in the pheromone table holds the estimated goodness of selecting a neighbor to forward a data packet to destination. The ultimate relationship between ants' characteristic and network routing is the main reasons behind the popularity of the application of ACO to routing problems. Literature in the form of research articles stated in chapter 2 shows the different implementations of ACO in network routing and also gave the indications of good performance and the strong similarities between two. By using these similarities, the research work presented in this thesis report has divided into three stages.

- I. In the first stage, a routing algorithm with the name ANTALG was presented, which was implemented over a static network and compared to the well-known routing algorithm AntNet.
- II. During the second stage of thesis work all the employed data structures and implemented code was shifted into a MANET environment with suitable changes in node characteristics present in NS-2.
- III. The Third stage implements a new protocol OANTALG over the existing structure of ANTALG which have the feature of orientation that guides the moving ants to move precisely in the direction towards the destination.

The data structure used in the three stages was same. During the shift over from stage one to stage two; the new node structure was adopted for a MANET environment which was in-build in the network simulator. While transferring from stage two to stage

three, an orientation factor was included in the probability calculation Equations which increases pheromone concentration in favorable paths and decreases pheromone concentration on less favorable paths. Because, the data structure used in the three stages are same, so, here in the next section, the details of the data structure have been presented.

3.2 Data Structures Used

Two sets of homogeneous mobile agents that act as artificial ants called, FANT and BANT are used. But they are differently situated in the environment in different fashions; that is, they can sense different inputs and they can produce different, independent outputs. They can be broadly classified as deliberative agents, because they behave reactively retrieving a pre-compiled set of behaviors, and they also maintain a complete internal state description. The proposed algorithm uses set of artificial ants and replaces the typical routing table in traditional algorithms with the table of probabilities called the pheromone table. The routing table is organized similar to the way it is maintained in vector-distance algorithms, but the entries in the this table are probabilistic values. The structure containing statistics about the local traffic and plays the role of a local adaptive model for the traffic toward each possible destination. The structure of the pheromone table is as shown in following Table 3.1:

Table 3.1: Pheromone table

Target Nodes	Preferred Neighbor Nodes			
	nn_1	nn_2	...	nn_m
tn_1	$\Omega(1,1)$	$\Omega(1,2)$	...	$\Omega(1,m)$
tn_2	$\Omega(2,1)$	$\Omega(2,2)$	...	$\Omega(2,m)$
....				
tn_n	$\Omega(n,1)$	$\Omega(n,2)$	...	$\Omega(n,m)$

The pheromone table is formulated using the probabilities extracted from the obtained pheromone value. Each entry in the table $\Omega(o,p), 1 \leq o \leq n, 1 \leq p \leq m$ represents the probability of selecting the p^{th} neighbor node $nn_p, 1 \leq p \leq m$ as the next node (among the preferred neighbor nodes) while moving towards the o^{th} destination called as target nodes (tn_o). Each ant creates a memory model to store the trip time of the nodes visited by it. The memory is used to represent the Route Information (RI) collected by the ant. The structure of the memory model is shown in Table 3.2:

Table 3.2: Ant Memory Structure

<i>NO_Add</i>	<i>TR_Time</i>
---------------	----------------

In memory model *NO_Add* represents the address of the node and *TR_Time* shows the trip time of the node visited by the ant. The proposed algorithm uses the structure shown in Table 3.3 for the packet to be used by FANT and BANT.

Table 3.3: Ant Packet Format

<i>PKT_Type</i>	SRC_Add	TAR_Add
PKT_Len	SQN_NO	ST_Time
RI_Max_No	MEM_RI	

Here *PKT_Type* represents the type of packet, i.e., packet to be used either by the FANT or by the BANT. SRC_Add and TAR_Add indicates the source and destination addresses respectively. PKT_Len denotes the length of the packet. SQN_NO denotes the sequence number of packet. ST_Time specifies the start time of the packet. RI_Max_No represents the maximum number of nodes presented in the route information and MEM_RI shows the size of memory required by the route information. The proposed algorithm also uses a traffic matrix to keep track of the traffic information as shown in Table 3.4, which helps in measuring the congestion information.

Table 3.4: Traffic Matrix

M_TT	V_TT	B_TT
------	------	------

Here M_TT represents the mean of the ant's trip time, V_TT indicates variance of traffic trip time and B_TT denotes the best trip time observed by the ant. On the basis of the traffic conditions, mean and variance of trip time may fluctuate vigorously. This matrix plays a vital job in updating the pheromone table used for routing the data.

3.3 ANTALG: Ant Routing Algorithm

Step 1: In the proposed ANTALG routing algorithm, two ants are used as: FANT which is created at the source S after a fix amount of time and moves towards the target, i.e., destination node D. The other ant is BANT, which is created at the destination. BANT follows the path in reverse order as traversed by FANT and updates the route table. The FANT which is headed towards the destination node D, presently available at the node i , follows the path through node j by probability (Ω_{ijd}) calculated as follows:

$$\Omega_{ijd} = \frac{\rho_{ijd} + \theta \Gamma_{ij}}{\theta |N_i| + (1 - \theta)} \dots \dots \dots (1)$$

Where (Ω_{ijd}) is computed as the normalized sum of the pheromone ρ_{ijd} with a heuristic value Γ_{ij} taking into account the length of the j -th link queue of the current node i . θ is the weight of the importance of the heuristic function with respect to the pheromone deposit. The heuristic value Γ_{ij} is a controlled value function of the length l_{ij} of the queue on the link connecting the node i with its neighbor j and number of neighbors $(|N_i|)$ of node i which is represented by:

$$\Gamma_{ij} = 1 - \left(\ell_{ij} / \sum_{i=1}^{|N_i|} \ell_{ij} \right) \dots \dots \dots (2)$$

Step 2: If the selected node is not among the previously chosen nodes, then the new selected node is added to the ant's memory as described in Table 3.2 above. If the selected node is among the already chosen nodes, then a cycle is detected and the nodes involved in the cycle are deleted from ant's memory.

Step 3: When an ant reaches the destination node, then it can estimate the correctness of the selected path. The correctness of the path can be defined according to an application requirement. This algorithm uses the ant's trip time and the parameters of the local statistical model to calculate the goodness of the path. Quantifying the goodness value, the destination node creates BANT.

Step 4: BANT takes the exact path by the corresponding ant, but in the opposite direction, and deposits the certain amount of pheromone $\rho(i)$ on reverse path to the source node. This amount is proportionate according to the procedure which is responsible for the integrity of the path. The pheromone modification Equations are as follows:

If (node was in the path of the ant) then

$$\rho(i) = \rho(i) + r * (1 - \rho(i)) + r^2 * \rho_{high} / C \dots\dots\dots(3)$$

else

$$\rho(i) = \rho(i) - r * \rho(i) - r^2 * \rho_{high} / (C * (|N_i| - 1)) \dots\dots\dots(4)$$

Where ρ_{high} is the value of highest pheromone value obtained so far in the iteration. $|N_i|$ is the number of nodes in the neighborhood and C is the constant which is dependent on to the simulation time in script. The experimental work has taken the value of C as 1000. Because all the calculations in experiments are carried out in milliseconds. r represents the reinforcement factor and is dimensionless constant whose value range from (0,1]. The value of r is computed as follows:

$$r = c1 \frac{t_{best_d}}{t_{i \rightarrow d}} + c2 \frac{t_{sup} - t_{best_d}}{(t_{sup} - t_{best_d}) + (t_{i \rightarrow d} - t_{best_d})} \dots\dots\dots(5)$$

where $t_{i \rightarrow d}$ is the trip time observed by newly created FANT from node i to the destination node d and t_{best_d} is the best time observed by the FANT while moving

towards destination d . c^1 and c^2 are real values in the interval $(0, 1)$ so that $c^1 + c^2 = 1$. t_{sup} represents the time observed by ant in the observation window.

ANTALG uses a variation of ant algorithms by using the highest pheromone value ρ_{high} in Equations (3) and (4) which will help in converging better routes quickly. This concept is motivated by ACS, Min-Max and rank based ant system in traveling salesman problem. Here pheromone deposits are considered only for the best path found. This concept will facilitate ANTALG to congregate better routes quickly. To the best of our knowledge, none of the ant based MANET algorithm has used the highest pheromone value ρ_{high} in pheromone modification Equations. So the proposed algorithm helps in finding better routes early.

3.4 Phases in ANTALG

ANTALG consists of three phases, which are route setup phase, route maintenance phase, and route recovery phase. The detailed working of the above defined algorithm is described as follows:

3.4.1 Route Setup Phase

When any source node wants to send data to a destination; the ANTALG creates FANT firstly at the source node and send the FANT towards the destination. After that source nodes are randomly chosen from the set of nodes and send towards the randomly chosen destinations. This will help the active nodes inside the network to take part in knowing the overall topology in the form of pheromone table just described ahead.

In the previous available ant based techniques only the destination node is randomly chosen which sometimes covers only a subset of the overall network. Because sometimes the path gets discovered early and selecting the random destination may not cover the overall topology till that time. So ANTALG helps in discovering the broader view of topology as compared to previously available ant based MANET techniques. Also in the few of the available ant based techniques the source node is selected on by one from the list of network nodes and then FANTs are broadcasted towards randomly

chosen destination to cover the broader view of topology. But this wastes a lot of time. However, in ANTALG the route setup process works till the discovery of the required path. After that no new source or destination nodes are chosen randomly. Only the FANTs and BANTs which are on the way of search, complete their task.

So, the designed algorithm helps in covering the broader view of topology in lesser time as compared to other ant based MANET algorithms. The process of route setup phase is illustrated as follows. FANTs are forwarded from the source node towards the destination node.

- I. When any midway node i receives the FANT then it checks the destination address of FANT, if node i is not the destination node: (a) node i adds own address and time FANT arrived to i , (b) node i adds the source address as the destination address for routing BANT, into routing table (c) node i broadcasts FANT to its neighbors again, if did not have a valid route to the destination.
- II. When node i receives duplicate FANT, i.e., with the same sequence number and source address or either the sequence number is less than or equal to max sequence number and route record of FANT includes the address of present node, then that FANT is dropped. Otherwise the node updates the max sequence number by new value and executes the steps shown in I.
- III. When the destination of FANT is same as that of i , route is discovered. FANT is discarded after retrieving relevant information from FANT, and BANT is sent on to the path followed by FANT. The pheromone table is updated as per BANT Equations to be defined ahead. When BANT reaches to the source S all the pheromone tables get updated.
- IV. When node i receives FANT and node i already have a route to the destination in its cache, then it adds its address to the memory of the FANT and creates BANT and send it in the reverse direction towards the source after adding the path information of the destination.

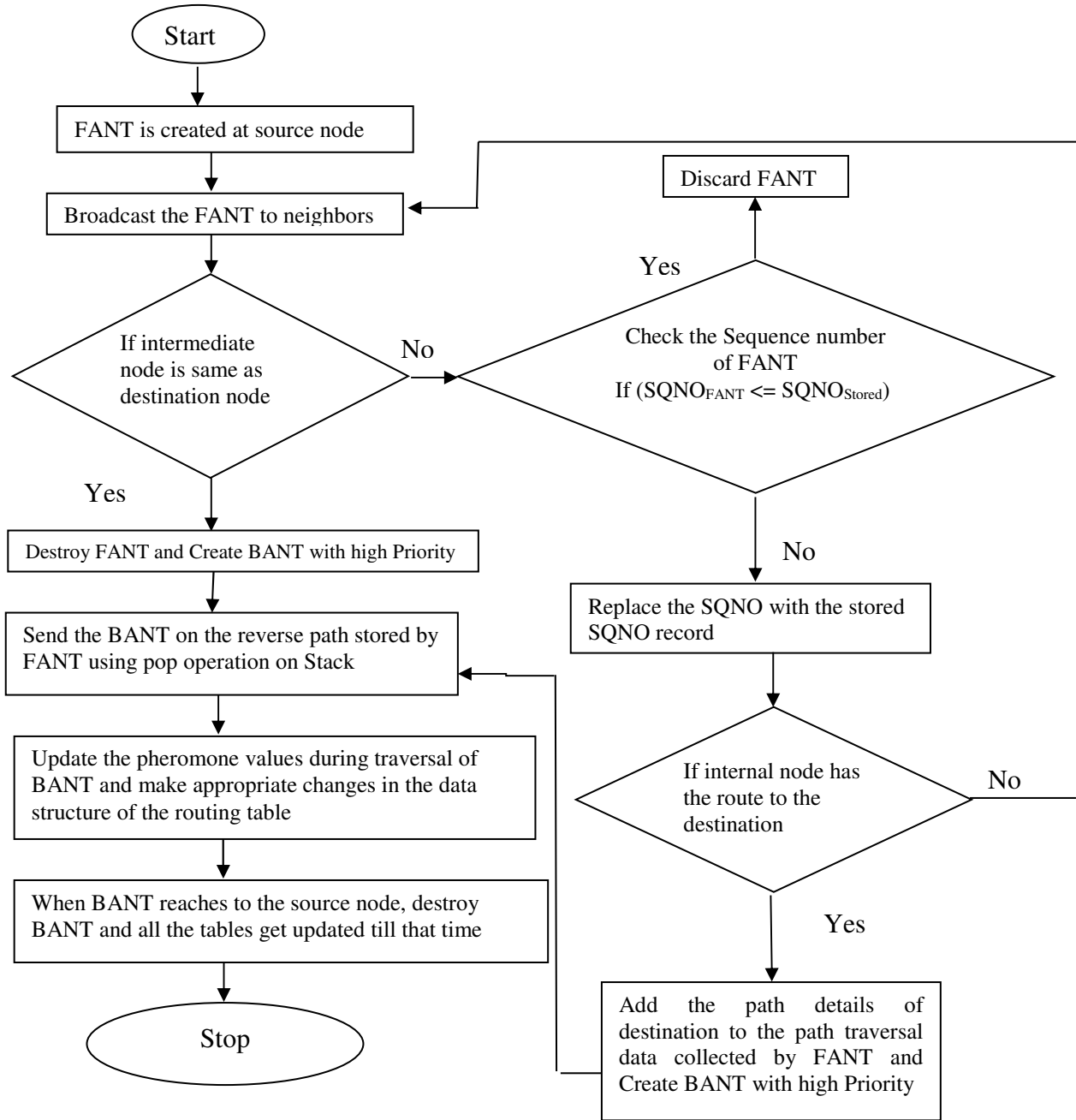


Figure 3.1: Route Setup Phase

3.4.2 Route Maintenance Phase

In order to maintain the established paths and to find better or alternative paths, a source node periodically dispatches FANTs in a proactive way at the rate lesser than the data sending rate. But most of the ant based algorithms send it according to the equivalent rate of sending data packets. A proactive FANT can be either being unicasted probabilistically or broadcasted. If there is no error in the network, then proactive

FANT chooses the next hop probabilistically to probe an established path. It also collects up-to-date information about the established path and correspondingly updates the pheromone values on the path. Like in AntHocNet a proactive FANT is also broadcasted with a small probability at the intermediate node to explore a new or alternative path.

If a node receives a proactive FANT, but it does not have any routing information to the destination, it broadcasts the proactive FANT. However the total number of broadcast allowed through the path toward the destination is limited to control the overhead. When there is congestion at any intermediate node the node i will retrace the path of FANT to inform to source S to change the route. Pheromone value is changed from the source and same value is updated in the routing Table, so that congested path is not further loaded. This information can be obtained from the heuristic function value Γ as calculated in Equation (2) which will decrease its value as the length of the queue increases.

When all the alternative routes fail, source node will then initiate a new route request again. When there is an error in the node, mainly due to change in location of mobile nodes, the pheromone value is made to 0, so that route is not followed by the broadcasting node.

Once the FANT and BANT have established the pheromone tracks for the communication between source and destination nodes, subsequent data packets are also used to maintain the path. As the data packet is transmitted over from one node to another node, it increases the pheromone value for that route entry using the increment Equation (3) as already explained above. All other pheromone values in the routing table decrease over time as per Equation (4).

3.4.3 Route Recovery Phase

A route can be invalidated due to movement of nodes or a link being broken. If a node does not receive the hello packet as a part of beaconing from its neighbor for a certain amount of time, the link is considered broken. In such a case, the node changes the

associated entries of pheromone in its routing table to zero, and broadcasts messages of link failure notification. All the neighbors on receiving the notification message update their routing table. The broken link will conduct a local repair process and tries to find an alternative path to the destination.

In the meantime, it buffers all the packets it receives. If the node finds a new path to the destination, it will send all the buffered packets to the destination via the newly found route. At the same time, a notification ant is sent to the source to inform the source node about the changes in the route. All nodes on the path that receives the notification ant update their routing table to make appropriate changes in routes. The source will replace the related path with the path value in the notification ant.

If the local repair could not find an alternative path, then an error message will be sent to the source node. After receiving an error message, the source node will begin a new path finding process by sending new FANT in search of the destination. ANTALG provides multiple paths, but only the path with the best pheromone value is used for communication. If the optimal path fails, it leads to choosing the next optimal path. Hence proposed ant algorithm does not break down even on failure of the optimal path.

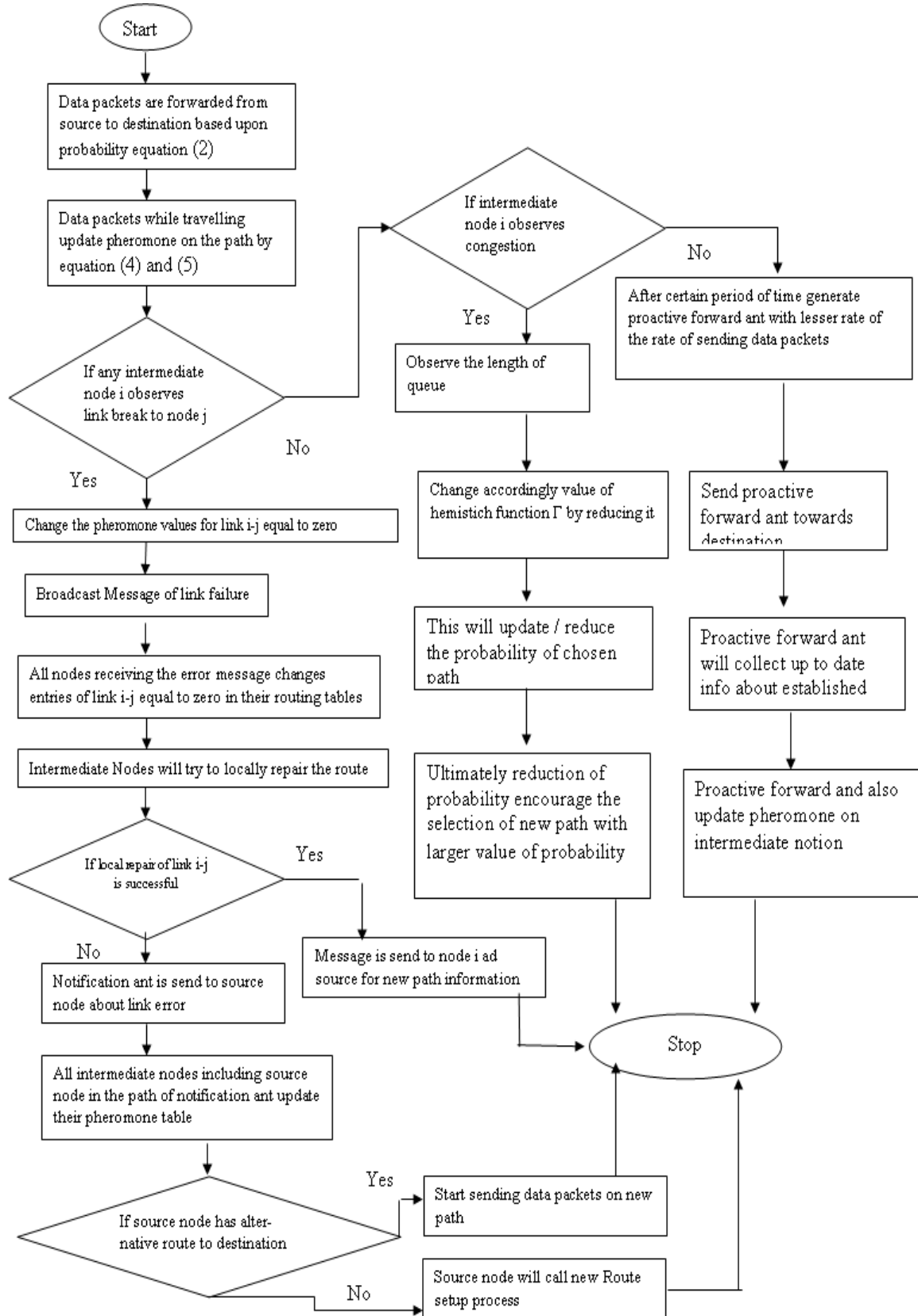


Figure 3.2: Route Maintenance & Route Recovery Phase

3.5 Complexity Analysis

The time needed by the packet while traveling from node x to node y is as follows.

$$T_{ixy} = ([\ell_x + SO(packet)] / Cap_{xy} + TT_{xy} \dots \dots \dots (6)$$

Where TT_{xy} is the transmission time of the link between nodes x and y . ℓ_x length of low priority queue of link x and y . Cap_{xy} is the capacity of the link x and y . $SO(packet)$ represents the size of the packet.

While traversing the path, every FANT computes the probability to choose the next node among all the neighbors (N_n), so it is represented by $O(N_n)$. Let us assume that the maximum hop count for any ant is H . In the worst case, FANT has to repeat all the above calculations at H different nodes, so the complexity formula is modified to $O(HN_n)$. Suppose that N_a is the number of ants generated by a node, thus the complexity is being tailored to $O(N_aHN_n)$. FANT prepares the stack containing the information of the nodes visited by it. FANT on reaching the destination node becomes the BANT and traverses the path in reverse order as followed by it. It needs to pop the stack to know the next hop to travel. The pop operation in the stack is $O(1)$. BANT must enter the one of output queues and complexity of a queue again can be measured as $O(1)$. Also, it should update the routing table for every neighbor N_n for the destination d which is $O(N_n)$. Suppose FANT searches for one distinct path and FANT or BANT pair updates only the routing tables at the source node for the given node. There is an equal probability of creating FANT with any one of the $N-1$ nodes as the destination. Hence, only one of $(N-1)$ FANTs is used to search for the shortest path between the source and the destination. Presuming that there are P number of paths available between the source and destination, so the overall complexity becomes $O(PN_aHN_n)$. Here P is considered as the maximum possible value of $P_{\max}$. $P_{\max}$ is upper bound of the maximum number of paths between the source node and destination node. The upper bound $P_{\max}$ on the total number of paths between source and destination node is $P_{\max} = [e(N-2)!]$ where e is a constant value equal to 2.7183.

3.6 Illustrative Example 1: ANTALG

Let MANETs are represented by $G=(V,E)$ having n nodes with a finite non empty set $V(V_1,V_2,\dots,V_n)$ of nodes and a non-empty set of links (E) representing two end points that can communicate with each other. Building routes from source node to destination using ANTALG routing protocol involve creation of ants onto source node and then sending them towards randomly selected nodes from list N of neighbors. Suppose A, B, C, D, E, F & G represent seven wireless nodes in a MANET environment. Let us assume that Node A wants to communicate with node G. There exist three different routes through which packets can be routed. Routing is the process by which packets can be routed from source to destination. Efficient routing protocol finds an optimal path in terms of Quality of Service (QoS). Here the links are assumed wireless in nature. Each edge in Figure 4.3 represents the pair of nodes which can directly communicate with each other in a bidirectional way.

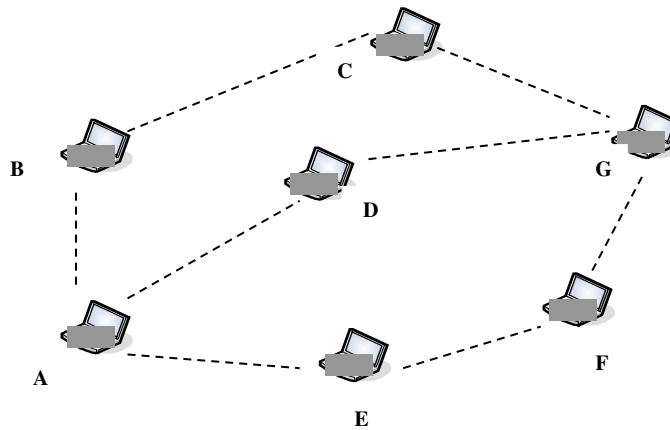


Figure 3.3: MANET representing set of nodes within the range for communication

In this example, node A is considered as source node and node G as destination node. When source A wants to communicate with destination G then FANTs are created at node A and send towards node G. Because there is no direct link between node A and G, so FANTs are broadcasted to all outgoing links, i.e., to nodes B, D and E. The FANTs are now sent to nodes C, F and G. Ultimately first FANT reach to destination node G by collecting all the information about intermediate node. BANT is created over destination node and send towards the source node A in the reverse order of nodes stored in the stack of FANT.

Let us further consider the pheromone table on few of the nodes in topology for better illustration. In the pheromone table, the values of pheromone are considered hypothetically by giving more weightage to the shorter and nearer paths as can be easily seen from example of topology. The sum of each row in table is kept as 1.0.

Table 3.5: Pheromone table of node A

Neighbors		B	D	E
Destination	B	0.8	0.1	0.1
	C	0.6	0.3	0.1
	D	0.8	0.1	0.1
	E	0.8	0.1	0.1
	F	0.1	0.3	0.6
	G	0.25	0.5	0.25

Table 3.6: Pheromone table of node G

Neighbors		C	D	F
Destination	A	0.25	0.5	0.25
	B	0.6	0.3	0.1
	C	0.8	0.1	0.1
	D	0.8	0.1	0.1
	E	0.1	0.3	0.6
	F	0.1	0.1	.08

Table 3.7: Pheromone table of node D

Neighbors		A	G
Destination	A	0.8	0.2
	B	0.6	0.4
	C	0.4	0.6
	E	0.6	0.4
	F	0.4	0.6
	G	0.2	0.8

Table 3.8: Pheromone table of node C

Neighbors		B	G
Destination	A	0.6	0.4
	B	0.8	0.2
	D	0.4	0.6
	E	0.5	0.5
	F	0.3	0.7
	G	0.2	0.8

Table 3.9: Pheromone table of node F

Neighbors		E	G
Destination	A	0.6	0.4
	B	0.5	0.5
	C	0.4	0.6
	D	0.4	0.6
	E	0.8	0.2
	G	0.2	0.8

Table 3.10: Pheromone table of node B			
Neighbors		A	C
Destination	A	0.8	0.2
	C	0.8	0.2
	D	0.6	0.4
	E	0.6	0.4
	F	0.5	0.5
	G	0.4	0.6

Table 3.11: Pheromone table of node E			
Neighbors		A	F
Destination	A	0.8	0.2
	B	0.6	0.4
	C	0.5	0.5
	D	0.6	0.4
	F	0.2	0.8
	G	0.4	0.6

According to the working of ANTALG, BANT reaching node D from node G updates the routing table of node D using Equations (3) and (4) as defined above. Assuming value of $r=0.1$, $C=1000$

If (node was in the path of the Ant) then

$$\rho(i) = \rho(i) + r * (1 - \rho(i)) + r^2 * \rho_{\text{high}} / C \dots\dots\dots(3)$$

else

$$\rho(i) = \rho(i) - r * \rho(i) - r^2 * \rho_{\text{high}} / (C * (|N_i| - 1)) \dots\dots\dots(4)$$

So putting the values of example assumed above in Equations (3), the pheromone on path D-G will be increased as follows:

$$\rho(i) = \rho(i) + r * (1 - \rho(i)) + r^2 * \rho_{\text{high}} / C \dots\dots\dots(3)$$

$$\begin{aligned} &= 0.8 + 0.1 * (1.0 - 0.8) + 0.1 * 0.1 * 0.8 / 1000 \\ &= 0.8 + 0.02 + 0.000008 \\ &= 0.820008 \end{aligned}$$

However the value of pheromone on other paths will decrease as follows:

$$\rho(i) = \rho(i) - r * \rho(i) - r^2 * \rho_{\text{high}} / (C * (|N_i| - 1)) \dots\dots\dots(4)$$

$$= 0.2 - 0.1 * 0.2 - 0.1 * 0.1 * 0.8 / (1000 * (2-1))$$

[ρ_{high} is equal to 0.8 (between 0.2 and 0.8) and $^{\mid Ni \mid}$ is the number of nodes in the neighborhood which are 2 because neighbor of node D are node A and node G respectively]

$$= 0.2 - 0.02 - 0.000008$$

$$= 0.179992$$

Table 3.12: Pheromone table of node D (<i>before updation</i>)			
Neighbors		A	G
Destination	A	0.8	0.2
	B	0.6	0.4
	C	0.4	0.6
	E	0.6	0.4
	F	0.4	0.6
	G	0.2	0.8



Table 3.12(a): Pheromone table of node D (<i>after updation</i>)			
Neighbors		A	G
Destination	A	0.8	0.2
	B	0.6	0.4
	C	0.4	0.6
	E	0.6	0.4
	F	0.4	0.6
	G	0.179992	0.820008

It can also note that the sum of pheromones in a row again becomes 1.0 after the updation,
i.e., $0.179992 + 0.820008 = 1.0$.

BANT moves and update the Pheromone table in the whole topology. After a periodic time, source and destinations are again randomly chosen and the whole network gets populated with FANTs and BANTs which will ultimately update the whole topology with the latest pheromone information. The above example clearly illustrates that the pheromone on the favorable path increases and decreases on the other paths. This will increase the probability of selection of path in Equations (3) because the probability of the path is directly proportional to the pheromone of the path. This example illustrates the accuracy and efficiency of the proposed scheme.

3.7 OANTALG: Orientation Based ANT Routing ALgorithm

Wang et al. [99] introduced an orientation factor into multicast routing and gave a new algorithm called MACA, which helps the search agents in finding the destination in the

direction of a target node. As already discussed in Section 1.7, the final shape of the proposed routing algorithm “OANTALG” is also carrying orientation factor and has the property of distributed learning. The proposed algorithm is partially prejudiced by the algorithms like AntNet [61], PERA [94], MACA [99], ADAA [100] and AntHocNet [121] that provide formative of forwarding of message packets (ant packets) from the source to the destination node in the direction of the target node. The new algorithm adds the orientation factor into ANTALG with slight changes as communicated in the next section.

3.8 Modifications Done in OANTALG from ANTALG

The data structure and basic steps used in the OANTALG are same as of ANTALG except a small difference that the new modification includes an orientation factor in the calculation of probability which guides the ants to move in the accurate direction towards the destination by emphasis of more pheromone on the better paths and reduces the pheromone over the less promising paths. The FANT, which is headed towards the destination node d , presently available at the node i , follows the path through node j by probability (Ω_{ijd}) calculated as follows:

$$\Omega_{ijd} = \frac{\rho_{ijd} + (\cos \alpha) \theta \Gamma_{ij}}{(\cos \alpha) \theta |N_i| + (1 - \theta (\cos \alpha))} \dots\dots\dots(7)$$

Where (Ω_{ijd}) is computed as normalized the sum of the pheromone ρ_{ijd} with a heuristic value Γ_{ij} taking into account the length of the link queue of the current node. θ is the weight of the importance of the heuristic function with respect to the pheromone deposited. $\cos \alpha$ represents the cosine value of the axial coordinates of the two orientation vector v_1 and v_2 computed as follows.

$$\cos \alpha = \frac{v_1 \cdot v_2}{|v_1| |v_2|} \dots\dots\dots(8)$$

3.9 Pseudo Code for OANTALG

The pseudo code of OANTALG algorithm is described as follows:

OANTALG : Orientation based ANT Routing ALGorithm

Input: Ant Set $\{a_1, a_2, a_3, \dots, a_m\}$, Number of nodes: $\{n_1, n_2, n_3, \dots, n_0\}$

Output: Edges forming path by selecting nodes with highest probability

1. **Begin**
 2. **Procedure** Node_Selection()
 3. while (SN = sel_node)
 4. **For** each source node
 5. Randomly select node from the domain of nodes
 6. **End for**
 7. **End Procedure**
 8. **Procedure** Initialize_Pheromone ()
 9. Select the pheromone value as $\rho_{ij} = 1.0 / |N_i|$
 10. **End Procedure**
 11. **Procedure** Send_FANT ()
 12. **if** (Current_Time == Time of sending FANT (Forward ant)) **then**
 13. Update the stack contents as stackF[top] = CN
 14. Increment top of the stack as top = top + 1
 15. **End if**
 16. Update the next node status as NN = NextSelect (CN , TN)
 17. **If** (NN == TN) **then**
 18. Create the Backward ant BANT
 19. Destroy forward ant FANT
 20. **else If** (NN == Already Visited)**then**
 21. Delete ant's Memory
 22. Clear the Stack contents
 23. Call Procedure Send_FANT()
 24. **else**
-

25. Assign current node as next node by assigning $CN = NN$

26. **End if**

27. **End Procedure**

28. **Procedure** Send_BANT ()

29. **while**(($TTL > 0$) and ($stackF[top] \neq NULL$))

30. Assign next node as $NN = stackF[top]$

31. Decrement top of the stack as $top = top - 1$

32. **if** ($NN == SN$) **then**

33. exit()

34. **else if** ($CN == NN$) **then**

35. Update pheromone as follows

36. **end if**

37. **If** (node was in the path of the ant) **then**

38. $\rho(i) = \rho(i) + r * (1 - \rho(i)) + r^2 * \rho_{high} / C$

39. **else**

40. $\rho(i) = \rho(i) - r * \rho(i) - r^2 * \rho_{high} / (C * (|N_i| - 1))$

41. **End if**

42. **else**

43. Destroy BANT

44. **End if**

45. **End while**

46. **End Procedure**

47. **Procedure** Next_Select (CN , TN)

48. **For** ($i = 1$ to $|N_i|$)

49. Call Orient (CN, TN)

50. Calculate $\Omega_{ijd} = \frac{\rho_{ijd} + (CT)\theta\Gamma_{ij}}{(CT)\theta|N_i| + (1 - \theta(CT))}$

51. **return** node_ID with highest value of Ω_{ijd}

52. **End For**

53. **End Procedure**

54. **Procedure** Orient (CN , TN)

55. $x1 = \text{get_x_coordinate}(CN)$

56. $y1 = \text{get_y_coordinate}(\text{CN})$

57. $x2 = \text{get_x_coordinate}(\text{TN})$

58. $y2 = \text{get_y_coordinator}(\text{TN})$

59. $v1.v2 = x1 * x2 + y1 * y2$

60. $|v1| = \sqrt{x1^2 + y1^2}$

61. $|v2| = \sqrt{x2^2 + y2^2}$

62. $CT = \frac{v1.v2}{|v1| . |v2|}$

63. **Return CT**

64.**END Procedure**

65. **End**

3.10 Illustrative Example 2: OANTALG

Figures 3.4(a)-(b) show the network topology used in MANETs which is represented by $G=(V,E)$ showing wireless nodes with a finite non empty set $V(V_1,V_2,\dots,V_n)$ of nodes and non-empty set of links (E) representing two end points that can communicate with each other because of direct communication range.

Figure 3.4(a) represents 2-D view of the network topology constructed with x-y coordinates that have the orientation attribute while Figure 3.4(b) represents the position of the nodes without plotting it on x-y axis. Building routes from the source node to the destination using OANTALG routing protocol involves the creation of ants onto the source node and then sending them towards randomly selected nodes from the list of neighbors. The proposed routing algorithm finds the optimal path based on various metrics like throughput, jitter, end- to- end delay, and total number of packets routed.

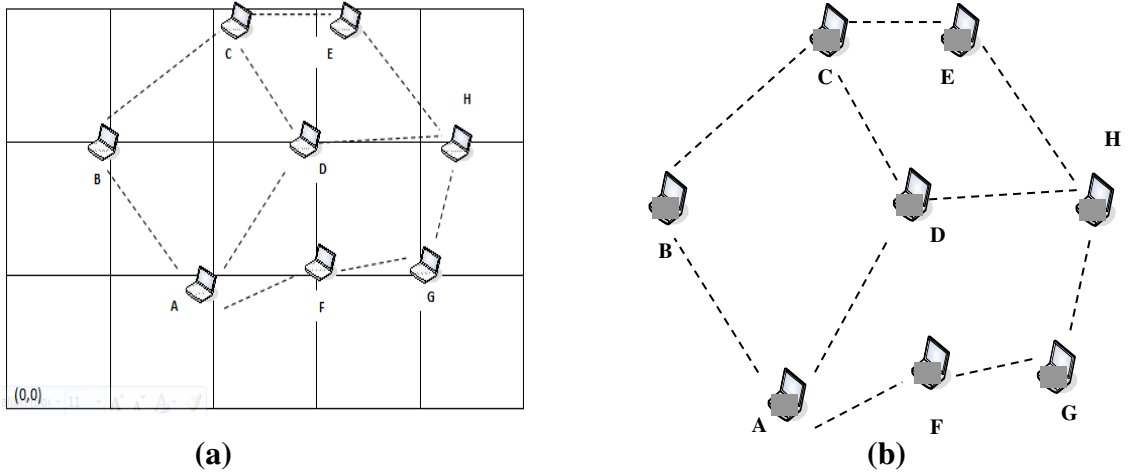


Figure 3.4(a)-(b): Generalized Network Topology in the proposed scheme

Suppose A, B, C, D, E, F, G & H represent eight wireless nodes in a MANET environment. Let us assume that node A wants to communicate with node E. There exist three different outgoing edges through which packets can be routed. Agents moving toward nodes B lay less pheromone as compared to the agents moving towards node D and node F due to the presence of the orientation factor. This can be seen from the Figure 3.4(a)- (b) that nodes D and F are in the direction towards the destination whereas node B is in reverse direction of the destination. Routing is the process by which packets can be routed from the source to the destination. Efficient routing protocol finds an optimal path in terms of Quality of Service (QoS). So, taking the assumptions as presumed above that node A is source node and node E is the destination node. Equations 7 defined below is used to compute the selection of next node with respect to the direction attribute. It gives the probability measure for selection of next node starting from the source node. Detailed explanation of the Equations is provided in Section 3.2 where the probability Ω_{ijd} which serves as the ultimate factor for deciding the next node is explained in detail. This section will illustrate the advantages of using the direction attributes for selection of the next node starting from the source node. Mathematically, it can be defined as follows:

$$\Omega_{ijd} = \frac{\rho_{ijd} + (\cos \alpha) \theta \Gamma_{ij}}{(\cos \alpha) \theta |N_i| + (1 - \theta (\cos \alpha))} \dots \dots \dots (7)$$

$\text{Cos}\alpha$ is the term, which is responsible for the orientation factor in Equations 7. Therefore, extending the example, it is assumed that all the factors except $\text{Cos}\alpha$ as same and observe how the orientation factor is responsible for increasing the probability value Ω_{ijd} on the way to the destination and decreasing the probability value Ω_{ijd} on non-favorable paths. Equations 8 derives the value of $\text{Cos}\alpha$ as follows:

$$\text{Cos}\alpha = \frac{v_1 \cdot v_2}{|v_1| \cdot |v_2|} \dots\dots\dots(8)$$

A 2-D environment has been assumed for the simulation work. The x-axis and y-axis are chosen with the help of right hand thumb rule. Middle finger and thumb point towards x-axis and y-axis defining the plane. The plane forms the quadrant containing the source as well as the destination with intersection of the thumb and middle finger near to the source and the fore finger pointing towards the direction of the destination. As there are only three outgoing nodes that can be chosen from the source node A, so let us assume the coordinate values of node A, B, F and D.

$$A(x, y) = (2, 1), B(x, y) = (1, 2), F(x, y) = (3, 1) \text{ and } D(x, y) = (3, 2)$$

$$\text{Cos}\alpha(A, B) = \frac{(2*1) + (1*2)}{\sqrt{2*2 + 1*1} + \sqrt{1*1 + 2*2}} = \frac{2+2}{\sqrt{5} + \sqrt{5}} = 0.80$$

$$\text{Cos}\alpha(A, F) = \frac{(2*3) + (1*1)}{\sqrt{2*2 + 1*1} + \sqrt{3*3 + 1*1}} = \frac{6+1}{\sqrt{5} + \sqrt{10}} = 0.992$$

$$\text{Cos}\alpha(A, D) = \frac{(2*3) + (1*2)}{\sqrt{2*2 + 1*1} + \sqrt{3*3 + 2*2}} = \frac{6+2}{\sqrt{5} + \sqrt{13}} = 0.995$$

So this can be clearly seen from the example quoted above that $\text{Cos}\alpha$ value on the path between node A and node D is the highest. So, this will encourage the values when multiplied in the numerator portion in Equations 7. Also, $(1 - \theta(\text{Cos}\alpha))$ factor in the denominator of Equations 7 will decrease its value and ultimately, the value of

probability Ω_{ij^d} will increase. Therefore, it is clear from the example shown above that how evident orientation factor plays an important role in deciding the right direction toward the destination.

The next chapter deals with the presentation of the experimental setup created for the simulation purpose and then present the experimental results. Section 4.1 presents the various performance metrics used for the comparison purpose to evaluate the proposed algorithm. Section 4.2 gives various parameter settings needed for the simulation. Finally, Section 4.3 briefs various assumptions considered for the experimental setup. Next chapter also computes the performance evaluation of the proposed algorithms with the well-known traditional and other available ant based routing protocols on the basis of the performance metrics. The simulation work is divided into three stages. So, the results are discussed in three phases, which are independently presented under three Sections 4.4, 4.5 and 4.6 respectively. Each section presents the conclusion of the phase in its last subsection.

Chapter 4

Experimental Setup and Comparative Results

4.1 Performance Metrics

A performance metric is a standard used in a routing algorithm to determine the effective value of measurement observed by a packet while travelling from the source node to the destination. Performance parameters used to measure and evaluate the effectiveness of the proposed algorithm are as follows:

- I. Throughput,
- II. Packet Loss & Delivery Ratio
- III. Data Packets Sent
- IV. Path Length
- V. End to End Delay
- VI. Jitter

4.1.1. Throughput

In communication networks, Throughput is the amount of digital data per time unit delivered over a physical or logical link. It is measured in bits per second (bits/s or bps), occasionally in data packets per second or data packets per timeslot.

- $\text{Throughput (Bytes / Sec)} = (\text{Size of Packet} * 8) / \text{Time to reach target node}$
- $\text{Throughput (Packets / Sec)} = \text{Number of Packets send successfully} / \text{Total time}$

Some of the results / contents of this chapter have been peer reviewed and accepted for publications in

- G. Singh, N. Kumar, A. K. Verma, "OANTALG: An Orientation based ANT Colony ALgorithm for Mobile Ad-hoc Networks", *Wireless Personal Communication. Volume 77, Issue 3, pages 1859-1884, DOI: 10.1007/S11277-014-1613-6 (SCI-Thomson Reuters Rated Journal with Impact Factor 0.428)*

4.1.2. Packet Drop and Delivery Ratio

It is the measure of the percentage of packets delivered successfully to the target nodes.

$$\text{Delivery Ratio} = (\text{No. of packets send} - \text{packets lost}) * 100 / \text{No. of packets send}$$

Or

$$\text{Delivery Ratio} = \text{No. of packets successfully reached} * 100 / \text{No. of packets send}$$

4.1.3. Data Packets Sent

Number of packets sent during the time of simulation were also measured to check that which of the algorithm can sent large quantities of data in the defined time interval.

$$\text{Packets Send} = \text{Number of Packets Successfully Send} + \text{Number of Packets lost}$$

4.1.4. Path Length

It is the length of the path based on the number of the hops.

$$\text{Path Length} = \text{Number of hops through which the packet routed to reach destination}$$

4.1.5. End- to-End Delay

End-to-End delay refers to the time taken for a packet to be transmitted across a network from source to destination. Data transmission seldom occurs only between two adjacent nodes, but via a path which may include many intermediate nodes. End to End delay is the sum of delays experienced at each hop from the source to the destination.

$$\text{End to end delay} = \text{Time spend on Hop}_1 + \text{Time spend on Hop}_2 + \dots + \text{Time spend on Hop}_n$$

4.1.6. Jitter

Jitter is defined as the variation in the time between packets arriving caused by network congestion, timing drift, or route changes. It can be measured as deviation or displacement of time between two peers.

$$\text{Jitter} = |T_s - T_m|$$

Where T_s is the standard time for a packet takes to reach the destination and T_m is the time observed by packet to reach the destination.

4.2 Parameter Settings

The experimental investigations are classified into three stages as already stated in Chapter 3. NS-2 was used to conduct the simulation work. The first stage implements the ANTALG in a static environment of 20 nodes in which 25 pairs of node made to communicate with each other populating the network with large number of packets. The second stage implemented ANTALG in a wireless environment by applying the appropriate modifications in the node structure of stage 1. In this stage, a small topology of 5 nodes is considered and ANTALG is compared with well-known MANET routing algorithms like AODV, DSR and ZRP. Because the topology is small, so 2 pair of nodes were made to communicate, i.e., 80 percentile of the available nodes were participating in the data transmission process. Stage 3 added one more feature as orientation with ANTALG to realize OANTALG. Orientation factor guides or motivates the moving agent (ants) and data packets to move into the right direction towards the destination.

Table 4.1: Stages in simulation work

Various Stages carried out during simulation work	
Scenario	Description
Stage 1	ANTALG simulated with a static network of 20 nodes and compared with famous ant based routing algorithm called as AntNet.
Stage 2	Shifted the work carried out in stage 1 into stage 2 by implementing ANTALG in a wireless environment with small a network size of 5 nodes. The results obtained were compared with traditional prominent protocols of MANET named as AODV, DSR and ZRP.
Stage 3	Orientation factor was added in this stage with ANTALG and new protocol called as OANTALG was came across. OANTALG was simulated with network of 40 nodes and compared with two traditional (AODV, DSR) and one ant based algorithm (HOPNET).

The simulation parameters used for execution of the new protocol in different stages are shown in Tables 4.2. Mostly parameters use the default settings. The other parameters which are changed are illustrated in the following Table.

Table 4.2: General Parameters Used during simulation in different stages

Parameter Name	Parameter Value (Stage1 / Stage2 / Stage3)
Simulation Start Time	0.5 / 10.0 / 25.0
Simulation End Time	20 / 150 / 180
Queue Type	DropTail / CMUPriqueue / Priqueue
Data Packet Type	FTP (Under TCP Class) in all Stages
Number of Nodes	20 / 5 / 40

Network Type	Wired / Wireless / Wireless
Number of Simultaneously Opened Connections	25 / 2 / 6 pairs of connections
Dimension of Topology	Flat / [500x400] / [1000x1000] [X x Y]
Algorithm Used for Comparison	Stage 1 : ANTALG / AntNet Stage 2 : ANTALG / AODV / DSR / ZRP Stage 3 : OANTALG / AODV / DSR / HOPNET
Node Speed	Stage 1 : Static Topology Stage 2 / Stage 3 : Between 10 m/s to 35 m/s [m/s : meter per second]
Mobility Model	Stage 1 : Static Topology Stage 2 : Random Way Point

4.3 Assumptions

The assumptions made in the experimental investigation are listed as follows:

- I. It is not mandatory that all the nodes in a network are moving at a same time. Some nodes get stabilized early as compare to other nodes.
- II. All the nodes in a network can generate traffic as control packets at any given time while moving according to its needs when required.
- III. Multiple hops are used as required by a data packet or control packet before arriving at the destination.
- IV. All the simulations follow the random waypoint movement model.
- V. All nodes, including the source, the destination and the intermediate nodes are mobile and are moving at an allocated speed specified during the coding of the algorithm using the 'setdest' command available in NS-2.

- VI. File transmission protocol, which is subclass of transmission control packets are used for generation of data packets in all the simulation scenarios.
- VII. All the links are presumed bidirectional in nature, i.e., any node can communicate with any node in two way fashion.

Next part of the thesis reports on experimental results and the major investigations obtained by simulation. All the simulation work was incorporated into the test bed of NS-2 (version 2.33). The comparison of experimental results was classified into three categories, according to the simulation work carried out in three stages presented by three different Sections.

- I. Section 4.4 outlines the comparative analysis of ANTALG with AntNet.
- II. Section 4.5 discusses the findings of the simulated ANTALG protocol in wireless networks with AODV, DSR and ZRP.
- III. Section 4.6 presents the new protocol OANTALG which includes orientation factor inculcated with the properties of ANTALG and compared the results with AODV, DSR and HOPNET.

4.4 Experimental Results of First Stage of Simulation Work

This Section presents the comparative analysis of the results obtained from the execution of topology under stage 1. In the simulation work of the first stage, the topology shown in Figure 4.1 is evolving. The given topology makes a static network with size of 20 nodes. The nodes are numbered from 0 to 19 for the identification intention. The simulation work is done in Linux environment. For the sake of similarity, the given topology is employed in all the protocols used for the purpose of comparison.

The experimental setup was executed for many times. The results of the few executions are taken for the purpose of comparison and are presented in following subsections. As per the assumptions stated in Section 4.3, all the links shown in Figure 4.1 are bi-directional in nature, i.e., the topology uses the full duplex channels. On

executing the program in NS-2, two types of files are produced. One file contains the data to be input to network animator that gives the output in the form of video containing the running screen similar to which is shown in Figure 4.1. Whereas other file is called as the trace file which is used to extract the useful results in the form of performance metrics like throughput, packet loss, end to end delay and jitter etc.

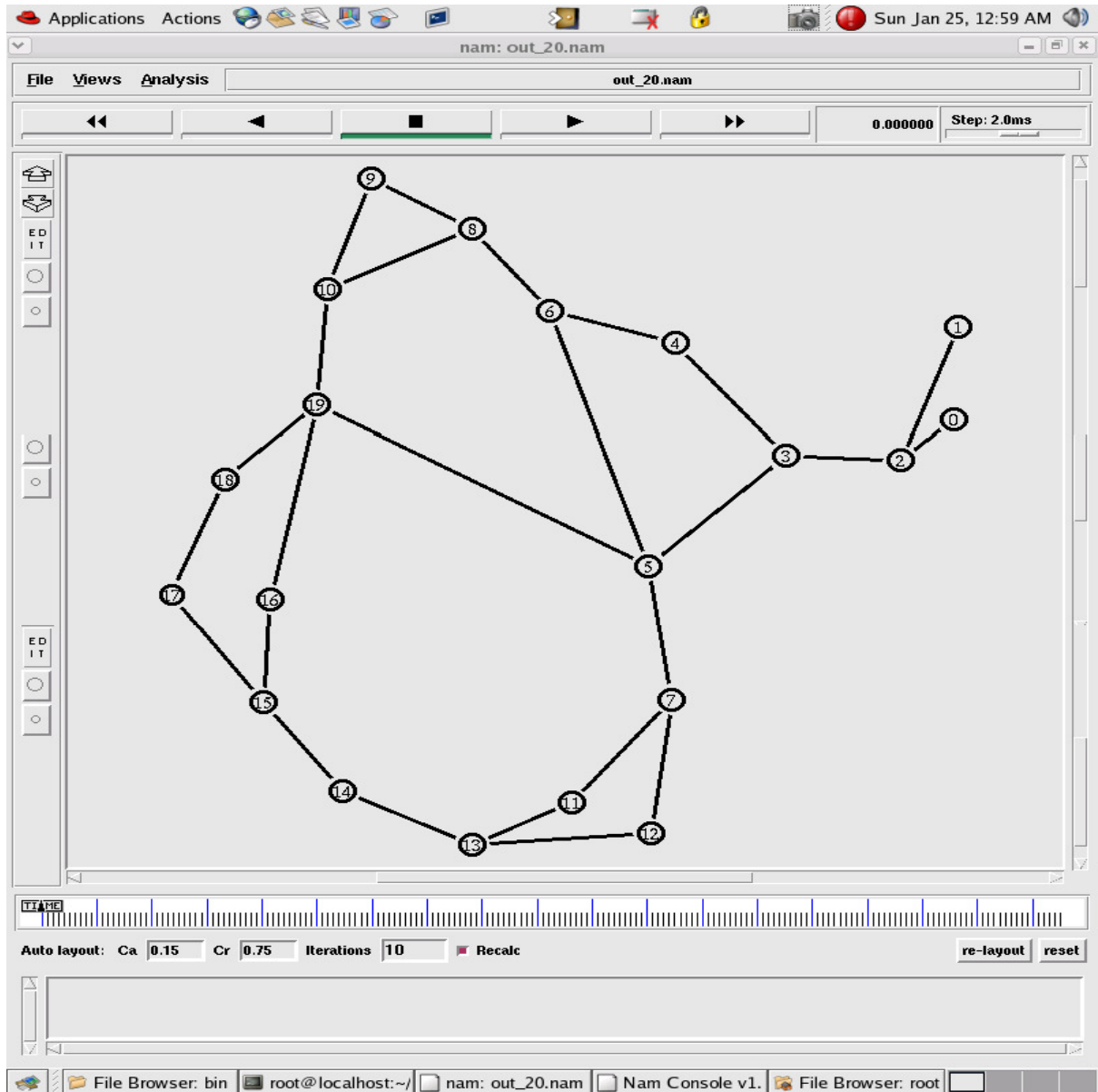


Figure 4.1: Screenshot of ANTALG in static network

4.4.1 Routing Table / Optimized Path Calculation

In the static topologies, one can easily use, see and print the routing tables. The simulation work presents two cases. In the first case, optimized or minimal found is found by ANTALG, whereas AntNet found a path which is lengthier than the path found by ANTALG. In the second case, both the algorithms find the same path, but the path found by ANTALG puts more emphases on the searched path by showing the better pheromone concentration.

4.4.1.1 Case 1 (Shorter Path Length)

In the first case, node 10 is considered as source node and node 12 is considered as destination node from the topology shown in Figure 4.1. Table 4.3 represents the routing table of node 10. In order to send the packets to destination 12 there are three outgoing edges at node 10. The pheromone concentration over the link to node 19 is more as compared to other outgoing edges. So, in both the algorithms, node 19 is chosen as the next node for moving to destination 12.

Table 4.3: Routing Table at node 10

Source Node 10 -----→ Destination Node 12						
AntNet			ANTALG			
Routing Table at node 10			Routing Table at node 10			
dest	next	phvalue	dest	next	phvalue	
0	19	0.240298	0	19	0.258546	
0	8	0.352512	0	8	0.433693	
.....						
11	19	0.805273	11	19	0.825981	
11	8	0.175056	11	8	0.142479	
11	9	0.019671	11	9	0.031540	
12	19	0.840551	12	19	0.668880	
12	8	0.146368	12	8	0.229254	
12	9	0.013081	12	9	0.101867	

13	19	0.898656	13	19	0.952206
13	8	0.097404	13	8	0.041542
13	9	0.003940	13	9	0.006252
Path - 10 – 19			Path - 10 - 19		

After moving to node 19, routing table of node 19 will be inspected to move towards the destination. Table 4.4 represents the routing table of node 19. On investigating the routing table, it is found that AntNet chooses 16 as the next node, whereas ANTALG chooses node 5 as the next node.

Table 4.4: Routing Table at node 19

Source Node 10 -----→ Destination Node 12					
AntNet			ANTALG		
Routing Table at node 19			Routing Table at node 19		
dest	next	phvalue	dest	next	phvalue
11	18	0.000619	11	18	0.001382
11	16	0.557441	11	16	0.299816
11	10	0.002227	11	10	0.041025
11	5	0.439713	11	5	0.657776
12	18	0.000354	12	18	0.000179
12	16	0.712868	12	16	0.237362
12	10	0.009726	12	10	0.037693
12	5	0.277052	12	5	0.724765
13	18	0.000002	13	18	0.000010
13	16	0.998232	13	16	0.608669
13	10	0.000031	13	10	0.005080
13	5	0.001735	13	5	0.386240
Path - 10 – 9 – 16			Path -10 – 19 – 5		

After moving to node 16, routing table of node 16 will be checked in order to move towards the destination by AntNet. However, the ANTALG will now peep into

routing table of node 5. Table 4.5 represents the two routing tables used by each algorithm. AntNet chooses node 15 as the next node because the pheromone concentration on this path is more as compared to other available options. But ANTALG opts node 7 as the next node for the packets to move to next node.

Table 4.5: Routing Table at node 16 (AntNet), node 5 (ANTALG)

Source Node 10 -----→ Destination Node 12					
AntNet			ANTALG		
Routing Table at node 16			Routing Table at node 5		
dest	next	phvalue	dest	next	phvalue
11	15	0.967165	11	19	0.000000
11	19	0.032835	11	7	1.000000
12	15	0.997172	11	3	0.000000
12	19	0.002828	11	6	0.000000
13	15	0.999998	12	19	0.000000
13	19	0.000002	12	7	1.000000
			12	3	0.000000
			12	6	0.000000
			13	19	0.000000
			13	7	1.000000
			13	3	0.000000
			13	6	0.000000
Path - 10 – 9 – 16 – 15			Path -10 – 19 – 5 – 7		

On reaching to node 15, routing table of node 15 will be examined for choosing the next node by AntNet. Node 14 is elected as the next node to reach the destination because of higher pheromone concentration on that edge. However, the ANTALG will review the routing table of node 7 and chooses node 12 as the next node to move towards the destination.

So, packets fired by ANTALG have ultimately reached the destination, but the packets transmitted by AntNet are still on the way to the destination. Table 4.6 shows

that packets started by AntNet will need to move node 13 in order to finally reach the destination. This example of running simulation makes us clear that the ANTALG favors the shorter or optimized paths as compared to AntNet which is a well-known algorithm that belongs to the category of ant routing.

Table 4.6: Routing Table at node 15 (AntNet), node 7 (ANTALG)

Source Node 10 -----→ Destination Node 12					
AntNet			ANTALG		
Routing Table at node 15			Routing Table at node 7		
dest	next	phvalue	dest	next	phvalue
11	17	0.000101	11	11	1.000000
11	16	0.000860	11	12	0.000000
11	14	0.999040	11	5	0.000000
12	17	0.000002	12	11	0.000000
12	16	0.000004	12	12	1.000000
12	14	0.999994	12	5	0.000000
13	17	0.000000	13	11	0.339157
13	16	0.000000	13	12	0.660843
13	14	1.000000	13	5	0.000000
Path - 10 – 9 – 16 – 15 – 14			Path -10 – 19 – 5 – 7 – 12		
So, Ultimately the path will be					
Path - 10 – 9 – 16 – 15 – 14 – 13 – 12					

4.4.1.2 Case 2 (Better Concentrated Pheromone Path)

In the second case, node 0 is considered as source node and node 14 is considered as destination node by taking the example of the topology shown in Figure 4.1. Table 4.7 represents the routing table of node 0. In order to send the packets to destination node 14 there exists only one outgoing node, i.e., node 2. That is why the pheromone concentration on the edge towards node 2 is 1.0 because there is no other outgoing edge. So, in both the algorithms, node 2 is chosen as the next node for sending the packets to destination 14.

Table 4.7: Routing Table at node 0

Source Node 0 -----→ Destination Node 14					
AntNet			ANTALG		
Routing Table at node 0			Routing Table at node 0		
dest	next	phvalue	dest	next	phvalue
1	2	1.000000	1	2	1.000000
2	2	1.000000	2	2	1.000000
3	2	1.000000	3	2	1.000000
4	2	1.000000	4	2	1.000000
5	2	1.000000	5	2	1.000000
6	2	1.000000	6	2	1.000000
7	2	1.000000	7	2	1.000000
8	2	1.000000	8	2	1.000000
9	2	1.000000	9	2	1.000000
10	2	1.000000	10	2	1.000000
11	2	1.000000	11	2	1.000000
12	2	1.000000	12	2	1.000000
13	2	1.000000	13	2	1.000000
14	2	1.000000	14	2	1.000000
15	2	1.000000	15	2	1.000000
16	2	1.000000	16	2	1.000000
17	2	1.000000	17	2	1.000000
18	2	1.000000	18	2	1.000000
19	2	1.000000	19	2	1.000000
Path - 0 – 2			Path - 0 – 2		

Packets on reaching node 2 examine the routing table of node 2 as shown in Table 4.8. On investigating the routing table of node 2, it is found that AntNet chooses node 3 as the next node with pheromone concentration of **0.997753**. However ANTALG chooses node 3 as the next node with pheromone concentration of **0.998245**.

This shows that pheromone concentration created by ANLTALG is more than pheromone concentration deposited by AntNet.

Table 4.8: Routing Table at node 2

Source Node 0 -----→ Destination Node 14					
AntNet			ANTALG		
Routing Table at node 2			Routing Table at node 2		
dest	next	phvalue	dest	next	phvalue
13	3	0.999577	13	3	0.999348
13	1	0.000211	13	1	0.000326
13	0	0.000211	13	0	0.000326
14	3	0.997753	14	3	0.998245
14	1	0.001124	14	1	0.000877
14	0	0.001124	14	0	0.000877
15	3	0.998790	15	3	0.994315
15	1	0.000605	15	1	0.002843
15	0	0.000605	15	0	0.002843
Path - 0 – 2 - 3			Path - 0 – 2 – 3		

After reaching to node 3, both the algorithms choose node 5 as the next node for reaching to destination as shown in Table 4.9. AntNet chooses the next node with pheromone concentration of **0.571247** and ANTALG chooses the next node with pheromone concentration of **0.592369**.

Table 4.9: Routing Table at node 3

Source Node 0 -----→ Destination Node 14					
AntNet			ANTALG		
Routing Table at node 3			Routing Table at node 3		
dest	next	phvalue	dest	next	phvalue
13	5	0.618293	13	5	0.595403
13	4	0.381703	13	4	0.404590
13	2	0.000004	13	2	0.000007

14	5	0.571247	14	5	0.592369
14	4	0.428699	14	4	0.407563
14	2	0.000054	14	2	0.000068
15	5	0.596102	15	5	0.552959
15	4	0.403867	15	4	0.446972
15	2	0.000031	15	2	0.000068
Path - 0 - 2 - 3 - 5			Path - 0 - 2 - 3 - 5		

After reaching to node 5, both the algorithms opts node 5 as the subsequent node for reaching to destination as represented by Table 4.10. AntNet chooses the next node with pheromone concentration of **0.977603** and ANTALG chooses the next node with pheromone concentration of **0.999569**.

Table 4.10: Routing Table at node 5

Source Node 0 -----> Destination Node 14					
AntNet			ANTALG		
Routing Table at node 5			Routing Table at node 5		
dest	next	phvalue	dest	next	phvalue
13	19	0.000000	13	19	0.000001
13	7	1.000000	13	7	0.999999
13	3	0.000000	13	3	0.000000
13	6	0.000000	13	6	0.000000
14	19	0.022396	14	19	0.000427
14	7	0.977603	14	7	0.999569
14	3	0.000000	14	3	0.000002
14	6	0.000001	14	6	0.000001
15	19	0.129994	15	19	0.185390
15	7	0.844743	15	7	0.814270
15	3	0.000001	15	3	0.000214
15	6	0.025262	15	6	0.000126
Path - 0 - 2 - 3 - 5 - 7			Path - 0 - 2 - 3 - 5 - 7		

After reaching to node 7, both the algorithms select node 11 as the subsequent node for reaching to destination as shown in Table 4.11. AntNet chooses the next node with pheromone concentration of **0.650434** and ANTALG chooses the next node with pheromone concentration of **0.517680**.

Table 4.11: Routing Table at node 7

Source Node 0 -----→ Destination Node 14					
AntNet			ANTALG		
Routing Table at node 7			Routing Table at node 7		
dest	next	phvalue	dest	next	phvalue
13	11	0.687865	13	11	0.511206
13	12	0.312135	13	12	0.488794
13	5	0.000000	13	5	0.000000
14	11	0.650434	14	11	0.517680
14	12	0.349566	14	12	0.482320
14	5	0.000000	14	5	0.000000
15	11	0.572055	15	11	0.514807
15	12	0.422846	15	12	0.439845
15	5	0.005098	15	5	0.045348
Path - 0 – 2 – 3 – 5 – 7 – 11			Path - 0 – 2 – 3 – 5 – 7 – 11		

After reaching to node 11, both the algorithms select node 13 as the next node for going to destination as shown in Table 4.12. AntNet chooses the next node with pheromone concentration of **0.997768** and ANTALG chooses the next node with pheromone concentration of **0.996771**.

Table 4.12: Routing Table at node 11

Source Node 0 -----→ Destination Node 14					
AntNet			ANTALG		
Routing Table at node 11			Routing Table at node 11		
dest	next	phvalue	dest	next	phvalue

13	13	0.999464	13	13	0.998770
13	7	0.000536	13	7	0.001230
14	13	0.997768	14	13	0.996771
14	7	0.002232	14	7	0.003229
15	13	0.992937	15	13	0.996035
15	7	0.007063	15	7	0.003965
Path - 0 - 2 - 3 - 5 - 7 - 11 - 13			Path - 0 - 2 - 3 - 5 - 7 - 11 - 13		

After reaching at node 13, both the algorithms select node 14 as the final node for reaching the destination as shown in Table 4.13. AntNet chooses the node 14 with pheromone concentration of **1.0** and ANTALG chooses the node 14 with pheromone concentration of 1.0.

Table 4.13: Routing Table at node 13

Source Node 0 -----> Destination Node 14					
AntNet			ANTALG		
Routing Table at node 13			Routing Table at node 13		
dest	next	phvalue	dest	next	phvalue
12	14	0.000000	12	14	0.000000
12	12	1.000000	12	12	0.999994
12	11	0.000000	12	11	0.000006
14	14	1.000000	14	14	1.000000
14	12	0.000000	14	12	0.000000
14	11	0.000000	14	11	0.000000
15	14	1.000000	15	14	1.000000
15	12	0.000000	15	12	0.000000
15	11	0.000000	15	11	0.000000
Path: -0-2-3-5-7-11-13-14			Path -0-2-3-5-7-11-13-14		

Tables 4.7 to 4.13 represent the path traversed by packets from node 0 to 14. However, packets followed the same path to the destination, but the routing tables of

ANTALG show better pheromone values. It means ANTALG emphasis more on better paths by depositing more pheromone on the promising path, thus leading to better and stronger decisions in the form of pheromone concentration.

4.4.2 Jitter

Another metric used to measure the performance is Jitter. It is measured as the absolute variation in the timings of end to end delay measured by different packets. The simulation work also compared ANTALG with AntNet on the basis of Jitter. The value obtained by five different executions is given in Table 4.14. After that the graph has been formulated to represent the values along with the average.

Table 4.14: Jitter surveyed in different executions

	Experiment_1	Experiment_2	Experiment_3	Experiment_4	Experiment_5	Average Jitter
AntNet	0.851940338	0.861880461	0.867536882	0.866526418	0.870469288	0.863670 677
ANTALG	0.86538114	0.876267252	0.859083291	0.841965725	0.850322426	0.858603 967

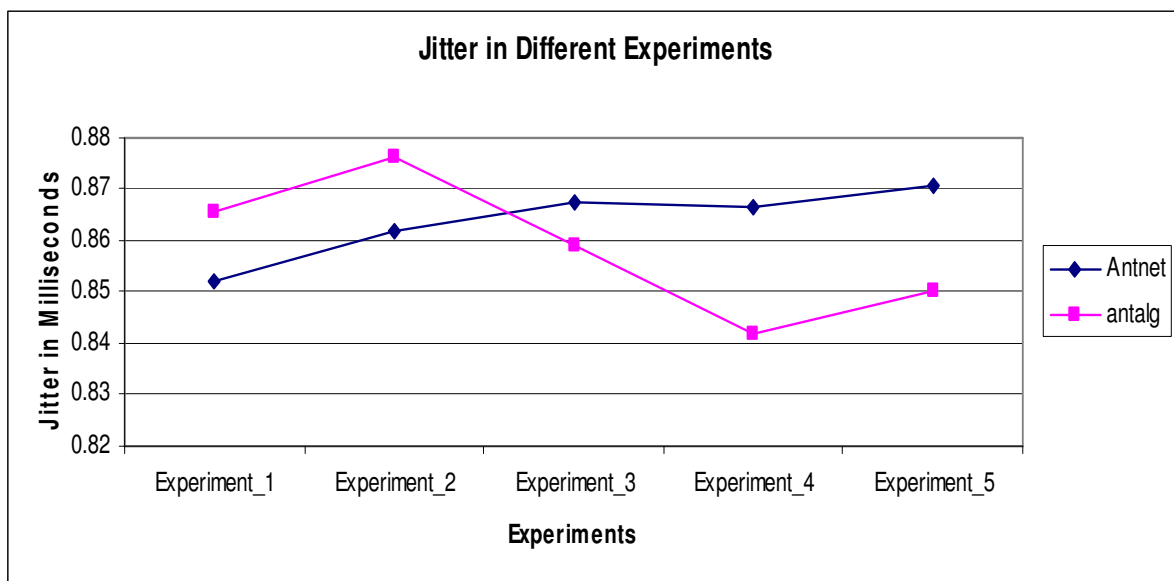
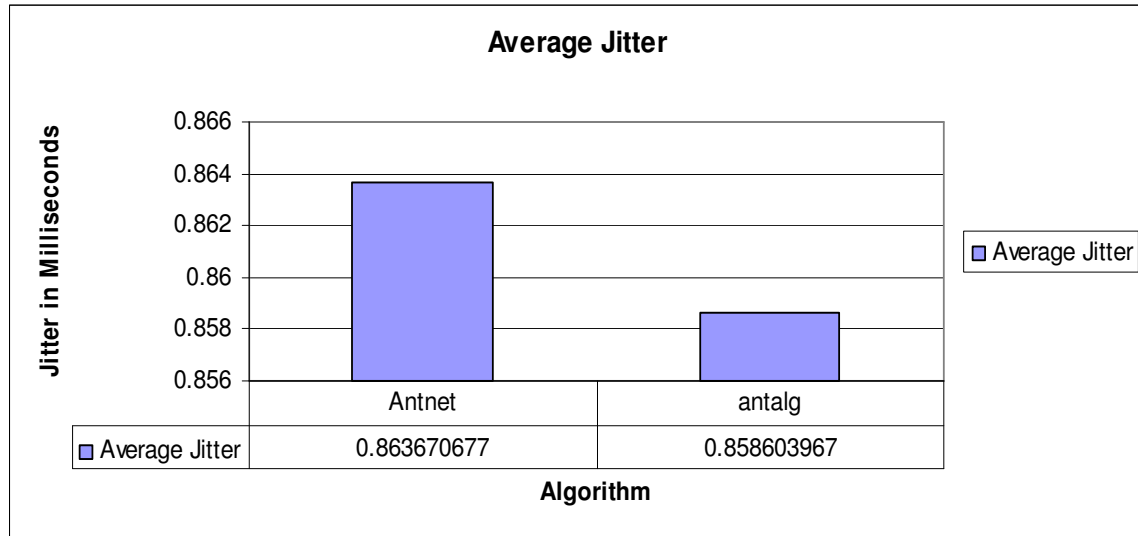


Figure 4.2: Jitter observed in different executions

**Figure 4.3:** Average Jitter

Figures 4.2 and 4.3 represent the jitter value achieved from different experiments and the average of the obtained values respectively. AntNet experiences more jitter as compared to ANTALG. This indicates that the packets in AntNet observe larger variation in the delay monitored during their travel.

4.4.3 End-to-End Delay

The end to end delay of all successfully delivered data packets is also measured. This performance metric gives another perspective on the overall performance of the simulated algorithms. As a general requirement, the delay observed by the packets should be minimized.

Table 4.15: End to End Delay monitored in different executions

	Experiment_1	Experiment_2	Experiment_3	Experiment_4	Experiment_5	Average End to End Delay
AntNet	1.246952505	1.258828282	1.244388413	1.244112767	1.255822582	1.25002091
ANTALG	1.224922436	1.249492899	1.239000115	1.22052448	1.246613336	1.236110653

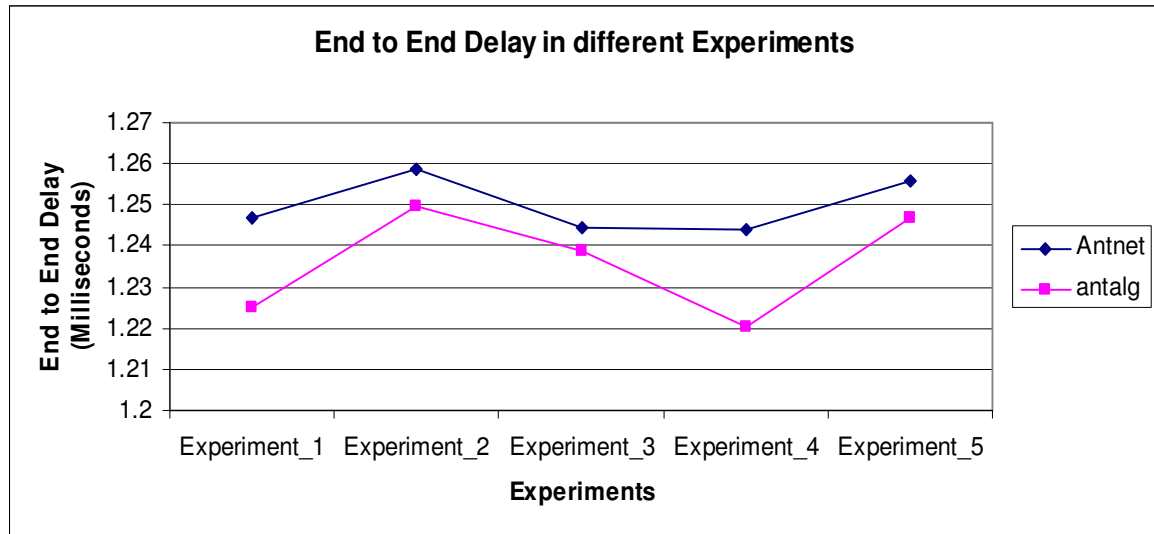


Figure 4.4: End to End Delay observed from different Experiments

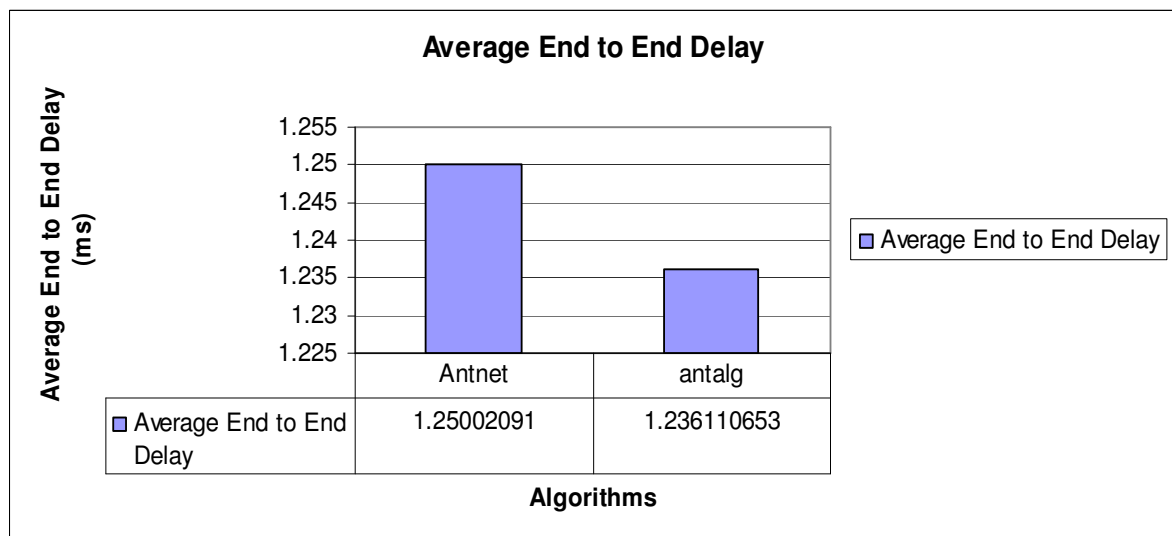


Figure 4.5: Average End to End Delay from different experiments

The results obtained by Figures 4.4 and 4.5 show that ANTALG has lesser end to end delay as compared to AntNet. This indicates that packets reach their destination earlier in ANTALG with lesser delay.

4.4.4 Control Packet Overhead

Control packet overhead is measured as the total number of traffic packets transmitted in the system to maintain the topology. A routing protocol should disburse as minimum control traffic as possible in order to successfully deliver data packets.

Table 4.16: Control Packet Overhead monitored in different executions

	Experiment_1	Experiment_2	Experiment_3	Experiment_4	Experiment_5	Total Control Packets Sent
AntNet	46965	47336	46742	46599	47223	234865
ANTALG	45966	46878	46557	45752	46800	231953

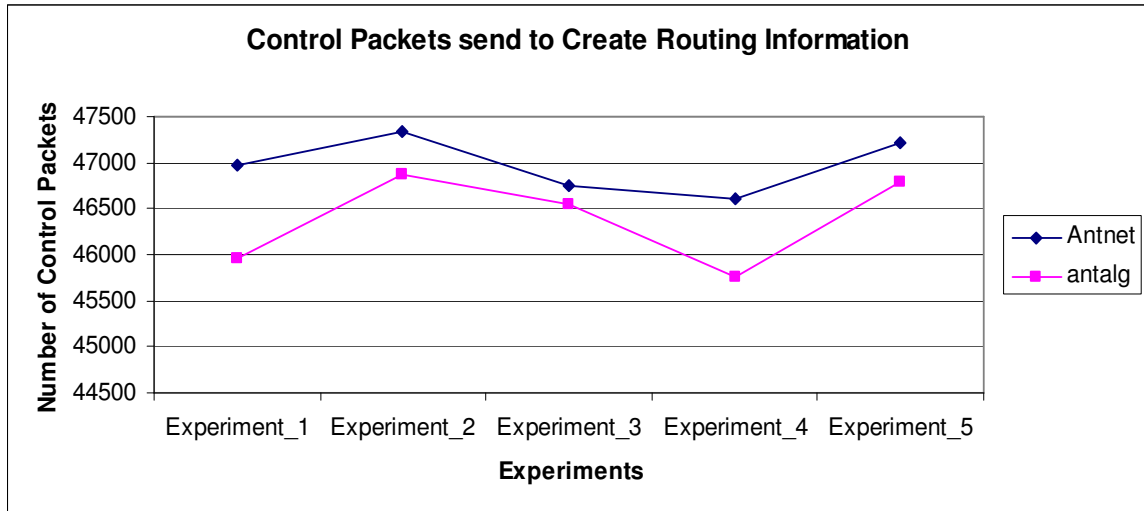


Figure 4.6: Control Packets sent in different experiments

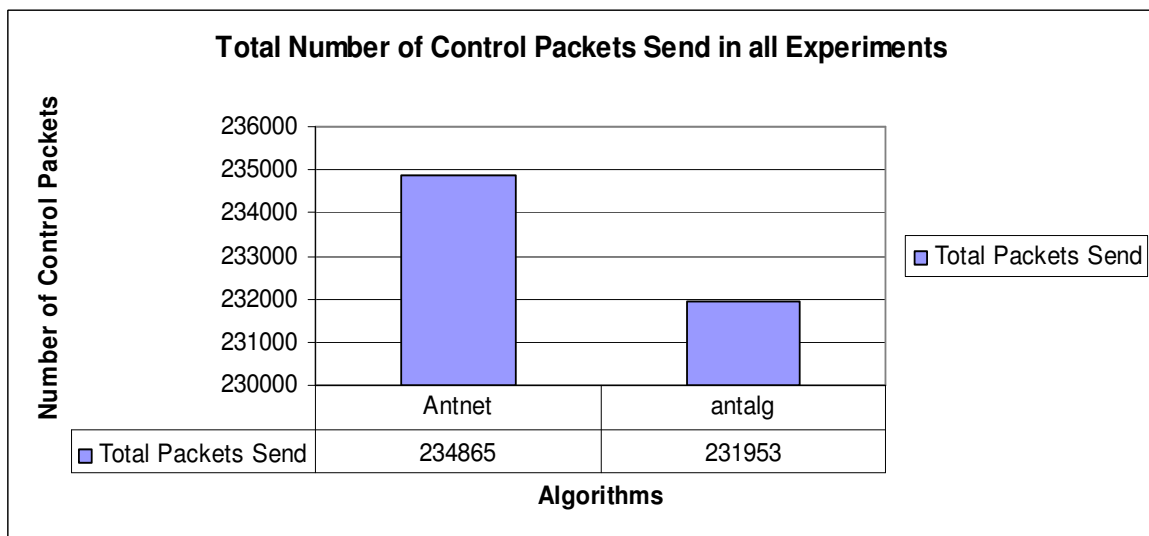


Figure 4.7: Average Control Packets sent calculated from different experiments

Figures 4.6 and 4.7 reveal that more number of control packets are transmitted in AntNet, which is the indication of the greater overhead than ANTALG. It means AntNet observe a larger number of traffic as compared to ANTALG.

4.4.5 Packet Loss

This Section presents summary of results for showing total number of packets lost by each routing protocol in different experiments during the time of simulations.

Table 4.16: Total number of packets lost in different executions

	Experiment_1	Experiment_2	Experiment_3	Experiment_4	Experiment_5	Total Packets lost
AntNet	6	5	4	1	5	21
ANTALG	3	3	4	2	3	15

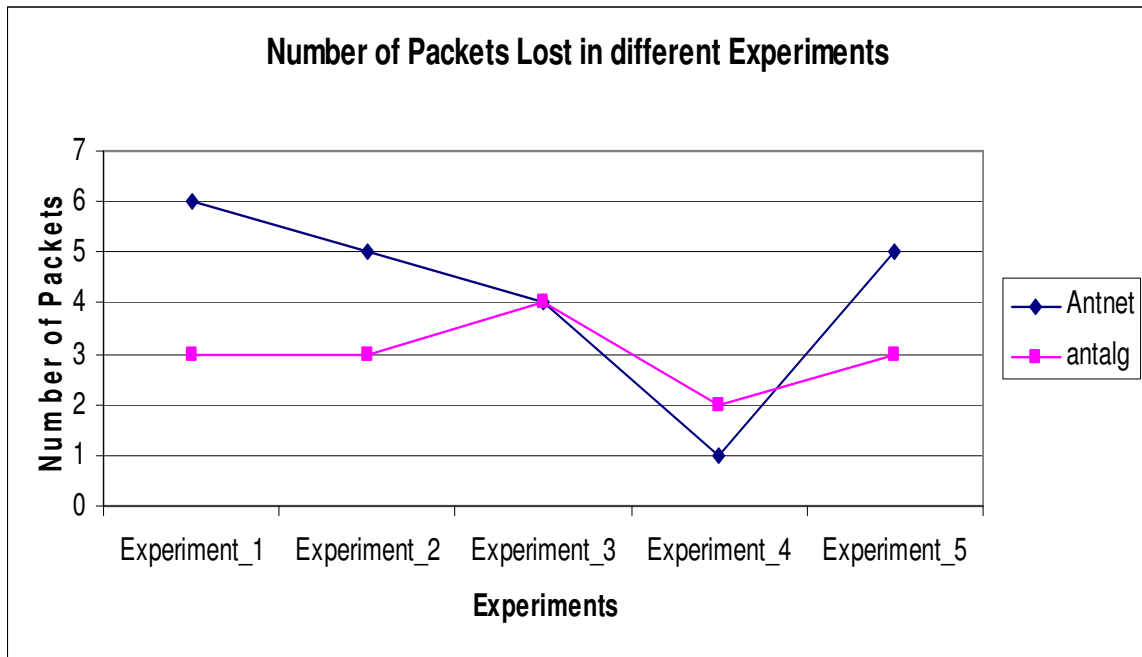


Figure 4.8: Number of Packets lost during different experiments

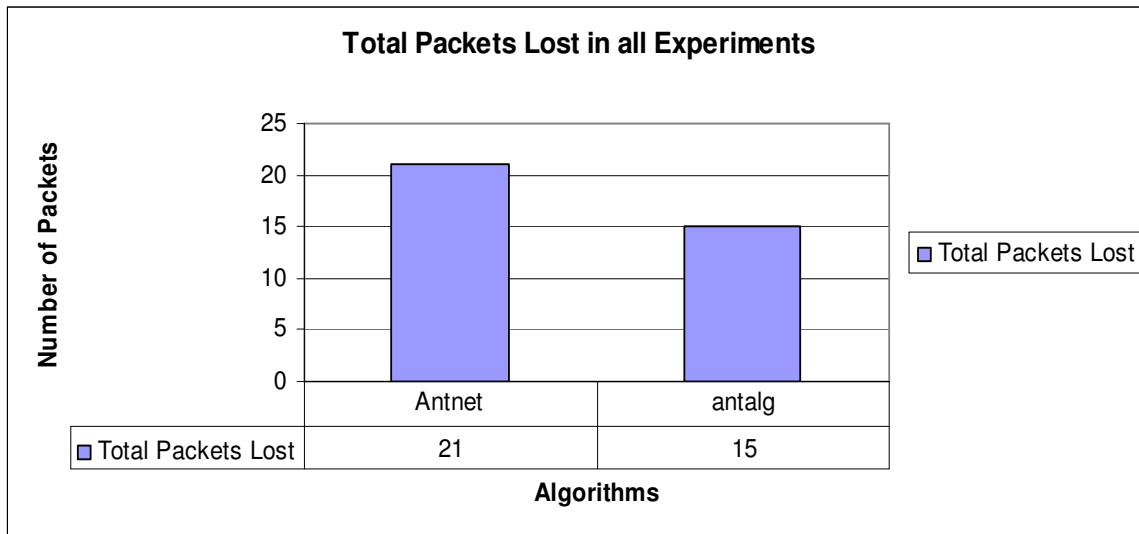


Figure 4.9: Total number of Packets lost in all the experiments

This can be easily stated from the graph of Figures 4.8 and 4.9 that fewer packets are dropped in ANTALG. These results signal towards more successful delivery of packets in ANTALG as compared to AntNet.

4.4.6 Delivery Ratio

This Section presents results obtained from experiments based on packet lost depending upon which one can find the delivery ration. Delivery ratio is the measure of a fraction of the total packets successfully delivered at the destination to the total number of packets transmitted.

Table 4.18: Delivery ratio in different experiments

	Experiment_1	Experiment_2	Experiment_3	Experiment_4	Experiment_5	Average Delivery Ratio
AntNet	99.98722453	99.98943721	99.99144239	99.99785403	99.98941194	99.99107402
ANTALG	99.99347344	99.99360041	99.99140838	99.99562861	99.99358974	99.99354012

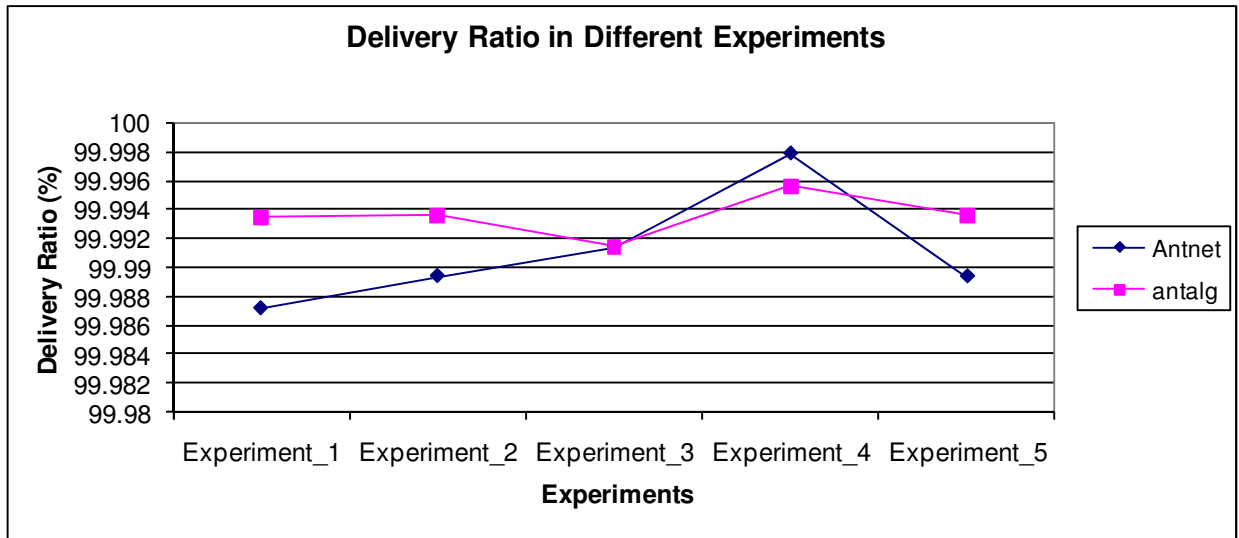


Figure 4.10: Delivery Ratio in all the experiments

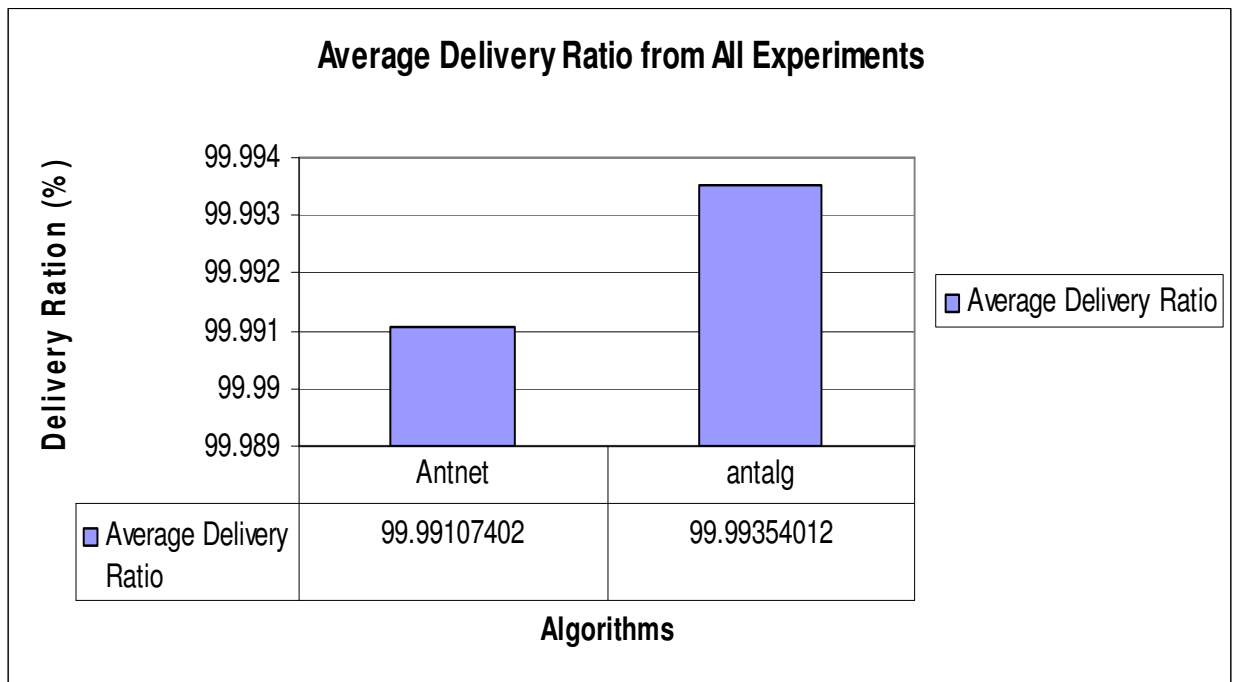


Figure 4.11: Delivery Ratio from all the experiments

The outcome of the Table 4.18, Figure 4.10 and Figure 4.11 reveals that delivery ratio is better in ANTALG as compared to AntNet. So, ANTALG performs better than AntNet on the basis of delivery ratio.

4.4.7 Conclusion at the End of Stage One

- I. ANTALG traces better routes than the AntNet in terms of shorter path and better quality paths in term of more pheromone concentration.
- II. Jitter is less in case of ANTALG as compared to AntNet. This shows there is less variation in the time needed to traverse the route.
- III. AntNet has higher end to end delay as compared to ANTALG. This depicts that the packets reach the destination with less time in case of ANTALG.
- IV. More number of control packets were sent in case of AntNet to create the topology information which indicates higher control overhead in AntNet and lesser in ANTALG.
- V. Lesser number of packets were dropped in ANTALG. It points that the ANTALG performs better than AntNet.
- VI. ANTALG has a better delivery ratio than AntNet.

4.5 Experimental Results of Second Stage of Simulation Work

This Section shifts the data structure and other implementations of wired network to a wireless network by doing nominal changes. The running screenshot of network in stage two is shown in Figure 4.12. The small wireless network of stage two comprises of 5 nodes. The simulation work is done in Linux environment. In this stage the simulation results obtained from the execution of ANTALG is compared with the simulation results obtained from the execution of famous protocols like AODV, DSR and ZRP. For the sake of similarity, the same scenario is employed in all the protocols used for the purpose of comparison. The experimental setup was executed according to the parameters as already discussed in the Table 4.2. As per the assumptions stated in Section 4.3, all the links are bi-directional in nature, i.e., the topology uses the full duplex channels for communication among the nodes.

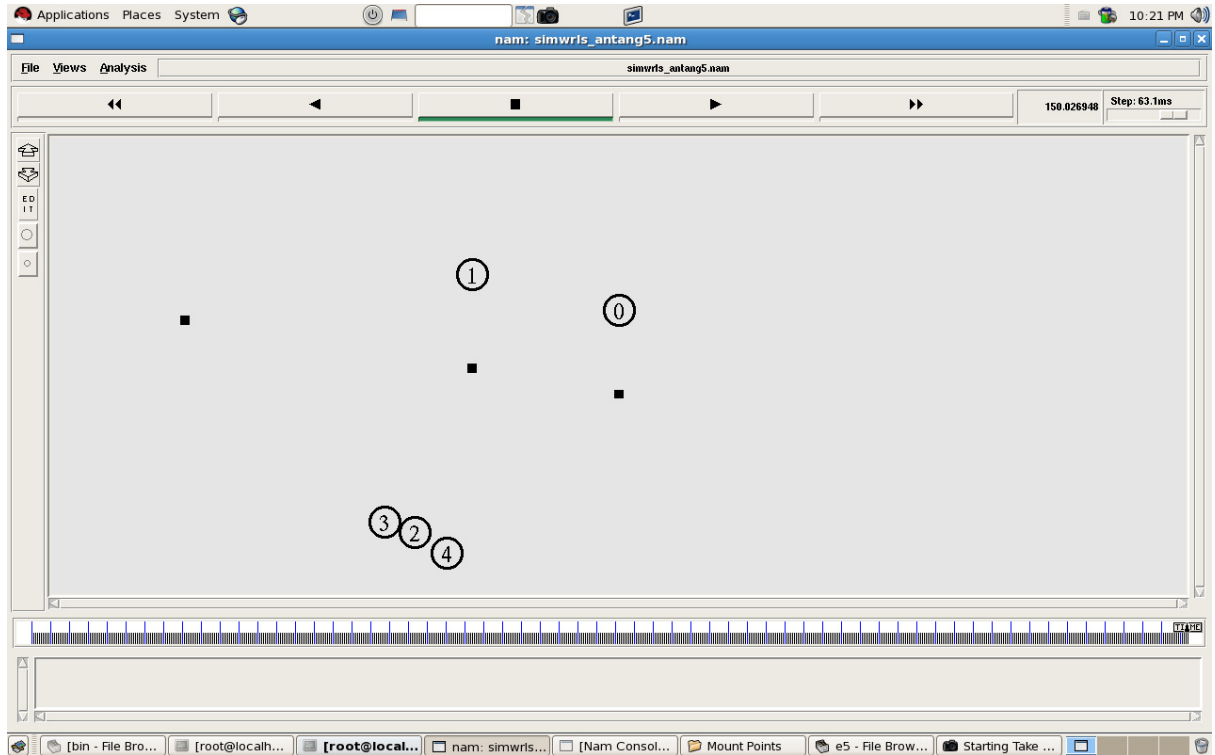


Figure 4.12: Running screenshot for wireless network running using ANTALG

4.5.1 Throughput

Throughput is a classic evaluation metric for routing algorithms. It is measured as the fraction of successfully delivered data packets to the total time of simulation. It is desirable that this metric should remain as high as possible under all circumstances.

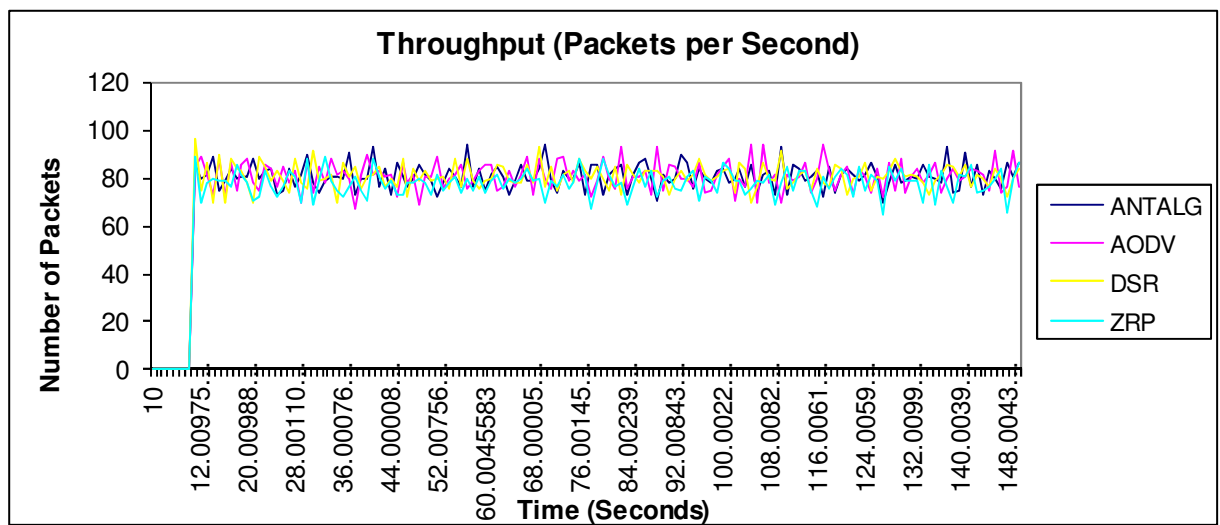


Figure 4.13: Throughput (Packets per Second)

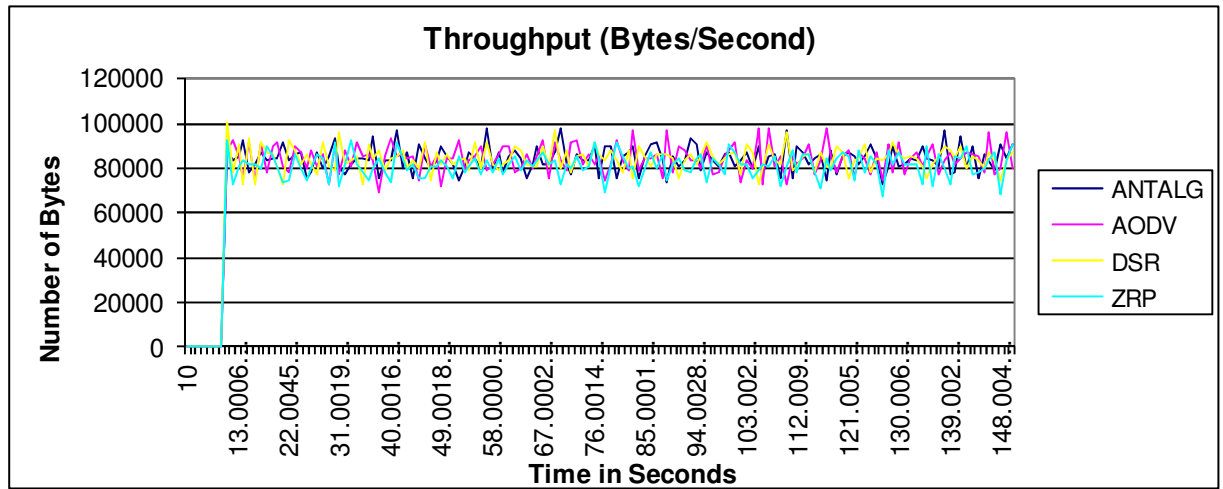


Figure 4.14: Throughput (Bytes per Second)

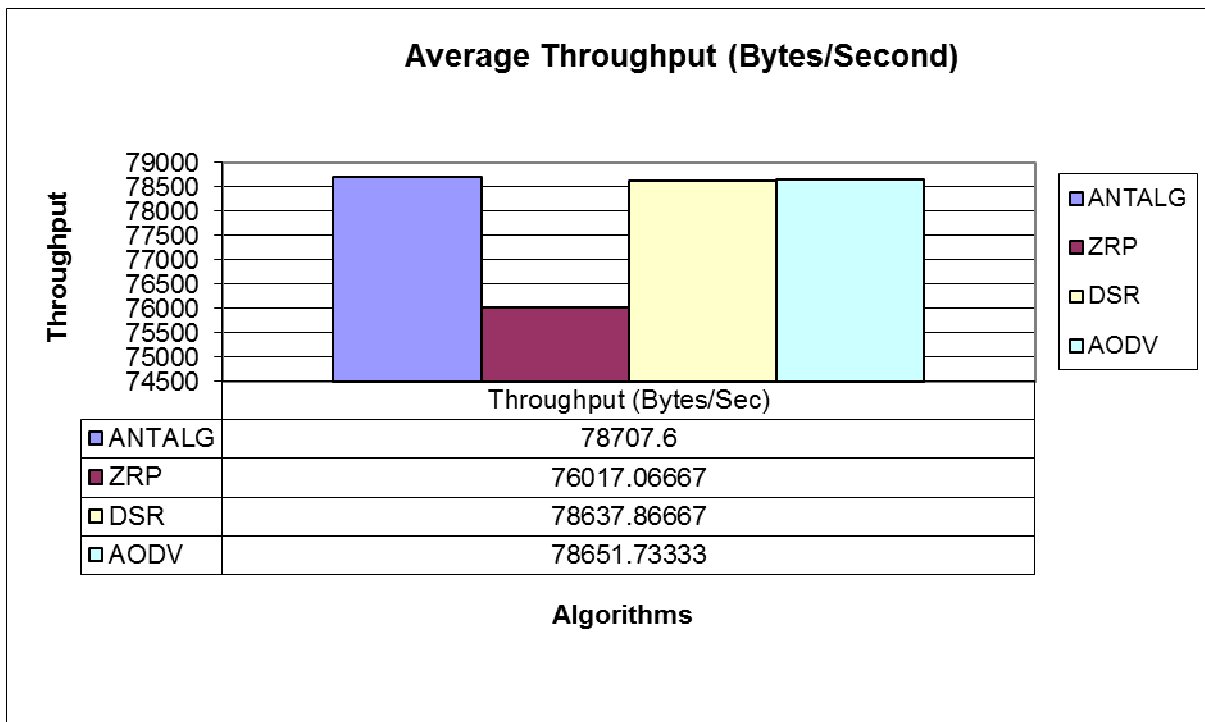


Figure 4.15: Average Throughput (Bytes per Second)

The throughput metric can be measured either in scale of packets per second or in scale of bytes per second. Figures 4.13 and 4.14 show the throughput in packets per second and throughput in bytes per second respectively. Study of the graphs for throughput clearly showed that the ANTALG take the lead. So ANTALG has shown

better performance when compared on the basis of throughput with AODV, DSR and ZRP algorithms.

4.5.2 Number of Packets Dropped and Delivery Ratio

This metric is another parameter used to compare the performance of the routing algorithms. As a desirable property lesser number of packets should be dropped while communication. After calculating the lost packets, delivery ratio can be calculated by finding the percentage of the successfully delivered packets to the total number of transmitted packets.

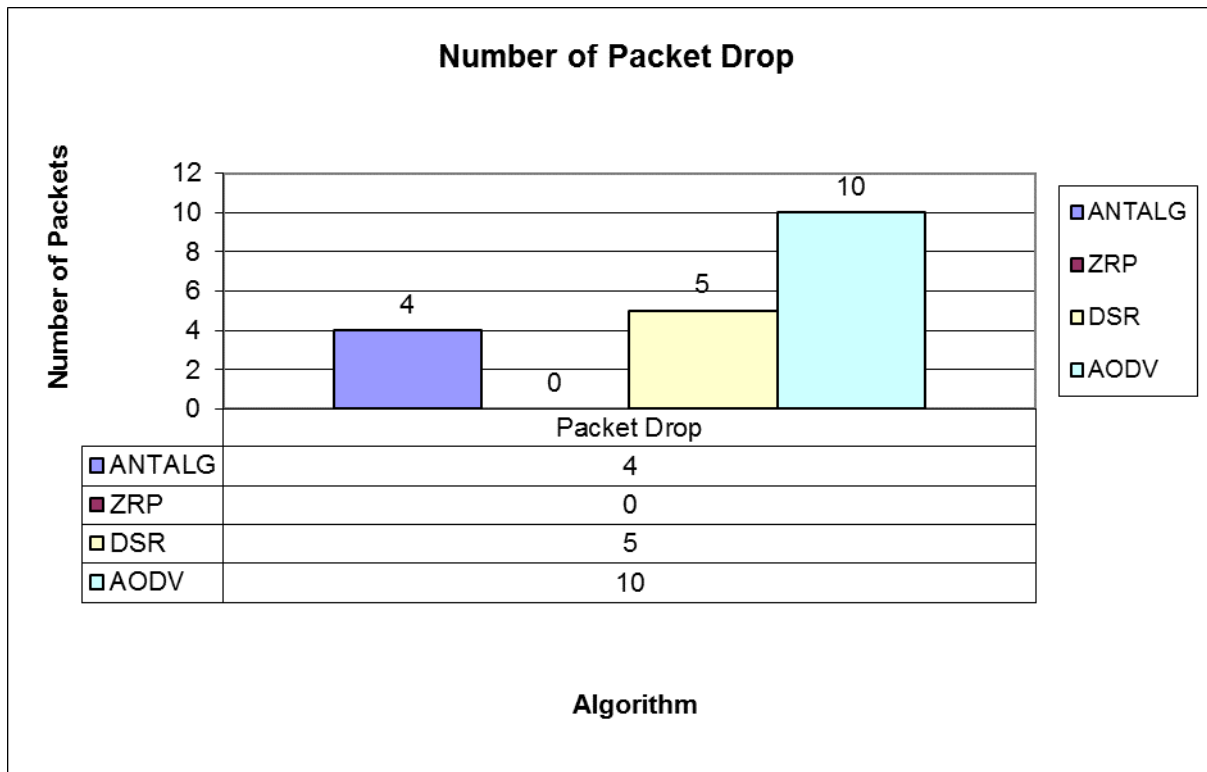


Figure 4.16: Number of Packets lost

The results shown by graphs for packet loss and delivery ratio in Figures 4.16 and 4.17 indicate that ANTALG is better than AODV and DSR algorithms but slightly lacks from ZRP algorithm. This is because ZRP algorithm is hybrid protocol and also uses a table driven approach which definitely favors these parameters. ANTALG is best in its category of reactive protocols.

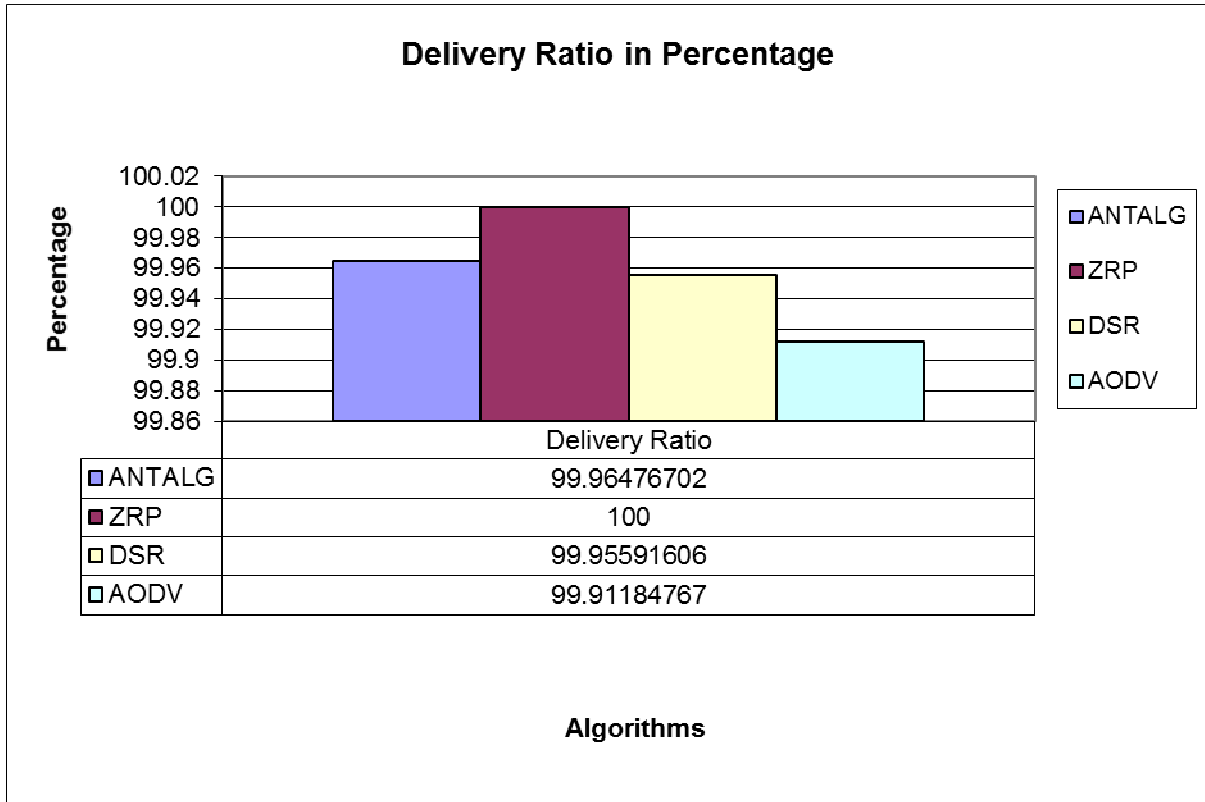


Figure 4.17: Delivery Ratio

4.5.3 Number of Packets Sent

With the help of this metric, one can easily find that which algorithm has the capacity for sending more packets. As a desirable property more number of packets must be send while communication.

The results of Figure 4.18 represent the total number of data packets send during the complete session of communication. During the simulation work, FTP packets were sent which fall under the class of TCP communication. TCP packets were used for study purpose because of their reliable nature in the process of communication. From the obtained graph it is quite clear that ANTALG has the capacity to send larger volume of packets as compared to the other protocols used for the purpose of comparison. So on the basis of this parameter ANTALG has better capability than AODV, DSR and ZRP. On visualizing the results it can be undoubtedly stated that ANTALG can send more number of data than the algorithms like AODV, DSR and ZRP.

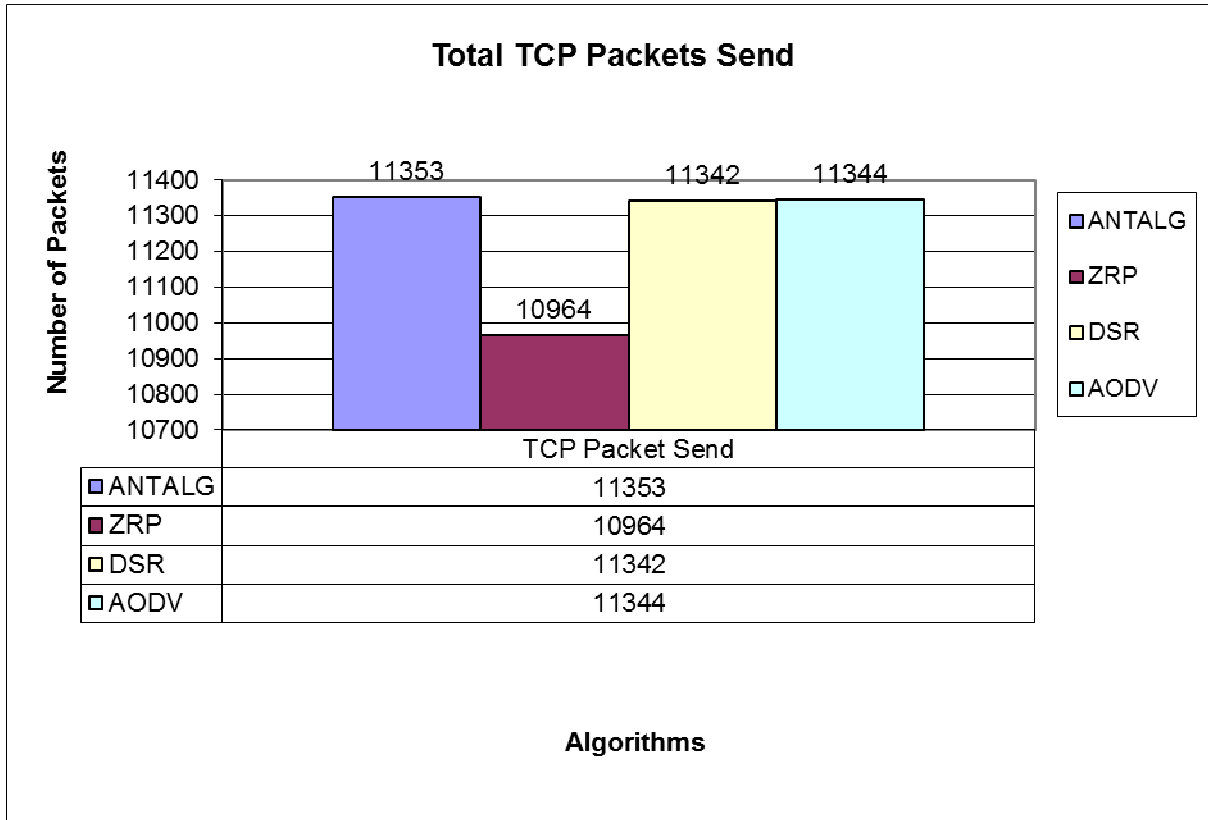


Figure 4.18: Total number of Data Packets Sent

4.5.4 Path Length

Packets move hop by hop during their travel from source node towards the destination node. It is desirable that a lesser number of hops should be covered to reach the ultimate end. This parameter is studied to estimate the better quality path in terms of a lesser number of hops taken by packets to reach the destination.

In this Section, the average number of hops is measured for a packet to successfully arrive at its destination. This is a relative measure of protocol efficiency, especially since the network is being optimized to minimize hop count.

The result of graph 4.19 show average path length in ANTALG is shorter in length when compared with ZRP, AODV and DSR. However ZRP has almost closer path length as compared to ANTALG but the study revealed that this is just because of the presence of the proactive nature of the ZRP along with its Hybrid approach.

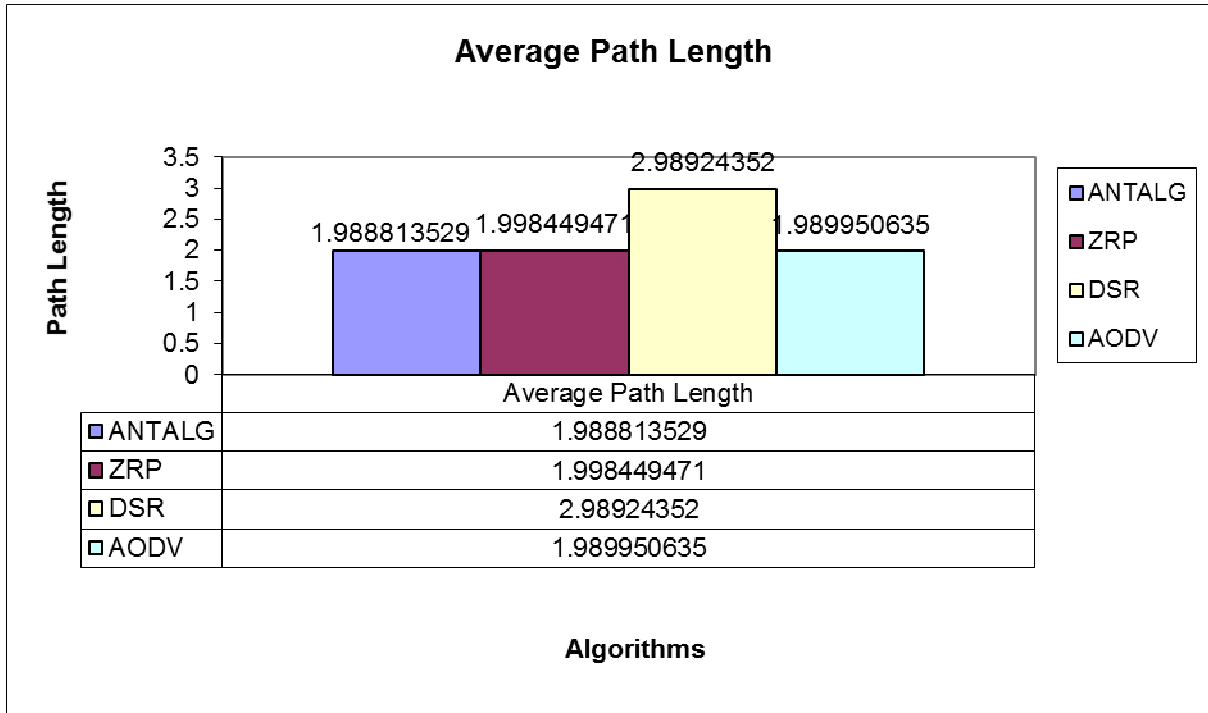


Figure 4.19: Average Path Length

4.5.5 End to End Delay

This Section presents a parameter which represents the time taken by a packet to reach the destination

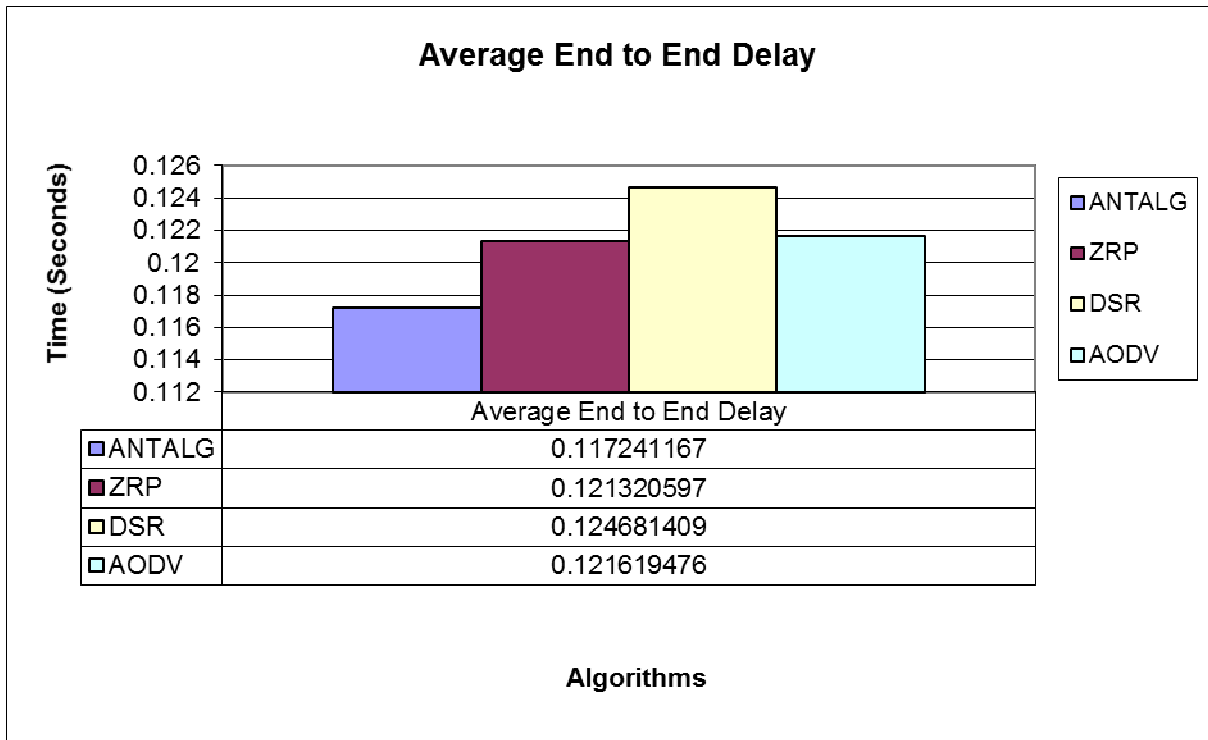


Figure 4.20: Average End to End Delay

End to End delay can be measured as the difference in the time the packet reach the destination for the time packet has started its journey. As general observation this parameter should be minimized. The results of end to end delay from Figure 4.20 specify that the ANTALG has better performance than the AODV, DSR and ZRP algorithms.

4.5.6 Jitter

Jitter is represented as the variation in the end to end delay observed by the packets from each other. This Section presents a metric which represent the consistency in the time taken by a packet to reach the destination with by observing the time of other packets.

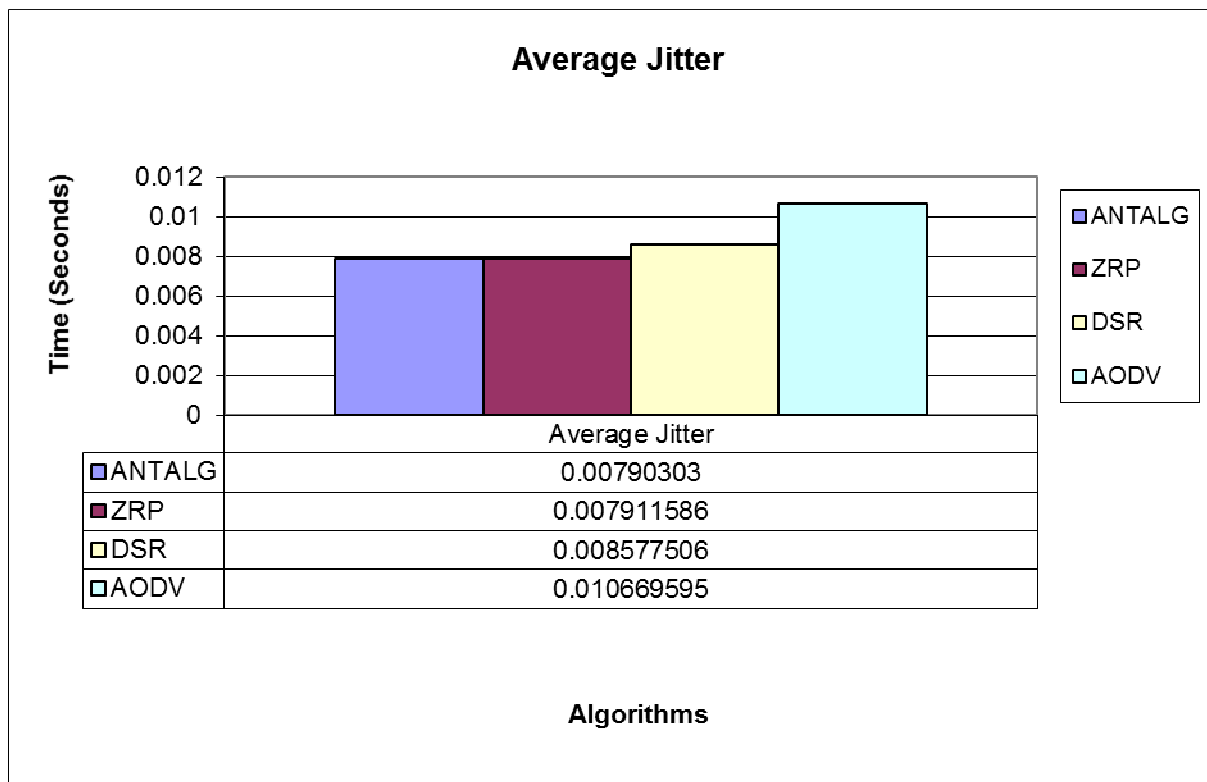


Figure 4.21: Average Jitter

Results of Figure 4.21 reveal that the packets in ANTALG observe lesser jitter as compared to other packets. However jitter observed by ZRP is closer to the value of ANTALG. This is due to that ZRP maintains the routing table all the time.

4.5.7 Conclusion at the End of Results from Stage Two

- I. Average jitter in ANTALG is better than all the other protocols.
- II. AODV, DSR & ZRP algorithms have a higher average of end to end delay as compared to ANTALG. This depicts that the packets reach the destination with less time in case of ANTALG.
- III. More number of data packets were sent in case of ANTALG during communication as compared to AODV, DSR & ZRP algorithms.
- IV. Lesser number of packets were dropped in ANTALG as compared to AODV & DSR. It points that the ANTALG performs better and has a better delivery ratio also. However ZRP has an even better delivery ratio as compared to ANTALG due to its more emphasis on the proactive nature than ANTALG.
- V. ANTALG traces better routes than the ZRP, AODV & DSR algorithm in terms of path length.
- VI. ANTALG has a better average throughput than AODV, DSR & ZRP.

4.6 Experimental Results of Third Stage of Simulation Work

In the stage three of implementation, new parameter as orientation factor was added to the tested work of stage two. The orientation factor helps the search packets or data packets to favorably move towards the destination by improving the pheromone concentration on the promising paths. However, it reduces the pheromone concentration on the paths which are not on the way towards destination node. After making amendments the algorithm is assigned a new name as OANTALG, i.e., Orientation based ANT ALGorithm in MANETs. The running screenshot of network in stage three is shown in Figure 4.22. The small wireless network of stage three comprises of 40 nodes. The simulation work is done in Linux environment. In this stage the simulation results obtained from the execution of OANTALG is compared with the simulation results obtained from the execution of traditional famous protocols like AODV, DSR and with another ant based protocol of its category HOPNET. For the sake of similarity, the same scenario is employed in all the protocols used for the purpose of comparison. The experimental setup was executed according to the parameters as already discussed

in the Table 4.2. As per the assumptions stated in Section 4.3, all the links are bi-directional in nature, i.e., the topology uses the full duplex channels for communication among the nodes.

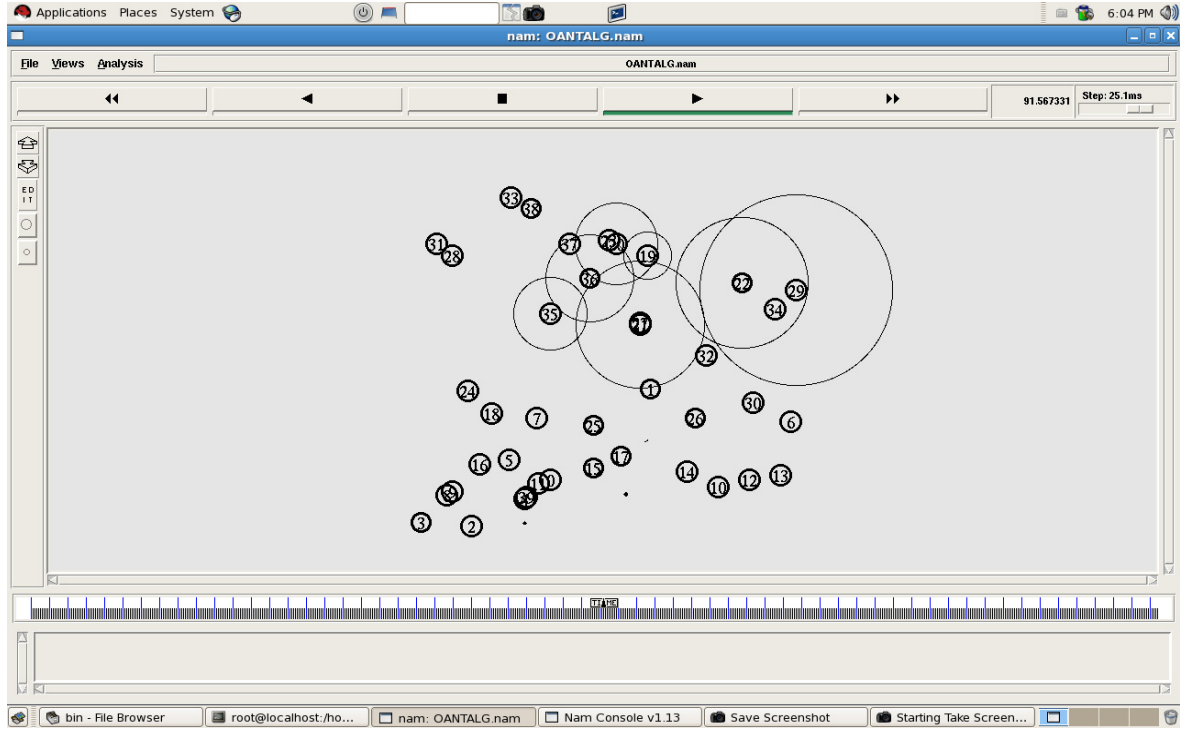


Figure 4.22: Running screenshot for wireless network running using OANTALG

4.6.1 Effect on Throughput

Figures 4.23 and 4.24 show the effect of the proposed scheme on throughput in bytes per second measured after every 20 seconds and average throughput as bytes per second after the end of communication session respectively. The results obtained show that the proposed protocol OANTALG has better performance as compared to AODV, DSR and HOPNET algorithms. OANTALG is a hybrid algorithm. Initially, it is reactive in nature, but when there is need to send data then the nature of the algorithm partially changes from reactive to proactive. Due to hybrid approach, the throughput of the proposed algorithm is better than AODV and DSR. The simulation work shows that it also outperforms another hybrid protocol HOPNET. This is because HOPNET infers hybrid approach since the initialization of the topology, but OANTALG becomes hybrid when there is a need of data transmission. So hello and other types of control packets in OANTALG are lesser than those in HOPNET. As a result, a large number of

data packets can take advantage of available bandwidth. That is the reason why the throughput of the proposed algorithm is better than the throughput of AODV, DSR and HOPNET. Specifically, throughput in OANTALG is 1.92 times better than HOPNET, 1.02 times better than AODV and 1.44 times better than DSR.

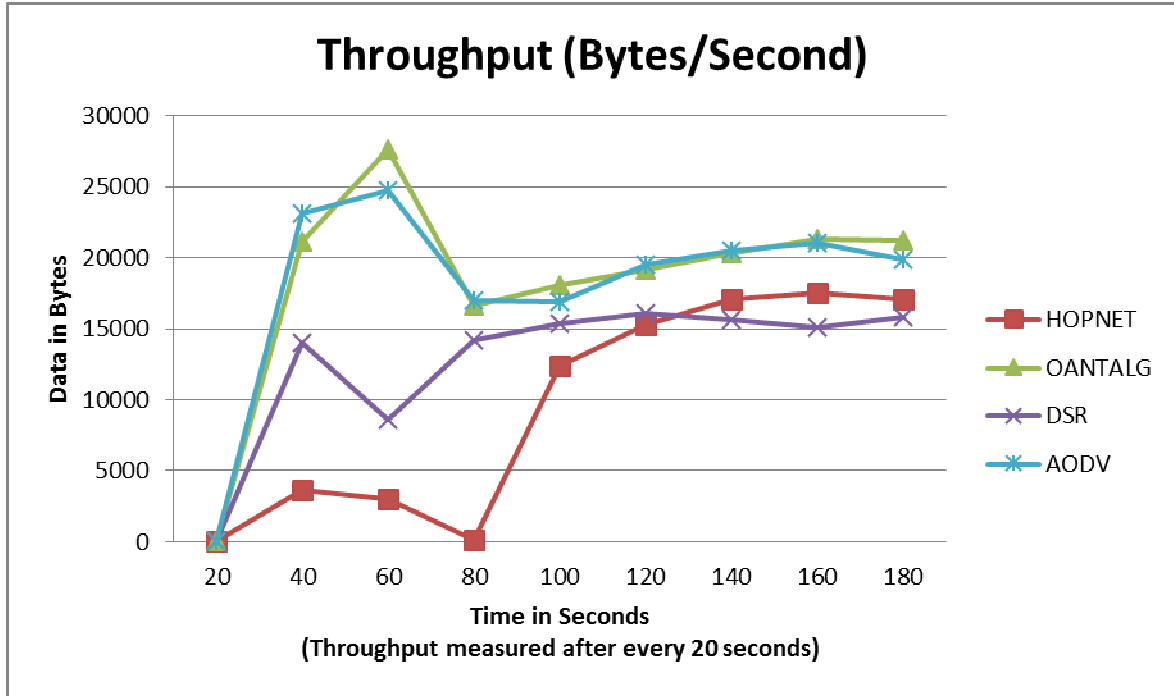


Figure 4.23: Variation in throughput measured in Bytes/Second after every 20 seconds

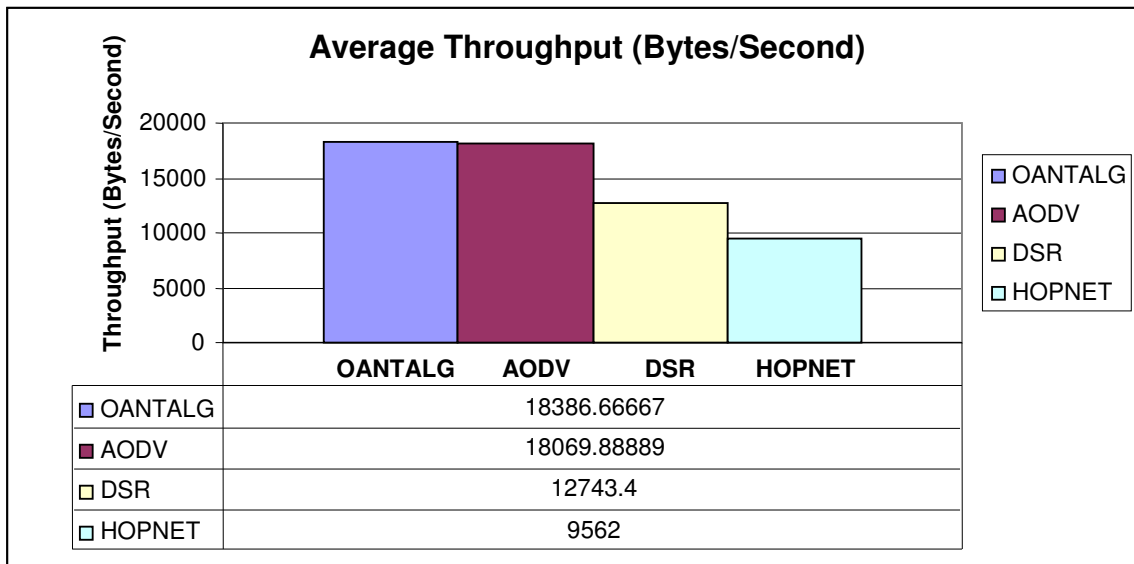


Figure 4.24: Average throughput after complete time

4.6.2 Effect on Packet Drop and Packet Delivery Ratio

Figures 4.25, 4.26 show the effect of the proposed scheme on packet's drop and packet delivery ratio. The results obtained show that the proposed OANTALG is better than AODV and DSR algorithms, but slightly less than HOPNET algorithm. This is because HOPNET algorithm uses table driven approach within zones, which definitely favors these parameters. As the proposed algorithm uses a hybrid approach, so fewer number of packets are dropped in OANTALG than AODV and DSR because of the availability of the pheromone table (similar to routing table in traditional protocols). However, HOPNET emphasizes more on proactive approach within zones; so less number of packets are dropped in this case. Consequently delivery ratio is better in OANTALG when weighed against AODV & DSR and somewhat lesser than HOPNET. From Figure 4.26, it can be concluded that the delivery ratio in OANTALG is 1.003 and 1.011 times better than the delivery ratio in AODV and DSR respectively.

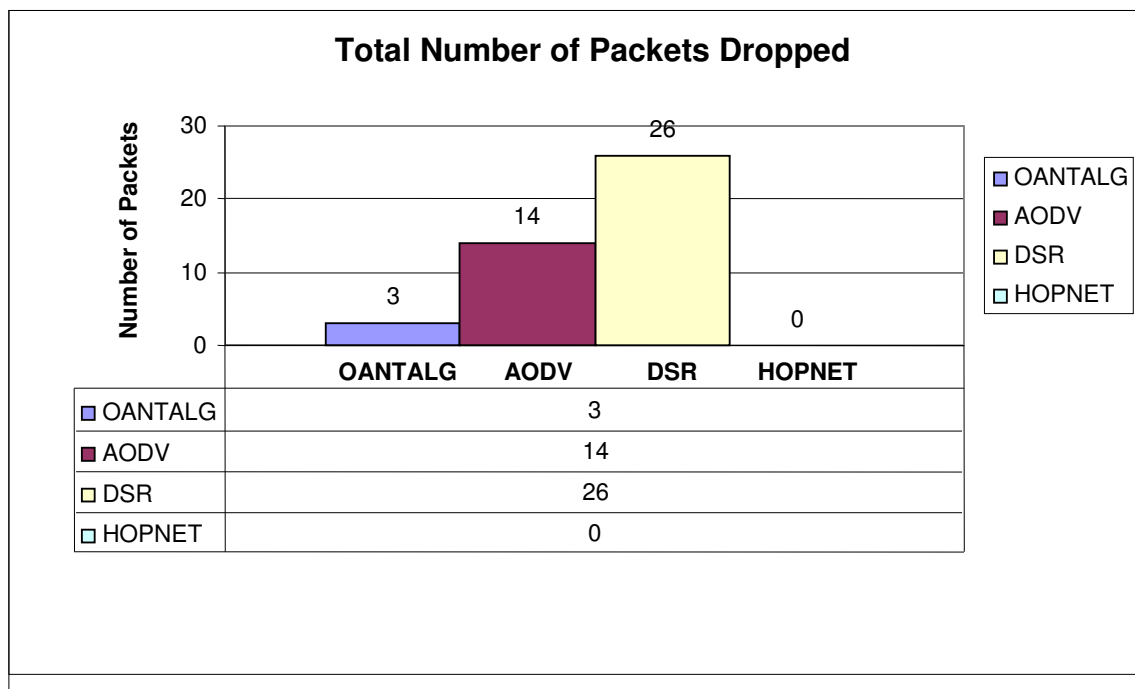


Figure 4.25: Total Number of Packets Drop

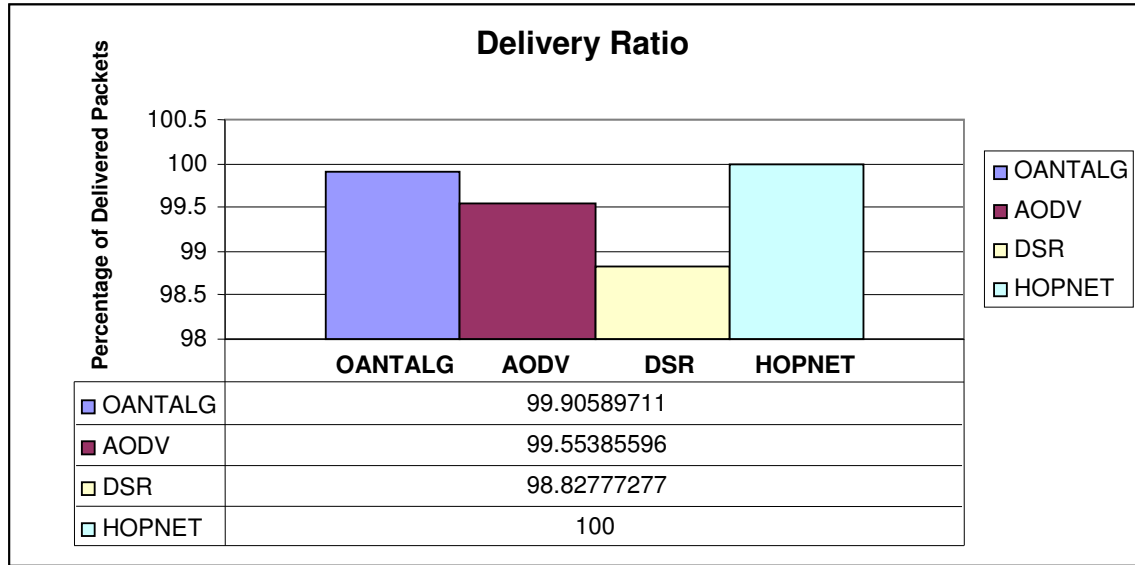


Figure 4.26: Delivery Ratio

4.6.3 Effect on Data Packets Sent

Figure 4.27 shows the effect of proposed the scheme on data packets sent. The results obtained show that the proposed OANTALG sends more number of data packets than the algorithms like AODV, DSR and HOPNET. The reason for sending higher packet rate is as explained above. OANTALG can approximately send 1.6% more packets send by AODV, 30.4% more packets than DSR and 37.9% more packets than HOPNET.

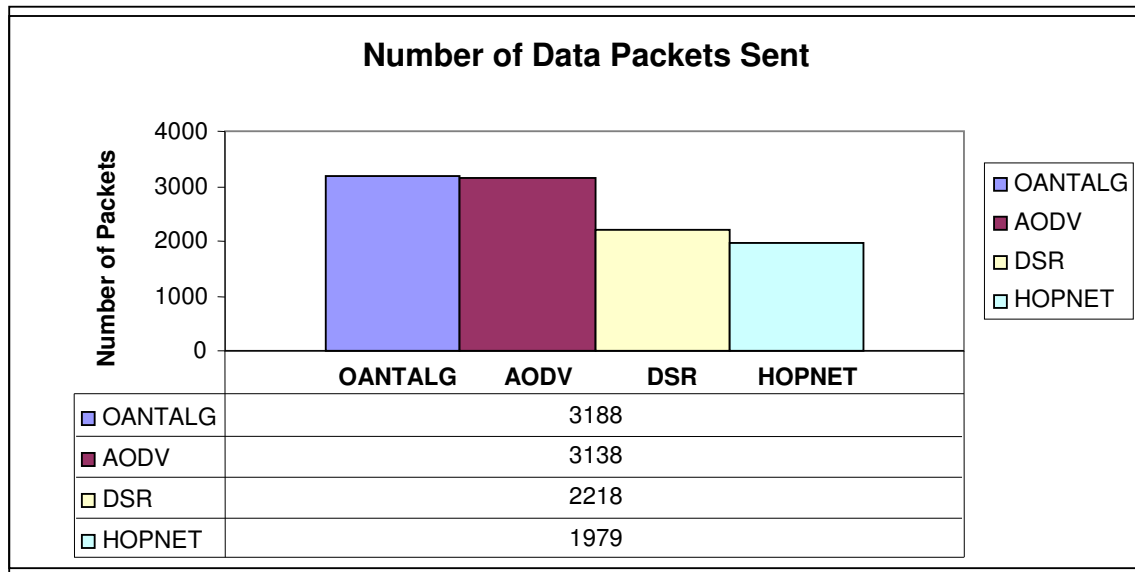


Figure 4.27: Total TCP Packets sent by each Protocol

4.6.4 Effect on Path Length

Figure 4.28 shows the effect of the proposed algorithm on the length of the path in terms of number of hops. The results obtained show that the proposed OANTALG has shorter path length when compared to AODV and DSR. However, HOPNET has lesser path length as compared to OANTALG, but the study revealed that this was just because of the more proactive nature of HOPNET. Because of the accessibility of pheromone table as described in Section 3.2; OANTALG can choose the optimal path as the route information. Pheromone also provides the positive feedback on the promising paths and negative feedback on the poor or congested paths. Average path length in OANTALG is 2.4 times more than HOPNET. However, average path length in AODV and DSR is 1.02 and 1.62 times more than the OANTALG.

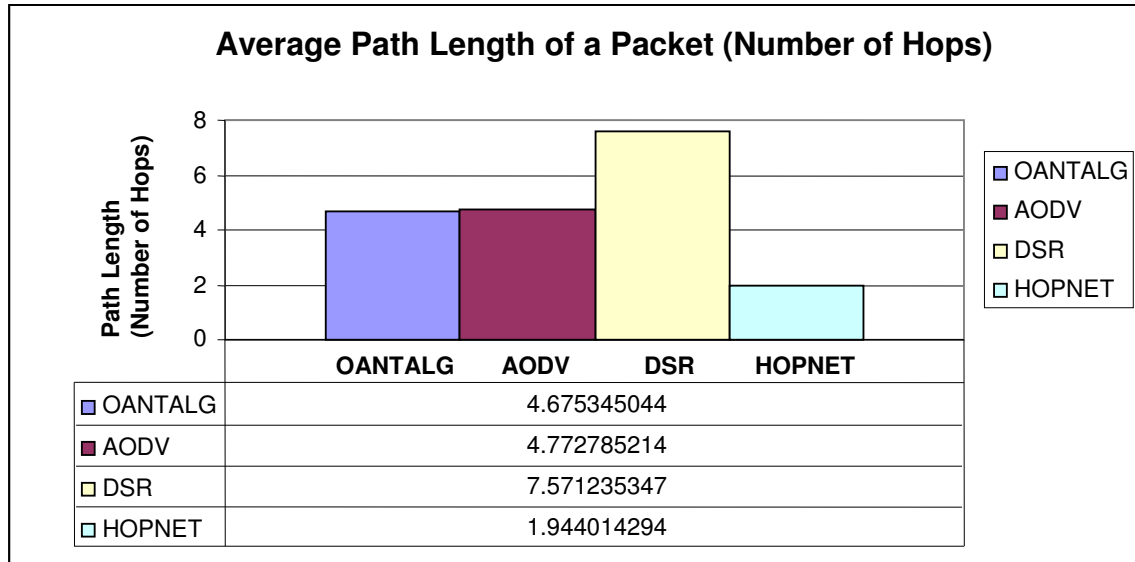


Figure 4.28: Average Path Length (Number of hops between source and destination)

4.6.5 Effect on End- to- End Delay

Figures 4.29 and 4.30 show the effect of the proposed scheme on End to End delay. The results obtained from Figure 4.30 show that the proposed scheme is better than the DSR with respect to Average End to End delay. However, AODV performs slightly better than OANTALG just because of its beckoning nature. Also, pure hybrid nature of HOPNET enables it with 26% less average end to end delay. Average end to end delay in DSR is 67% more than its value from OANTALG.

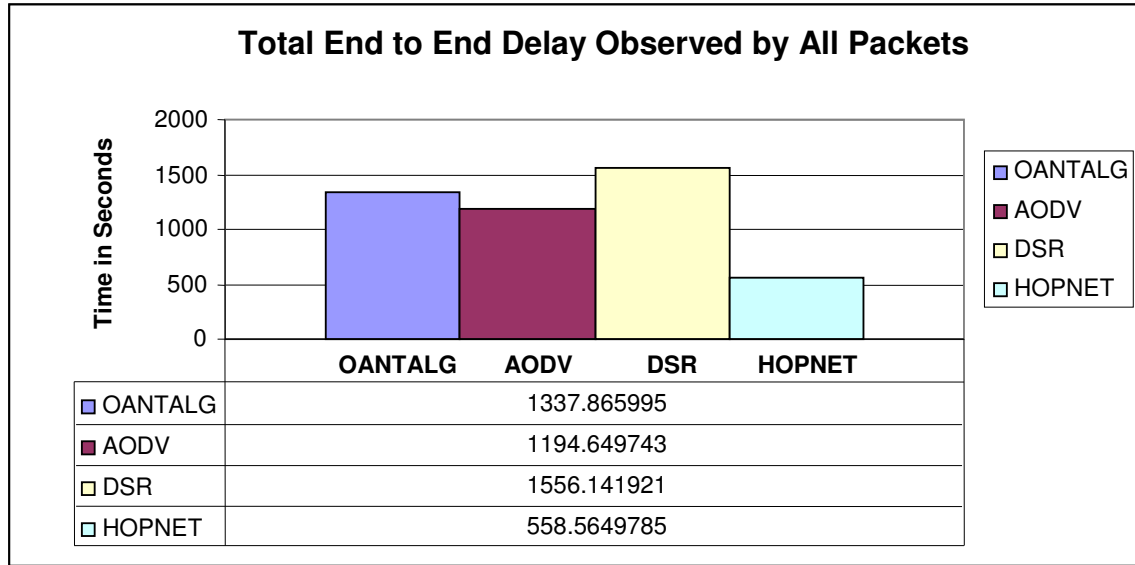


Figure 4.29: Total End- to- End Delay between the source and destination

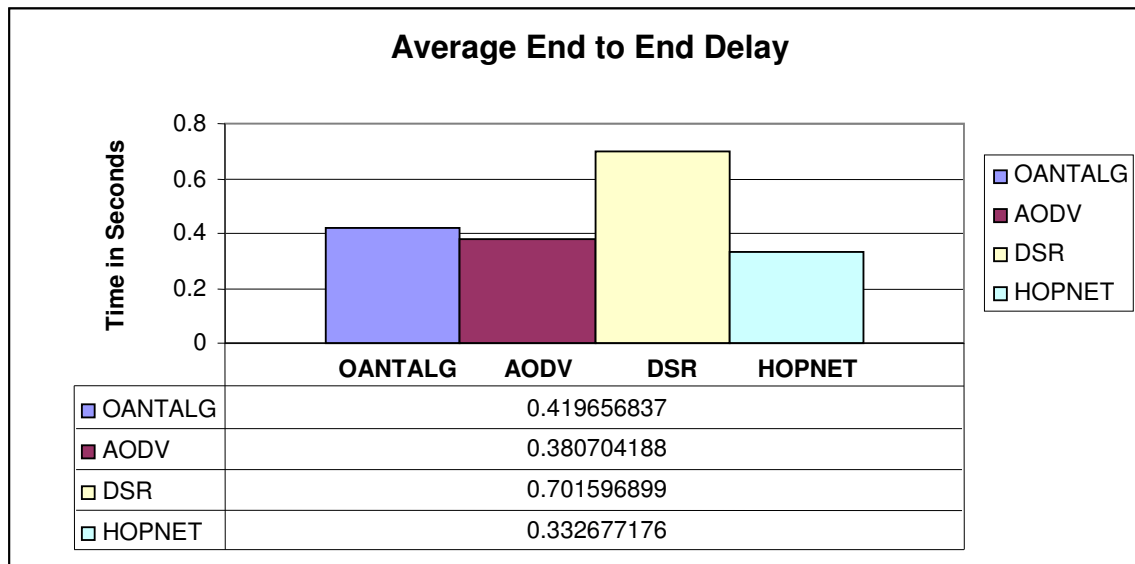


Figure 4.30: Average End- to- End Delay between all sources and destinations

4.6.6 Effect on Jitter

Figures 4.31, 4.32 show the effect of the proposed scheme on total and average jitter. The results obtained show that the proposed scheme has less jitter as compared to traditional and ant based routing MANET's algorithms. Jitter is measured as the fluctuation of end-to -end delay from one packet to the other packet. Results of Figure 4.32 show that the average jitter in OANTALG is better than all other approaches in its

category. Average jitter in OANTALG is 1.42, 3.57 and 1.26 times better than AODV, DSR & HOPNET routing algorithms respectively.

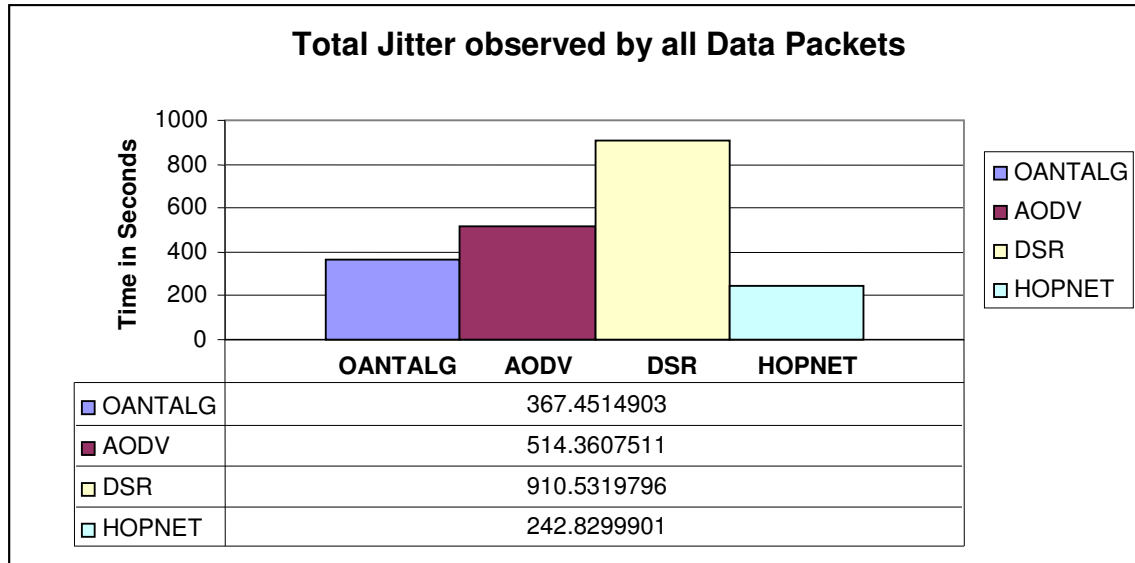


Figure 4.31: Total Jitter observed between the source and destination

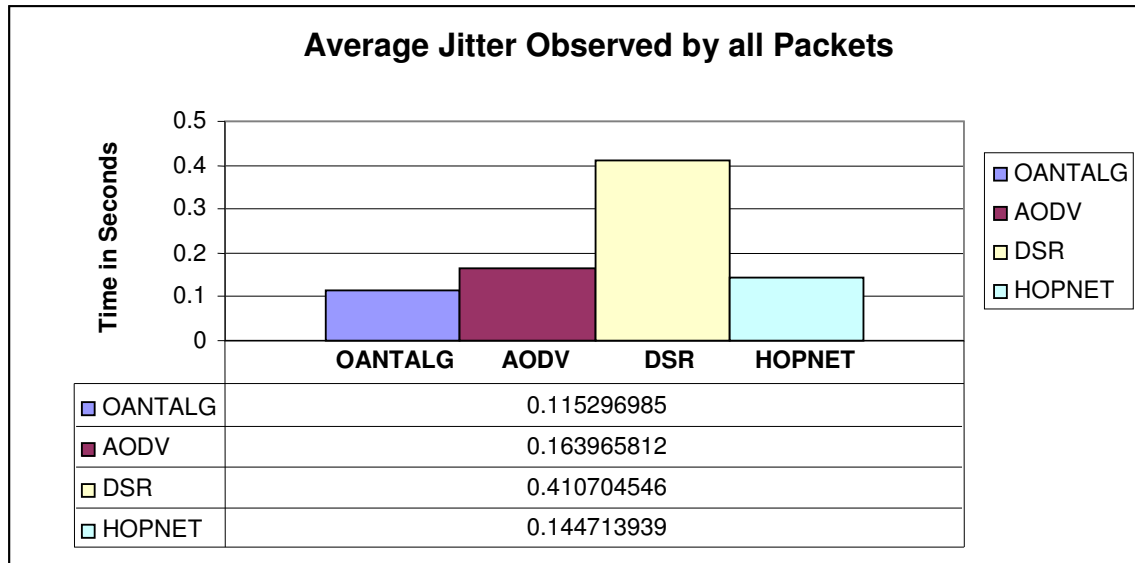


Figure 4.32: Average Jitter between all sources and destinations

4.6.7 Conclusion at the End of Results from Stage Three

- I. OANTALG has better throughput than AODV, DSR & HOPNET.
- II. Lesser number of packets were dropped in OANTALG as compared to AODV & DSR. It indicates that the OANTALG can perform better in terms of delivery

ratio. Although, HOPNET has even better delivery ratio as compared to OANTALG due to its more proactive nature than OANTALG.

- III. Average jitter in OANTALG is better than all the other protocols.
- IV. DSR has higher average of end to end delay as compared to OANTALG. However, OANTALG slightly lacks from AODV and HOPNET on comparing it on the basis of average and to end delay.
- V. More number of data packets was sent in case of OANTALG during communication as compared to AODV, DSR & HOPNET.
- VI. OANTALG traces better routes than the AODV & DSR in terms of path length. But it lacks from HOPNET on this matter due to presence of proactive factor in HOPNET all the times.

This work has addressed the problem of routing in MANETs using ACO and has proposed a new Orientation based ANT ALgorithm (OANTALG) in MANETs. The designed algorithm has used the concept of orientation factor and highest pheromone value to avoid local minima problem which exists in most of the previous solutions. OANTALG algorithm operates using reinforcement learning for optimal routing behavior in MANETs. Table 4.19 shows a relative comparison of various traditional and ant based algorithm in contrast to the proposed algorithm with respect to various parameters. The proposed algorithm is better than the other algorithms considered for comparison like AODV, DSR and HOPNET. On the whole, there are six major metrics/factors based on which OANTALG is compared to AODV, DSR and HOPNET. Table 7.1 shown below, refers to the Figures of comparison.

Table 4.19: Relative comparison of all algorithms with respect to various parameters

Algorithms	OANTALG	AODV	DSR	HOPNET
No of TCP Packets Sent	3188	3138	2218	1979
Average Throughput (Bytes/Second)	18386.67	18069.89	12743.4	9562
Packets Dropped	3	14	26	0
Average End to End Delay	0.419657	0.380704	0.701597	0.332677
Average Jitter	0.115297	0.163966	0.410705	0.144714
Average Path Length	4.675345	4.772785	7.571235	1.944014

After evaluating Table 4.19, it can be clearly stated that OANTALG outperforms AODV on five factors out of the six parameters used for comparison. It outperforms DSR on all the six factors presented in the Table 4.19. AODV has proven to be better than OANTALG on the basis of only one parameter, but it is better than HOPNET on the basis of two parameters.

Table 4.20: Percentage Improvement of OANTALG over other Algorithms

Algorithms	AODV	DSR	HOPNET
No of TCP Packets Sent	1.6% more	30.4% more	37.9% more
Average Throughput (Bytes/Second)	1.79% more	30.7% more	48% more
Packets Dropped	5 times more packets are dropped	9 times more packets are dropped	-----
Average End to End Delay	----	67% more	-----
Average Jitter	42% less	256% less	26.3 % less
Average Path Length	2.14% more	62% more	

HOPNET has performed the worst when compared on the basis of average throughput and numbers of packets sent, but OANTALG has never shown worst performance when compared on the above six parameters. In summary, it can be concluded that OANTALG is better than the other algorithms used for the comparison and can be used in wide areas of applications as compared to their counterparts. The percentages of improvement in the proposed OANTALG algorithm with respect to the other algorithms are represented in Table 4.20. The results obtained show that OANTALG has performed better than the other algorithms of its category with respect to various performance evaluation parameters.

5. Conclusion and Future Work

MANETs are highly dynamic and distributed in nature. As a special type of network, MANETs have received the attention from a large community of researchers in recent years. Researchers are actively working on many research projects concerned with MANETs. With the emergence of wireless networking and developments in MANETs, new tactics in routing algorithms are required. The research indicates that the algorithms inspired by SI and nature, especially ACO are getting a lot of interest in solving combinatorial problems. Routing in MANETs is also a type of combinatorial problem where there is always a need of an optimized solution. Even a slight improvement in these solutions can lead to higher company profits.

The thesis presented the novel nature inspired routing algorithms derived from the ant colonies with orientation factor guiding the data and control packets progressing towards the destination to move in a right direction. This work explored the potential of the nature-inspired networking in developing the solutions to the routing problem in MANETs. The agents in ACO inspired routing algorithms communicate indirectly through the process called stigmergy. The agents provide positive feedback to the favorable paths by depositing more pheromone on the better links. However, the agents also convey negative feedback through evaporation and aging mechanisms, which avoid stagnation.

This thesis work tried to cover the state-of-the-art studies about ACO algorithm and its application to routing problems in MANETs. The simulation results and the discussion presented in the previous chapter lead us to understand the various properties of the proposed algorithms. The newly designed and developed architecture used the *concept of orientation to guide floating packets in right direction* and *highest pheromone value to avoid local minima problem*, which exists in most of the previous solutions. The new algorithms operate using reinforcement learning for optimal routing behavior in MANETs. The results obtained show that the new algorithms are better than the other algorithms with respect to various performance evaluation parameters.

As discussed earlier that the simulation work passes from three different stages. Each stage crosses a certain level of work in order to complete the overall target.

- I. In the first stage of development, the results were compared with the well known routing protocol AntNet in a wired network. The results obtained were found encouraging because of better routing table solutions in the terms of shorter path and better concentrated pheromone values as perceived in Section 4.4.
- II. After obtaining these wonderful results the implemented data structure and packet formats were transferred into MANET environment and compared with the traditional algorithms like AODV, DSR and ZRP. Again the results obtained by executing ANTALG were found very encouraging which can also be viewed in Section 4.5.
- III. In the last phase of development, an orientation factor was included with the implemented scheme in MANET. The orientation factor helped the control and data packets to flow into appropriate direction towards destination. OANTALG has shown better performance than HOPNET, AODV and DSR protocols as exposed by result of tables 4.19 and 4.20 on the basis of various metrics during comparison.

It has been a great learning experience to study and develop such a result oriented ant based protocol. At the last, it can be stated that thesis work was a success in developing the new MANET routing algorithms containing orientation factor which simulates the ACO behavior with highest pheromone value and elucidate the best understanding of the routing problem.

In the future, some of the security issues and stagnation avoidance technique can be integrated with this method to increase the efficiency of MANETs with respect to security and routing.

References

- [1]. P. Krishnamurthy, K. Pahlavan, “Principles of Wireless Networks: A Unified Approach”, Pearson with Prentice Hall Publication, edition 3rd, ISBN 0-13-093003-2, 2001.
- [2]. D. Bertsekas, R. Galager, “Data Networks”, Prentice Hall, New Jersey, Edition 2nd, Prentice Hall, ISBN: 0132009161, 1992
- [3]. C. K. Toh, “Ad Hoc Mobile Wireless Networks”, Protocols and Systems, Prentice Hall, PTR, ISBN: 0130078174, 2002.
- [4]. S. R. Das, C. E. Perkins, E. M. Royer, “Performance Comparison of Two On-Demand Routing Protocols for Ad-hoc Networks”, Proceedings of IEEE Computer and Communications Societies (INFOCOM '00), Tel Aviv, Israel, Volume 1, pages 3-12, 2000.
- [5]. M. Abolhasan, T. Wysock, E. Dutkiewicz, “A Review of Routing Protocols for Mobile Ad-hoc Networks”, International Journal Of Ad-hoc Networks, Volume 2, Issue 1, pages 1-22, 2004.
- [6]. T. Camp, J. Boleng, V. Davies, “A Survey of Mobility Models for Ad-hoc Networks Research”, Journal of Wireless communication & Mobile Computing (WCMC), Volume 2, Issue 5, pages 483–502, 2002.
- [7]. N. K. Cauvery, K. V. Viswanatha, “Enhanced Ant Colony Based Algorithm for Routing in Mobile Ad-hoc Network”, AINN 2008: International Conference on Artificial Intelligence and Neural Networks, Bangkok, Thailand, Proceedings of World Academy of Science, Engineering and Technology, Volume 36, pages 30-35, 2008.
- [8]. B. Bhargava, X. Wu, Y. Lu, W. Wang, "Integrating Heterogeneous Wireless Technologies: A Cellular-Assisted Mobile Ad-hoc Network", Mobile Networks and Applications: Special Issue on Integration of Heterogeneous Wireless Technologies, Volume 9, pages 393-408, 2004.
- [9]. T. C. Homero, P. A. S. Khan, P. J. C. Ramírez, "Accurate Modelling of VoIP Traffic QoS Parameters in Current and Future Networks with Multifractal and Markov Models", Mathematical and Computer Modelling Journal, Volume 57, Issues 11-12, pages 2832-2845, 2013.

- [10]. V. Seethalakshmi, G. Mohan Kumar, "A Survey of Routing Protocols in Mobile Ad-hoc Network", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, pages 1340-1346, 2013.
- [11]. G. V. Kumar, Y. V. Reddy, M. Nagendra "Current Research Work on Routing Protocols for MANET: A Literature Survey", International Journal on Computer Science and Engineering (IJCSE), Volume 2, Issue 3, pages 706-713, 2010.
- [12]. Y. Lu, W. Wang, Y. Zhong, B. Bhargava, "Study of Distance Vector Routing Protocols for Mobile Ad Hoc Networks." in the proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom'2003), Texas, March 2003.
- [13]. J. Yang, W. Zhao, M. Xu, B. Xu, "A Multipath Routing Protocol Based on Clustering and Ant Colony Optimization for Wireless Sensor Networks", Journal of International Journal of Computer Network and Information Security, pages 49-59, 2009.
- [14]. V. K. Verma, R. C. Joshi, B. Xei, D. P. Aggarwal, "Combating the Bloated State Problem in Mobile Agents Based Network Monitoring Applications", Computer Networks, Volume 52, Issue 17, pages 3218-3228, 2008.
- [15]. L. K. Awasthi, M. Mishra, R. C. Joshi, "A Weighted Check Pointing Protocol for Mobile Distributed Systems", International Journal of Ad-hoc and Ubiquitous Computing (IJAHUC), Volume 5, Issue 3, pages 137-149, 2010.
- [16]. R. Kumar, M. Dave, "A Framework for Handling Local Broadcast Storm Using Probabilistic Data Aggregation in VANET", Wireless Personal Communications, Volume 72, Issue 1, pages 315-341, 2013.
- [17]. G. S. Pavani, L. G. Zuliani, H. Waldman, M. Magalhaes, "Distributed Approaches For Impairment-Aware Routing and Wavelength Assignment Algorithms in GMPLS Networks", Computer Networks, Volume 52, pages 1905–1915, 2008.
- [18]. A. Habib, M. Khan, B. Bhargava, "Edge-to-Edge Measurement-based Distributed Network Monitoring", Computer Networks, Volume 44, Issue 2, pages 211-233, 2004.
- [19]. W. Wang, Y. Lu, B. Bhargava, "On Security Study of Two Distance Vector Routing Protocols for Mobile Ad Hoc Networks ." in the proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom'2003), Texas, March 2003.

- [20]. G. Singh, A. Kaur, "Using Random Way Point Mobility Model to Compare The Performance Of Reactive And Proactive Routing Protocols Of MANETs in Ns-2", In the proceedings of National Conference on Futuristic Trends in Computing, Communication & Information System – FTTCCIS and Yamuna Journal of Technology & Business Research - Volume 3. Issue (1-2), pages 5.1-5.7, 2013.
- [21]. A. Panisson, A. Barrat, C. Cattuto, W. V. D. Broeck, G. Ruffo, R. Schifanella, "On the Dynamics Of Human Proximity For Data Diffusion In Ad-hoc Networks", Ad-hoc Network, Volume 10, Issue 8, pages 1532-43, 2011.
- [22]. D. Kaur, N. Kumar, "Comparative Analysis of AODV, OLSR, TORA, DSR and DSDV Routing Protocols in Mobile Ad-hoc Networks", International Journal of Computer Network and Information Security, Volume 3, pages 39-46, 2013.
- [23]. G. S. Aujla, S. S. Kang, "Comprehensive Evaluation of AODV, DSR, GRP, OLSR and TORA Routing Protocols with Varying Number of nodes and Traffic Applications over MANETs", IOSR International Journal of Computer Engineering (IOSR-JCE), Volume 9, Issue 3, Pages 54 -61, 2013.
- [24]. A. Kumar, R. Singh, "Mobile Ad-hoc Networks Routing Optimization Techniques Using Swarm Intelligence", International Journal of Research in IT and Management, Volume 1, Issue 4, pages 13-32, 2011.
- [25]. M. Parsapoor, U. Bilstrup, "Ant Colony Optimization for Channel Assignment Problem in a Clustered Mobile Ad-hoc network", Advances in Swarm Intelligence Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, Volume 7928, pages 314-322, 2013.
- [26]. B. Roy, S. Banik, P. Dey, S. Sanyal, N. Chaki, "Ant Colony based Routing for Mobile Ad-hoc Networks towards Improved Quality of Services", Journal of Emerging Trends in Computing and Information Sciences, Volume 3, Issue 1, pages 10-14, 2011.
- [27]. B. Verma, S. Kulkarni, "Fuzzy Logic Based Interpretation and Fusion of Colour Queries", Fuzzy Sets and Systems, Volume 147, Number 1, pages 99-118, 2004.
- [28]. S. L. O. B. Correia, J. Celistino Junior, O. Cherkaoui, "Mobility-aware Ant Colony Optimization Routing for Vehicular Ad-hoc Networks", IEEE WCNC 2011 – Network, pages 1125-1130, 2011.
- [29]. J. Jain, R. Gupta, T. K. Bandhopadhyay, "Ant Colony Algorithm in MANET-Local Link Repairing of AODV", proceedings of 3rd IEE International

- Conference on Electronics Computer Technology (ICECT), pages 270-273, 2011.
- [30]. V. Kumar, P. Balasubramanie, "Ant Colony Optimization Using Hierarchical Clustering in Mobile Ad-hoc Networks", *European Journal of Scientific Research*, Volume 61, Number 4, pages 549-560, 2011.
- [31]. J. C. Bansal, H. Sharma, K. V. Arya, A. Nagar, "Memetic Search in Artificial Bee Colony Algorithm". *Journal of Soft Computing*, Volume 17(10), pages 1911-1928, 2013.
- [32]. M. Dorigo, "Optimization, Learning and Natural Algorithms", PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1992
- [33]. F. Dressler, O. B. Akan, "A Survey on Bio-inspired Networking", *International Journal of Computer Networks*, Volume 54, Issue 6, pages 881-900, 2010.
- [34]. F. Ducatelle, G. D. Caro, L. M. Gambardella, "Ant Agents for Hybrid Multipath Routing in Mobile Ad-hoc Networks", In the proceedings of Second Annual Conference on Wireless On-demand Network Systems and Services, 2005. (WONS 2005), Manno-Lugano, Switzerland, pages 44 - 53, 2005.
- [35]. F. Correia, T. Vazal, V. J. Lobo, "Models for Pheromone Evaluation In Ant Systems for Mobile Ad-hoc Networks", *First International Conference on Emerging Network Intelligence-2009*, Sliema, Malta, pages 85-90, 2009.
- [36]. M. Lin, X. Yubin, M. Weixiao, "An Ant Colony Based Congestion Elusion Routing Scheme for MANET", in the proceedings of IEEE Globecom 2011.
- [37]. M. Dorigo, M. Birattari, T. Stutzle, "Ant Colony Optimization : Artificial Ants as a Computational Intelligence Technique", *IEEE Computational Intelligence Magazine*, Volume 1, Issue 4, pages 28-39, 2006.
- [38]. B. Bullnheimer, R. F. Hartl, C. Strauss, "A New Rank-Based Version Of The Ant System: A Computational Study", *Central European Journal for Operations Research and Economics*, Volume 7, Issue 1, pages 25-38, 1999.
- [39]. A. Sampath, T. C, Sabu M. Thampi, "An ACO Algorithm for Effective Cluster Head Selection", *Journal of Advances in Information Technology*, Volume 2, Number 1, pages 50-56, 2011.
- [40]. H. Al-Zurba, T. Landolsi, M. Hassan, F. Abdelaziz, "On the Suitability of using Ant Colony Optimization for Routing Multimedia Content Over Wireless Sensor Networks", *International journal on applications of graph theory in wireless Ad-*

- hoc networks and sensor networks (GRAPH-HOC) Volume 3, Issue 2, pages 15-35, 2011.
- [41]. R. Singh, D. K. Singh, L. Kumar, "Swarm intelligence based approach for routing in mobile Ad-hoc networks", International Journal of Science and Technology Education Research, Volume 1, Issue 7, pages 147 – 153, 2010.
- [42]. S. Marwaha, J. Indulska, M. Portmann, "Biologically Inspired Ant-Based Routing In Mobile Ad-hoc Networks (MANET): A Survey" Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, Queensland Res. Lab. (QRL), Univ. of Queensland, Brisbane, QLD, Australia, pages 12-15, 2009.
- [43]. S. B. Wankhade, M. S. Ali, "Ant Based Techniques for QoS Routing in Mobile Ad-hoc Network: An Overview", International Journal - Advanced Networking and Applications Volume 3, Issue 2, Pages 1094-1097, 2011.
- [44]. P. Deepalakshmi, S. Radhakrishnan, "An Ant Colony Based Multi Objective Approach to Source-Initiated QoS Multicasting Method for Ad-hoc Networks", International Journal - Advance. Soft Computing Applications, Volume 3, Issue 2, pages 1-18, 2011.
- [45]. T. White, "Swarm Intelligence And Problem Solving In Telecommunications". Canadian Artificial Intelligence Magazine, Volume 41, Pages 14-16, 1997.
- [46]. F. Correia, T. Vazao, "Simple Ant Routing Algorithm Strategies for a (Multipurpose) MANET Model", International Journal of Ad-hoc Network, Volume 8, Issue 8, Pages 810-823, 2010.
- [47]. S. M. Senouci, M. Naimi, "New Routing For Balanced Energy Consumption In Mobile Ad-hoc Networks", In PE-WASUN '05: Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks. New York, NY, USA: ACM Press, Montreal, Quebec, Canada, pages 238–241, 2005.
- [48]. R. Baskaran, P. V. Paul, P. Dhavachelvan, "Ant Colony Optimization for Data Cache Technique in MANET", Proceedings of International Conference in Advances in Computing & Advances in Intelligent Systems and Computing, Springer India, Volume 174, pages 873-878, 2013.
- [49]. S. Kaur, R. S. Sawhney, R. Vohra, "MANET Link Performance Parameters using Ant Colony Optimization Approach", International Journal of Computer Applications, Volume 47, Number 8, pages 40–45, 2012.

-
- [50]. B. Roy, S. Banik, P. Dey, N. Chaki, S. Sanyal, "Ant Colony Based Routing for Mobile Ad-hoc Networks Towards Improved Quality of Services", *Journal of Emerging Trends in Computing and Information Sciences*, Volume 3, Number 1, pages 10-14, 2012.
 - [51]. R. Schoonderwoerd, O. Holland, J. Bruten, "Ant-Based Load Balancing in Telecommunications Networks", *Proceedings of 1st International Conference on Autonomous agents*, Marina Del Rey, CA, pages 209-216, 1997
 - [52]. B. Biskupski, J. Dowling, J. Sacha, "Properties and Mechanisms of Self-Organizing MANET and P2P Systems", *ACM Transactions on Autonomous and Adaptive Systems*, Volume 2, Issue 1, Article 1, 2007.
 - [53]. M. Stojmenovic, "Swarm Intelligence for Routing in Ad-hoc Wireless Networks", Chapter 7 in: *Security and Routing in Wireless Networks* (Y. Xiao, J. Li, Y. Pan, eds.), Nova Science Publishers, pages 167-188, 2005.
 - [54]. S. Sharma, M. Singh, G. Singh, "Simulated proposed finer algorithm based on Ant colony optimization in wired networks", *Yamuna Journal of Technology and Business Research*, 2012.
 - [55]. E. Bonabeau, F. Henaux, S. Guerin, D. Snyers, P. Kuntz, G. Theraulaz, "Routing in Telecommunications Networks with 'Smart' Ant-Like Agents", *Proceedings of the second international workshop on Intelligent Agents for Telecommunication applications*, Paris, France, 1999.
 - [56]. P. Sharma, K Kotecha, "Optimization in Stagnation Avoidance of ACO based routing of Multimedia Traffic over Hybrid MANETs", *IJCST Volume 1. 2*, Issue 2, Pages 260-265, 2011.
 - [57]. M. Meisel, V. Pappas, L. Zhang, "A Taxonomy of Biologically Inspired Research in Computer Networking", *Computer Networks: The International Journal of Computer and Telecommunications Networking by Elsevier North-Holland, Inc.* New York, NY, USA, Volume 54, Issue 6, 2010.
 - [58]. S. A. Ladhake, S. A. Thakare, "A Review: Ant Based Ad-hoc Routing Protocols", *International Journal of Emerging Technology and Advanced Engineering*, Volume 2, Issue 1, pages 373 – 375, 2012.
 - [59]. M. Dorigo, G. D. Caro, "AntNet: A Mobile Agents Approach to Adaptive Routing," *Tech. Report*, IRIDIA- Free Brussels University, Belgium, 1997.
 - [60]. B. N. Keshvamurthy, D. Taushniwal, B. K. Eshwar, "Hiding Co-occurring Prioritized Sensitive Patterns Over Distributed Progressive Sequential Data

- Streams”, *Journal of Network and Computer Applications*, Volume 35, Issue 3, pages 1116-1129, 2012.
- [61]. G. D. Caro, M. Dorigo, “AntNet: Distributed Stigmergic control for Communication Networks”, *Journal of Artificial Intelligence Research*, Volume 9, pages 317-365, 1998.
- [62]. G. D. Caro, M. Dorigo, “Ant Colonies for Adaptive Routing in Packet-Switched Communications Networks”, In the proceedings 5th international conference of Parallel Problem Solving from Nature, Amsterdam, Springer-Verlag London, Pages 673-682, 1998.
- [63]. G. D. Caro, M. Dorigo, “Two Ant Colony Algorithms For Best-Effort Routing In Datagram Networks”, In The Proceedings of 10th International Conference on Parallel and Distributed Computing and Systems (PDCS’98), Las Vegas, NV, 1998.
- [64]. G. S. Pavani, H. Waldman, “Traffic Engineering And Restoration In Optical Packet Switching Networks By Means Of Ant Colony Optimization”, In The Proceedings Of Third International Conference on Broadband Communications, Network and Systems (BroadNets 2006), San Jose, CA, 2006.
- [65]. G. S. Pavani, H. Waldman, “Evaluation Of An Ant-Based Architecture For All-Optical Networks”, In The Proceedings of 10th Conference on Optical Network Design and Modelling (ONDM’06), Copenhagen, Denmark, 2006.
- [66]. S. Sharma, M. Singh, G. Singh, “Realistic Inspection Of Proposed Ant Algorithm With AntNet Algorithm Using NS-2”, *International Journal of Research in IT, Management and Engineering*, Volume 2, Issue 6, pages 146-156, 2012.
- [67]. H. Wang, Z. Shi, A. Ge, C. Yu, “An Optimized Ant Colony Algorithm Based On The Gradual Changing Orientation Factor For Multi-Constraint Qos Routing”, *Journal Of Computer Communications*, Volume 32, Issue 4, 2009.
- [68]. G. S. Pavani, L. G. Zuliani, H. Waldman, M. Magalhaes, “Distributed Approaches For Impairment-Aware Routing And Wavelength Assignment Algorithms in GMPLS Networks”, *Computer Networks* 52, pages 1905–1915, 2008.
- [69]. H. Shokrani, S. Jabbehdari, "A Survey of Ant- Based Routing Algorithms for Mobile Ad-hoc Networks", *International Conference on Signal Processing*

- Systems algorithm at Islamic Azad University, North Tehran branch, Tehran, Iran, pages 311-317, 2009.
- [70]. M. Sailaja, R. K. Kumar, P. S. R. Murty, P. E. S. N. KrishnaPrasad, "A Study on Routing Algorithms Derived From the Nature for MANETs", International Magazine on Advances in Computer Science and Telecommunications IMACST: Volume 2, Number 1, pages 31-37, 2011.
- [71]. A. K. Gupta, H. Sadawarti, A. K. Verma, "MANET Routing Protocols Based on Ant Colony Optimization", International Journal of Modeling and Optimization, Volume 2, Issue 1, pages 42-49, 2012.
- [72]. G. Singh, N. Kumar, A. K. Verma, "Ant Colony Algorithms in MANETs: A Review", Journal of Network and Computer Applications, Volume 35, Pages 1964–1972, 2012.
- [73]. H. Matsuo, K. Mori, "Accelerated Ants Routing in Dynamic Networks", 2nd International Conference on Software Engineering, Artificial Intelligence, Networking & Parallel/Distributed Computing, pp.333–339, Nagoya, Japan, 2001.
- [74]. O. Hussein, T. Saadawi, "Ant Routing Algorithm For Mobile Ad-hoc Networks (ARAMA)", in the Proceedings IEEE International Conference on Performance, Computing, and Communications Conference, pages 281 – 290, 9-11 , 2003
- [75]. S. S. Dhillon, X. Arbona, P. V. Mieghem, "Ant Routing in Mobile Ad-hoc Networks", 3rd international conferences on networking and services (ICNS 2007), Athens, pages 67-74, 2007
- [76]. S. Prasad, Y. P. Singh, C. S. Rai, "Swarm Based Intelligent Routing for MANETs", International Journal of Recent Trends in Engineering, Academy Publishers (ACEEE), Volume 1, Number 1, pages 153-158, 2009.
- [77]. A Chowdhury, S. Das, "Automatic Shape Independent Clustering Inspired By Ant Dynamics", International Journal of Swarm and Evolutionary Computation, Volume 3, pages 33-45, 2012.
- [78]. P. Chakraborty, G. G. Roy, S. Sinha, S. Bose, A. Mondal, S. Das, "Automatic Shape Independent Clustering Inspired By Ant Dynamics", In The Proceedings of International Workshop on Machine Intelligence Research organized by Machine Intelligence Research Labs, Pages 64-74, 2009.

- [79]. B. R. Sujatha, M. V. Sathyanarayana, "PBANT-Optimized ANT Colony Routing Algorithm For MANETs", *Global Journal of Computer Science and Technology*, Volume 10, Issue 2, pages 29-34, 2010.
- [80]. G. S. Sharvani, A. G. Ananth, T. M. Rangaswamy, "Efficient Stagnation Avoidance For MANETs with Local Repair Strategy Using Ant Colony Optimization", *International Journal of Distributed and Parallel Systems (IJDPS)*, Volume 3, Number 5, pages 123-137, 2012.
- [81]. G. S. Sharvani, A. G. Ananth, T. M. Rangaswamy, "Ant Colony Optimization Based Modified TERMITE Algorithm (MTA) with efficient stagnation avoidance strategy for MANETS", *International journal on applications of graph theory in wireless Ad-hoc networks and sensor networks (GRAPH-HOC)* Volume 4, Number 2/3, 2012.
- [82]. M. Heusse, D. Snyers, Sylvain Guerin, Pascale Kuntz, "Adaptive Agent Driven Routing and Load Balancing in Communication Networks", *Advances in complex systems*, Volume 1, Volume 2, pages 237-254, 1998.
- [83]. M. Heusse, D. Snyers, Y. Kermarrec, "Adaptive Agent Driven Routing in Communication Networks: Comparison with A Classic Approach", *Advances in complex systems*, Volume 2, Issue 3, pages 237-219, 1999.
- [84]. M. Roth, S. Wicker, "Termite: Ad-hoc Networking with Stigmergy", *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, Volume:5, pages 2937 – 2941, 2003.
- [85]. M. Roth, S. Wicker, "Termite: A Swarm Intelligent Routing Algorithm for Mobile Wireless Ad-hoc Networks", *Journal of Studies in Computational Intelligence*, Volume 31, pages 155-184, 2006.
- [86]. S. Rajagopalan, C. Jaikao, C. C. Shen, "Unicast Routing for Mobile Ad-hoc Networks with Swarm Intelligence," *University of Delaware, Tech. Report*, Pages 2003-07, 2003.
- [87]. S. Rajagopalan, C. C. Shen, "ANSI: A Swarm Intelligence-Based Unicast Routing Protocol for Hybrid Ad-hoc Networks", *Journal of Systems Architecture*, Volume 52, Issue 8-9, pages 485-504, 2006.
- [88]. N. Sengottaiyan, R. Somasundaram, S. Arumugam, "A Modified Routing Algorithm for Reducing Congestion in Wireless Sensor Networks", *European Journal of Scientific Research*, Volume 35, Issue 4, pages 529-536, 2009.

- [89]. S. B. Wankhade, M. S. Ali, "Route Failure Management Technique for Ant Based Routing in MANET", International Journal of Scientific & Engineering Research, Volume 2, Issue 9, Pages 1-5, 2011.
- [90]. N. Umapathi, N. Ramaraj, R. A. Mano, "A Proactive Ant Colony Algorithm for Efficient Power Routing using MANET", International Journal of Computer Applications, Volume 58, Number 20, pages 33-36, 2012.
- [91]. R. Singh, D. K. Singh, L. Kumar, "Performance Evaluation of ACO Based on Demand Routing Algorithm for Mobile Ad-hoc Networks", International Journal of Engineering Science and Technology (IJEST), Volume 3, Number 3, pages 1809-1815, 2011.
- [92]. M. Gunes, U. Sorges, I. Bouazzi, "ARA – The Ant-Colony Based Routing Algorithm for MANETs", In the proceedings of Proceedings. International Conference Parallel Processing Workshop, pages 79-85, 2002.
- [93]. M. Gunes, O. Spaniol, "Ant-Routing-Algorithm for Mobile Multi-Hop Ad-hoc Networks" In the Book Of Network control and engineering for QoS, Security and Mobility II, Kluwer Academic Publishers Norwell, MA, USA, Volume II, Number 1, pages 120–138, 2003.
- [94]. J. S. Baras, H. Mehta, "A Probabilistic Emergent Routing Algorithm for Mobile Ad-hoc Networks," In WiOpt'03: Modeling and Optimization in Mobile, Ad-hoc and Wireless Networks, pages 68-73, 2003.
- [95]. A. Kumar, R. Singh, "Mobile Ad-hoc Networks Routing Optimization Techniques Using Swarm Intelligence", International Journal of Research in IT & Management (IJRIM), Volume 1, Issue 4, pages 2231-4334, 2011.
- [96]. M. Heissenbuttel, T. Braun, "Ants-Based Routing In Large Scale Mobile Ad-hoc Networks", Presented at 13th ITG/GI-Fachtagung Kommunikation in Verteilten Systemen (KiVS), Leipzig, Germany, pages 91-99, 2003.
- [97]. X. Zheng, W. Guo, R. Liu, "An Ant Based Distributed Routing Algorithm for Ad-hoc Networks (ADRA)". International Conference on Communication Circuits and System, ICCCAS – 2004, pages 412-417, 27-29 2004.
- [98]. S. Kamali, J. Opatrny, "A Position Based Ant Colony Routing Algorithm for Mobile Ad-hoc Networks", Journal Of Networks – Academy Publishers, Volume 3, Number 4, pages 31-41, 2008.

- [99]. H. Wang, Z. Shi, S. Li, "Multicast Routing for Delay Variation Bound Using A Modified Ant Colony Algorithm", *Journal of Network and Computer Applications*, Volume 32, Issue 1, Pages 258–272, 2009.
- [100]. X. Yang, L. Layuan, C. Chuanhui, "Application Research Based Ant Colony Optimization for MANET", *Proceedings of IEEE International Conference on Wireless Communications, Networking and Mobile Computing 2006 (WiCOM 2006)*, Wuhan, pages 1-4, 2006.
- [101]. P. Deepalakshmi, S. Radhakrishnan, "Ant Colony Based QoS Routing Algorithm for Mobile Ad-hoc Networks", *International Journal of Recent Trends in Engineering*, Volume 1, Number 1, pages 459-462, 2009.
- [102]. R. Gupta, "RSAR: Ring Search based Ant Routing for MANETs", *International Journal of Computer Applications*, Volume 38, Number 11, pages 22-26, 2012
- [103]. P. Sarala, D. Kalaiselvi, "Multipath Dynamic Source Routing with Cost and Ant Colony Optimization for MANETs", *International Journal of Applied Engineering Research*, Dindigul, Volume 1, Number 1, pages 83-91, 2010.
- [104]. A. M. A. Elmoniem, H. M. Ibrahim, M. H. Mohammed, A. R. Hedar, "Ant Colony and Load Balancing Optimizations for AODV Routing Protocol", *International Journal of Sensor Networks and Data Communications*, Volume 1, Pages 1-14, 2011.
- [105]. L. Rosati, M. Berioli, G. Reali, "On Ant Routing Algorithms in Ad-hoc Networks with Critical Connectivity", *Ad-hoc Networks*, Volume 6 Issue 6, Pages 827-859, 2008.
- [106]. N. Sengottaiyan, R. M. Somasundaram, S. Arumugam, "A Modified Routing Algorithm for Reducing Congestion in Wireless Sensor Networks", *European Journal of Scientific Research*, Volume 35, Issue 4, pages 529-536, 2009.
- [107]. M. Singh, J. Agrawal, S. Sharma, "Dynamic Adaptive Routing Algorithm Based on Ant Algorithm in Combination with Queuing Network Analysis", *International Journal of Computer Science Issues (IJCSI)*, Volume 9, Issue 3, Number 2, pages 444-448, 2012.
- [108]. B. Nancharaiah, B. Chandra Mohan, "Routing in Mobile Ad-hoc Networks (MANETS) using Fuzzy Aided Ant Colony Optimization (ACO) Technique", *Journal of Theoretical and Applied Information Technology*, Volume 52, Number 2, pages 227-235, 2013.

- [109]. F. Correia, T. Vazal, "Simple Ant Routing Algorithm", In the proceedings of International Conference on In Information Networking ICOIN 2008, Busan, pages 1–8, 2008.
- [110]. L. Yinsheng, H. Chen, X. Zheng, C. F. Tsai, J. H. Chen, N. Shah, "A Service-Oriented Travel Portal and Engineering Platform", Expert System Appl., Volume 38, Issue 2, pages 1213-1222, 2011.
- [111]. R. Asokan, A. M. Natragan, C. Venketesh, "Ant Based Dynamic Source Routing Protocol to Support Multiple Quality of Service (QoS) Metrics in Mobile Ad-hoc Networks" International Journal of Computer Science and Security, Volume 2, Issue 3, 2011.
- [112]. S. Sofat, A. K. Gupta, "Performance Based Analysis of ACO Based DSR and DSR", International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS), Volume 4, Issue 6, Pages 553-557, 2013.
- [113]. D. Sutariya, P. Kamboj, "A Survey of Ant Colony based Routing Algorithms for MANET", European Scientific Journal, European Scientific Journal Volume 3, pages 82-91, 2013.
- [114]. S. Sofat, A. K. Gupta, "Ant Colony Based Dynamic Source Routing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 10, Pages 721-728, 2013.
- [115]. E. Khosrowshahi-Asl, M. Noorhosseini, A. Saberi, "A Dynamic Ant Colony Based Routing Algorithm for Mobile Ad-hoc Networks", Journal Of Information Science And Engineering, Volume 27, 1581-1596, 2011
- [116]. P. B. Pankajavalli, N. Arumugam, "BADSR: An Enhanced Dynamic Source Routing Algorithm for MANETs Based on Ant and Bee Colony Optimization", European Journal of Scientific Research, Volume 53, Number 4, pages 576-581, 2011.
- [117]. E. Osagie, P. Thulasiraman, R. K. Thulasiram, "PACONET: improved Ant Colony Optimization Routing Algorithm for Mobile Ad-hoc Networks", 22nd International Conference on Advanced Information Networking and Applications (AINA 2008), Okinawa, pages 204-211, 2008.
- [118]. S. J. Gudakahriz, S. Jamali, E. Zeinali, "NISR: A Nature Inspired Scalable Routing Protocol for Mobile Ad-hoc Networks", IJCSET, Volume 1, Issue 4, pages 180-184, 2011.

- [119]. S. Marwaha, C. K. Tham, D. Srinivasan, "Mobile Agents Based Routing Protocol for Mobile Ad-hoc Networks", Proceedings of the IEEE Global Communications Conference (GlobeCom 02), Taipei, Taiwan, pages 198-209, 2002.
- [120]. A. Sabari, K. Duraiswamy, "Multiple Constraints for Ant Based Multicast Routing in Mobile Ad-hoc Networks", Journal of Computer Science, Volume 5, Issue 12, Pages 1020-1027, 2009.
- [121]. G. D. Caro, F. Ducatelle, L. M. Gambardella, "AntHocNet: An Ant-Based Hybrid Routing Algorithm for Mobile Ad-hoc Networks," Proceedings of Parallel Problem Solving from Nature (PPSN VIII), Volume 3242 of LNCS, pages 461-470. Springer -Verlag, 2004.
- [122]. P. Sharma, Y. Karkhanawala, K. Kotecha, Member, IACSIT, "Bandwidth Constrained Routing of Multimedia Traffic over Hybrid MANETs using Ant Colony Optimization", International Journal of Machine Learning and Computing, Volume 1, Issue 3, Pages 242-246, 2011.
- [123]. P. Deepalakshmi and S. Radhakrishnan, "An Ant Colony-Based Multi Objective Quality of Service Routing for Mobile Ad-hoc Networks", EURASIP Journal on Wireless Communications and Networking, DOI: 10.1186/1687-1499-2011-153, pages 1-12, 2011.
- [124]. M. Poojary, B. Renuka, "Ant Colony Optimization Routing to Mobile Ad-hoc Networks in Urban Environments", International Journal of Computer Science and Information Technologies, Volume 2, Number 6, pages 2776-2779, 2011.
- [125]. Z. Yuan-Yuan, H. Yan-Xiang, "Ant Routing Algorithm for Mobile Ad-hoc Networks Based on Adaptive Improvement," Proceedings of International conference on Wireless Communications, Networking and Mobile Computing, 2005, Volume 2, Pages 678 – 681, 2005.
- [126]. D. Sivakumar, R. S. Bhuvaneswaran, "Proposal on Multi Agent Ants Based Routing Algorithm for Mobile Ad-hoc Networks", IJCSNS International Journal of Computer Science and Network Security, Volume 7, Number 6, pages 261-268, 2007.
- [127]. J. Wang, E. Osagie, P. Thulasiraman, R. K. Thulasiram, "HOPNET: A Hybrid Ant Colony Optimization Routing Algorithm for Mobile Ad-hoc Network," Journal Of Ad-hoc Networks, Volume 7, Issue 4, pages 690-705, 2009.

- [128]. S. Sharma, M. Singh, G. Singh, "Swarm intelligence Based Comparative Scrutiny Of Different Routing Algorithms in MANETs", *International journal of research in engineering and applied sciences*, Volume 2, Issue 2, pages 1818-1831, 2012.
- [129]. A. M. Okazaki, A. A. Frohlich, "Adapting HOPNET Algorithm for Wireless Sensor Networks", in the proceedings of 9th International Information and Telecommunication Technologies Symposium, IEEE Latin America Transactions, Volume 8, Issue 4, Pages 30-35, 2009.
- [130]. M. K. Rafsanjani, S. Asadinia, F. Pakzad, "A Hybrid Routing Algorithm Based on Ant Colony and ZHLS Routing Protocol for MANET", Springer-Verlag Berlin Heidelberg, pages 112–122, 2010.
- [131]. S. Asadinia, M. K. Rafsanjani, A. B. Saeid, "A Novel Routing Algorithm Based-on Ant Colony in Mobile Ad-hoc Networks", 3rd international Conference on Ubi-media computing (U-Media), Jinhua , pages 77-82, 2010.
- [132]. J. A. P. Martins, S.Luis O. B. Correia, J. C. Junior, "Ant-DYMO: A Bio-Inspired Algorithm for MANETS", 17th IEEE international conference on telecommunications, Doha, pages 748-754, 2010
- [133]. M. Woo, "Performance Analysis of An Ant-based Routing Algorithm with Enhanced Path Maintenance for MANETs", In the proceedings of the Seventh International Conference on Wireless and Mobile Communications (ICWMC 2011), Luxembourg, pages 157-162, 2011.
- [134]. D. Dhull, S. Dhull, "An Improved Ant Colony Optimization (IACO) Based Multicasting in MANET", *International Journal of Inventive Engineering and Sciences (IJIES)* ISSN: 2319–9598, Volume 1, Issue 3, pages 8-12, 2013.
- [135]. S. Modi, J. Prithviraj, "Performance Comparison of IACO,AODV Networking Routing Protocols", *International Journal of Smart Sensors and Ad-hoc Networks (IJSSAN)*, pages 33-37, 2011
- [136]. N. Shah, K. M. Chao, A. N. Godwin, A. James, "An Abstract Knowledge Based Approach to Diagnosis and Recovery of Plan Failure in Multi-Agent Systems", *Advanced Engineering Informatics* Volume 21, Issue 2, pages 183-190, 2007.
- [137]. R. Singh, M. Dave, "Antecedence Graph Approach to Check Pointing for Fault Tolerance in Mobile Agent Systems" *IEEE Trans. Computers*, Volume 62, Issue 2, pages 247-258, 2013.

- [138]. N. Kumar, R. Iqbal, N. Chilamkurti, A. James, "An Ant Based Multi Constraints QoS Aware Service Selection Algorithm in Wireless Mesh Networks", *Simulation Modelling: Practice and Theory* Volume, Volume 19, Issue 9, pages 1933-1945, 2011.
- [139]. H. Wu, K. Sun, "Improved Ant Colony Classification Algorithm Applied to Membership Classification", *Advances in Swarm Intelligence Lecture Notes in Computer Science*, Springer-Verlag Berlin Heidelberg, Volume 7928, pages 278-287, 2013.
- [140]. D. Karthikeyan, M. Dharmalingam, "Ant Based Intelligent Routing Protocol for MANET", In the proceedings of Pattern Recognition, Informatics and Medical Engineering (PRIME-2013), pages 11-16, 2013.
- [141]. G. Singh, N. Kumar, A. K. Verma, "OANTALG: An Orientation based ANT Colony ALGORITHM for Mobile Ad-hoc Networks", *Wireless Personal Communications*, DOI: 10.1007/S11277-014-1613-6, 2014.
- [142]. M. Rana, R. K. Singh, "An Optimization Algorithm for Congestion Control", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 4, Issue 1, pages 266-274, 2014.
- [143]. B. Kadri, M. Feham, A. Mhammed, "Efficient and Secured Ant Routing Algorithm for Wireless Sensor Networks", *International Journal of Network Security*, Volume 16, Number 2, Pages 149-156, 2014.
- [144]. R. T. Thivya Lakshmi, R. Srinivasan, G. S. Raj, "An Efficient Algorithm for Improving Qos in MANETs", *International Journal of Scientific and Research Publications*, Volume 4, Issue 2, Pages 1-6, 2014.
- [145]. A. R. Sardar, M Singh, R. R. Sahoo, K. Majumder, J. K. Sing, S. K. Sarkar, "An Efficient Ant Colony Based Routing Algorithm for Better Quality of Services in MANET", *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol. I*, In the book of *Advances in Intelligent Systems and Computing*, Volume 248, pages 233-240, 2014
- [146]. M. Zhang, C. Xu, J. Guan, R. Zheng, Q. Wu, H. Zhang, "A Novel Bio-inspired Trusted Routing Protocol for Mobile Wireless Sensor Networks," *KSII Transactions on Internet and Information Systems*, Volume 8, Number 1, 2014.
- [147]. The Network Simulator NS-2. <<http://www.isi.edu/nsnam/ns/>>.