Reliable Routing Protocol For Mobile Ad Hoc Network

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

In

Computer Engineering

By

Kajal Shambhubhai Patel

129990907006

Under Supervision of

Dr J. S. Shah

GUJARAT TECHNOLOGICAL UNIVERSITY
AHMEDABAD

[January – 2018]

Reliable Routing Protocol For Mobile Ad Hoc Network

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

In

Computer Engineering

By

Kajal Shambhubhai Patel

129990907006

Under Supervision of

Dr J. S. Shah

GUJARAT TECHNOLOGICAL UNIVERSITY
AHMEDABAD

[January – 2018]

DECLARATION

I declare that the thesis entitled **Reliable Routing Protocol For Mobile Ad Hoc Network** submitted by me for the degree of Doctor of Philosophy is the record of research work carried out by me during the period from …… to ……..…….. under the supervision of **Dr J S Shah** and this has not formed the basis for the award of any degree, diploma, associate ship, fellowship, titles in this or any other University or other institution of higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis. I shall be solely responsible for any plagiarism or other irregularities, if noticed in the thesis.

Signature of the Research Scholar : …… Date:…..……

Name of Research Scholar: …… Place : ……

CERTIFICATE

I certify that the work incorporated in the thesis **Reliable Routing Protocol For Mobile Ad Hoc Network** submitted by Shri / Smt. / Kumari **Kajal Shambhubhai Patel** was carried out by the candidate under my supervision/guidance. To the best of my knowledge: (i) the candidate has not submitted the same research work to any other institution for any degree/diploma, Associateship, Fellowship or other similar titles (ii) the thesis submitted is a record of original research work done by the Research Scholar during the period of study under my supervision, and (iii) the thesis represents independent research work on the part of the Research Scholar.

Signature of Supervisor: …… Date: ……

Name of Supervisor: ……Place: ……

Originality Report Certificate

It is certified that PhD Thesis titled **Reliable Routing Protocol For Mobile Ad Hoc Network** by **Kajal Shambhubhai Patel** has been examined by us. We undertake the following:

a. Thesis has significant new work / knowledge as compared to already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.

b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.

c. There is no fabrication of data or results which have been compiled / analysed.

d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.

e. The thesis has been checked using **turnitin** (copy of originality report attached) and found within limits as per GTU Plagiarism Policy and instructions issued from time to time (i.e. permitted similarity index <=25%).

Signature of the Research Scholar : …… Date: ……..…..

Name of Research Scholar: ……..... Place : ……..

Signature of Supervisor: ……..... ... Date: ……....

Name of Supervisor: ……..…. Place: ……

plagarism check1

ORIGINALITY REPORT

3%	1%	4%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Advances in Intelligent Systems and Computing, 2016. Publication	2%
2	web.cs.hacettepe.edu.tr Internet Source	1%
3	Sapna, , M Sharma, and H Kaur. "Performance evaluation of hybrid network using RIP", 7th International Symposium on High-capacity Optical Networks and Enabling Technologies, 2010. Publication	1%

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	< 1%
EXCLUDE BIBLIOGRAPHY	ON		

PhD THESIS Non-Exclusive License to GUJARAT TECHNOLOGICAL UNIVERSITY

In consideration of being a PhD Research Scholar at GTU and in the interests of the facilitation of research at GTU and elsewhere, I Kajal Shambhubhai Patel (Full Name of the Research Scholar) having (Enrollment No.) 129990907006 hereby grant a non-exclusive, royalty free and perpetual license to GTU on the following terms:

a) GTU is permitted to archive, reproduce and distribute my thesis, in whole or in part, and/or my abstract, in whole or in part (referred to collectively as the "Work") anywhere in the world, for non-commercial purposes, in all forms of media;

b) GTU is permitted to authorize, sub-lease, sub-contract or procure any of the acts mentioned in paragraph (a);

c) GTU is authorized to submit the Work at any National / International Library, under the authority of their "Thesis Non-Exclusive License";

d) The Universal Copyright Notice (©) shall appear on all copies made under the authority of this license;

e) I undertake to submit my thesis, through my University, to any Library and Archives. Any abstract submitted with the thesis will be considered to form part of the thesis.

f) I represent that my thesis is my original work, does not infringe any rights of others, including privacy rights, and that I have the right to make the grant conferred by this non-exclusive license.

g) If third party copyrighted material was included in my thesis for which, under the terms of the Copyright Act, written permission from the copyright owners is required, I have obtained such permission from the copyright owners to do the acts mentioned in paragraph (a) above for the full term of copyright protection.

h) I retain copyright ownership and moral rights in my thesis, and may deal with the copyright in my thesis, in any way consistent with rights granted by me to my University in this non-exclusive license.

i) I further promise to inform any person to whom I may hereafter assign or license my copyright in my thesis of the rights granted by me to my University in this nonexclusive license.

j) I am aware of and agree to accept the conditions and regulations of PhD including all policy matters related to authorship and plagiarism.

Signature of the Research Scholar:

Name of Research Scholar:

Date: Place:

Signature of Supervisor:

Name of Supervisor:

Date: Place:

Seal:

Thesis Approval Form

The viva-voce of the PhD Thesis submitted by Shri/Smt./Kum ……. (Enrollment No. ……) entitled …… …… was conducted on …….…… (day and date) at Gujarat Technological University.

(Please tick any one of the following option)

☐ The performance of the candidate was satisfactory. We recommend that he/she be awarded the PhD degree.

☐ Any further modifications in research work recommended by the panel after 3 months from the date of first viva-voce upon request of the Supervisor or request of Independent Research Scholar after which viva-voce can be re-conducted by the same panel again.

(briefly specify the modifications suggested by the panel)

☐ The performance of the candidate was unsatisfactory. We recommend that he/she should not be awarded the PhD degree.

(The panel must give justifications for rejecting the research work)

---------- ----------

Name and Signature of Supervisor with Seal (1) (External Examiner 1) Name and Signature

---------- ----------

(2) (External Examiner 2) Name and Signature (3) (External Examiner 3) Name and Signature

ABSTRACT

Mobile Adhoc NETwork (MANET) helps us in setting up a network of mobile nodes like laptop, smart phones, tablet etc. without the need of any infrastructure. We can develop a temporary network in the battle field, forest, hilly area, meeting rooms, disaster area etc. whenever the need arises. There is no need of any Access Point (AP) or Base Station (BS) to build MANET. The nodes in this network can move freely and change their position and thus the topology of the network at any time. Nodes are battery operated and resource constrained. MANET uses wireless media for data communication. There is no specialized router used in MANET. Each and every node has to act as a router to forward data from source node to a destination node. The routing protocols used for wired network cannot be used for MANET due to the aforementioned characteristics of it. The routing protocols like DSDV, AODV, OLSR, DSR, etc. designed for MANET consider only dynamically changing network topology. These basic routing protocols do not consider any security issue while routing. Thus MANET is vulnerable to many security attacks as nodes uses wireless media and has to depend on unknown intermediate nodes for routing their packets. The attacks like packet drop, intentional packet delay before forwarding, eavesdropping, DoS attacks, packet modification, fabrication and replication of packets etc. can be done by intermediate nodes. These attacks may compromise confidentiality and disturb the network operation which may lead to a failure of the whole network. Apart from the aforementioned hard security threats, the MANET is also vulnerable to soft security threats like low quality of service, wrong information delivery or advertisement and malicious/malfunction activities. These soft threats are associated with behaviour of the intermediate nodes. Since the nature of the system is open, such behaviours are difficult to control. Hence, such soft threats are also difficult to detect.

To detect/avoid attacker nodes many researchers have come up with routing protocols which use various cryptographic approaches. The cryptographic approaches are very complex and have huge computational overhead on node, which is not suitable for resource constraint mobile nodes. Additionally cryptography based solutions are binary solutions. The nodes either pass or fail the security checks. In a MANET, behaviour of a node changes continuously. These changes may occur due to malicious behaviour of the nodes/hardware failure/mobility of the node. The cryptographic approaches cannot detect

such continuously changing behaviour of nodes. To solve the problem, researchers come up with a trust based routing solutions. For measuring reliability, we can use the trust value associated with each node. On almost all existing trust based routing schemes, communication parameters like number of successful sessions, packet forwarded between two nodes, number of packets dropped or delayed, response time, battery life, mobility of node etc. are used for calculating trust value of a node. Researchers also use various methods for aggregating these parameters to calculate trust values like a weighted sum model, Bayesian model, fuzzy model, Markov chain based model, etc. Most of existing trust based routing protocols create a route based on trust of nodes and gives only one trustworthy route. If this route breaks, we need to re-establish other route which may add overhead. Also, all existing trust based routing protocols used most trusted nodes while the route is established. This may add extra burden to trustworthy nodes only and free other nodes from routing. Hence, trustworthy nodes may overburden in routing only and could not do their own task.

In order to address these issues we can think of searching multiple trusted route between same source to destination during route formation. This allows us to use other available routes, if any route fails and also if we use some trusted route simultaneously, we may distribute the load of routing between multiple nodes. A source node will not re request for route until all trusted routes are broken or expired. In proposing a routing scheme which we named TMA-AODV (Trust based Mobility Aware AODV), we have used a number of packets dropped by a node, the number of packets delayed by a node before forwarding and number of time node move out of the active route due to its movement (i.e. number of time a node is being reason of link break) for trust computation. The first two parameters protect a network from any type of packet drop and packet delay attack. And the latter parameter helps us to provide a stable route. For calculating the trust value, we have used weighted sum model. We have chosen AODV protocol, as it performs efficiently in both static as well as the dynamic network. For the implementation, we have used the OPNET simulator 11.0 academic edition. We have performed various analyses to understand the working of ad hoc network and AODV routing. We have implemented message drop attack and message delay attacks to study its effect on route discovery time and throughput of the network. Later on, we have investigated the proposed routing scheme (Trust based Mobility Aware-AODV) to measure route discovery time and throughput in the presence of message drop/delay attack and with and without mobility. The results of investigations

show improvement in throughput and route discovery time. Route discovery time is found to be large at the beginning as the proposed scheme (TMA-AODV) has to search for multiple trusted routes from source to destination and uses them for simultaneous data transmission. The other reason could be, the initial learning period required by TMA-AODV to collect enough observations at the nodes. Once all routes found, route discovery time improves compare to AODV. The throughput has also improved as we are detecting and avoiding packet drop and packet delay nodes while routing using TMA-AODV. The final results of the experiments with AODV and TMA-AODV shows that the TMA-AODV outperforms the traditional AODV in the presence of mobile nodes and attacker nodes. This confirms that TMA-AODV can be used in place of AODV with resource constrained, error-prone and highly dynamic mobile ad-hoc networks.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my research supervisor, Dr J S Shah, for the constant support to my PhD research work. I wish to thank him for his excellent guidance, patience, motivation, enthusiasm, and extraordinary knowledge along with providing the excellent atmosphere for doing my research work. I would also like to thank him and the university who allowed and supported me to do the research in the area of MANET security using the trust based approach. His guidance really helped me at each and every stage of my research work and writing of this thesis. I could not imagine a better advisor for my PhD study.

My Doctorate Progress Committee members also guided me through all these years. I would like to thank Dr D R Patel and Dr M A Zaveri, who have reviewed my research work time to time being DPC committee members and suggested me the valuable correction and improvement in my research work. They have also helped me by giving invaluable comments and asking detailed and deep questions during the review sessions.

My sincere thanks also goes to my previous principal Prof M V Garach and my current principal Dr G J Vala, who have allowed me doing this research work as a part time candidate along with my job responsibilities. They also helped me to provide various facilities like books, journal papers, internet and computing facilities, fund for attending trainings and conferences etc., which have been very useful for this thesis and research work. I convey my gratitude to all faculties and staffs of my department and other departments of my institute who directly and indirectly supported me for doing this research work.

My special thanks to Mr J M Shah, Prof Jayesh Solanki, Prof J S Dhobi, Prof M T Savaliya, Prof H M Diwanji, Prof T J Raval, Prof C A Patel, Prof Kalpesh Patel, Prof. Mita Pareekh and Prof Dipen Contractor, who have supported me for completing my research work. I also thank to my family and my son, Aryan, without whose constant love and care it would be difficult to complete this work. Lastly, I wish to thank one and all who helped me in finishing and furnishing my research work.

Table of Content

Chapter 1	**Mobile Ad hoc Network**		**1**
	1.1	Introduction	**1**
	1.2	Setting up wireless Network	**1**
	1.3	Characteristics Mobile Ad hoc Network	**2**
	1.4	Applications of Mobile Ad hoc network	**3**
	1.5	History of MANET	**4**
	1.6	Routing in MANET	**5**
		1.6.1 Proactive Routing Protocols	**7**
		1.6.2 Reactive Routing Protocols	**9**
		1.6.3 Other Routing Protocols	**12**
	1.7	Selection of base routing protocol for research.	**12**
Chapter 2	**Security Challenges in MANET and their possible solutions**		**14**
	2.1	Introduction	**14**
	2.2	Challenges in MANET	**14**
	2.3	Security Requirements	**15**
		2.3.1 Hard Security Services	**15**
		2.3.2 Soft Security Services	**16**
	2.4	Threats	**16**
		2.4.1 Attacks (Hard Security Threats)	**16**
		2.4.2 Misbehaviour (Soft Security Threats)	**19**
	2.5	Countermeasures	**19**
		2.5.1 Prevention Techniques: Cryptographic approach	**20**
		2.5.2 Detection Techniques: Intrusion Detection System based approach	**21**
		2.5.3 Trust based approach	**23**

2.6 Selection of Trust based approach for detecting attacks for this thesis **23**

Chapter 3 **Trust Management** **26**

3.1 Trust **26**
3.2 Definition of trust **27**
3.3 Digital representative of Trust **27**
3.4 Trust factors **27**
3.5 Trust Metrics **28**
3.6 Characteristics of trust **28**
3.7 Trust modelling and Trust management **29**
3.8 Trust Computation Engines **30**
3.8.1 Summation model **30**
3.8.2 Average model **30**
3.8.3 Bayesian model **31**
3.8.4 Belief Model **31**
3.8.5 Fuzzy Model **32**
3.8.6 Markov chain based trust model **32**
3.9 Use of Trust management in MANET **33**
3.10 Selection of Trust Computation Engine for this thesis **33**

Chapter 4 **Literature Review** **35**

4.1 Introduction **35**
4.2 State of the Art **36**
4.3 Comparison of Existing Trust based routing approaches for MANET **46**
4.3.1 Network Parameters used for calculating trust value **46**
4.3.2 Trust computation Engines used **47**
4.3.3 Attacks/Misbehaviour detected **47**
4.4 Survey Conclusion **48**

Chapter 5 **Proposed Trust Based Routing Model** **50**

5.1 Problem Statement **50**

5.2 Scope of our Research **51**

5.3 Objective of our Research **51**

5.4 Original Contribution By the Thesis **52**

5.5 Proposed System **55**

5.5.1 Proposed System Architecture **55**

5.5.2 Features of proposed routing algorithm **56**

5.6 System Diagrams and Algorithms **58**

5.6.1 Context Flow Diagram of proposed system **58**

5.6.2 Algorithm of proposed system **59**

5.6.3 Flowchart of traffic monitoring module **64**

5.6.4 Flowchart of proposed routing module **66**

5.7 Parameters for Performance Measurement **68**

Chapter 6 **Implementation Of Attacks And Its Effect On MANET** **69**

6.1 Research Methodology used for this thesis implementation **69**

6.2 Malicious node models **70**

6.3 Implementation of Malicious nodes **71**

6.3.1 Implementing malicious nodes in OPNET **71**

6.4 Effect of packet drop attack on AODV routing **74**

6.4.1 Simulation environment with 6 packet drop attacker nodes **75**

6.4.2 Simulation environment with 12 packet drop attacker nodes **78**

6.4.3 Simulation environment with 24 packet drop attacker nodes **81**

6.4.4 Concluding Remarks **82**

6.5 Effect of packet drop and packet delay attack on AODV routing **84**

6.5.1 Simulation environment with 3 packet drop and 3 packet delay attacker nodes **84**

6.5.2 Simulation environment with 6 packet drop and 6 packet delay attacker nodes **87**

6.5.3 Simulation environment with 12 packet drop and 12 packet delay attacker nodes **90**

6.5.4 Concluding Remarks **92**

6.6 Effect of mobile nodes with AODV routing **94**

6.7 Effect of packet drop and delay attack on AODV routing with mobile nodes **96**

6.7.1 Simulation environment with 3 packet drop attacker nodes, 3 packet delay attacker nodes and 16 mobile nodes **97**

6.7.2 Simulation environment with 6 packet drop attacker nodes, 6 packet delay attacker nodes and 16 mobile nodes **101**

6.7.3 Simulation environment with 12 packet drop attacker nodes, 12 packet delay attacker nodes and 16 mobile nodes **103**

6.7.4 Concluding Remarks **105**

6.8 Summary **107**

Chapter 7 **Implementation Of Proposed Routing Protocol And Its Analysis** **111**

7.1 Implementation of Proposed trust based routing protocol (TMA-AODV) for MANET **111**

7.2 Comparison of AODV and TMA-AODV without any attacker nodes and with mobility **115**

7.2.1 AODV and TMA-AODV without any attacker nodes and with fixed nodes **116**

7.2.2 AODV and TMA-AODV without any attacker nodes and with mobility **116**

7.3 Effect of TMA-AODV in MANET in presence of packet drop attacker nodes **118**

7.3.1 TMA-AODV with 6 packet drop attacker nodes **118**

7.3.2 TMA-AODV with 12 packet drop attacker nodes **121**
7.3.3 TMA-AODV with 24 packet drop attacker nodes **123**
7.3.4 Concluding Remarks **125**
7.4 TMA-AODV in presence of packet drop and packet delay attacker nodes **127**
7.4.1 TMA-AODV with 3 packet drop and 3 packet delay attacker nodes **127**
7.4.2 TMA-AODV with 6 packet drop and 6 packet delay attacker nodes **129**
7.4.3 TMA-AODV with 12 packet drop and 12 packet delay attacker nodes **131**
7.4.4 Concluding Remarks **134**
7.5 TMA-AODV in presence of mobile nodes, packet drop and packet delay attacker nodes **135**
7.5.1 TMA-AODV with 3 packet drop and 3 packet delay attacker nodes and 16 mobile nodes **136**
7.5.2 TMA-AODV with 6 packet drop and 6 packet delay attacker nodes and 16 mobile nodes **138**
7.5.3 TMA-AODV with 12 packet drop and 12 packet delay attacker nodes and 16 mobile nodes **140**
7.5.4 Concluding Remarks **142**
7.6 Conclusion **144**

Chapter 8 **Conclusion And Future Enhancement** **147**

8.1 Conclusion of the thesis **147**
8.2 Future enhancements **150**

References **151**

List of Publications **155**

List of Abbreviation

Abbreviation	Full Form
LAN	Local Area Network
MANET	Mobile Ad hoc Network
VANET	Vehicular Ad hoc Network
IEEE	Institute of Electrical and Electronics Engineers
PDA	Personal Digital Assistant
PC	Personal Computer
DARPA	Defence Advanced Research Projects Agency
PRNET	Packet Radio Network
MHz	Mega Hertz
GLOMO	Global Mobile Information System
ATM	Asynchronous Transfer Mode
NTDR	Near Term Digital Radio
DSDV	Destination Sequence Distance Vector
OLSR	Optimized Link State Routing
MPR	Multi Point Relay
DSR	Dynamic Source Routing
AODV	Adhoc On demand Distance Vector
RREQ	Route Request
RREP	Route Reply
RERR	Route Error
GRP	Geographic Routing Protocol
GPS	Global Positioning System
PIN	Personal Identification Number
MAC Address	Medium Access Control Layer Address
MAC	Message Authentication Code
MD5	Message Digest 5
MIC	Message Integrity Code
HMAC	Hash-based Message Authentication Code
SEAD	Secure Efficient Ad hoc Distance vector
SAODV	Secure Adhoc On demand Distance Vector
PKI	Public Key Infrastructure
CA	Certificate Authority
MOCA	Multiprocessor On-line Competitive Algorithm
MDA	Maximum Degree Algorithm
IDS	Intrusion Detection System
FSM	Finite State Machine
DoS attack	Denial Of Service attack
CPU	Central Processing Unit
PA	Packet Acknowledge
PP	Packet Precession
GR	Gratuitous Route replies
BL	Black Lists
SG	Salvaging

Abbreviation	Full Form
3-D	Three Dimension
PI	Past Interactions
PR	Peer Recommendations
QoS	Quality of Service
SREQ	Service Request
SREP	Service Reply
ARMAN	Ant Routing for Mobile Ad Hoc Networks
TMA-AODV	Trust based Mobility Aware-Ad hoc On demand Distance Vector
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
CRC	Cyclic Redundancy Check
PD	Permissible Delay.
Po	number of packets observed by a node
PF	number of packets successfully forwarded by a node
PD	number of packets delayed by a node
PER	number of link break by a node
OPNET	Optimized Network Engineering Tools
FTP	File Transfer Protocol

List of Figures

Figure Number	Caption
Figure 1.1	Setup of wireless network
Figure 1.2	Multi hop routing in ad hoc network
Figure 1.3	Categories of MANET routing protocols
Figure 1.4	Example of DSDV routing table on node A
Figure 1.5	RREQ packet broadcasting Source node A and destination node B
Figure 1.6	RREP packet from destination B to source A
Figure 5.1	Architecture of Proposed system
Figure 5.2	Context Flow Diagram of TMA-AODV routing
Figure 6.1	Malicious node model in OPNET (from OPNET)
Figure 6.2	ip_dispatch process used by the malicious node model at IP module (from OPNET)
Figure 6.3	Declaration of counter in malicious ip_dispatch process(from OPNET)
Figure 6.4	Experiment setup without attacker node (image from OPNET)
Figure 6.5	Experiment setup with 6 attackers (packet drop) nodes (image from OPNET)
Figure 6.6	Configuration of FTP traffic for ad hoc nodes (image from OPNET)
Figure 6.7	Comparison of Route Discovery Time (6 packet drop attacker)
Figure 6.8	Comparison of Throughput (6 packet drop attacker)
Figure 6.9	Experiment setup with 12 attackers (packet drop) nodes (image from OPNET)
Figure 6.10	Comparison of Route Discovery Time (12 packet drop attacker)
Figure 6.11	Comparison of Throughput (12 packet drop attacker)
Figure 6.12	Experiment setup with 24 attackers (packet drop) nodes (image from OPNET)
Figure 6.13	Comparison of Route Discovery Time (24 packet drop attacker)
Figure 6.14	Comparison of Throughput (24 packet drop attacker)
Figure 6.15	Average Route Discovery Time Vs number of attacker graph (drop attack)
Figure 6.16	Average Throughput Vs number of attackers graph (drop attack)
Figure 6.17	Experiment setup with 6 attackers (3 packet drop + 3 packet delay) nodes (image from OPNET)
Figure 6.18	Comparison of Route Discovery Time (3 packet drop + 3 packet delay)
Figure 6.19	Comparison of Throughput (3 packet drop + 3 packet delay)

Figure Number	Caption
Figure 6.20	Experiment setup with 12 attackers (6 packet drop+ 6 packet delay) nodes (image from OPNET)
Figure 6.21	Comparison of Route Discovery Time (6 packet drop+ 6 packet delay)
Figure 6.22	Comparison of Throughput (6 packet drop+ 6 packet delay)
Figure 6.23	Experiment setup with 24 attackers (12 packet drop+12 packet delay) nodes (image from OPNET)
Figure 6.24	Comparison of Route Discovery Time (12 packet drop+12 packet delay)
Figure 6.25	Comparison of Throughput (12 packet drop+12 packet delay)
Figure 6.26	Average Route Discovery Time Vs number of attackers graph (drop+delay attack)
Figure 6.27	Average Throughput Vs number of attackers graph (drop+delay attack)
Figure 6.28	Experiment setup without attacker node and with 16 mobile nodes (image from OPNET)
Figure 6.29	Experiment setup with 6 attacker nodes (3 packet drop + 3 packet delay+ 16 mobile nodes) (image from OPNET)
Figure 6.30	Comparison of Route Discovery Time (3 packet drop + 3 packet delay+ 16 mobile nodes)
Figure 6.31	Comparison of Throughput (3 packet drop + 3 packet delay+ 16 mobile nodes)
Figure 6.32	Experiment setup with 12 attacker nodes (6 packet drop+ 6 packet delay+ 16 mobile nodes) (image from OPNET)
Figure 6.33	Comparison of Route Discovery Time (6 packet drop+ 6 packet delay+ 16 mobile nodes)
Figure 6.34	Comparison of Throughput (6 packet drop+ 6 packet delay+ 16 mobile nodes)
Figure 6.35	Experiment setup with 24 attacker nodes (12 packet drop+12 packet delay + 16 mobile nodes) (image from OPNET)
Figure 6.36	Comparison of Route Discovery Time (12 packet drop+12 packet delay nodes+ 16 mobile nodes)
Figure 6.37	Comparison of Throughput (12 packet drop+12 packet delay nodes+ 16 mobile nodes)
Figure 6.38	Average Route Discovery Time Vs number of attackers graph (drop+delay attack+mobility)
Figure 6.39	Average Throughput Vs number of attackers graph (drop+delay attack+mobility)
Figure 7.1	my_aodv_rte process model (from OPNET)
Figure 7.2	Route table related update and trust table related data structure in my_aodv.h file
Figure 7.3	Route reply and route error related update in my_aodv_pkt_support.h file
Figure 7.4	Trust table related functions in my_aodv_ptypes.h file
Figure 7.5	Comparison of Route Discovery Time with AODV and TMA-AODV (6 packet drop attacker nodes)

Figure Number	Caption
Figure 7.6	Comparison of Throughput with AODV and TMA-AODV (6 packet drop attacker)
Figure 7.7	Comparison of Route Discovery Time with AODV and TMA-AODV (12 packet drop attacker nodes)
Figure 7.8	Comparison of Throughput with AODV and TMA-AODV (12 packet drop attacker)
Figure 7.9	Comparison of Route Discovery Time with AODV and TMA-AODV (24 packet drop attacker)
Figure 7.10	Comparison of Throughput with AODV and TMA-AODV (24 packet drop attacker)
Figure 7.11	Comparison average route discovery time of AODV and TMA-AODV in presence of packet drop attack
Figure 7.12	Comparison of average throughput with AODV and TMA-AODV routing in presence of packet drop attack
Figure 7.13	Comparison of Route Discovery Time with AODV and TMA-AODV (3 packet drop +3 packet delay attacker)
Figure 7.14	Comparison of Throughput with AODV and TMA-AODV (3 packet drop +3 packet delay attacker)
Figure 7.15	Comparison of Route Discovery Time with AODV and TMA-AODV (6 packet drop + 6 packet delay attacker)
Figure 7.16	Comparison of Throughput with AODV and TMA-AODV (6 packet drop + 6 packet delay attacker)
Figure 7.17	Comparison of Route Discovery Time with AODV and TMA-AODV (12 packet drop + 12 packet delay attacker)
Figure 7.18	Comparison of Throughput with AODV and TMA-AODV (12 packet drop + 12 packet delay attacker)
Figure 7.19	Comparison of average route discovery time of AODV and TMA-AODV in presence of packet drop and delay attack
Figure 7.20	Comparison of average throughput with AODV and TMA-AODV routing in presence of packet drop and delay attack
Figure 7.21	Comparison of Route Discovery Time with AODV and TMA-AODV (3 packet drop + 3 packet delay attacker and 16 mobile nodes)
Figure 7.22	Comparison of Throughput with AODV and TMA-AODV (3 packet drop + 3 packet delay attacker and 16 mobile nodes)
Figure 7.23	Comparison of Route Discovery Time with AODV and TMA-AODV (6 packet drop + 6 packet delay attacker and 16 mobile nodes)
Figure 7.24	Comparison of Throughput with AODV and TMA-AODV (6 packet drop + 6 packet delay attacker and 16 mobile nodes)
Figure 7.25	Comparison of Route Discovery Time with AODV and TMA-AODV (12 packet drop + 12 packet delay attacker and 16 mobile nodes)
Figure 7.26	Comparison of Throughput with AODV and TMA-AODV (12 packet drop + 12 packet delay attacker and 16 mobile nodes)
Figure 7.27	Comparison of average route discovery time of AODV and TMA-AODV in presence of mobile nodes and packet drop/delay attack
Figure 7.28	Comparison of average throughput with AODV and TMA-AODV routing in the presence of mobile nodes and packet drop/delay attacks

List of Tables

Table Number	Caption
Table 2.1	Comparison of possible security solutions for MANET routing
Table 3.1	Comparison of trust computation engine
Table 4.1	Comparative analysis of existing trust based routing schemes
Table 4.2	Abbreviations used in table 4.1
Table 6.1	Effect of packet drop attack on AODV routing and MANET
Table 6.2	Effect of packet drop and delay attack on AODV routing and MANET
Table 6.3	Effect of mobility on route discovery time of AODV and throughput of MANET
Table 6.4	Effect of packet drop and delay attack on AODV routing and MANET in presence of mobile nodes
Table 7.1	Comparison of AODV and TMA-AODV in absence of attack and mobility
Table 7.2	Comparison of average throughput and average route discovery time with AODV and TMA-AODV routing
Table 7.3	Improvement in Average throughput and Average route discovery time with TMA-AODV (6 packet drop attackers)
Table 7.4	Improvement in average throughput and average route discovery time with TMA-AODV (12 packet drop attackers)
Table 7.5	Improvement in average throughput and average route discovery time with TMA-AODV (24 packet drop attackers)
Table 7.6	Average Result obtained when Packet drop
Table 7.7	Improvement in Average throughput and Average route discovery time with TMA-AODV (3 packet drop +3 packet delay attacker)
Table 7.8	Improvement in average throughput and average route discovery time with TMA-AODV (6 packet drop + 6 packet delay attacker)
Table 7.9	Improvement in average throughput and average route discovery time with TMA-AODV(12 packet drop + 12 packet delay attacker)
Table 7.10	Average result obtained when Packet drop and delay
Table 7.11	Improvement in Average throughput and Average route discovery time with TMA-AODV (3 packet drop + 3 packet delay attacker and 16 mobile nodes)
Table 7.12	Improvement in average throughput and average route discovery time with TMA-AODV (3 packet drop + 3 packet delay attacker and 16 mobile nodes)
Table 7.13	Improvement in average throughput and average route discovery time with TMA-AODV (12 packet drop + 12 packet delay attacker and 16 mobile nodes)
Table 7.14	Average Result obtained when Packet drop+delay+mobility
Table 7.15	Improvement with packet drop attack
Table 7.16	Improvement with packet drop attack and packet delay attack
Table 7.17	Improvement with packet drop attack, packer delay attack and mobility
Table 8.1	Improvements with TMA-AODV compare to AODV

List of Appendices

Appendix A	Trust table related library functions
Appendix B	Manuscript of published papers

CHAPTER 1

Mobile Adhoc Network

1.1 Introduction

The wireless network is a network setup which uses radio frequency signals for communication among nodes of network [3]. It can be a Wi-Fi network, a wireless LAN, a Mobile Ad hoc network (MANET) or a Vehicular Ad hoc network (VANET). Nowadays the wireless ad hoc networks are getting popular for setting up network in laboratories, meeting rooms, hostel building etc. as they are easy to setup and no cabling or pres existing infrastructure is involved. The battery operated nodes like laptop, tablet, smart phone etc. can be both an end system as well as a router in such network.

1.2 Setting up wireless Network

Wireless networks are getting popularity because of easiness in its set up and usage. No cable is required to setup or use the network [4]. To setup wireless network, we need a node with wireless adapter and a wireless access point or wireless router as shown in following figure1.1 [3].

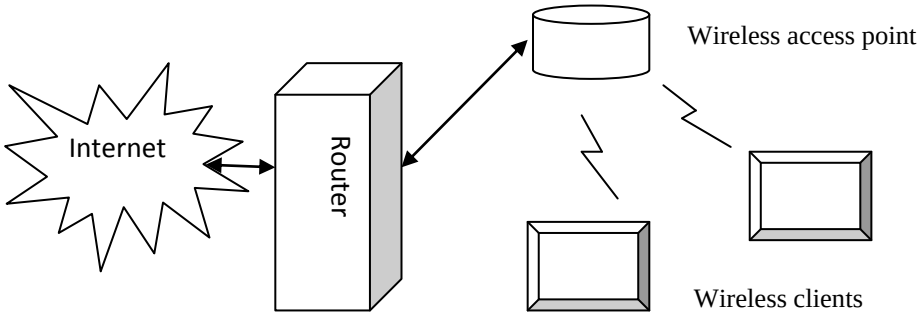


Figure 1.1 Setup of wireless network [3]

When a node in wireless network wants to send data, the binary data converted into a radio frequency signal and sent to the receiver node using its wireless adapter. The receiving node converts the radio signal back to binary data for processing[3]. The IEEE 802.11 standard declares two modes of operation of wireless network: Infrastructure mode and Ad

hoc mode. The network shown in figure 1.1 is the example of wireless network operates in infrastructure mode. In this mode each node of wireless network directly connects with its nearest access point (based on the range of the access point) and all access points are connected through an existing wired network. The wireless source node sends packet to its access point (access point with which node is associated), and it is the responsibility of access point to send packet to the destination node. In the Ad hoc mode all wireless nodes are directly connected with each other without the help of any access point. Wireless networks which operate in ad hoc mode are also called Ad hoc networks [3] [4] [5].

The infrastructure based wireless LAN solutions allows user with mobile devices to use internet provided at Malls, University, Airports and many other public places. This conventional wireless network needs fixed pre-existing infrastructure and central administration, which involved a lot of money and time to setup and maintenance [2].

1.3 Characteristics Mobile Ad hoc Network

Mobile devices like Laptop, PDAs, tablets PCs, smart phones, digital cameras, etc. becomes more lightweight, user friendly, cheap and powerful. This leads to a new alternative network where all mobile devices are connected with each other and communicate in Ad Hoc mode [2]. A source node can send message directly to all nodes which are within its radio range (S1 to D1). If the destination node is outside the radio range of the source node, the intermediate nodes are helping the packet to reach destination node as shown in following figure 1.2(S2 to D2) [8]. In figure 1.2, each circle represents a radio range of a node which is located in its center. All nodes in the network coordinate with each other for packet transfer. There is no pre-existing infrastructure required. No central administration is needed. Each node is self administrated and self configured. Network topology changes dynamically due to movements of mobile nodes. The mobile node can enter and exit the network at any time. Such a network is called Mobile Ad Hoc Network (MANET) [2] [4] [5].

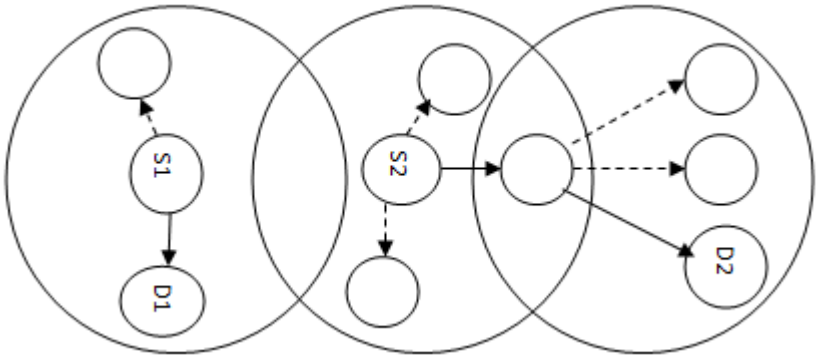


Figure 1.2 Multi hop routing in ad hoc network [8]

Mobile ad hoc networks are distributed in nature. It is a network of lightweight wireless nodes, which are battery operated, hence energy constrained. The ad hoc network is created on temporary basis. They use wireless media (radio signals) for communication. Each node in this network acts as a router. Routing in this network is difficult because of dynamic topology which changes with movement of nodes. MANET uses multi hop routing for packet forwarding. From the above discussion, we came out with following characteristics of Mobile Ad hoc network [1][2][5].

1) Self organising and self administrating network
2) No need of pre existing infrastructure
3) Dynamically changing topology
4) Energy constrained network
5) Limited bandwidth
6) Multi hop routing

1.4 Applications of Mobile Ad hoc network

The followings are some of the applications of MANET [2].

- Military communication and operation in battlefield.
- Automated battlefield
- Search and rescue operation after disaster
- Supporting doctors and nurses in hospitals
- Vehicular services- road and accident guidance
- Taxi- cab network

- Network of visitors at airport, conference and meeting room
- Home/Office equipment network
- Virtual classroom
- Multi user game
- Tracking animal movement in Zoo, sanctuaries etc.

1.5 History of MANET

The word ad hoc is a Latin word which means “for this only” [4]. In 1973 first generation Ad hoc network was developed by DARPA (Defence Advanced Research Projects Agency). DARPA started research on possibility of use of packet switching radio communication among computers and came up with packet radio network. This DARPA PRNET used from 1973 to 1987 was an experimental network, which was robust and reliable. They use radio frequency for data communication. 20 frequencies between 1718.4 MHz and 1840.0 MHz are used for channel selection. They use Omni directional, spread spectrum, half duplex transmission at 400kbps and reception at 100kbps. They collect various network parameters and used them to debug, and monitor the network. Routing protocols are also designed to assure correct, reliable and fast communication [1].

The second generation ad hoc networks were developed during 1980 to 1993. In 1980s as a part of the survivable adaptive network program an ad hoc network was developed. Their aim was to provide a packet switch network for battlefield elements without setting up any infrastructure. The GLOMO project had also developed an ad hoc network during this era. The aim of GLOMO project was to develop a mobile environment in defence information infrastructure by providing connectivity and data access services to wireless mobile users. They developed self organising and self healing network with a multi hop routing protocol. They use ATM over wireless for data communication. The other ad hoc network, developed as second generation ad hoc network was NTDR (Near term Digital Radio) system. They were experimental mobile packet switching radio network that linked Tactical operations centre in brigade area. They were also self organising and self healing. They used to provide a data transport facility to Army battle command system [1].

The third generation of ad hoc network started in 1990s. In 1990s with the invention of laptops and other wireless computing devices the concept of the commercial ad hoc network has introduced. The commercial use of mobile ad hoc network becomes popular

after Bluetooth and Sensor network. Different routing protocols suitable for mobile ad hoc network were introduced by the researcher. Devices, applications and protocols are developed for ad hoc network based on characteristics of the MANET. The MANET is widely used for various civilian applications like business, parking area, forest etc. [1].

1.6 Routing in MANET

Routing is one of very important basic operations of any network. Routing is a process of searching a path for a packet on which it travels to reach from its source to destination [13]. Routing techniques are designed based on the architecture and characteristics of the network [14] [8]. Mobile adhoc network are wireless network with all or some mobile nodes. MANET doesn't have any wireless router or access point. Each node acts as a router to send or receive data packets to or from the other nodes. In a MANET, the node uses multihop routing for communication. In a MANET, the routing is challenging due to the dynamically changing topology of a network as nodes are freely moving. Routing is one of the critical network operations in this network. The following unique characteristics of the mobile adhoc network make routing very challenging [8][13][14].

1) Asymmetric links: In MANET, the radio signals are used as a communication media. Two nodes can communicate with each other through radio signal. The wireless link between two nodes is not always bidirectional. Meaning, if a node A sends data to node B, it is not always necessary that node B can also send data to node A. This is due to the moving nature of the mobile nodes that breaks the communication link in-between. Due to such link routing will be very difficult [8][13].
2) Low bandwidth: The wireless channels use for MANET is low bandwidth channels. They are also shared by all nodes. The routing should be designed in such a way that it will add minimum routing overhead and spare more bandwidth for data transmission [8][14].
3) Resource constrained nodes: The nodes in MANET are battery operated. To ensure long life of the node, Routing protocol should be energy efficient [8][13][14].
4) Dynamically changing topology: It is a major problem while designing routing protocols for MANET. As the network topology constantly changing with time it is very difficult to update routing table or link information frequently without introducing network traffic or computing overhead [8][11][12][13]1[4].

5) Interference: Due to usage of wireless media, one transmission may affect or interfere other transmission in the same radio range. The nodes within same radio range can overhear the communication of each other. This should be taken care while designing routing protocols for security reasons [8][14].

In a MANET, the development of routing protocols started with two popularly used routing schemes for wired network: Distance vector and link state [14]. Distance vector algorithm needs less storage and computationally efficient compare to link state mechanism [14][17]. This protocol suffered from count to infinity problem due to loops [14]. This problem can be solved using various inter-nodal coordination mechanisms. With ad hoc network such coordination among network nodes become difficult due to frequent topology changes [13][14]. The Link state algorithm is not suffering from the looping problem [14][21]. They need to maintain latest link information of the entire network topology at each and every node. This introduces a huge communication and storage cost in case of mobile ad hoc network due to dynamically changing topology [14][21].

The routing protocols designed for wired networks, which has large routing overhead are not compatible for MANET because wired network has fixed topology and continuous power supply[13][17]. Routing in adhoc network is a very vast area. In this thesis, we are limiting our study to unicast routing only. In adhoc network nodes are not aware of the topology of the network. The main aim of routing protocol should be updating each node of a network about the other nodes[13][14][17]. This could be done by frequently broadcasting link information from each node and updating the routing table based on that. The node should announce its status to all other nodes and also get information about the status of other nodes. The mechanism used for constructing and updating routing table on each node is very important for an efficient routing scheme [13][14].

The all routing protocols designed for MANET are mainly classified into two broad categories: proactive routing protocols, and reactive routing protocol (shown in following figure 1.3) [8][15].

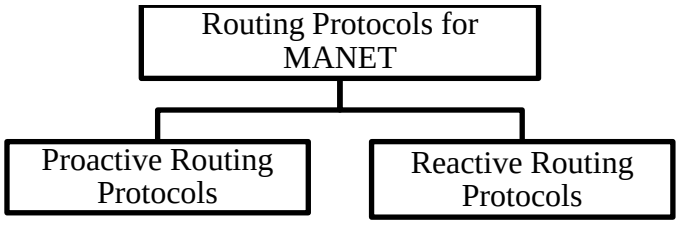


Figure 1.3 Categories of MANET routing protocols[8]

1.6.1 Proactive Routing Protocols

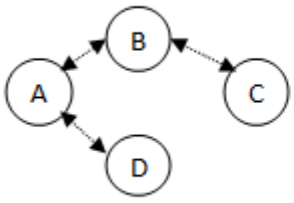
It is a table driven routing protocol. Proactive routing protocols maintain precise information about routes in routing table on each node [13][14]. It tries to evaluate all existing routes in a network and continuously update the routing table accordingly. Each node maintains up to date information towards all destinations and periodically broadcast this information to the entire network [13][14][17]. In this type of protocol when source node has data to send to any destination, route to that destination already found in its routing table and can immediately be used [14][17]. This makes data transmission fast. For maintaining up-to-date routing information on each node introduce a huge overhead in the network. Some of the most popular proactive routing protocols are DSDV and OLSR [13].

Destination Sequence Distance Vector (DSDV)

It is a table driven routing scheme. It is based on a bellman ford algorithm [81]. The contribution of this algorithm is to solve count to infinity problem [13][14]. Each entry in the routing table has a unique sequence number. If a link is being used then the corresponding sequence number is even, otherwise corresponding sequence number is odd [13]. The sequence number is generated by destination node and sent to all with a next update message. Route information is distributed among nodes of the network by sending the whole route table periodically by each node and update information more frequently [18]. Each node maintains a route table storing route towards all the nodes of the network [18]. The entries in the route table are destination node, next hop node, total hops for that destination, sequence number, and install time [13][14]. The example of the DSDV routing table is shown in figure 1.4.

In DSDV, when any node receives route to any destination, it compares the sequence number (new) of receiving route update with sequence number (old) stored for that destination in its route table [13][18]. If the new sequence number is greater than the old sequence number the route table updated. If both sequence numbers are same, the better

route (with less hop count) will be stored in the route table. This protocol is useful for MANET with small numbers of nodes [13]. The regular route update of DSDV needs battery power and bandwidth of network frequently so this protocol is not suitable with a highly dynamic network with large number of nodes [13].



Destination	NextHop	hops	Seq.No.	InstallTime
A	A	0	124	1200
B	B	1	3B6	1500
C	B	2	A42	1600
D	D	1	112	1700

Figure 1.4 Example of DSDV routing table on node A [13]

Optimized Link State Routing (OLSR)

This protocol is based on link state routing algorithm [21][22]. It is also a table driven algorithm just like DSDV [13]. To maintain changes in network topology, periodically messages are exchanged among nodes of the network. OLSR is an optimal way of implementing link state routing. To optimize link state routing, OLSR reduces size of announcement messages/control packets and number of transmission used for flooding the message in whole network [21][22]. OLSR uses a multi point selector to identify multi point relay (MPRs) nodes who flood the message to all nodes of the network [13][21][22]. It is not necessary to broadcast messages to all the nodes. Instead, the node multicast messages to MPRs. The MPRs are the nodes in network selected in such a way that they are connected with all nodes of the network [13]. Thus MPRs used to control number of transmission of link update messages when flooding [21][22]. We can use this protocol with a large number of nodes in MANET [13][21]. The limitation of this protocol is, they assumes that the link between nodes are bi directional which is not always true with wireless media (radio frequency signals) [21]. Also, sometimes the removal of redundant flooding becomes problematic when it is used with a network with large packet drop rate [21].

1.6.2 Reactive Routing Protocols

Reactive protocols search for a route when needed. They are not maintaining route information on each and every node of the network [13][14]. The search for a route is initiated by flooding a route request packet in the network when needed. They don't have the overhead of maintaining the global routing table [13][14][17]. They quickly react to topology changes of the network due to movement of the nodes [17][19][20]. Though there are advantages of reactive routing, they are having some limitation also. They introduce network traffic due to flooding mechanism [13][14]. The route finding process takes more time compared to proactive routing. Some of the most popular examples of reactive routing protocols are DSR and AODV [13][14].

Dynamic Source Routing (DSR)

It doesn't use periodic announcement by node to inform all other nodes about topology changes [11][12][13][20]. It computes and maintains the route when needed. In this scheme the sender node will initiate the route searching process, when it has data to send to any destination [20]. As a result of this search, sender obtain a complete sequence of nodes travelling through which packet can reach to the destination. This node sequence is stored in a packet header and used by each node forwarding a data packet to the next node until the destination node is reached [13][20]. There are two stages of DSR: route discovery and route maintenance [13]. A source node starts route discovery by broadcasting route request packet which is received by all the nodes available in its radio range. The destination node receives this packet and send route reply packet to source node with list of sequences of nodes from source to destination [11][12][13][20]. While route is in use the source node monitors the link status of the whole route [13][20]. This is called route maintenance. If any link break found during the monitoring, route discovery process started to discover the new route [13][20]. In DSR, the route is a part of the packet, so there is no problem of loops [13].

Adhoc On demand Distance Vector (AODV)

In this routing scheme, the route from source node to destination node is discovered only on demand [13][14]. AODV uses the concept of distance vector routing protocol [13]. It solves count to infinity problem using sequence number with each update message [9][13]. Actually AODV uses features of both DSDV and DSR routing protocols [13][14]. It uses

on demand route discovery and route maintenance processes from DSR and hop by hop routing and sequence number from DSDV [13][19]. In this routing each node maintains a route table which stores entries for all active routes toward destinations through neighbours and two counters: broadcast ID and sequence number [9][13][14].

When a node wants to send data to any destination, it will search for a valid route toward the destination node in its route table [9][13][14][19]. If route found, node will use that route for sending data packets. Otherwise the node initiates a route discovery process by incrementing its broadcast id and broadcast route request packet (RREQ) to its neighbours [9][12][19]. RREQ has source address, source sequence number (broadcast id), destination address, destination sequence number and hop count fields [13][14]. The broadcast id and source address uniquely identifies the RREQ. On receiving an RREQ, the neighbour nodes check for valid route toward the destination nodes mentioned in the RREQ in their routing tables. If any neighbour node has a valid route to the destination node, it creates RREP packet and sends it back to the source node on the same path from where the RREQ comes [9][13][19]. If no valid route available at any neighbour, they further broadcast RREQ to their neighbours. While RREQ travels from one node to other nodes, each node sets up a reverse path towards source node by recording address of node from where RREQ comes. This way of keeping track of the path is called reverse path setup [13][14][19].

Once RREQ reaches to a destination node, the destination node generates RREP with destination sequence number and send it back to the source node using reverse path [9][12][13][14][19]. When RREP travels back to source node each intermediate node will record the node address and destination sequence number to store forward path from source to destination. This is called setting up of forward path [13][14][19]. After receiving the RREP packet (old) if an intermediate node receives other RREP (new) for the same route request, it checks for a destination sequence number of both RREP. It records and process new RREP only if the destination sequence number is larger in new RREP or both destination sequence numbers are same with low hop count in new RREP. This ensures loop free route [12][13][14]. When a source node receives RREP packet, it stores it in its routing table and use the route for sending data packets. During data transmission, all nodes in a route send hello packets to their neighbours to maintain the route [9][13][14][19].

If any intermediate node of an active route leaves the network due to movement or failure of a node, it will not send hello packet and, hence link break found [13][19]. The node who will detect link break will send the RERR packet to source node about route failure [13][19]. After receiving RERR packet source node will reinitiate route discovery process by broadcasting new RREQ packets [13]. The following figure 1.5 and figure 1.6 shows the flow of the route request and route reply packets in AODV [23] .

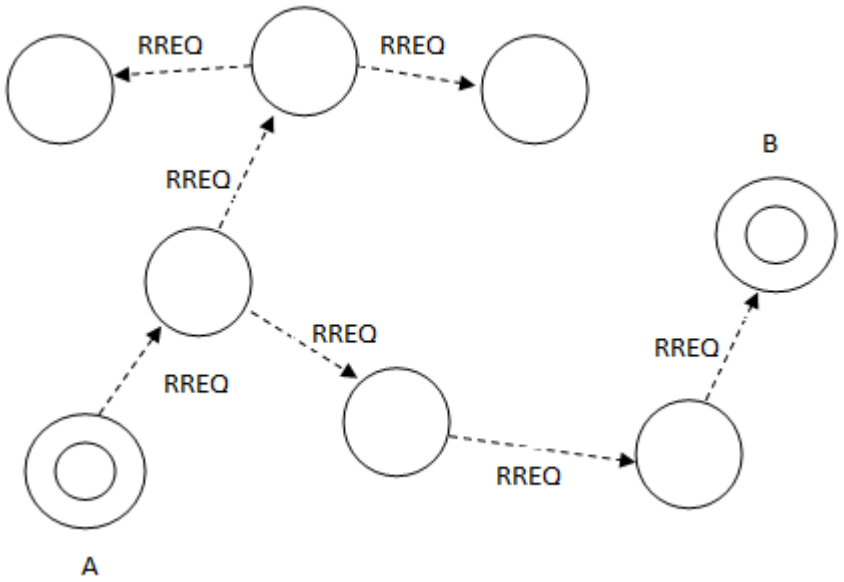


Figure 1.5 RREQ packet broadcasting Source node A and destination node B. [23]

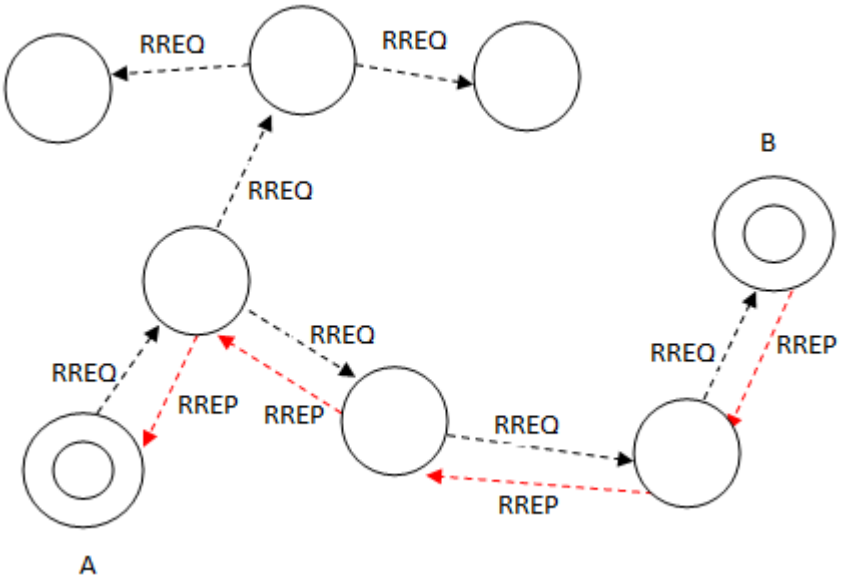


Figure 1.6 RREP packet from destination B to source A [23]

Though AODV is one of the widely used routing protocols in MANET, it also has some limitations [14]. It has large route formation latency. It is vulnerable to misuse by internal as well as external nodes [13][14].

1.6.3 Other Routing Protocols

Apart from these two types of routing protocols, researchers also come up with other routing protocols. Some routing protocols are proactive as well as reactive (hybrid routing protocols) for example Zone Routing protocol [10]. In this routing the nodes are distributed among zones and to find route within a zone (intra zone) they do proactive routing and to search, route outside the zone (inter zone) they use reactive routing [10]. Some routing protocols are position based routing protocols like GRP (Geographic Routing Protocol)[6]. In GRP, each node finds out its own position and the position of other nodes using GPS devices, when it need to send data and directly send packets in the direction of destination node via intermediate nodes [6][7]. In GRP there is no need to maintain a routing table [6][7]. The main approach of Geographic Routing Protocol is greedy forwarding. In greedy forwarding approach packet will be sent to the node which is the nearest neighbour of the destination node [7]. The intermediate nodes also use the same approach until a packet delivered to the destination node [7]. This approach fails when a packet reaches to a void node. The void node is a node with no neighbours near to the destination node [6][7].

1.7 Selection of base routing protocol for research

After studying mechanism, advantages and limitations of all above fundamental routing protocols used for MANET, we came to the following conclusion regarding their application in the wireless network scenarios.

DSDV: table driven, topology changes advertise/broadcast by the node, low route discovery latency, works well when a small number of network node and less mobility [14].

OLSR: table driven, optimize by reducing control packet size and controlled flooding (removing redundancy in flooding) of link information using MPR (Multipoint relay) nodes, low route discovery latency, works well with large network also. When the network has a large packet drop rate some node may never get control packet due to removing redundancy in flooding [13].

DSR: on demand, works well with high mobility, more routing traffic and routing overhead when used with low mobility, no routing table maintained, large route discovery latency [13]

AODV: on demand, route table maintained at each node, work well with high mobility, moderate mobility and no mobility, compare to DSR low route discovery latency [14]

In this thesis, our goal is to design a mobility aware trust based routing protocol for mobile adhoc network. This protocol should work well with and without mobile nodes in the network. DSR works well with highly dynamic environment [7][11]. With low or no mobility it unnecessarily produces large routing overhead [7][13]. DSDV and OLSR works well with low mobility (less topology changes) [5][13]. In case of high mobility, they introduce large overhead [13]. AODV works well with both low mobility and high mobility in the network, hence we chose AODV routing protocol as a base protocol for our thesis [5][7][11][12][14].

CHAPTER 2

Security Challenges In Mobile Adhoc NETwork And Their Possible Solutions

2.1 Introduction

Mobile Adhoc Network (MANET) is self administered network of mobile nodes [1] [2]. There is no permanent framework or topology of this network [1] [2]. After the invention of IEEE 802.11 and Bluetooth technologies the commercial utilization of MANET is possible[1]. MANET can be used in various sensitive applications like home automation, rescue operation, education, etc. [24]. These applications demand secure network operations to provide their services. MANET is highly susceptible to security attacks at each network layer due to its unique characteristics and hostile infrastructure [24]. In such hostile environment the node cannot be monitored and open to attack by attackers [24] [25].

2.2 Challenges in MANET

The unique characteristics of MANET, which we have discussed in chapter 1 introduced many designing challenges at each layer of the network [24]. At the physical layer, we must take care that node should cope up with frequent changes of links due to movement of nodes in the network [24]. At the data link layer, we must deal with hidden and/or exposed nodes, fair access of bandwidth and packet collision as a part of Media Access Control [24] [25]. At the network layer proper routing scheme should be defined to ensure data packet forwarding from source node to destination node [24] [25]. At transport layer the packet loss and packet delay due to dynamic topology and wireless media must be handled properly [24]. Application layer protocols should be designed to address frequent disconnectivity issues [24]. Apart from the challenges in designing basic network operations at each layer, MANET has also many security related issues due to its characteristics like dynamically changing topology, use of wireless media, bandwidth

sharing, no controlling authority/ administrator, multihop routing, energy constraint nodes [24]. Also, during routing a source node has to take the help of intermediate nodes for packet forwarding to the destination node [9] [13] [14]. If any intermediate node is not cooperating i.e. not forwarding packets to save its battery life, may create problem during routing [24][25]. Such intermediate nodes can also be attacker nodes and may expose the network to security attacks [24][25].

2.3 Security Requirements

2.3.1 Hard Security Services

Same as the other networks, a security goal of MANET is to provide authentication, confidentiality, integrity, availability, and non-repudiation [26]. These services are called hard security services[75].

Authentication: It is necessary because attacker node can masquerade as a trusted node and can get private information or disturb the operation of the network. In the network, each node must verify identity of another node before starting a communication [26][28].

Confidentiality: Illegal access of data transmitted in the network and eavesdropping should be handled to ensure the confidentiality. In a MANET, sensitive data may pass through many intermediate nodes before reaches to the destination node. Cryptographic algorithm can be used to ensure confidentiality[26][28].

Integrity: Integrity assures that data transmitted on the network cannot be modified by any attacker node which is an intermediate node between sender and receiver nodes. Various one way hash algorithm along with cryptography algorithms can be used to ensure integrity of data[26][29].

Availability: The node or a network service will be unavailable due to Denial of service attack or misbehaviour of a node in the network. An attacker may use jamming technique at the physical layer to jam the network at the physical layer. Availability ensures that the nodes and services must be available for the legitimate user of the network [26].

Non-repudiation: This requirement ensures that a message received at destination node must be from specific source only and source node ensures that the message must be received by the specific destination node only. It ensures the action committed between two nodes cannot be denied. This can be implemented using digital signature [26].

2.3.2 Soft Security Services

The MANET is an open system. It is not administrated by any central authority. In such system, it will be very difficult to define security policy. We need to define ethical norms for the participants of such open systems. We need to design a system which enforces these norms. This introduced soft security services like reliability, access control, quality of information and malicious/malfunctioning activities [76]. The conventional security approaches cannot be used to implement soft security services [75][76]. It can be effectively implemented using trust based approaches [76].

2.4 Threats

In a mobile adhoc network, the mobile nodes are not in control of any centralized administrator [1][2]. Each node works independently and communicates with other nodes of the network using radio signal [1]. The attacker nodes may take control on any mobile node and compromise the security of the whole network[24][26][27]. Due to the usage of radio signal, the nodes within same radio range can overhear the traffics of each other[2]. This also may lead to various attacks [26]. Unlike wired networks, attacker node need not get physical access of network[26]. Use of wireless link makes network traffic susceptible to eavesdropping and interferences by attackers [26][27].

In a MANET, nodes can enter or exit the network at any time and from anywhere. Nodes can also move freely. This leads to frequent changes in topology of the network. In such scenario, it will be very difficult to differentiate between normal behaviour and malicious behaviour of nodes [24][26][27].

The other threat on basic operation of MANET is misbehaviour of intermediate nodes. The intermediate node does not cooperate in routing by not following standard routing operation specifications to disturb the normal network operation [24][25][26].

2.4.1 Attacks (Hard Security Threats)

The security attacks on MANET are classified into two main categories: active attacks and passive attacks [24]. Passive attacker nodes only study the traffic pattern between two nodes. By studying traffic they try to discover vulnerable information and use them to damage or disturb the network [26]. It is very difficult to detect passive attacks because they do not make any changes in observed packets[26]. The active attacker nodes are the

attackers which disturb the network operation by changing the network traffic, i.e. changes packet content [24][25][26]. Active attackers can perform deletion, modification, replication, redirection or fabrication in data or control packets travelling in the network[26]. The nodes which intentionally attack the network operation are malicious nodes [26].

Passive Attacks

They are non destructive attacks. They will not disturb the regular operation of the network. They study the network traffic for knowing channel usage pattern, the location information of node, packet content and address information of node[25][26]. In a MANET, this observation is easy because all the nodes within the same radio range share the bandwidth[26]. The attacker node will collect all observed information about the node/network to attack the network in future. It is very difficult to detect such attacks. The eavesdropping, traffic analysis, spoofing, etc. are the passive attacks [26].

Eavesdropping: It is an attack on the physical layer of the MANET. The goal of this attack is to read and get secret information like password, PIN, private key, address information, etc. from the packet [26]. The other data which attacker interested are size of data exchange, time of data transmission, frequency of communication between two nodes, etc. [26]. This information can be used by an attacker to disturb network in the future [26][28].

Traffic Analysis: It is a data link layer attack in MANET. Attacker node listens traffic of others using wireless card which operates in promiscuous mode and some special software to analyse the frequency and load of traffic in the network[26][28]. This analysis is used to find the topology of the network, location of the node, activities by the node, the role of the node in activity, etc. [26].

Active Attacks

These attacks disturb the routine operation of the MANET. The attacker modifies the content of packet and resend it to get access of the resource as an authenticated user [28]. These attacks can be detected using several mechanisms [26].

Jamming: It is a physical layer attack in MANET. In this attack, to disturb the communication between two nodes, the attacker will send signals with the same strength and speed of that communication [26].

Sleep Deprivation Attack: The attacker node targets a node of the network and continuously sends packet to it to keep it busy in sending or processing packets[26][28]. Thus, the battery power of nodes and bandwidth of the network is not properly utilized[33].

Black Hole Attack: It is an attack on network layer on MANET. This attack actually disturbs the routing protocol of the MANET. The attacker node declares a valid and shortest route to the destination and enforces a source to follow that route. When the data packets come to the attacker node, it will be modified or dropped or mistreated by attacker node[24][25][26].

Greyhole attack: This is an attack on network layer of the MANET. This attack has two phases. In the first phase, the attacker node advertises itself having an optimal route to the destination node and be an intermediate node of a route during route formation[31]. In the second phase it will interrupt some of the incoming packets from source with a certain probability[31]. It may drop packets to certain destination while forwarding all other packets. Another way of doing greyhole attack is to drop the packet during some time period, and then switch to normal behaviour[31]. Due to such pattern of attack this attack is difficult to detect [31].

Worm hole attack: This is also a network layer attack. Two attacker nodes coordinate with each other to attack the MANET. These two nodes established a worm hole link and tunnel routing traffic on different position of network[30]. This attack modifies the topology of the network and creates routes which actually do not exist [26][28].

Node Isolation attack: It is an attack which disconnects communicating nodes from the other nodes of the network. These attacker nodes do not cooperate in spreading link information in link based routing and hence isolate some specific nodes of a network from the other nodes. It is an attack on the routing in MANET [26].

Denial of Service attack and flooding attack: These are attacks on data link layer of the network. This attack makes the network useless by consuming all resources or by utilizing whole bandwidth in some useless tasks[28][33]. The attacker node inserts a lot of fake packets in network. Hence the performance of the network is degraded [26][28].

Rushing attack: This is an attack on demand routing protocols in MANET. During route discovery wherever attacker node received route request packet, it immediately forward

this route request to all other nodes in network[32]. The forwarding is done in such a quick manner that the legitimate route request reaches late. Thus, the attacker node will be an intermediate node of each route and source will never get secure route [32].

Spoofing Attack: It is an activity of attacker node, which hides its own identity by changing the IP address or MAC address in packet to perform an attack. It may disturb the current communication in network without disclosing its identity [28].

2.4.2 Misbehaviour (Soft Security Threats)

In a MANET, the routing algorithm assumes that all nodes are working in cooperation with each other during route formation [24]. This assumption is not always true. Routing protocols designed for MANET are not enforcing cooperation or coordination of nodes[24][26]. Sometimes the nodes in MANET act selfishly. Instead of cooperating in routing, they reject to forward the packet further in the network and save its own resources and battery life[24][25][26]. They are not disturbing operations of other nodes, but they simply show selfish behaviour. They are using the facilities provided by all other cooperative nodes and consume their resources, but don't use its own resources for other nodes. Such nodes in MANET are misbehaving/ selfish nodes[24][25]. To detect such behaviour, we need to define some ethical norms for system and enforce the nodes to follow these norms [76]. The trust based approaches can be best solution to detect the soft security threats [75][76].

2.5 Countermeasures

To secure mobile adhoc network, our first attempt should be to prevent the attacks whenever possible[24]. However, in some cases, prevention mechanism cannot work. We should use the detection mechanism to detect the attacker nodes and inform the nodes of network about this attack. The prevention approach is a proactive approach while detection approach is reactive solution for securing MANET [26].

In this thesis, we are addressing the attacks on the network layer, which are the attacks on routing and packet forwarding [24][25]. In a MANET, at network layer both data packets and control packets are vulnerable to the attacks [24].

2.5.1 Prevention Techniques: Cryptographic approach

Most of the active attacks described in section 2.4 can be prevented using authentication techniques with routing protocol in MANET. If any node wants to participate in routing, it has to be an authenticated user of the network[26]. The authenticated user is a node of network, which ensures to follow the specification of routing protocol and not misbehave [26]. At each routing phase, the protocol enforces authentication and thus, avoid the unauthenticated or attacker nodes to participate in routing and implement attacks on the network [26].

For authentication, we have to use cryptographic techniques, which can be symmetric (private key) cryptography and asymmetric (public key) cryptography [26][34]. In symmetric cryptography node uses a cryptographic algorithm to generate digital evidence of user authorization of the packet using a secret key[26]. One example of such algorithm is MD5 algorithm which generates Message Authentication Code (MAC) using the message and the secret key and send this code with a message which will be verified for authenticating user at the destination using the same secret key[34][35][36]. Ariadne uses symmetric cryptography to authenticate node and protect data [34]. Secure routing protocol [35] assumes a security association between source and destination node. With each route request packet from source to destination is appended with message integrity code (MIC) computed using HMAC algorithm and a secret key (already shared between source and destination node)[35]. At destination node, the MIC is regenerated using HMAC algorithm and secret key and compared with MIC with the route request packet. If match found, the packet is coming from valid user. Same way destination also sends a route reply packet with MIC code which will be verified at source node [35]. SEAD (Secure Efficient Ad hoc Distance vector) routing protocol secures DSDV routing protocol [38]. It uses one way hash function to authenticate hop count in advertised route messages and route updates[38]. Each node also authenticates the sender of update message using message authentication code. SEAD protocol assumes that the secret key is already shared between the sender-receiver pair [38]. SAODV also uses message authentication code to secure fixed field of routing control packets and one way hash function to secure hop count [39].

In public key cryptography, the node generates digital signature and send it along with a routing packet for authentication. At receiving end this signature will be verified to prove the identity of the sender node [26]. The main problem of usage of public key

cryptography with MANET is a key distribution problem. In a MANET, there is no central administrator. Each node works independently. So, services like certification authorities, key servers, etc. are very difficult to implement in MANET [26].

The key distribution problem is addressed by many researchers using distributed key management systems [41][42][43][44[45]. Some researcher come up with an n-party version of Diffie Hellman key exchange algorithm in which n nodes combinely helps to generate a common key at each node [40][41]. Some research suggests usage of encrypted key exchange protocols [42]. In this scheme, a secret key is generated using shared passwords. A most common problem with all these solutions is, some initial information must be shared and distributed among each node of a network before using the protocol [26]. [43] proposed a public key infrastructure (PKI) for MANET. It uses distributed certificate authority concept. A subset of nodes of MANET works as a server and collectively act as a Certificate authority [43]. Each CA generates the partial signature and collects a partial signature from all other servers [43]. MOCA framework creates a cluster of CA by accessing the security and other physical characteristic of the nodes [44]. [45] came up with Maximum Degree Algorithm (MDA) which is a completely independent concept of public key management scheme for MANET. It allows each and every user to generate their public-private key pair and to perform authentication [45]. It does not require any Certificate Authority. This scheme is used for only data security. We cannot secure multihop routing using MDA and MOCA [44][45].

2.5.2 Detection Techniques: Intrusion Detection System based approach

Prevention techniques can be used to provide security services like confidentiality, authentication, non repudiation and integrity. This technique is not suitable to provide availability service in MANET. Also, when the attacker modifies its patterns, it will be difficult to prevent them. In such cases detection of attack approach works. Intrusion Detection System (IDS) is designed and installed in the network, which continuously monitors the incoming and outgoing traffic on the network to identify malicious activities [26][27]. Once an attack is detected, the IDS will start process for preventing it or process to minimize the damage from that attack[26]. IDS also inform us, about the techniques attackers used to perform attacks. After detecting attacks, it will also suggest possible prevention or mitigation approaches[27]. IDS designed for wired networks cannot be used with MANET. The traditional wired network uses IDS to provide protection from external

attacks [24][25][46]. In a MANET, each node is independent. The wireless channel used for communication in MANET is also open to all. There is not any point/place in MANET from where we can monitor the activities of the whole network. There is no clear line of separation between inside and outside network[25]. In a MANET all network operations are performed assuming that all nodes are cooperative. In such hostile environment, it is not always necessary that, a malicious node disturbs the network. Sometime a legitimate node may also refuse forwarding packet due to limited battery power, hardware failure, etc. Thus, IDS should be specially designed for a MANET to keep all the above discussed characteristics of the MANET in mind [24][46].

In this section, we will discuss various IDS designed for the MANET.

Specification based IDS: They are used to detect the activities of a node which is not as per the specification of routing protocol in MANET[26]. If any node is not working as per the specified rule of routing protocol, an intrusion is detected[26]. In [47] the states of the FSM (Finite State Machine) are defined based on AODV specifications for the route discovery process. Each node maintains a forwarding table for each neighbour node. Each route request and route reply packets sent within a range of node is monitored by a node. The node analysed the packet and check if specifications are followed or not. A specification based IDS are used to detect the modification or forge attacks [27][47]. These IDS can not detect the attacker nodes which performs the attacks but does not violate the specification of routing protocol directly like some DoS attacks[26].

Anomaly based IDS: They are used to detect abnormal behaviour of the system like CPU usage, execution frequency of a command, etc. [26]. To identify abnormal behaviour, we have to define normal behaviour of the system. The normal behaviour of the system changes frequently with time. Sometime normal activities are also identified as abnormal activities (false positive). However, this IDS is useful to detect unknown attacks. [48] introduced anomaly based IDS which monitored incoming and outgoing packet frequency on a node to identify blackhole and other dropping attacks.

Misuse-Based IDS: They use signature of known attacks to compare them with current activities on node to identify the attack [26]. It is an efficient IDS with low false positive. The only limitation of this IDS is, they can not identify new attacks[26]. They maintain a signature database of attacks which should be frequently updated with new attack

signature[26]. In [49], the IDS is proposed for AODV routing to detect dropping, flooding and route disruption attacks.

Promiscuous Monitoring Based IDS: In MANET nodes uses wireless media and they can overhear traffic within their communication range [24][25][26]. If a node operates Promiscuous mode, it can overhear packets to or from all their neighbours within its radio range [26]. In promiscuous monitoring based IDS, the node detects the malicious activities of their neighbours by monitoring their traffic [26]. This scheme used to detect packet drop and packet modification related attacks. In [50] authors introduced a Watchdog and Pathrater techniques to detect packet drop and modification with DSR routing.In [51] the Watchdog and Pathrater method was extended to detect attacks with other routing protocols also.

The conventional security solutions like cryptography and IDS can not easily be used with MANETs [25][26][27]. None of the solutions, which we discussed in the above paragraphs is the best solution if we consider the characteristics of MANET [27]. All solutions are specific to routing protocol and prevent/detect specific attacks only [26][27]. No solution works well if we consider limited resource of a MANET node [27]. The researcher should develop a solution to keep the features of the node and the characteristics of MANET in mind [27].

2.5.3 Trust based approach

Trust based routing protocols are developed by many researchers for wireless ad hoc network [52-66]. In trust based routing, each node observes the behaviour of each of its neighbour and records them. The observed parameters are used to calculate trust value of a node using mathematical models. Before communicating with any neighbour, a node will check whether the node is trustworthy or not. For that, node calculates trust value of that node based on recorded parameters and makes decision. All existing trust based approaches are discussed in section 4.2 and compared in section 4.3.

2.6 Selection of Trust based approach for detecting attacks for this thesis

In a mobile ad hoc network the nodes are battery operated and processing capability and memory of the mobile node is limited [24]. We need to concentrate on a security solution

which is light weight and does not incur more overhead and communication cost [27]. The desirable features of a security mechanism should be lightweight, distributed (decentralized), reactive and fault-tolerant [28].

The cryptographic algorithms are computationally complex. Thus, usage of a cryptographic algorithm in MANET introduces large computational overhead [26][27][34]. The cryptographic solutions are binary solution. They either provide full security or no security. In a MANET, the misbehaving nodes or selfish node can not be easily detected using cryptography based solutions. The nodes behave as a legitimate user of system and cooperate in network operation initially and hence passed the cryptographic security check [76]. However, after some time, nodes may act maliciously or selfishly [76]. This may be due to hardware failure or misbehaviour of node. The cryptography based algorithms can not detect/ prevent such behaviour [75][76]. Hence, cryptography can not be an effective approach for securing MANET[75]. The trust based approach can be the best solution for enforcing nodes of MANET to behave normally and avoid soft security threats discussed in section 2.4.2 [75][76].

To provide security in MANET, the attacks should be prevented or detected [26]. The attacks on network layer should be handled to avoid network failure [24][26]. To provide security to routing protocols, researchers come up with many solutions based on (1) Cryptographic approaches (2) Intrusion Detection System based approaches and (3) Trust based approaches[27]. The table 1.1 shown below gives you comparison of these three approaches.

Table 2.1 Comparison of possible security solutions for MANET routing [27]

Security solution / Attacks/ Design challenges	Cryptographic solution	IDS based solution	Trust based solution
Drop	Yes	Yes	Yes
Delay	No	Yes	Yes
Data modification	Yes	No	Yes
Link break	No	Yes	Yes
Battery life	No	Yes	Yes
Computation overhead	Large	Large	Relatively low
Administrative authorities	Required	Required	Not Required
Soft security Threats	Not address	Can address	Can address

The Cryptographic approaches and IDS which we discussed in section 2.5, need a lot of computational overhead, which is not appropriate for the resource constraint mobile node. Also, for IDS we need to continuously monitor the network traffic. Unlink wired network, we don't identify a single point from where we can monitor whole network traffic. In a MANET, each node has to monitor network traffic. This also adds overhead on each node. The trust based approach uses network parameters which are recorded, stored and used to calculate trustworthiness of node only when required. This avoids unnecessary overheads of sniffing of the packets or computational overhead of calculating public-private keys as with cryptographic solutions. Hence, at a first glance, the trust based approach seems to be the best approach. Compare to previous two approaches trust based approach is more suitable for securing mobile ad hoc network from network layer attacks. Many researchers introduced trust based routing protocols for MANET which are extensions of existing routing protocols [52-66]. The goal of our research is to develop a secure routing protocol, which avoid the consequences of the link breaks due to mobility of the nodes. At network layer most destructive malicious activities by the attacker is packet drop and packet delay of data and control packets. In this thesis, we aim to detect packet drop and packet delay attacker nodes using trust based approach. We also want to add a mechanism to avoid the nodes which are more involved in link break due to mobility, during route formation. After studying table 2.1, we opt trust based approach to design our proposed secure routing scheme, which is thoroughly discussed in the next chapter.

CHAPTER 3

Trust Management

3.1 Trust

Trust is a very important parameter for a social life. In social life, we can see trust among people who are working in coordination and cooperation for the benefit of each other. Social trust can be based upon the past experience or the reputation of a person [68]. Our trust on a person constantly changes with time as we are dealing with him/her in different circumstances [67]. In Business, trust is supported by a legal framework. The person will be punished with legal action and destroy his reputation, if not trustworthy in the financial issues of business [67]. The existing legal framework is not globalized, hence it will not be useful for electronic commerce [67]. With rapid usage of electronic commerce and distributed computing, the security of digital transaction of users will be very important [67][68]. Electronic commerce and distributed applications are used globally so, it will be very difficult to apply the legal framework of a specific state or country. Thus, the need of trust relationship between digital entities who interact for any one or both entities' benefit should be important. The development of a system using which we can establish trust relationship between digital system is very critical [67][68]. It can be useful for MANET, peer to peer systems, grid computing and cloud computing services which need security and privacy due to its unique application and characteristics[67][69]. It is very difficult to collect fair information to calculate trust of a digital system in an open environment, where the entities can be easily attacked. It is very challenging to implement a trust model which performs the mapping of social trust into digital trust [67]. Trust is used to improve digital system security and privacy. To incorporate trust in a digital system, a number of problems and their solutions should be addressed [67][69][70][71][72][73]. For example, trust modelling, trust evaluation, reputation system, trustworthy user interface design, etc.[67][68].

3.2 Definition of trust

Social trust is, a person's belief on the other person or product [67]. The social trust has two aspects: cognitive and affective [68]. According to [69], trust is a qualified belief by a trustor with respect to the competence, honesty, security and dependability of a trustee within a special context. Jøsang [74] defines trust in two ways: Evaluation trust and Decision trust. Evaluation trust is a belief that an entity will perform in the expected way, i.e. will bring expected benefit and not make any unexpected harm [74]. Decision trust is a level of trust in which one entity is depending on other's decision and action accepting risk of negative outcomes [74].

3.3 Digital representative of Trust

In a trust computing system, trust is viewed as a property of a system [67][74]. This property should be modelled, specified and accessed[74]. Trust can be used to assess entity based on specified standard within a context. Trust involves two entities: a trustor and a trustee[74]. The trustor is one who wants to assess trustworthiness of an entity and take decisions based on that assessment within a given context [74]. A trustee is an entity which is assessed by others to decide whether to use it or not. The trustee should put light on its own honesty, reliability, quality of service, etc. positive features [67][74].

In digital systems, trust can be represented as reputation, rating or recommendation [68] [74]. Reputation is generalized belief of community on the trustworthiness of a specific entity. It is computed based on past experience of the entity with others within a specific context [67]. The rating is computed based on trustor's own experience with trustee based on past experience [68]. The recommendation is the opinion of any trusted entity about the trustee [68]. Reputation can be used by an entity when it is newly entered or doesn’t have any past experience with trustee and not trusting any other nodes for recommendation [74]. The rating is the direct calculation of trust using recorded past experiences [68][74]. The recommendation is used when a trustor doesn't have any direct experience with trustee [67][68].

3.4 Trust factors

The factors which influence the trust of trustor on trustee can be objective as well as subjective properties of trustee [67][74]. The objective properties of trustee which

influence trust is integrity, reliability, competence, security, dependability, timeliness, behaviour and strength [67]. The subjective properties can be honesty, kindness and goodness [67]. This also depends on the context, i.e. a situation or environment in which the trust is assessed [74].

3.5 Trust Metrics

To evaluate trust, different metrics can be used. The trust metric can be classified in following categories [75].

1) Trust scale: It uses continuous or discrete values to measure the level of trust [75]. With this approach, one may use threshold based approaches to decide the trustworthiness [75]. For example, if the calculated trust value of an entity is more than the threshold, the entity will consider trustworthy [75].

2) Trust facets: It uses more than one values to define trustworthiness of an entity. In [76], the triplet (b, d, u)$\in$[0,1], where b + d + u = 1 is used to represent trust. The b, d, and u stand for belief, disbelief, and uncertainty respectively.

3) Trust logics (probability, fuzzy): It uses a probabilistic approach to measure trust. For example, the ratio between the total number of successfully forwarded packets and total number of incoming packets on a node can be as trust metric [67]. Some researchers also use beta distribution in which negative events and positive events are used to calculate the trust value [52] [53] [58] [63] [64]. Fuzzy logic can also be used to represent trust. In this approach a range is decided and a label is assigned to each range. For example, [-1.0 to 1.0] can be used for very low trust, [1.0 to 2.0] used for low trust, [2.0, 3.0] used for moderate and so on [61].

3.6 Characteristics of trust

The most important characteristics which play important role in trust modelling are [67][74]:

a) Trust is directed: It is a direct relationship between a trustor and trustee.
b) Trust is subjective: Trust on same entity changes from one person to another. For same person if the situation changes, trust will also be changed with time.
c) Trust is context-dependent.

d) Trust is measurable.
e) Trust depends on history.
f) Trust is dynamic.
g) Trust is conditionally transferable.
h) Trust can be a composite property.

3.7 Trust modelling and Trust management

It is a procedure of specifying, assessing and setting up a trust relationship among entities of a digital system by calculating trust [67][75]. There are three sources of trust modelling process: experience, recommendation and reputation [67][68][74]. It is a technical method using which, we can digitally represent trust. It is deployed by trust management process. Trust management is a continuous and automated process of collecting the required information to calculate trust value for making trust relationship decision, and evaluating that trust relationship based on criteria [67]. It includes four aspects: trust establishment, trust monitoring, trust assessment and trust control and reestablishment [67]. Trust establishment is a process of gathering evidence of an entity and establish trust relationship between a trustor and trustee using calculated trust value[67][76]. In trust monitoring, trustor continuously monitors the activities of trustee to collect evidence for trust assessment[67]. Trust assessment is a process which assesses the current trust relationship between a trustor and trustee and decide to change them or not[67][74]. In trust control and reestablishment, the trustor will take necessary action to control and re-establish trust relationship if it is broken or will be broken [67]. From the above discussion, we can conclude that the trust management is a combination of trust modelling and trust evaluation [67]. For actual trust modelling, we may consider following attributes of the system [67][68][74].

- Recommendations, reputations, feedback from others.
- Personal experience
- Reputation of trustor.
- Context factors like time, distance, transaction context, community context
- Policy factors like accepted level of recommendations.

3.8 Trust Computation Engines

Trust computation engines are used to aggregate various observations collected by a node to calculate the trust value. The most popular approaches are summation model, average model, Belief model, Fuzzy model and Bayesian model [74] [77].

3.8.1 Summation model

It is the simplest way of calculating the trust value from collecting evidence. It simply adds the observed parameter's value to calculate direct trust. For indirect trust it collects opinion from others and add them to calculate the trust value [74]. The equation (3.1) [74] can be used to calculate trust using this model. Tx(Y) is a trust of node X on node Y. Pi(Y) is the value of parameter P_i observed by node X for node Y.

$$\mathrm{Tx(Y)} = \sum_{\mathrm{i=0}}^{\mathrm{n}} \mathrm{Pi(Y)} \qquad (\mathrm{eq.}\ 3.1)$$

Where n is the number of parameters observed and P_i is the value of the parameter.

The weighted sum model is a variation of summation model which is widely used by researcher to calculate the trust value based on the importance of parameter in trust [65]. In this model a weight factor with a value from 0 to 1 is associated with each parameter showing the priority or importance of that factor based on the application for which trust value is calculated. For this, actual parameter value is multiplied with its associated weight factor and they are added in final trust value as shown in equation (3.2) [65]. Here, n is the number of parameters observed and Pi is the value of the parameter and Wi is a weight factor associated with that parameter. The sum of all weight factors should be 1.

$$\mathrm{Tx(Y)} = \sum_{\mathrm{i=0}}^{\mathrm{n}} \mathrm{Wi} * \mathrm{Pi(Y)} \qquad (\mathrm{eq.}\ 3.2)$$

3.8.2 Average model

It is also a simple approach to calculate trust value of a node. In this model, the average of all observed parameters is calculated for computing direct trust value. Also, for the indirect trust value, we may do average of opinion /recommendation collected from all nodes [74].

3.8.3 Bayesian model

To make any important decision, an entity takes an advice from other entities who have expertise in the field or knowledge. These experts also give their advice based on accumulated knowledge, experience and other information [78]. The automation systems that take such decision are called expert systems. Probabilistic model can also be used to implement an expert system, in which we can consider the uncertain expert knowledge to take a decision. Probabilistic model can use either classical approach in which based on repeated trials probable outcome can be found out, or Bayesian model which uses degree of person's belief that an event is occurred based on past experiences [52] [58] [64]. Bayesian model is widely used to calculate trust value of a mobile node from collecting evidence and past experiences [52] [53] [58] [64]. This model is based on Bayes' rule that is used to calculate conditional probability of b given a, from the conditional probability of a given b using equation (3.3) [52][53].

$$P(b|a) = \frac{P(a|b) * P(b)}{P(a)} \qquad (eq.3.3)$$

From Beta distribution, trust can be calculated as an equation (3.4) [52][53][58][64].

$$T = \frac{p + r_{\text{base}}}{n + p + r_{\text{base}} + s_{\text{base}}} \qquad (eq.3.4)$$

Here, p is the number of positive evidences, n is the number of negative evidences and $r_{base}=s_{base}=1$

3.8.4 Belief Model

The Subjective logic trust model is introduced by A Josang [76]. The term opinion is used to represent subjective belief between two entities. An opinion can be calculated using probability which includes uncertainty. The traditional trust model does not use uncertainty. If a node doesn't collect enough evidence about any other node, it must be uncertain about that node's trustworthiness [76]. In subjective logic trust is represented using belief, disbelief and uncertainty. Opinion is a vector containing three components which defined as $W^{A}_{B}=(b^{A}_{B}, d^{A}_{B}, u^{A}_{B})$ denotes a node A's opinion about any node B's

trustworthiness in MANET. Here, first component corresponds to belief, the second component is for disbelief and third shows uncertainty. Also $b^{A}_{B} + d^{A}_{B} + u^{A}_{B} = 1$. To calculate values of b^{A}_{B}, d^{A}_{B} and u^{A}_{B} a node will collect evidence which are positive evidence p or negative evidence n. The equations (3.5), (3.6) and (3.7) are used to compute values of b^{A}_{B}, d^{A}_{B}, and u^{A}_{B} [76] using p and n.

$$b^{A}_{B} = \frac{p}{p + n + 2} \qquad \text{(eq. 3.5)}$$

$$d^{A}_{B} = \frac{n}{p + n + 2} \quad \text{where } u^{A}_{B} \neq 0 \qquad \text{(eq. 3.6)}$$

$$u^{A}_{B} = \frac{2}{p + n + 2} \qquad \text{(eq. 3.7)}$$

3.8.5 Fuzzy Model

In [61], authors used fuzzy approach as trust model. This uses fuzzy logic for trust calculation. It does not include only extreme cases of node's trust worthiness, Trusted or Untrusted but also includes the values in between these two states. For example 0.24 of trust, 0.50 of trust, trusted, untrusted.

3.8.6 Markov chain based trust model

This model is used to predict a trust value of a node from current behaviour of the node. The predicted trust value can be used only for a short period of time. Based on the current predicted state this model is used to identify the malicious behaviour of the node. Node's state changes from one to another according to Markov chain. Author used five tuple Markov model to estimate trust value of each node as shown below[79].

$\Omega = (R, V, Q, \Lambda, \prod)$

R *is a set of normal state* $\{r_1, r_2, r_3 \ldots r_N\}$

V is a set of malicious state $= \{v_1, v_2, v_3 \ldots v_M\}$

$Q = \{qij\}$is a K X K matrix where $K = M + N$

qij represent a transfer from i to j where i, j ϵ R U V

Λ is a set of parameters observed and based on which state changes.

$\prod$ is $\{\prod_1, \prod_2, \prod_3 \ldots \prod_{M+N}\}$ set of node's initial state

$$\prod_i = P0\{x(t) = r_i\} \quad 1 < i < N$$

$$\prod_j = P0\{x(t) = r_j\} \quad N + 1 \leq j \leq N + M \text{ and } \sum_{i=1}^{M+N} \prod i = 1$$

3.9 Use of Trust management in MANET

In any distributed system including MANET, we can apply trust management to enhance security, reliability and other quality attributes of the system [67]. The use of trust management can be used to [67],

1) Detect malicious activities in the system. For example, attacker nodes of the system.
2) Help in decision making while doing some important operation ex. Routing
3) Select an entity which gives us maximum benefit. For example, in a MANET, we can choose the best route. Using trusted entities we can improve performance of the system. For example, In a MANET if we choose static node over mobile node while routing, we can achieve more stable route.
4) Improve Quality of Service of the system by using trust management.

3.10 Selection of Trust Computation Engine for this thesis

In this thesis, we have compared all trust computation engines, which we have discussed in section 3.8. This comparison is shown in table 3.1. The summation and average model are most simple model. They are simple to understand[74]. They simply add the events happened on the node. If the event is positive it is added otherwise subtracted[74]. We can also give priority to the event while adding them to get the final trust value [65][82]. By giving priority, we can define importance to specific behaviour of a node [65][82]. We can identify the frequent behavioural changes using summation model[65]. With average model, we are dividing the sum by the total number of events[74]. Hence, sometimes it will be difficult to capture behavioural changes [74]. The Bayesian model predicts the behaviour of the node by calculating the predictive probability using past successful and unsuccessful interactions [82]. The limitation of this method is that they assume fixed behaviour of each node, which is not realistic in a situation where node changes their behaviour with time [82].

Table 3.1 Comparison of trust computation engine

	Simple	Give Priority to the behaviour	Need large observations	Identify unstable behaviour	Need more resource
Summation model	Yes	Yes	No	Yes	No
Average model	Yes	Yes	No	No	No
Bayesian model	No	No	No	No	Yes
Belief model	No	No	No	Yes	No
Fuzzy model	Yes	No	Yes	Yes	Yes
Markov chain based model	No	No	Yes	Yes	Yes

The belief model is complex to understand [74][76]. We also can not give priority to certain behaviour in this model. The fuzzy model uses heuristic formulas for calculating trust. It is a computationally complex model. It can detect behavioural changes or unstable behaviour of the node [74]. The Hidden Markov Model needs large history of interaction [82]. Hence, it is not a practical solution for the mobile adhoc node [82]. The solution of this issue can be collection of feedback from neighbours to compensate shortage of information about the node[82]. However, this solution leads to further misbehaviour like unfair feedback [74].

After studying above table 3.1, we have decided to use a summation model with weight associated with each category of observation to aggregate them for calculating the trust value.

Chapter 4

Literature Review

4.1 Introduction

Mobile Adhoc NETwork is an infrastructure less, self organising, self managing network with the limited resources available at each node. Nodes are battery operated and they use a wireless radio frequency for communication. Due to these features, MANET are vulnerable to various security attacks (Hard security threats). In a MANET, nodes work in cooperation to route packets of other nodes. As nodes are not controlled by any administrator, nodes change their behaviour time to time. Sometime they cooperate in network operation, some other time they may not cooperate and behave selfishly (soft security threats). These changes in behaviour are due to selfish behaviour of a node or hardware failure on a node or lack of battery power. In this thesis, we are studying security at network layer which includes routing and data packet forwarding. To prevent or detect attacks, we may choose cryptographic solutions, but the same cannot effectively be used for soft security threats. Also, the cryptographic solutions need more computing overhead, which is not suitable for resource constrained wireless nodes in MANET. Additionally, the intrusion detection based solution cannot be used as it required one traffic monitoring point to observe incoming and outgoing traffic on the network. IDS also needs more battery life. The other approach to detect attacks in MANET is trust based approach. In this approach, the node observes the behaviour of their neighbours and records them. They use the recorded information to decide whether to trust them or not for packet forwarding. Compare to cryptography based and IDS based solutions, the trust based approach is light weight and can detect continuously changing behaviour of nodes. Many researchers have come up with various trust based routing schemes to secure routing in MANET. In this literature survey, we studied the works of some trust based routing approaches developed in the last decade. We have compared them based on the network parameters used by them to calculate trust value, the technique used to aggregate these parameters to compute the trust value and attack detected by their routing approaches.

4.2 State of the Art

In [53], the authors introduced a trust model in which, each node has a trust agent running on it. The trust agent collects various evidences, filter them and assign weights to them for computing trust. Trust agent performs three functions: Trust derivation, quantification and computation. In trust derivation, agent on a node will collect information from all others in passive mode. It will not use any special inquiry packet for collecting evidence. This agent gathers evidence and does an analysis of forwarded, received and overheard packets from neighbours. The parameters recorded are Frame received, data packet forwarded, control packet forwarded, data packet received, control packet received, connection established, data forwarded and data received. These collected parameters classify an event into trust categories. In quantification state, the values collected are quantified from -1 to 1 which is a continuous range from distrust to trust. Also, trust value of each category is also quantified. The trust computation function assigns weights to events or evidence that were collected and quantified. The weight is assigned based on the application which demands trust. The weight also changes with time. Also, node will assign the weight using its own condition and state. The weight can be from 0 to 1. Less weight means less important. The trust can be calculated using equation (4.1) [53].

$$Tx(Y) = \sum_{i=1}^{n} Wx(i) * Tx(i) \quad \text{(eq. 4.1)}$$

$Wx(i)$ is weight of ith category and $Tx(i)$ is trust of x in ith category. The trust categories discussed in this trust model [53] are Packet Acknowledge(PA), Packet Precession(PP), Gratuitous route replies(GR), Black Lists(BL), and Salvaging(SG). Hence, equation (4.1) is expanded as equation (4.2) [53].

$$Tx(Y) = Wx(PA) * Tx(PA) + Wx(PP) * Tx(PP) + Wx(GR) * Tx(GR) + Wx(BL) * Tx(BL) + Wx(SG) * Tx(SG) \quad \text{(eq. 4.2)}$$

This trust value is calculated and used to find the most trustworthy path from the source to the destination.

In [56], trust is represented as opinion which is a 3-D vector defined as following equation(4.3).

$$W^{A}_{B}=(b^{A}_{B}, d^{A}_{B}, u^{A}_{B}), \text{ where } b^{A}_{B}+d^{A}_{B}+u^{A}_{B}=1 \qquad \text{(eq. 4.3)}$$

Here, the b^{A}_{B} corresponds to belief, the d^{A}_{B} is for disbelief and the u^{A}_{B} shows uncertainty. These values are calculated based on negative (n) and positive (p) evidences observed for other nodes using given equations (4.4), (4.5) and (4.6) [56][76].

$$b^{A}_{B} = \frac{p}{p+n+2} \qquad \text{(eq. 4.4)}$$

$$d^{A}_{B} = \frac{n}{p+n+2} \quad \text{where } u^{A}_{B} \neq 0 \qquad \text{(eq. 4.5)}$$

$$u^{A}_{B} = \frac{2}{p+n+2} \qquad \text{(eq. 4.6)}$$

Here n is number of negative event recorded and p is number of positive events recorded for node B on node A. A node will collect opinion of all neighbours about other node and combine them using trust combination techniques. The two combination operations are used in this scheme: Discounting combination and Consensus combination. Discounting combination is used when node A wants an opinion about node C and node B gives opinion about node C to A. A must have opinion about node B and A will combine opinions A to B, B to C to obtain recommendation A to C. In Consensus combination node A ask opinion about node C to others. The other nodes may have a different opinion. Some may give a contrary opinion about a node. To combine such opinions, consensus combination is used. During route discovery source node(N1) broadcasts a RREQ with trust vector. On receiving the RREQ node N2 will ask opinion about node N1 to all neighbours and combine opinion using the trust combination. This opinion is a vector containing b, d and u. For u>0.5, N2 does not believe on N1. This is because uncertain(u) factor is more than 0.5 hence receiver node doesn't believe on sender. Hence N2 request for a N1's certificate for authentication. When d>0.5 i.e. disbelief factor is more than 0.5, N2 will not trust N1 and ask for N1's certificate for authentication. When b>0.5 (belief is more than 0.5), N2 will trust N1 and rebroadcast RREQ. Otherwise in all other cases, N2 ask for a N1's certificate for authentication.

In [52], author proposed a method, in which for calculating direct trust each node observes the neighbour node by monitoring their cooperation in packet forwarding. After transmitting data or control packet, the node will be in promiscuous mode. As soon as it hears the packet forwarded by its immediate neighbour, it will check it for any modification. If the packet is modified, the counter Pp is incremented otherwise decremented. If within a certain time limit packet is not forwarded by neighbour, the P_A is incremented otherwise it is decremented. A direct trust of a node x on node y is Txy which is defined using equation(4.7) [52].

$$T_{xy} = w(P_A) * P_A + w(P_P) * P_P \quad \text{(eq. 4.7)}$$

P_A is used to detect packet drop attack and Pp is used to detect packet modification attack. For propagating trust information of other nodes, RREQ is sent with the trust information of node from where it is received. Thus, the trust values of a node are propagated to other nodes. The DSR (Dynamic Source Routing discussed in section 1.6.2) routing will use this trust value to find out the route with maximum trust level and ensure attack free route by bypassing low trusted nodes.

In [60], the trust value is calculated using time based past interaction and peer recommendations. For calculating trust based on past interactions (PI), the equation (4.8) [60] is used.

$$PIx,y = 1 - \frac{1}{\max[\{Ws * SIxy - Wu * UIxy\}, 0] + 1} \quad \text{(eq. 4.8)}$$

PIx,y is a past interaction based trust value of node y to x. SI_{xy} is the number of successful interactions between x and y. UI_{xy} is the number of unsuccessful interactions between x and y. W_s and W_u are time dependent positive numbers and represents weight. If the number of successful interactions is less than the number of unsuccessful interactions ($SI_{xy}<UI_{xy}$) then $PIx,y = 1 - \frac{1}{\max[\{\text{negative value}\},0]+1} = 1 - \frac{1}{0+1} = 0$. If SIxy>>UIxy then $PIx,y = 1 - \frac{1}{\max[\{\text{positive value}\},0]+1} \approx 1$.

$$PRx, y = \frac{\sum_{i=1}^{n} \mathrm{TVx, i} * \mathrm{TVi, y}}{n-1} \qquad \text{(eq. 4.9)}$$

Trust using Peer recommendations(PR) can be calculated using the equation(4.9) [60]. The peer recommendation of node y at node x is calculated using the trust value of node i at x and trust value of y at node i. In equation (4.9), $\mathrm{TVx, i}$ is the trust value of node i at x and the $\mathrm{TVi, y}$ is the trust value of y at node i. The final trust value can be computed using equation (4.10)[60].

$$\mathrm{Tvx, y} = \frac{PIx, y + PRx, y}{2} \qquad \text{(eq. 4.10)}$$

This is a trust management scheme for distributed wireless sensor network. This scheme is simple and flexible. It doesn't required large storage space and complex computation at sensor node.

In [63], authors proposed trust management protocol for MANET which addresses two important areas: Trust bias minimization and application performance maximization. They integrate social as well as QoS parameters to calculate trust. Trust will be calculated as social trust and Qos trust. Social trust is calculated using social ties measured by intimacy and healthiness. QoS trust uses energy of node and cooperativeness (successful packet forwarded). Trust value of other node is a real number from 0 to 1, where 1 indicates full trust, 0.5 ignorance and 0 complete distrust. Intimacy is measured when two nodes have large number of interaction (direct or indirect) like packet routing and packet forwarding. Healthiness checks whether the node is malicious or not. Energy will be residue energy of a node, i.e. survivability of a node. Cooperative means cooperation of a node in routing. Each node will accurately assess their one hope neighbour by monitoring, overhearing and snooping traffic. The nodes will observe the nodes within their radio range and overhear the transmission power and packet forwarded by them for Δt time and calculates energy and cooperativeness. Trust aggregation is done in such a way that trust bias gets minimize. For trust aggregation input is direct rust and indirect trust. Indirect trust is opinion provided by 1 hop neighbours. To avoid slandering attack [83] while collecting opinion, threshold based filtering may use in which opinion from neighbour with value more than threshold will be considered. Relevance based trust approach is used in which

recommendation with high value in component x are considered for trust aggregation. The equation (4.11) [63] is used to calculate trust value.

$$T_{i,j}^{x}\ (t) = \beta 1\ T_{i,j}^{direct,x}(t) + \beta 2\ T_{i,j}^{indirect,x}(t) \qquad (eq.4.11)$$

Here $\beta 1$ and $\beta 2$ are weight factors and $\beta 1 + \beta 2 = 1$. x is a parameters ie intimacy, healthyness, energy, cooperativeness. i is an assessor and j is one hop neighbour of i. For each parameter x, i.e. intimacy, healthiness, energy and cooperativeness a direct trust component is computed. After calculating trust of each parameters, they will be aggregated using equation (4.12) [63]. Here, $W^x$is priority or weight assigned to each parameter (x).

$$Ti,j(t) = \sum_{x} W^x\ T_{i,j}^{x}\ (t) \qquad (\text{eq.}\ 4.12)$$

In [65], authors developed a trust based service discovery process in MANET, with less overhead and identify malicious behaviour of servers. Their approach will successfully discover all the service providers in network and then apply trust model to select most trustworthy service provider. They use AODV routing protocol for service discovery. For that they have modified RREQ packet as SREQ in which they add 32 bit service name. They also modified RREP packet as SREP with new fields service lifetime, service name, server address and trust value. Each node also maintains a service table with service ID, address, lifetime, service name, response time and trust stored in it. For calculating trust of each server following parameters are used: mobility of node with negative weight to server, battery life, response time, life time of service, packet drop with negative weight. Trust value of server x will be calculated at client node c using equation (4.13)[65].

$$\mathrm{T(Xc)} = \sum_{k=0}^{n} \mathrm{Wk} * \mathrm{Pk} \qquad (\text{eq.}\ 4.13)$$

Where Wk is weight assigned to parameter, Pk is observed value of parameter and T(Xc) is trust calculated by node c for server X. Once client receives reply from all servers, it will choose server with highest trust value. Also, dynamically trust value of each route will be calculated by client and change route, if found more trusted server than existing one.

In [54], authors proposed a secure routing scheme for MANET routing which is based on AODV. They concentrate on packet drop and packet delay attack. They used trust based approach in which trust is calculated using various network parameters and a weight is also assigned to them. The parameters they used are data packet dropped, data packet forwarded, number of packets delayed, control packets dropped and remaining energy of a node. Trust is calculated using weighted sum model. In AODV routing, a route request (RREQ) is broadcasted from source to destination with the help of intermediate nodes. In reply, the destination node will send a route by sending RREP on the same path from where the RREQ came. The RREP is modified by adding a new field trust value of path in it. It stores the summation of trust value of all intermediate nodes in it. When RREP reached at source, it has trust value of a whole path stored in it. Also, in AODV destination node gets multiple RREQ from all possible paths from source to destination. In this scheme destination will reply all RREQ with trust value of path in it. From all RREP the source will select the path with maximum trust value. Also periodically trust value of each path recalculated. And route with highest trust value will be used as a new route for packet forwarding by source.

In [58], authors proposed a probabilistic trust model for pervasive computing. Trust is computed as a probability that nodes interact satisfactory with their neighbours. They calculate both direct trust and indirect trust and combine them to get final trust value of a node. For direct trust Bayesian inference is used. Any node A's trust on B is probability that the B act well while A interact with B. The value can be from 0 to 1. For recording observation two parameters are used n_s (total number of satisfactory interaction) and n_u (total number of unsatisfactory interaction). In beta distribution they take $\alpha = n_s+1$ and $\beta=n_u+1$. Trust can be calculated using following equation (4.14)[58].

$$TA(B) = E\big(f(x;\alpha,\beta)\big) = \frac{\alpha}{\alpha + \beta} = \frac{n_s + 1}{n_s + n_u + 2} \qquad (eq.\,4.14)$$

Number of successful interaction is measured as number of packets successfully forwarded by a node. For calculating indirect trust recommendation is collected from neighbours. Let i nodes send recommendation about node B collected at node A. n_s^m is number of satisfactory interaction with node B provided by node m. So total satisfactory interaction in recommendations are $n_s^r = \sum_{k=1}^{i} n_s^k$. Similarly, $n_u^r = \sum_{k=1}^{i} n_u^k$ is calculated. For indirect

trust α =$n_s^r + 1$=$\sum_{k=1}^{i} n_s^k$+1 and β=n_u^r+1=$\sum_{k=1}^{i} n_u^k$+1. For calculating final trust, which aggregate direct observation and indirect recommendations, they have substituted $\alpha = \text{ns} + n_s^r + 1$ and $\beta = \text{nu} + n_u^r + 1$ in equation (4.14) and got eqation (4.15)[58]. Also they subtitute $n_s^r = \sum_{k=1}^{i} n_s^k$ and $n_u^r = \sum_{k=1}^{i} n_u^k$ in equation (4.15) and derived final trust equation (4.16)[58].

$$\text{TA(B)} = \frac{\text{ns} + n_s^r + 1}{(ns + n_s^r + 1) + (nu + n_u^r + 1)} \qquad (eq.4.15)$$

$$\text{TA(B)} = \frac{\text{ns} + \sum_{k=1}^{i} n_s^k + 1}{\left(ns + nu + \sum_{k=1}^{i} n_s^k + \sum_{k=1}^{i} n_u^k + 2\right)} \qquad (eq.4.16)$$

Authors used iterative filtering method for avoiding false recommendation. This technique will exclude all dishonest recommendation and use only recommendation provided by honest node for indirect trust calculation. This is a threshold based filtering scheme in which average of all collected recommendation is calculated. If difference of any recommendation and average recommendation is greater than s (threshold), the recommendation is considered as false. Authors also use weight which changes with time to calculate n_s and n_u. This is used to give more weight to recently collected observations than past observations.

In [57] authors proposed a trust model which ensures reliability, integrity and trust worthiness of data provided by a sensor node. They uses trust vote to earn trust on each node. Each node maintains a counter trust_vote which will incremented if successful message transmission. Each node performs three actions 1) when node A sends a message to node B, Node A will creates an entry in its trust table for node B. 2) when node B forward a message to next level , node A overhear it and compare it with original packet sent. It measures changes in the forwarded packet to see whether there is any modification or packet is not at all forwarded within certain time. If packet is not forwarded or modified, node A will record the untrust vote for node B in trust table, otherwise A will record a trust entry for node B. 3) If un trust entries for any node in trust table reaches above the threshold, that node will be declared as malicious node and inform to others by broadcasting a message.

In [61], authors proposed an approach which incurs less cost of trust evaluation using less memory and power. It also detects and prevents malicious, selfish and faulty nodes. For calculating trust on each node, they use past interaction observed in specific time window Δt. They actually observe number of successful interactions and unsuccessful interactions for Δt period of time. Once time laps the window shift right for one time unit. Thus it drops the observation of first time unit and considers observation of new time window. The window length depends on the network setup. Thus trust value of node y observed at node x is calculated using equation (4.17) and (4.18) [61].

$$Txy = \left[100 * \left(\frac{Sxy}{Sxy + Uxy}\right)\left(1 - \frac{1}{Sxy + 1}\right)\right] \qquad (eq.4.17)$$

$$Txy = \frac{100\ Sxy^2}{(Sxy + Uxy)(Sxy + 1)} \qquad (eq.4.18)$$

Where the value lies between 0 and 100. After calculating trust value, node is classified based on following

Trusted 100-f ≤ Txy≤100

Uncertainty 50-g ≤Txy<100-f

Untrusted 0≤Txy<50-g

f is half of average value of all trusted node

g is ⅓ of average of all trusted node.

For evaluation recommendation, node will broadcast recommendation request to all trusted node. Let j nodes are trusted or uncertain. Then node x calculate indirect trust on y using equation(4.19)[61].

$$Txy = \frac{\sum_{i \in trusted\ or\ uncertain}(Tx, i * Ti, y)}{100 * j} \qquad (eq.4.19)$$

In [66], authors present a light weight trust based routing which consumes less computational resources. It will use locally available information on each node. This protocol will detect black hole and grey hole attack. On each node, the trust value of all neighbour nodes will be calculated. Node i can compute trust on neighbour j using equation (4.20)[66],

$$T_{\mathrm{i}}(j) = \alpha T_{i(self)}(j) + \beta T_{i(neighbor)}(j) \qquad (eq.4.20)$$

$T_{i(self)}(j)$ is the j's behaviour observed by node i and $T_{i(neighbor)}(j)$ is j's trust collected from all neighbours of i which are also neighbours of j. The equation (4.22) and (4.21) are used to compute them respectively [66].

α+β=1 and 0≤ α,β≤1

$$T_{i(neighbor)}(j) = \frac{1}{n}\sum_{k=1}^{n} T_{k(self)}(j) \qquad (eq.4.21)$$

This is an average of existing trust from n number of neighbours on j.

$$T_{i(self)}(j) = \frac{\sum_{k=0}^{N-1} forwarded(k)}{\sum_{k=0}^{N-1} toForwarded(k)} \qquad (eq.4.22)$$

The node i observe all the packets sent to node j by itself and by other common neighbours by hearing traffic in promiscus mode. Packet forwarded is a counter which stores packets from node i successfully forwarded by node j. Packet toForwarded is a counter which stores number of packets from neighbour successfully forwarded by node j. Also, these observations are measured periodically for each slot or window. Total slots are N. And this trust value is used to calculate reliable route from source to destination during routing.

In ARMAN scheme [64], authors have seen trust as a subjective probability and no dependency on any third party. It uses direct information coming from personal observation of agents and indirect information coming from other agents. Trust computation is performed in three parts. 1)obtains direct observation between truster and trustee. 2)collects second hand information provided by set of neighbours 3)integrate first hand and second hand information using Dempster Shafer theory[87]. To avoid malicious second hand information it uses Similarity view. It is based on an assumption that if two agents observe an event in the same way, they have similar views. This scheme uses the number of packets successfully forwarded by a node as positive observation and packet drop or modification in packet before forwarding as negative observations. The reputation value is a mean of the beta distribution between[86] two nodes(truster and trustee).

In [88], authors proposed trust based AODV routing protocol for MANET. In this scheme trust value is associated with each node. The trust can be calculated using estimated energy of node, success rate of a node when data transmission and mobility of a node. The energy of a node is estimated by subtracting energy used for sending, receiving and processing the packets on the node from the current energy of the node. The mobility of a node is estimated using distance and speed of a neighbour node which can be computed using strength of signal required to send data to that neighbour and received data from that neighbour. For computing the node's success rate in data transmission, the algorithm consider ratio of total packets successfully send and total number of packet sent for data as well as control packets. During route formation the algorithm considers the node with maximum trust value in route and hence, the most secure and stable route is formed. This algorithm does not use multiple paths simultaneously to improve the performance of network. They also overburden the most trust worthy nodes in routing.

In [89], authors proposed TES-AODV (Trust and Energy Supported-AODV) routing protocol for the MANET. This routing protocol uses both trust based approach for stable/reliable route and MD5 algorithm for signing each message sent by a node for securing it. Trust value is associated with each node which can be calculated using energy level and successful packet transmission rate of that node. The successful packet transmission rate of a node can be computed by finding ratio of number of successfully send packet and total sent packet. Three such ratios are computed for RREQ packets, RREP packets and data packets which are added to find final ratio. The energy level of each node is initialized with 100. Each time packet is sent or received the energy consumption will be subtracted from current energy level. During route formation the node which has trust value above the threshold value are considered in route. Also before sending packet each node uses MD5 algorithm to sign the packet for security. The authors do not considered mobility of a node which also is one of the reasons of frequent link break. Also at a time only one route is used for data transmission.

4.3 Comparison of Existing Trust based routing approaches for MANET

4.3.1 Network Parameters used for calculating trust value

Each proposed technique[53,56,52,60,63,65,54,58,57,61,66,64,88,89] uses number of packet forwarded by the node as one of the parameters for calculating trust value. In MANET, each node will forward the packet to their immediate neighbours and get passive acknowledge from them by overhearing the transmission of that next hop neighbour on the route. This is because all the nodes in the same radio range can receive each transmitted packet by a node located in their range [61]. If the overheard packet matches the sent packet, means packet is successfully forwarded. Some researchers also compare this overheard packet with sent packet to check for any modification [53][63][58][61][64]. However, this approach introduce large overhead for comparing two packets each time, which is not suitable for resource constrained nodes in MANET. Some researcher [54, 63, 65, 88, 89] uses remaining battery life as one of parameter for trust value calculation (more battery life means more trust). This will help to choose the more stable route from source to destination. Some algorithms use the number of packet dropped by the node [54,58,63,65] for calculating trust. If more number of packets dropped less trust value is assigned. Thus packet drop attacks (grey hole and black hole attacks) are easily detected. To detect unnecessary delay in packet forwarding, many approaches [54,58,63] use number of packet delayed by the node to calculate its trust value. More number of packets delayed before forwarding on a node may reduce the trust value of that node and thus detects packet delay attacks(a type of jelly fish attack). The trust based algorithms proposed in [58,63] also used number of successful communication session between nodes in trust computation formula and quantify intimacy (social trust) relationship between them. From the above survey, we have observed that the trust models use number of packet successfully forwarded by a node, number of packets dropped at node, number of packet delayed at node and remaining battery life of node to calculate trust value of a node. All existing schemes observe one or more of these parameters of each neighbour or collect opinion based on one or more of these parameters to calculate trust value.

After studying all existing scheme we conclude that there has been no work done that uses routing error packet sent by a node to calculate trust. This parameter is important because

if a node initiates an RERR packet means there is link break from that node. This link break may occur due to mobility, hardware failure, insufficient battery life or intentional packet drop by the next hop neighbour node. The node intentionally drops the packet to save its resources. The node in active route, initiates RERR packets when the next hop node is not responding the packets forwarded to it. When link break occurs, the node tries to do some attempt for link repair. If after attempts, route is not repaired, an RERR packet is uni casted to the source node of the active route. On receiving RERR packet, the source node reinitiates route discovery process. During routing, if we omit such next hop neighbour nodes, which are responsible for the link break i.e. which are responsible for more RERR packets, we may get more stable route. Thus we may reduce route discovery attempt and thus reduce network traffic.

4.3.2 Trust computation Engines used

Trust computation engines are used to aggregate various observed parameters collected by the node to calculate trust value. The most popular approaches are discussed earlier in section 3.7 of chapter 3. The table 4.1 shows the trust computation engine used by various existing trust based routing schemes.

4.3.3 Attacks/Misbehaviour detected

Each proposed technique [52][53][54][56][57][58][60][61][63][64][65][66][88][89] successfully detects an attack which drops either data or control packets. The techniques proposed in [53][63][58][61][64][89] detects any modification done by attackers in packet before forwarding them. In [89], authors use MD5 algorithm to digitally sign the packet to secure it. The delay in packet forwarding is detected in [54][58][63] as they record total number of packet delayed by node and used it to calculated trust value. Most of the trust based routing approaches use most trusted route from source to destination for forwarding data packet. Using trust based routing scheme, we can detect all the attacks which include dropping of packets, modifying content of packets, and delaying packets before forwarding them further.

The trust based approach proposed in [58][64] uses direct trust values calculated at node using node's personal experience by observing neighbour nodes and indirect recommendation about a node provided by the neighbour nodes. If any node tries to provide false recommendation about any other attacker node, it will be detected in this scheme [58][64]. For detecting false recommendation they use similarity rule. After

receiving a recommendation from all, average of them found and if any individual recommendation is widely different than the average recommendation then it will not be considered. The [88] uses mobility of node for calculating trust value of a node. The mobility is calculated by measuring strength of signal received from the each neighbour node and strength of signal required to send packet to neighbour nodes [88].

Table 4.1 Comparative analysis of existing trust based routing schemes

Scheme	**Parameters used**	**Routing protocol**	**Attacks detected/Prevented**	**Approach**	**Trust computation Engine used**
Pirzada & Mcdonald [53]	NPR, NPFS	DSR	DR, MD	D	Bayesian Model
Xiaoqi Li et al.[56]	NPFS	AODV	DR	D & I	Belief Model with Subjective Logic
Pirzada et al.[52]	NPFS	DSR	DR	D	Bayesian Model
R A Shaikh et al.[60]	NPFS	AODV	DR	D & I	Average
I RChen et al.[63]	RE, NPFS, NSS, NPDR, NPD	AODV	DR, DL, MD	D	Weighted sum model
Gohil Bhumika et.al. [65]	RE, RT, NPDR, MOB	AODV	Load balancing, DR	D	Weighted sum model
Vinesh Patel et al. [54]	NPFS, NPDR, NPD, RE	AODV	Load balancing, DR, DL	D	Weighted sum model
Sun & Denko [58]	NPFS	AODV	Detects false recommendations, DR,MD	D & I	Bayesian Model
Zia [57]	NPFS	AODV	DR	D	Deterministic model
Riaz Ahemad Shaikh et al.[61]	NPFS	AODV	DR, MD	D & I	Fuzzy model
Guemkam et al[64]	NPFS	AODV	Detects false recommendations, DR,MD	D & I	Bayesian Model
G. Dhananjayan & J. Subbiah [88]	NPFS,RE,MOB	AODV	DR	D	Bayesian Model
S. Sridhar et al[89]	NPFS, RE	AODV	DR, MD	D	Bayesian Model

Table 4.2 Abbreviations used in table 4.1

NPFS	number of packet forwarded successfully	RE	Residue Energy	D	Direct Observation
NPD	number of packet delay	DR	Packet drop attack	I	Indirect Recommendation
NPDR	number of packet dropped	DL	Packet delay attack	RT	Response time
NSS	number of successful session	MD	Packet modification attack		
NPR	number of packet received(to be forwarded)			MOB	Mobility

4.4 Survey Conclusion

The study of existing trusts based routing schemes led us to the following conclusion: There has been no scientific work found which uses a routing error packets (RERR packets) sent by the node to calculate trust. This parameter is important because if the

intermediate node of any active route initiates RERR packet, it means there is a link break from that node. The initiator of the RERR packet knows the address of the unavailable node due to which link is broken. While creating RERR packet, if the node appends IP address of unavailable node with that packet, all the neighbours of the node and receivers of RERR can find out the culprit (the reason for link break) and record it. In a MANET, one of the main reasons for link breaks is the movement of nodes. The RERR packet is created by a node of an active route when it finds a link break. The node will send the RERR packet to the source node to inform it about route failure. In response to that the source node has to rediscover the route towards the same destination. We aim to include number of time a node has broken the route in the trust calculation to detect and avoid such node during route formation and give stable route.

We have also notice that no scheme uses multiple trusted routes simultaneously to send data packets. Some scheme [54] finds multiple trustworthy routes, but they use most trustworthy route at a time. If we found more than one trustworthy route between same source and destination pair and use them simultaneously to send data, it will improve performance of overall network.

All existing schemes use the most trustworthy nodes during route establishment in routing process. This may add burden to all trustworthy nodes. They utilize almost all of their resources in cooperating in routing process and hence their own work will suffer. There has been no scientific work found which address this issue. In this thesis, we search for multiple trustworthy routes and use them simultaneously for sending data packets. Hence load is distributed among all the trusted nodes.

In the next chapter, we introduce the design of an AODV based routing protocol for MANET that uses RRER packets for recording trust parameter of a node. The protocol also uses the multiple trustworthy routes to distribute the load of data transfer among other trusted node.

CHAPTER 5

Proposed Trust Based Routing Model

5.1 Problem Statement

In this thesis, we have addressed the problem of the frequent link break of the route during routing in a MANET. This link break may happen due to the movement of nodes. Due to such frequent link break, it becomes very difficult for a routing protocol to maintain a route between source and destination. For each link break, routing protocol has to discover a new route which is time consuming and may degrade the performance of the routing protocol and hence the network. We have also addressed the problem of trust based routing schemes in a MANET, which always consider most trustworthy nodes in a route. Due to this behaviour of trust based routing, the trustworthy nodes of the network are overburdened in routing only and they cannot perform their routine task efficiently. To avoid this issue we, try to search multiple trustworthy routes between the same source and destination and use them simultaneously to send data packet. The frequent link break of route problem is also solved using this approach. We are searching and using more than one route for the same source and destination. When any route breaks, we can use the other available routes to send our data. Thus, the performance of routing can be improved. We have also included movement of node as one of the parameters to calculate node's trust value. More movement leads to more link breaks as the nodes in movement will change their position quite frequently. This makes the existing links broken as the node may go out of the range of the current neighbouring nodes. Hence, the more movement the less trust. To the best of our knowledge, there has no work done to address above discussed problems in mentioned approach. Hence, in this thesis, we try to solve following research problem: " How can we build a routing mechanism in a mobile ad hoc network that avoids malicious/selfish nodes in the route and give relatively stable and load balanced route considering the resource constraint nature of mobile node?" Stable route means route containing majority nodes with less mobility so that less link break. For Load balancing multiple trusted routes from source to destination are found and used simultaneously sending data packets.

5.2 Scope of our Research

We define our scope as:

1) Developing selfish/malicious node in a wireless network which performs packet drop and packet delay attacks.
2) Developing a secure and stable routing protocol (Trust based Mobility Aware-Ad hoc On demand Distance Vector: TMA-AODV) for MANET. The proposed algorithm will work on the network layer.
3) We also assume that packet forwarded by a wireless node is received by all the nodes who are in the range of the sender node. Thus, each node monitors the network traffic of their neighbours.
4) Providing the proof of concept by improving the route discovery time of routing and the throughput of the network with TMA-AODV. Route discovery time is improved as a result of simultaneous usage of multiple trustworthy routes and more stable routes. Throughput is improved with TMA-AODV in the presence of Drop and Delay attack. The improvement with TMA-AODV is to compare with AODV routing in the presence and absence of mobile nodes.

5.3 Objective of our Research

The objective of this research is to

1) Analyse the routing protocols for MANET in a network simulator and determine the choice of a routing protocol for the later research purpose.
2) Analyse the selected routing protocol with UDP and TCP traffic.
3) Implement the packet drop and the packet delay attacks and study their effect on routing and network parameters.
4) Survey the literature of various trust based routing schemes for comparing parameters used for trust calculation, trust computation engine used and security provided by them against various attacks.
5) Find out research gap and come to the final problem statement from the literature survey.
6) Propose a new secure and stable routing scheme with less overhead.
7) Implement the proposed routing scheme and compare it with standard AODV routing scheme.

5.4 Original Contribution By the Thesis

This thesis discusses the current trust based routing approaches and their comparative analysis for followings: Parameters used for calculating trust value, Attacks detected and Trust computation engine used for calculating the trust value. We have proposed a trust based routing protocol for mobile ad hoc network which detects both packet drop and delay activities of malicious/selfish node and establishes a stable route which has less link breaks. We used the weighted sum model to calculate trust value from the observed parameters because it is simple and incur less computational overhead. Also proposed routing scheme searches multiple trusted paths from the same source to destination and all trustworthy paths are used simultaneously to distribute load among multiple nodes of a network. We observe the route discovery time and the throughput of the network in the absence and presence of the malicious nodes and mobility in the network. We have also shown improvement in these observed parameters with our proposed routing protocol.

In proposed routing protocol (TMA-AODV) each node monitors traffic to and from their neighbours and stored observed values in trust table. For each neighbour node has an entry in trust table. The values observed and stored in trust table for each neighbour node is used to calculate trust value of that neighbour. This trust value specifies that how much that neighbour node is trustworthy. And this trust value will be calculated and used while the route is established from a source node to the destination node. When a source node has data to send to any destination node, it creates an RREQ packet with destination node id and send this packet to all its neighbours. If any neighbour node is the destination node, it will create an RREP packet with trust value and send it to the source node. Otherwise, they forward RREQ packet to their neighbours. When the RREQ packet received at the destination node, for each received RREQ packet a separate RREP packet is created and trust value of next hop neighbour is calculated. The trust value of the RREP is initialized with the calculated trust value. Then RREP packet is sent to the next hop neighbour, who further calculates trust value of its next hop neighbour and adds it to RREP trust value and sends the RREP to its next hop neighbour until RREP reaches at the source node. Thus, while RREP is received at each intermediate node, it will calculate the trust value of its next hop neighbour and add it into route's trust value. So at the end source node has multiple routes towards the destination with different trust values of each route. The source node will calculate average of trust of each route and use that average as a threshold. And alternatively choose routes having greater trust value than the threshold value. Thus the

load is distributed among more than one route and there will be less chance of route failure. We have modified the RREP packet to accommodate trust value in it. We have also added one field in the route table entry for storing the trust value of the route. The trust of route depends on the trust value of all intermediate nodes. Let a route **r** consists of **l** intermediate nodes. x_1,x_2,x_3---x_l where xi is i^{th} intermediate node of the route r. The trust value of the route r (Tr) can be calculated using the equations (5.1) and (5.2),

$$T_r = T_{x1} + T_{x2} + \ldots\ldots + T_{xl} \quad (\text{eq. } 5.1)$$

$$T_r = \sum_{i=1}^{l} T_{xi} \quad (\text{eq. } 5.2)$$

We have also modified the standard RERR packet (Route error packet), which is sent by a node of a route which detects the link break. The RERR packet is sent to the source node to inform it about route break. Originally, the RERR packet doesn't have any field which stores the node's identity who is responsible for route break (A node who is a part of route but not available). In our proposed routing algorithm, we have appended a field which stores IP address of the node that is breaking the route. The RERR packet is received by all the nodes and their neighbours which forwarding it to source node. They read the IP address of the node that is responsible for link break and increment link break count of that node by one and record it in the trust table.

For calculating the trust value in trust model, we have used a number of packets observed, number of packets successfully forwarded, number of packets delayed, and the number of link break due to that node. For detecting the amount of packet drop we take the difference of the number of packets observed and the number of packets successfully forwarded parameters. For detecting packet delay attack, we are interested in calculating the time taken by a node from the arrival of the packet to forwarding that packet further. If this calculated time is above the permissible delay, then our protocol detects it as a packet delay attack, i.e. A node is intentionally delaying a packet before forwarding it. Permissible delay of a packet on node is calculated by adding two delays: processing delay and transmission delay. Processing delay is the time taken by a node to process the header of receiving a packet (CRC check etc.) and decides the output link to further forward it.

The transmission delay depends on the length of packet in bits and bandwidth of the link [80]. So, Permissible Delay is calculated using equation (5.3)[80]

$$PD = D_p + D_t \quad \text{(eq. 5.3)}$$

Where D_p is the time taken by node to process the packet after receiving it and D_t is the time taken by node to forward the complete packet. PD stands for Permissible Delay.

Let our ad hoc network has N number of nodes. Any random node i of a network has M numbers of neighbours. The trust table at node i has total M entries in it. One for each neighbour. The node i's trust about node j can be calculated using the values stored in a trust table at node i for neighbour j.

Let $T_i(j)$ is a trust of node i about node j(j is neighbour of i). The trust $T_i(j)$ is calculated using equation (5.4).

$$T_i(j) = -W1 * (Po_j - PF_j) - W2 * PD_j - W3 * PER_j \quad (eq. 5.4)$$

Po_j: number of packets observed for a neighbour node j,
PF_j: number of packets successfully forwarded by neighbour node j,
PD_j: number of packets delayed at neighbour node j,
PERj: number of link break due to neighbour node j,

Here W_1, W_2, and W_3 are the weight factors. $W_1+W_2+W_3=1$ and $0<= W_1, W_2, W_3<=1$. W_1 is the weight of detecting packet drop at the node which is very important as a packet drop at an intermediate node is a serious issue. W_2 is weight related to packet delay detected on the node which is less serious compare to packet drop attack. W_3 is weight related to link break due to a node. Values of weights are calculated using observed parameter values of each neighbour, using the following equations.

$$X = Po_j - PF_j \quad \text{(eq. 5.5)}$$

$$Y = PD_j \quad \text{(eq. 5.6)}$$

$$Z = PER_j \quad \text{(eq. 5.7)}$$

$$W_1 = \left(\frac{X}{X+Y+Z}\right) \quad \text{(eq. 5.8)}$$

$$W_2 = \left(\frac{Y}{X+Y+Z}\right) \quad \text{(eq. 5.9)}$$

$$W_3 = \left(\frac{Z}{X+Y+Z}\right) \quad \text{(eq. 5.10)}$$

In our algorithm, weight values are calculated when the trust value of any node is calculated on the node.

5.5 Proposed System

5.5.1 Proposed System Architecture

In the proposed system, we have designed following modules which constitute the architecture of our proposed system shown in figure 5.1.

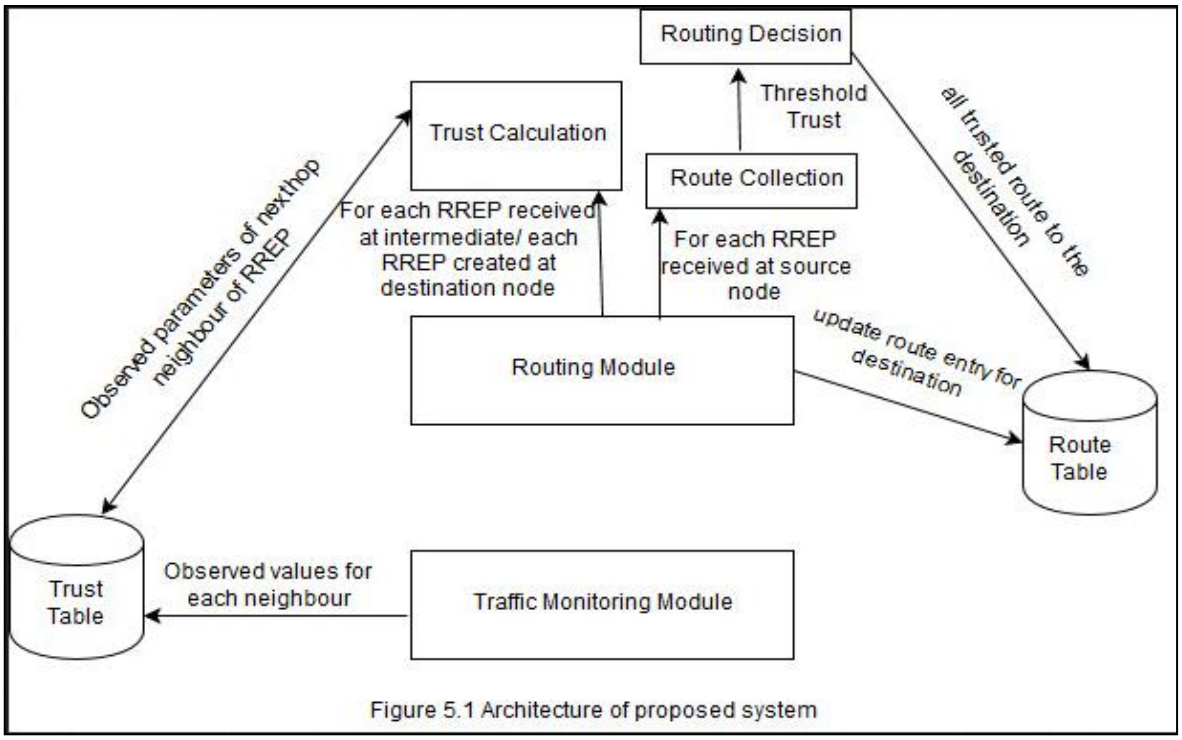


Figure 5.1 Architecture of proposed system

Our proposed routing scheme has three major modules: traffic monitoring, routing module and trust calculation. In traffic monitoring module, node continuously monitors the behaviour of all the nodes within its radio range. During monitoring, node detects events like packet drop, packet delay, and number of route broke by the node. The node records the observed events of each neighbour in its local trust table. Later, this data is used to

calculate the trust value of the specific neighbour. In routing module, RREQ, RREP and RERR packets are received and processed. The node processes the received RREQ packets same as standard AODV routing. We have appended a 32 bit address field in RERR packet of standard AODV routing that stores the IP address of the node who is not available and breaks the link. When link breaks, during data transmission a node from where route is not available, generates an RERR packet with IP address of a node who is not responding and send that RERR packet towards source node. In wireless media, all the neighbours of the node also receive the RERR packet and process it to modify the link break count of that node (who breaks the link) in their trust table.

We have appended 32 bit field in the RREP packet of standard AODV to store trust value of a route. On receiving RREP packet, the node will store route from that node to a destination node in its route table. Before forwarding RREP packet to the next hop neighbour, the node will call trust calculation module which calculates trust value of the next hop neighbour using observed values stored in its local trust table. This trust value is added into the trust value stored in an RREP, and the RREP with the new trust value is sent to the next hop neighbour. This process is repeated on each node until the RREP received at the source node.

When the RREP packet received at the source node, it waits until all RREP received. After receiving all RREP packets, source node will calculate average of trust values of all RREP. This average value will be the threshold trust value. All the routes having a trust value greater than the threshold trust value will be simultaneously used by the source node for sending the data packets. Also, these routes are stored in the routing table of the source node. In case of link break, the broken route will be deleted from the routing table and will not be used. Unlike AODV, the link break will not initiate route discovery process. Instead of that the other trusted routes are used. The route discovery process started only when all trusted routes are broken.

5.5.2 Features of proposed routing algorithm

1) Our proposed routing scheme is an extension of standard AODV routing.
2) Calculates trust value using the weighted sum model.
3) The parameters used for trust calculation are packets observed, packet successfully forwarded, number of packets delayed before forwarding and number of link break by a node.

4) More than one trusted routes are found and responsibility of data packet forwarding is distributed among them.
5) Permissible delay of a packet on node is calculated by adding two delay: processing delay and transmission delay.
6) Parameters considered for performance measurement are Throughput and Route discovery time.
7) Each node which is a part of active route observes traffic forwarded by all of its neighbours.
8) Each node maintains a trust table having an entry for each neighbour.
9) A new library is created having functions for adding, removing and getting entries of trust table.(refer Appendix A).
10) The 32 bit field is appended in RREP packet which is used for storing the trust value of the route.

Type (8bit)	R (1 bit)	A (1 bit)	Reserved (9 bits)	Prefix size (5 bits)	Hop count (8 bit)
Destination IP Address					
Destination Sequence Number (32 bits)					
Originator IP Address (32 bit)					
Lifetime (32 bits)					
Trust of route(32 bits)					

11) Format for trust table.

NodeID	Total pkt	Suc_forward	Delayed	Total_err

12) A 32 bit field is appended in the RERR packet format which is used to store the IP address of the node that broke the route.

Type(8 bit)	N (1 bit)	Reserved (15 bit)	Destination count (8 bits)
Unreachable Destination IP address(32 bit)			
Unreachable Destination Sequence Number(32 bit)			
Additional Unreachable Destination IP Addresses (if needed) (32 bit)			
Additional Unreachable Destination Sequence Numbers (if needed) (32 bit)			
IP Address of the node who is responsible for link break (32 bit)			

5.6 System Diagrams and Algorithms

5.6.1 Context Flow Diagram of proposed system

Our proposed routing protocol (TMA-AODV) can be used in mobile ad hoc network for detecting and avoiding misbehaviour of the nodes. The context diagram of the proposed routing scheme is shown in figure 5.2. There are mainly three categories of nodes involve in working of this scheme: source node, destination node and intermediate node. The source node is the initiator of the route discovery process in this routing scheme. It creates an RREQ packet, and broadcast it to all its neighbours. The neighbour nodes further broadcast RREQ to their neighbours until RREQ reaches at destination node. At destination node for each received RREQ, an RREP is created and sent back to the source node on the same path from where the RREQ came. The destination node will calculate the trust value of the next hop neighbour and add that value into the RREP. Each intermediate node which receives the RREP, further calculates the trust value of their next hop neighbour and add that trust value into RREP before forwarding it. This process is repeated on each intermediate node until RREP reaches at the source node. The source node receives multiple RREPs with a trust value attached to each. The source node calculates average of all trust values received with RREPs and simultaneously uses the RREPs with trust value more than the average trust value for data transmission. We can see one more user of the system, i.e. node, in context flow diagram. This node can be a source node, a destination node or an intermediate node. All nodes which are active can observe the traffic to and from their neighbours and record these observations in their local trust table. These recorded observed parameters are used by routing to calculate the trust value of any neighbour node.

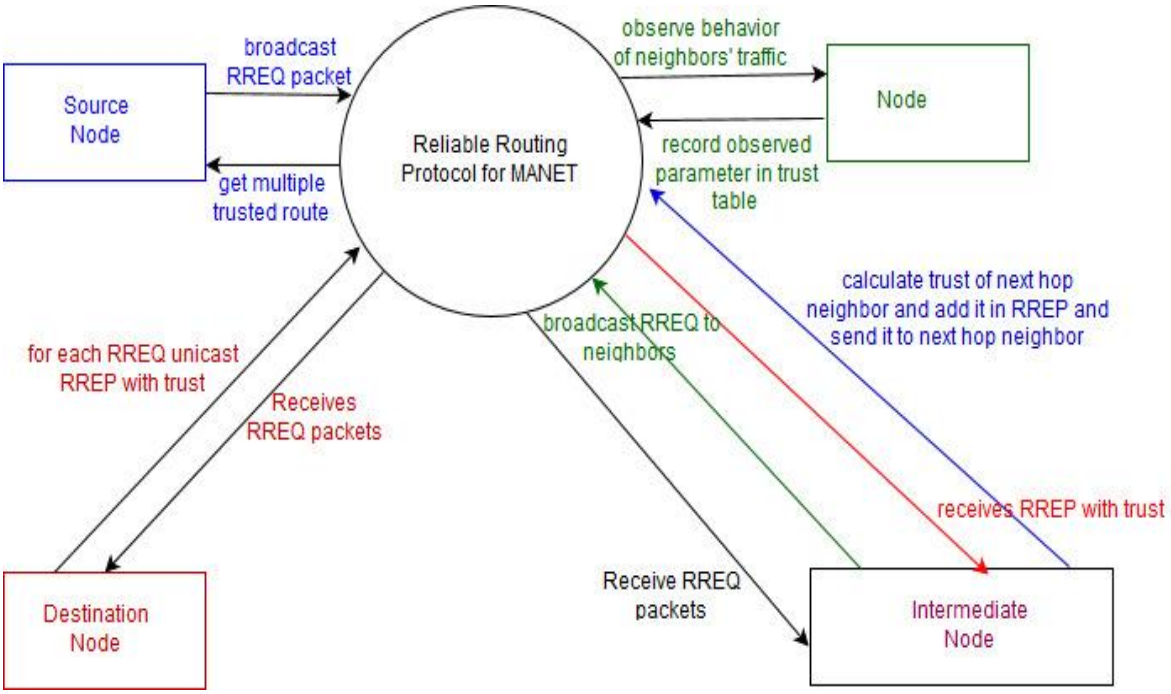


Figure 5.2 Context Flow Diagram of TMA-AODV routing

5.6.2 Algorithm of proposed system

The Trust based Mobility Aware AODV (TMA-AODV) routing protocol, which we have proposed in this thesis has mainly three modules: Traffic monitoring, Trust based routing and Trust calculation. The traffic monitoring module runs on each node which records the network parameters for each neighbour in the trust table by observing their traffic. Trust calculation is a function which receives all observed parameters of a node and returns a calculated trust value of the node. This function aggregates all parameters as a one trust value. The Trust based routing is the extension of AODV routing, which uses the trust calculation module to take routing decisions while forming route from source to destination. The algorithms for all these three modules are given below.

(1) Traffic monitoring

At each active node of a mobile ad hoc network, a process (Traffic_Monitor) is running which observes the traffic behaviour of all of its neighbours and records them in the trust table locally available at the node.

Entry in trust table for each neighbour consists of

Poi: number of packets observed for a neighbour node,

PFi: number of packets successfully forwarded by a neighbour node,

PDi : number of packets delayed at a neighbour node,

PERi: number of link break due to a neighbour node,

ALGORITHM : Traffic_Monitor

Input: NIL

Output : Updated_Trust_table: Change/Add an entry in trust table for each neighbour node.

Step 1: Begin

Step 2:Wait for a packet broadcasted by a neighbour

Step 3: If (packet coming from a neighbour)

Step 4: Creates a thread that receives the packet

Step 5: Search for entry for the neighbour in the local trust table.

Step 6: If (entry found in the trust table)

Step 7: Create a pointer R to that record.

Step 8: Else

Step 9: Create an entry for the neighbour in trust table pointed by R

Step 10: End if

Step 11: R.Poi++

Step 12: Compare received packet id with packet sent to the neighbour node

Step 13: If(match found)

Step 14: R.PFi++

Step 15: If (packet forwarded with the delay)

Step 16: R.PDi++

Step 17: End if

Step 18: End if

Step 19: Save R in trust table

Step 20: If (packet is RERR packet)

Step 21: Get IP address of the node appended in RERR

Step 22: Search for that node in trust table

Step 23: If (entry for that node found in the trust table)

Step 24: Create a pointer R to that record.

Step 25: Else

Step 26: Create an entry for that node in trust table pointed by R

Step 27: End if

Step 28: R.PERi++

Step 29: End if

Step 30: Save R in the trust table.

Step 31: Goto Step 2

Step 32:Else

Step 33: Goto Step 2

Step 34:End if

Step 35:End

(2) Proposed Trust based routing Protocol

ALGORITHM: TMA_AODV_ROUTING

Inputs: source_node, dest_node

Output: multiple paths from source_node to dest_node with the trust value of each path on source_node's routing table.

Step 1: Begin

Step 2: At source_node RREQ packet is created with dest_node address.

Step 3: source_node broadcasts RREQ packet to all its neighbours.

Step 4: For each neighbour node of source_node

Step 5: While (RREQ not reached to dest_node)

Step 6: RREQ is further broadcasted

Step 7: end loop

Step 8: RREP is created at dest_node with trust_val=0;

Step 9: next_node=dest_node

Step 10: While(next_node is not source_node)

Step 11: next_node= next hop neighbour

Step 12: search for entry of next_node in trust table

Step 13: Calculate trust_val of next_node

Step 14: Update RREP with trust_val+=calculated trust_val

Step 15: Unicast RREP to next_node

Step 16: End loop

Step 17: if (next_node is source_node)

Step 18: Route is stored in the routing table of source_node

Step 19: End if

Step 20: End for

Step 21: threshold_value= average of trust value of all routes.

Step 22: For each route

Step 23: if (trust value of route <threshold_value)

Step 24: remove the route from the routing table

Step 25: end if

Step 26: end for

Step 27: while (source has a data packet to send)

Step 28: for j=1 to N: number of routes from source to destination in the routing table

Step 29: use route j to send data packet

Step 30: end for

Step 31: end while

Step 32: End

(3) Trust Calculation

For calculating the trust value, following algorithm is used.

ALGORITHM: Trust_Calculate

Input: Node j whose trust value is needed

Output: $TV_{i,j}$: Trust value of a node j at node i

Step 1: Begin

Step 2: If entry for node j is found in trust table of node i

Step 3: Read the trust table on a node i and get observed and stored value for given node j

Step 4: Po_j: number of packets observed for a neighbour node j,

Step 5: PF_j: number of packets successfully forwarded by neighbour node j,

Step 6: PD_j: number of packets delayed at neighbour node j,

Step 7: PERj: number of link breaks due to node j,

// Calculate the W1,W2 and W3 at a node using observed values.

Step 8: $X=(Po_j-PF_j)$

Step 9: $Y=PD_j$

Step 10: $Z=PER_j$

Step 11: $W_1=(X/(X+Y+Z))$

Step 12: $W_2=(Y/(X+Y+Z))$

Step 13: $W_3=(Z/(X+Y+Z))$

Step 14: $TV_{i,j}= -W1*(Po_j-PF_j)- W2 * PD_j -W3 * PER_j$

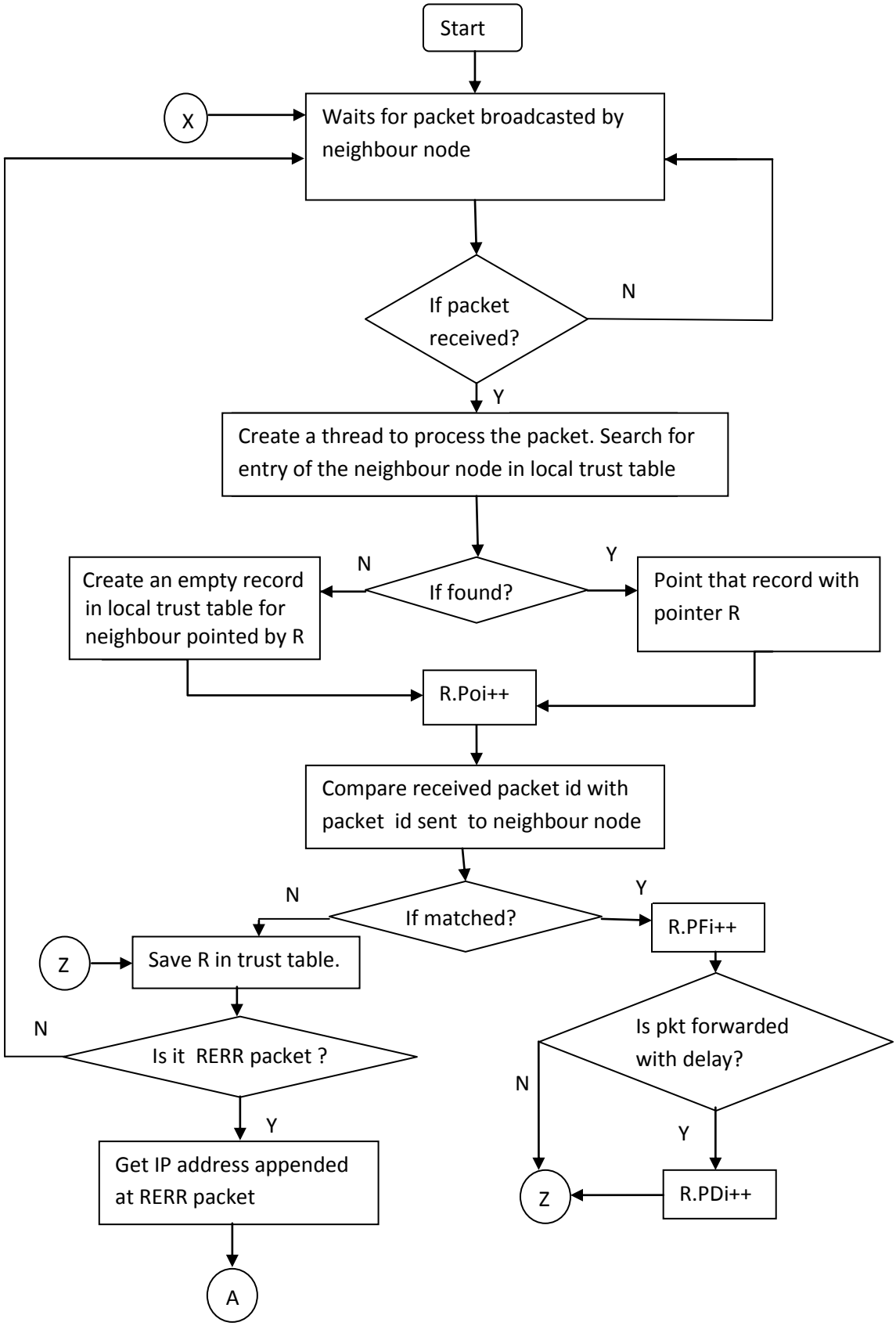
Step 15: Else

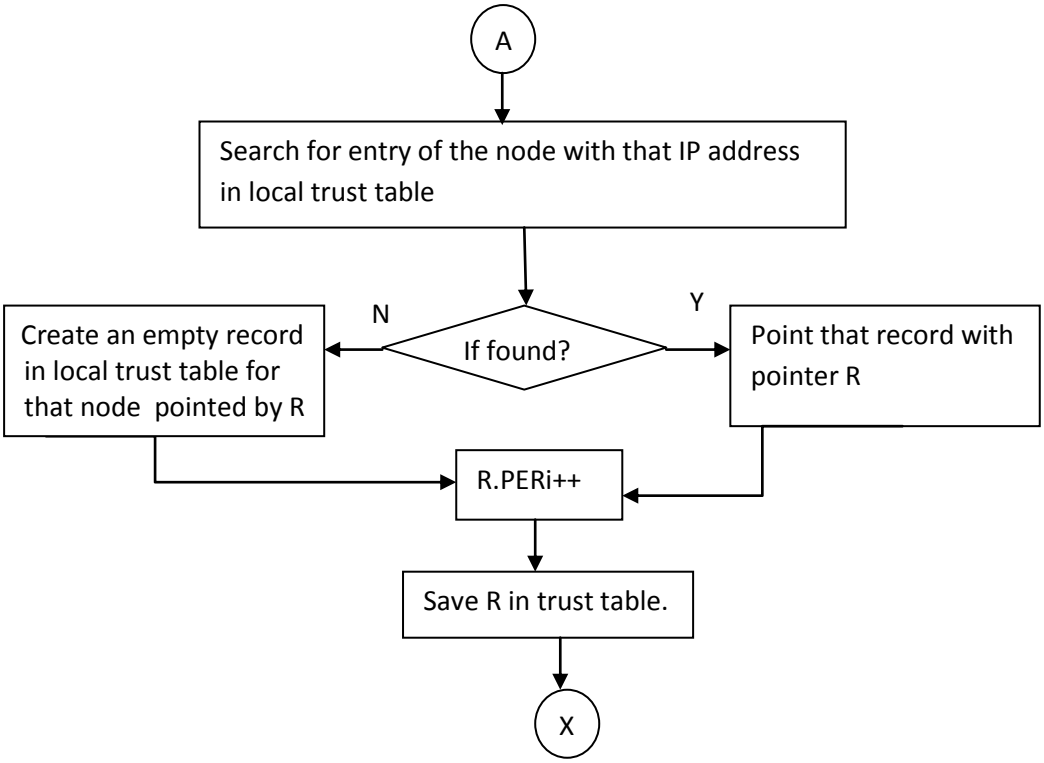
Step 16: $TV_{i,j}=$ threshold_trust

Step 17: End if

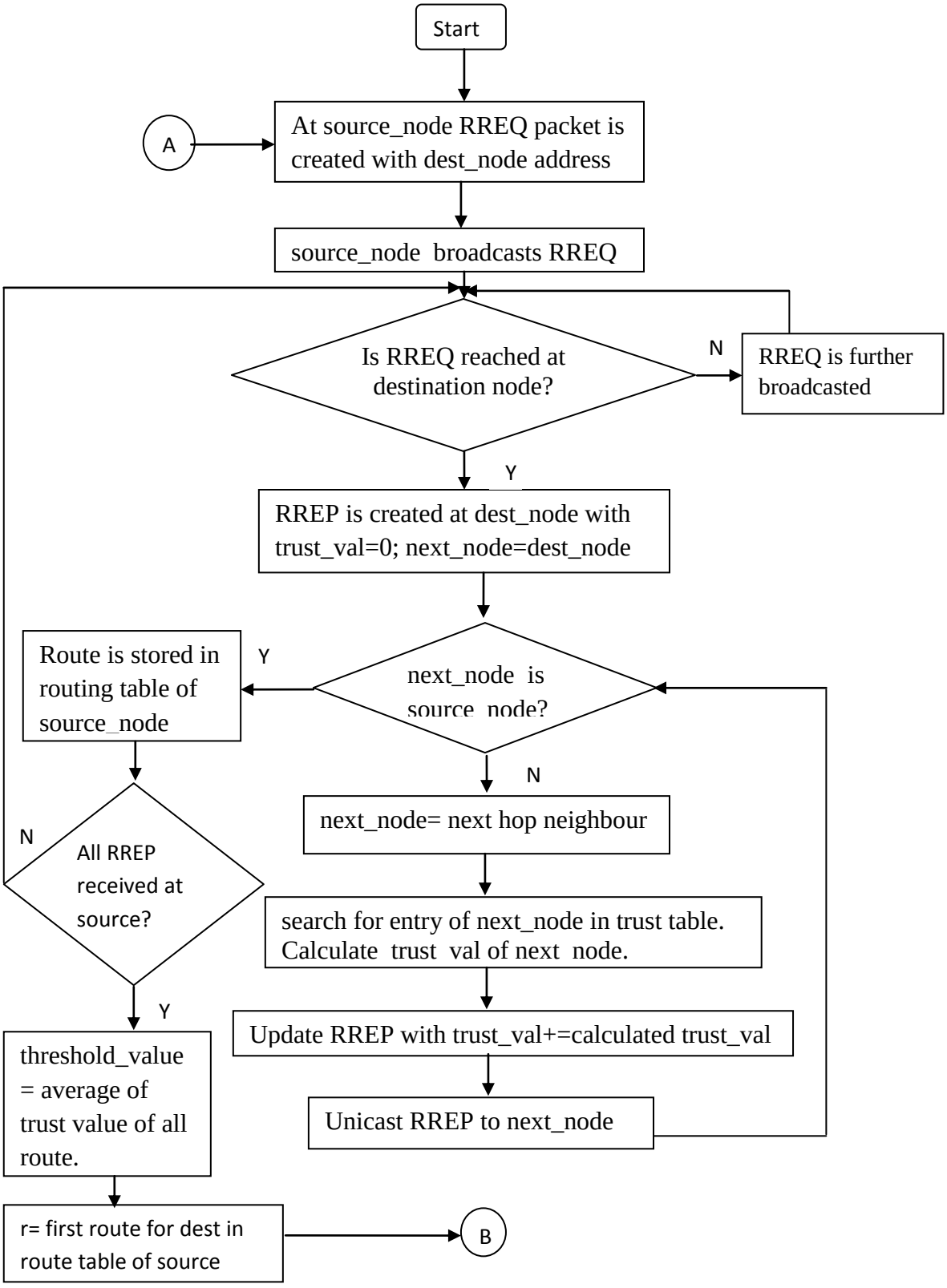
Step 18: End

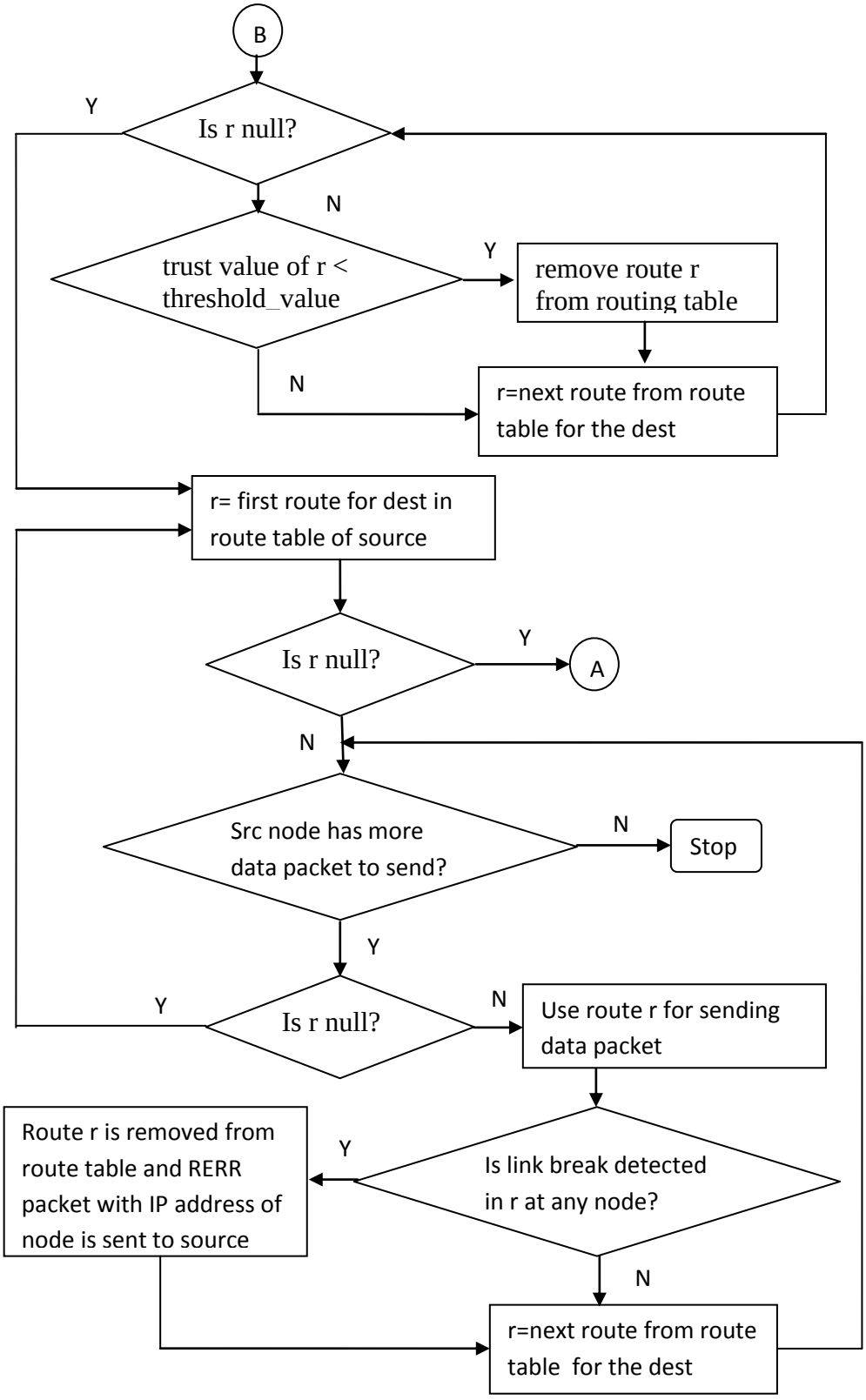
5.6.3 Flowchart of traffic monitoring module.




A
Search for entry of the node with that IP address in local trust table
If found?
N
Y
Create an empty record in local trust table for that node pointed by R
Point that record with pointer R
R.PERi++
Save R in trust table.
X

5.6.4 Flowchart of proposed routing module.




B
Is r null?
Y
N
trust value of r < threshold_value
Y
remove route r from routing table
N
r=next route from route table for the dest
r= first route for dest in route table of source
Is r null?
Y
A
N
Src node has more data packet to send?
N
Stop
Y
Is r null?
Y
N
Use route r for sending data packet
Is link break detected in r at any node?
Y
Route r is removed from route table and RERR packet with IP address of node is sent to source
N
r=next route from route table for the dest

5.7 Parameters for Performance Measurement

After introducing above trust based mobility aware AODV(TMA-AODV) routing in the MANET, we have considered the following parameters for performance measurement.

1) Throughput: the total data bits received by the destination node per second.
2) Route discovery time: the time taken by a source node to establish a route from the source node to the destination node for sending data packets.

We have used throughput for performance analysis because it is one of the important parameters to measure performance of any network. The various protocols for any network are always designed to improve throughput of the network. The throughput is measured in bits per second. It indicates the speed of data transmission.

The route discovery time is the time taken by a routing protocol to search a route from a source node to the destination node. An efficient routing protocol must have small route discovery time.

We want to compare the route discovery time taken with our proposed routing protocol against the time taken with standard AODV routing. We also want to compare the throughput of MANET with our proposed routing protocol against the throughput with standard AODV routing.

CHAPTER 6

Implementation Of Attacks And Its Effect On MANET

In this chapter, we have implemented the packet drop attack and the packet delay attack in an OPNET simulator and study its effect on an AODV routing within MANET. We have created few network scenarios to study the effect of the packet drop attack, then we have added the packet delaying attacker nodes in the network scenario and studied its effect on the AODV routing and the MANET. In section 6.1, we have discussed the research methodology that we used to implement our routing approach. In section 6.2, we have discussed about the malicious node models and its behaviour. The section 6.3 shows the implementation of the malicious node models in the simulator environment. In section 6.4, we have shown the result graphs with the packet drop attack. In section 6.5, we have shown the results showing the effect of packet drop and packet delay attack on AODV routing and MANET. We have also studied the effect of mobile nodes on the AODV routing and the MANET. The experimental setup and the results showing the effect of mobile nodes on the AODV routing within MANET are discussed in section 6.6. We have also studied the effect of the packet drop and the packet delay attack in the presence of the mobile node on the AODV routing and the MANET in section 6.7. The focus of this chapter is to cover following question.

How packet dropping attacker nodes, packet delaying attacker nodes and mobile nodes affect the performance of MANET with AODV as a routing protocol?

6.1 Research Methodology used for this thesis implementation

1) We have studied various literatures related to trust based routing in wireless network and done a comparative analysis to find out research gap and problem statement.
2) The literature survey helped to define an objective of the research.

3) We have used OPNET 11 for implementing our proposed algorithm (Trust based Mobility Aware-AODV : TMA-AODV) and performing all experiments/ comparative analysis.
4) To implement our proposed routing protocols, and attacks in OPNET, we need to do following major steps. (a) Create/modify the behaviour of wireless nodes to implement attacker nodes. (b) Build/modify routing protocol and implement our proposed routing scheme.
5) In OPNET, we have implemented various network scenarios and collect results and export those result values in MS Excel and use it to draw various graphs and comparisons.

Our research is **Qualitative** as we have implemented a trust based routing scheme which detects and avoids malicious/selfish activities in network and give stable route with less computation overhead and without incurring other communication cost.

Our research is **experimental** as we have set up the MANET network scenarios with TCP network traffic and attacker nodes (packet drop and packet delay) and prove the fairness of our proposed algorithm comparing results with standard AODV routing.

6.2 Malicious node models

We use a MANET model in OPNET to simulate AODV network. In the MANET, we have created two node models, which perform the packet drop and the packet delay attack respectively. The packet drop node model periodically drops data as well as control packets to disturb the network. The packet delay node model introduces random delay (50-150 ms) before forwarding each packet further. After creating the above node models, we have compared the performance of the AODV routing protocol (Route discovery time) and Wireless LAN (throughput) by creating an experimental setup with and without attacker nodes and study effect of attacker nodes. The packet drop attack disturbs the network by not forwarding the incoming packet further in the network. The packet drop attack frequently breaks the route between a source node and the destination node and hence, it reduces the throughput of the network and increases the route discovery time in a routing. The delay attacker node receives the packet and forward them as it is, but with some delay. Due to this delay the overall throughput of the network is reduced and the route discovery time is also increased.

Apart from these attacker nodes, the mobility of nodes also affects the performance of the network. In a MANET, the node of a network can move freely. The routing protocol of the network has to take care of changes of node position in network with time. Whenever the node which is a part of active route changes its position, route breaks. The route breakage information is sent to the source node by sending route error message. The source node has to search for new routes. This may decrease the throughput of the network. If the routing protocol avoids the mobile nodes of the network during route formation, there are less chances of route break and the throughput of the network may improve.

6.3 Implementation of Malicious nodes

In this thesis, we have implemented two malicious activities of a node. One is packet drop and the other is packet delay. We have implemented above both attacker nodes separately.

The packet drop attacker node is a node which is a part of active route in the MANET and periodically drops the data and the control packets. We have implemented malicious node in such a way that it updates some of the packets at lower layer such that they are dropped by higher layer.

The packet delay attacker node is a node which is a part of active route and introduced random delay before forwarding each packet. For implementing this attack, we have called the delay function before forwarding each packet to the next node.

6.3.1 Implementing malicious nodes in OPNET

In OPNET 11.5, we have updated a node model **wlan_wkstn_adv,** which comes with a standard **wireless LAN** package as **Drop_wlan_wkstn_adv_new** and **Dly_wlan_wkstn_adv**. In **Drop_wlan_wkstn_adv_new** the modification is done in such a way that the node accept first 50 packets and forward it as it is and modify next 20-30 packets in such a way that they are dropped at the higher layer. In **Dly_wlan_wkstn_adv,** a random number between 50 to 150 is generated and each packet is delayed for that many milliseconds before forwarding it.

The structure of node model is same as **wlan_wkstn_adv**. We have changed a process model **IP_Dispatch** of the drop node model and the delay node model as **IP_Dispatch_Drop_new** and **IP_Dispatch_Dly** respectively. In the modified IP dispatch

process, we have modified a function named **ip_dispatch_forward_packet().** This function is called to forward incoming packets to higher layer by the **ip** module of the node model when a packet arrives. We have also modified **ip_rte_datagram_higher_layer_forward** which is called by **ip_dispatch_forward_packet** function to forward the packets to the higher layer. The node model and the process model are shown in figure 6.1 and 6.2 respectively. In drop packet node model, we have created a state variable count_var to count incoming packets as shown in figure 6.3.

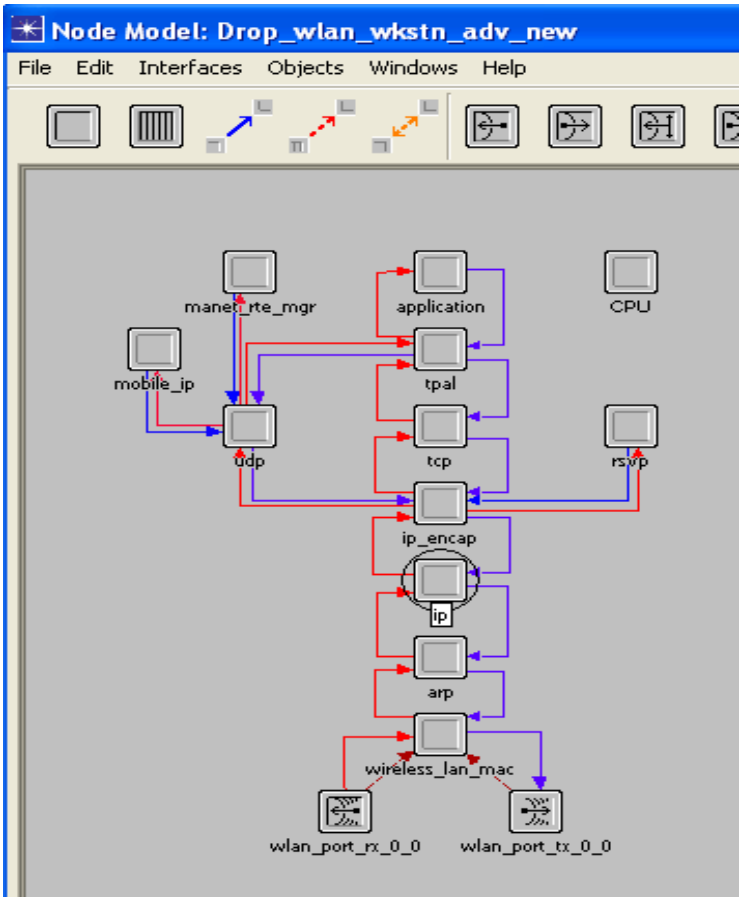


Figure 6.1 Malicious node model in OPNET (from OPNET)[84][85]

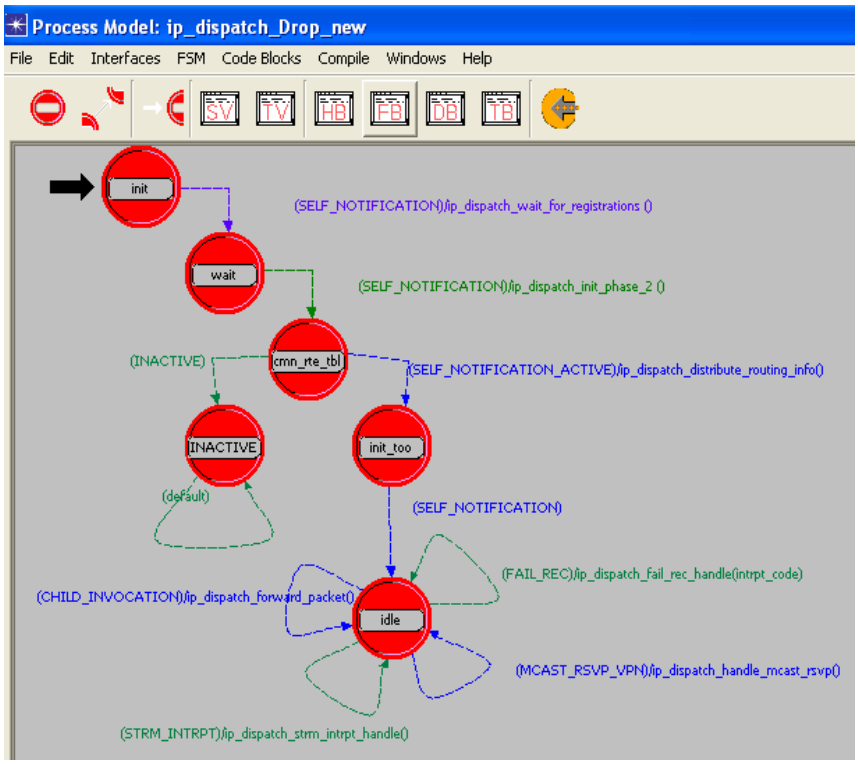


Figure 6.2 ip_dispatch process used by the malicious node model at IP module (from OPNET)[85]

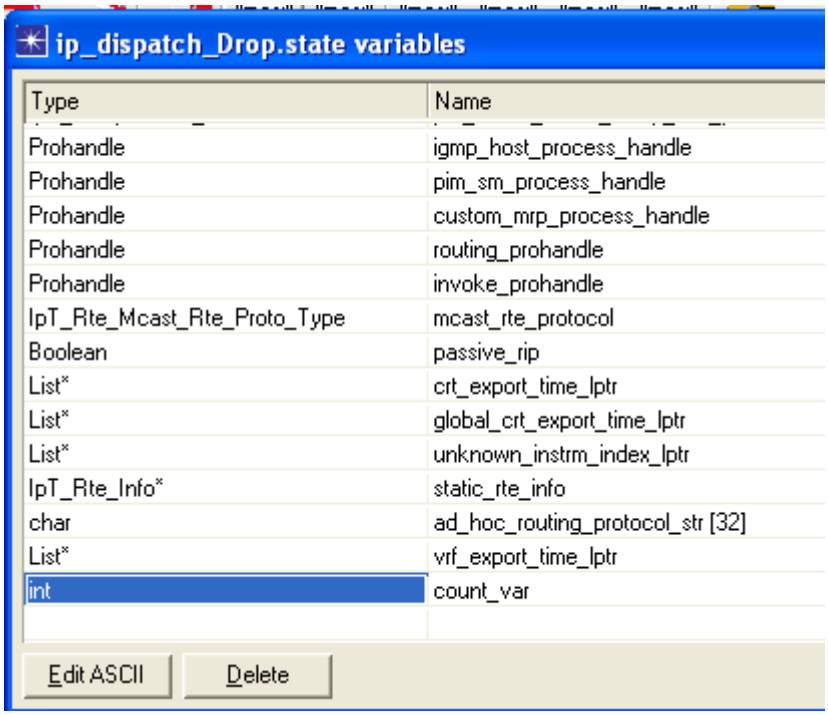


Figure 6.3 Declaration of counter in malicious ip_dispatch process(from OPNET) [85]

```
static void ip_dispatch_forward_packet (void)
	{
	……
	……
	count_var++;
	if (module_data.ip_ptc_mem.child_pkptr == OPC_NIL)
		{
		/* Packet to forward to "higher layers", which might include	*/
		/* some of the special child processes in IP.			*/
		Packet * pkptr = (Packet *)op_pro_argmem_access ();
		ip_rte_datagram_higher_layer_forward (pkptr);
		}
	else
		{
		/* Extract packet sent from child and have routing process	*/
		/* take care of it.						*/
		op_pro_invoke (routing_prohandle, module_data.ip_ptc_mem.child_pkptr);
		}
	FOUT;
	}



static void ip_rte_datagram_higher_layer_forward (Packet *frag_pk_ptr)
	{
		/*  Obtain a handle on the information carried in the "fields"	*/
		/* data structure in the incoming IP datagram.			*/
		op_pk_nfd_access (ip_pkptr, "fields", &pkt_fields_ptr);
		……
		……
		/* Set the destination address before sending the packet. */
		/* Only set the destination address if the field is not set. */
		if (! inet_address_valid (pkt_fields_ptr->dest_addr))
		{
			if(count_var<50)
			{
			pkt_fields_ptr->dest_addr=inet_address_copy (intf_ici_fdstruct_ptr->dest_addr);
			}
			else
			{
			pkt_fields_ptr->dest_addr=inet_address_copy (intf_ici_fdstruct_ptr->dest_addr);
			pkt_fields_ptr->src_addr = inet_address_copy (intf_ici_fdstruct_ptr->dest_addr);
			if(count_var==80)
				count_var=0;
			}
		}
		……
		……
	}
```

6.4 Effect of packet drop attack on AODV routing

After implementing packet drop attacker node model in OPNET, we have tested it by creating a wireless LAN scenario. These tests are implemented to see the effect of different packet dropping attacker nodes present in the network on route discovery time and in turn,

the overall throughput of the network. Each wireless LAN scenario has total 70 wireless nodes. The 69 nodes are working as an FTP client and one node is an FTP server. In the beginning, all the 69 client nodes send the FTP request to the FTP server for a file and in response to that the server will start sending data to each client node. All the nodes, including the FTP server are wireless node. They use the AODV routing as a routing protocol. In this experiment, we have studied the effect of our implemented packet drop attacker node model in MANET. For this study, we have created three malicious scenarios. In the first scenario, we have put 6 packet drop attacker node. In second and third scenario we have added 12 and 24 packet drop attacker nodes respectively. We also have created the same scenario with no attacker node and compared route discovery time and throughput of all malicious scenarios with results obtained with a non malicious scenario.

6.4.1 Simulation environment with 6 packet drop attacker nodes

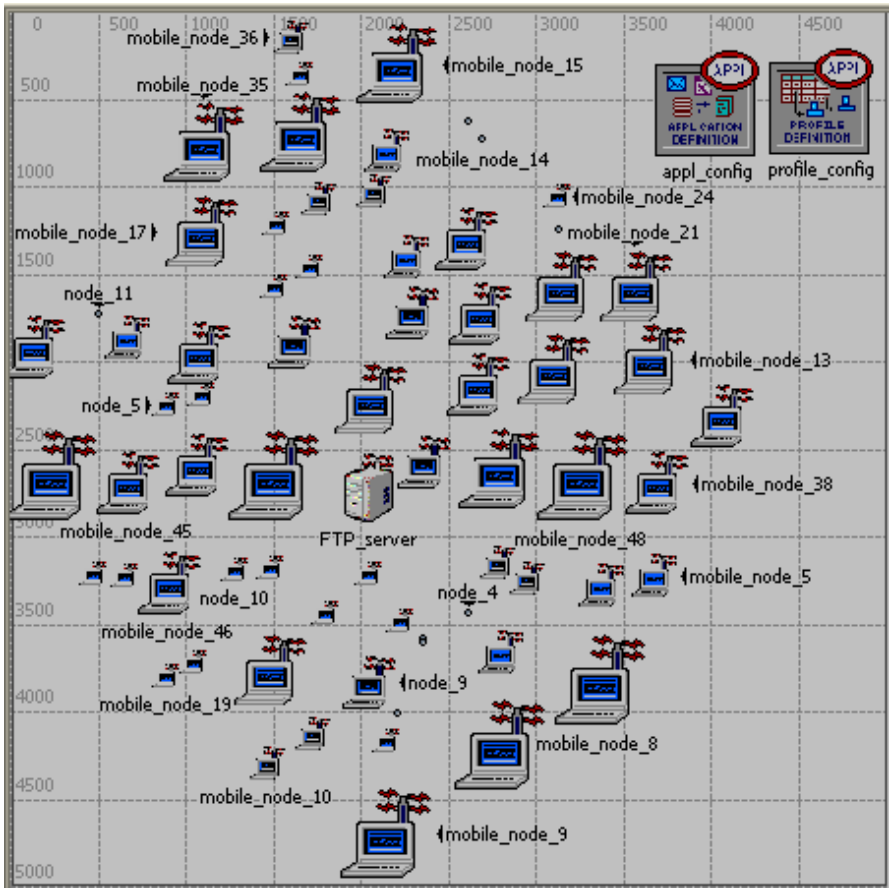


Figure 6.4 Experiment setup without attacker node (image from OPNET)[85]

In this experiment we have created two scenarios as shown in figure 6.4 and figure 6.5. The normal scenario contains all standard wireless nodes with AODV routing protocol. And the malicious scenario contains 6 attacker nodes encircled in figure 6.5 with AODV as

routing protocol. The attacker node performs a packet drop attack which we have implemented.

The traffic used for simulation is TCP traffic. We have used 69 wireless nodes and one FTP server. The simulation runs for 30 minutes. All the nodes in the wireless LAN are fixed node. All nodes in the network are configured to run multiple FTP sessions. TCP traffic is generated by configuring the Standard FTP Applications (Application Config object)[84][85] shown in figure 6.6.

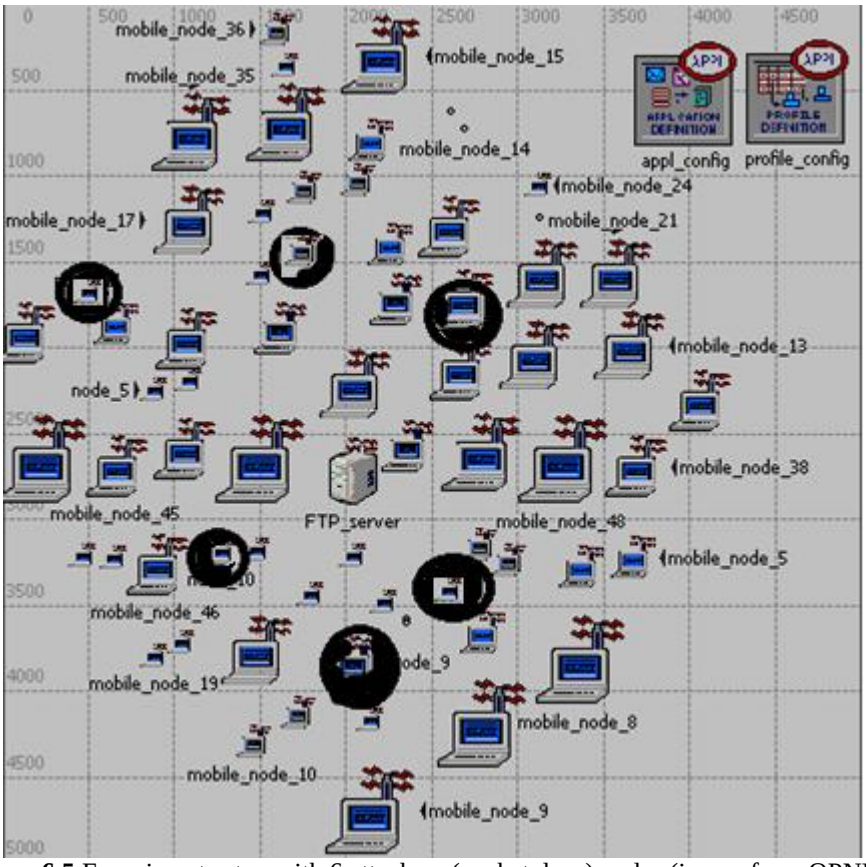


Figure 6.5 Experiment setup with 6 attackers (packet drop) nodes (image from OPNET)

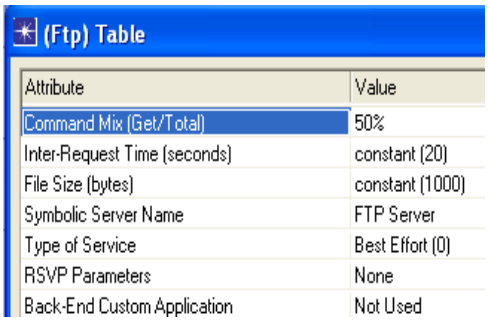


Figure 6.6 Configuration of FTP traffic for ad hoc nodes (image from OPNET) [84][85]

The result obtained after the experiment is compared. The route discovery time and the throughput of both scenarios are compared using the graph shown in figure 6.7 and the figure 6.8. From the graph we can conclude that, in the presence of packet drop attacker nodes the throughput of the network is degraded and the route discovery time is increased. In figure 6.7, we can see that initially, the route discovery time in the presence of the attacker nodes is almost same as the route discovery time of network without any attacker node. This is because our packet drop attacker node model initially, forwards some of the packets successfully and then starts dropping some packets. Again, after dropping some packets, node model behaves normally. The packet drop attacker node can be part of a route. However, after some time it starts dropping packets, which may break a route. When route breaks the routing algorithm has to search for a new route, which involves overhead. Due to this route discovery time and throughput of the network may degrade. Also, due to data packet loss by attacker nodes, the throughput of the network degrades in the presence of packet drop attacker nodes. The average route discovery time in the presence of 6 packet drop attacker nodes is (0.109340915 Sec) which is more compared to the average route discovery time (0.074528738 Sec) of a same network without any attacker nodes. The average throughput obtained in the presence of 6 packet attacker nodes is (308703.7 bps) which is less than the average throughput (350220.4 bps) obtained by the same network without any attacker nodes. Thus, the performance of the network is degraded in the presence of 6 packet drop attacker nodes.

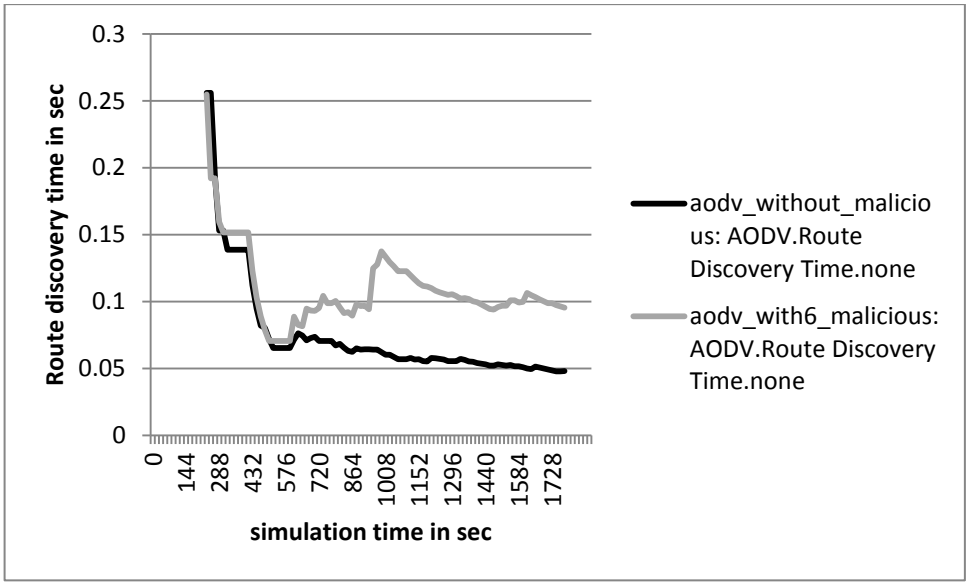


Figure 6.7 Comparison of Route Discovery Time (6 packet drop attacker)

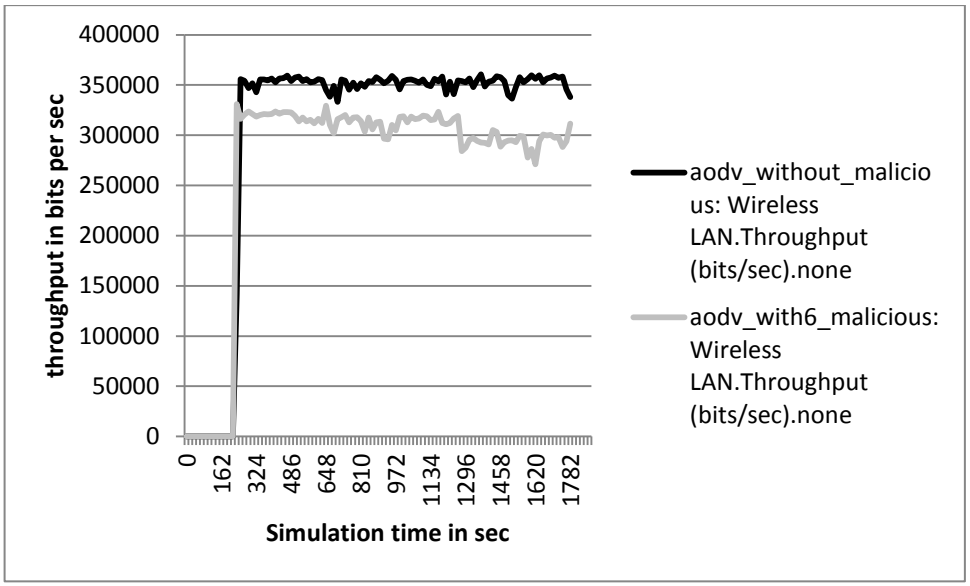


Figure 6.8 Comparison of Throughput (6 packet drop attacker)

6.4.2 Simulation environment with 12 packet drop attacker nodes

After studying the effect of 6 attacker nodes out of 69 nodes in a MANET, we have increased the packet drop attacker nodes from 6 to 12 and create one more scenario as shown in figure 6.9. In figure 6.9 encircled nodes are the packet drop attacker nodes. The traffic and all other parameters are same as previous scenarios.

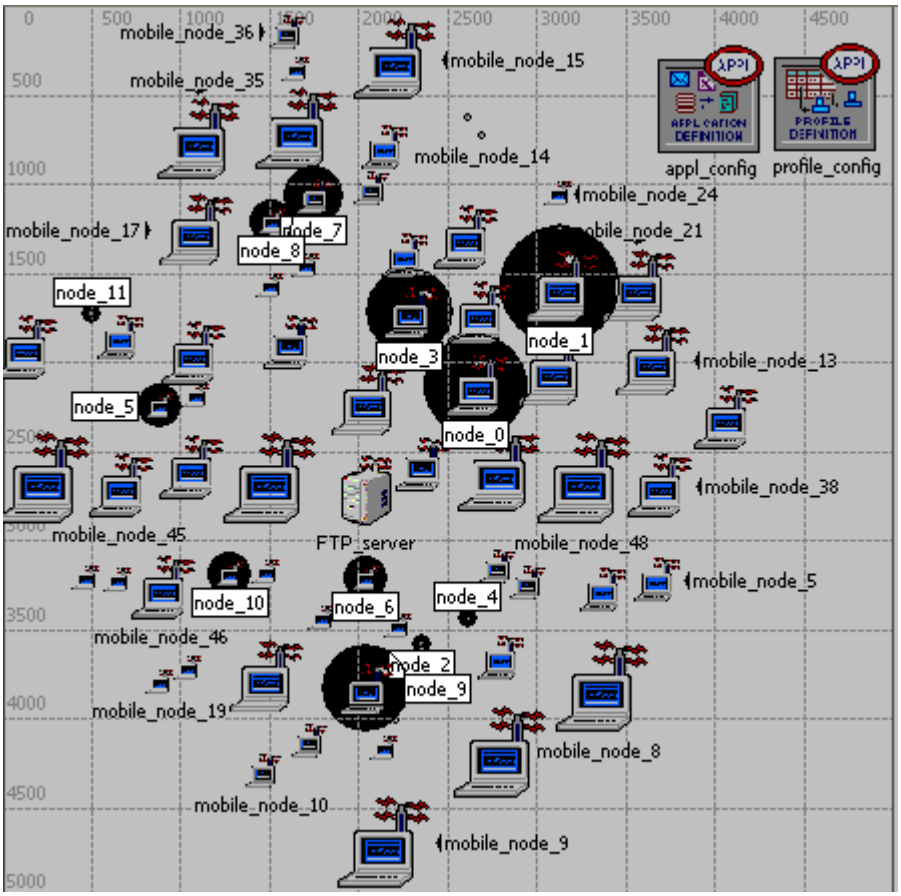


Figure 6.9 Experiment setup with 12 attackers (packet drop) nodes (image from OPNET)

The comparison of throughput and route discovery time in the absence of attacker node and with 12 packet drop attacker nodes is shown in figure 6.11 and figure 6.10 respectively. The route discovery time of malicious scenario with 12 packet drop attacker nodes is increased due to more packet loss. This packet loss occurs due to increase in packet drop attacker nodes in the network. The packet loss can be data packet loss and control packet loss. The data packet loss introduces more link breakage during routing, which increases the routing overhead in malicious scenario and hence reduces the throughput of the network. Also, control packet loss unnecessarily delays route formation process, which increases route discovery time. The average route discovery time with 12 packet attacker nodes is (0.298733205 sec) which is more than double compared to the average route discovery time(0.109340915 sec) of the same network with 6 packet drop attackers. The average throughput (304709.4 bps) of this network scenario is decreased than the average throughput (308703.7 bps) of the same network with 6 packet drop attacker nodes. This is due to increase in the packet drop attacker nodes in the network.

The more packet drop attacker nodes leads to more data and control packet loss, which affect the performance of the network.

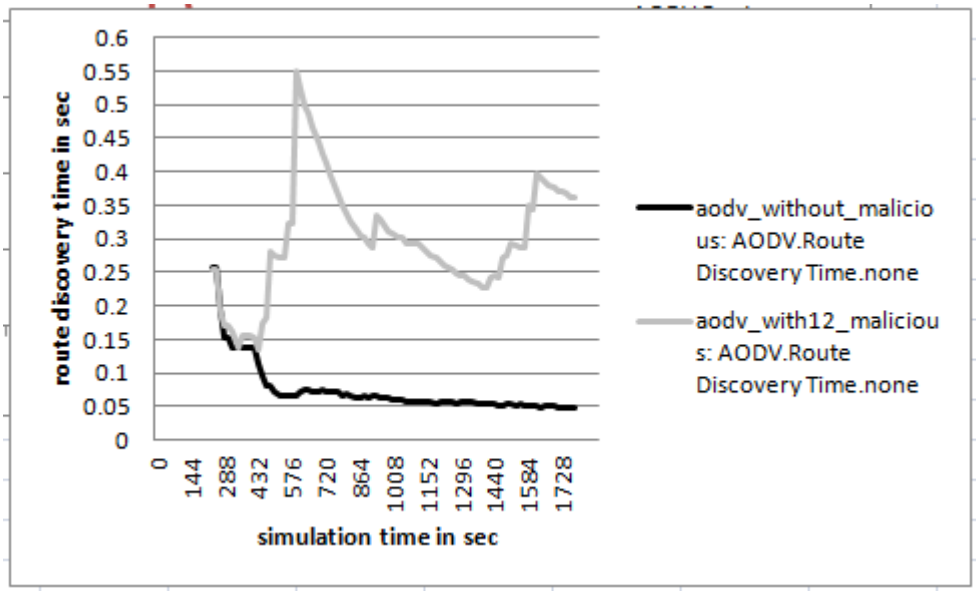


Figure 6.10 Comparison of Route Discovery Time (12 packet drop attacker)

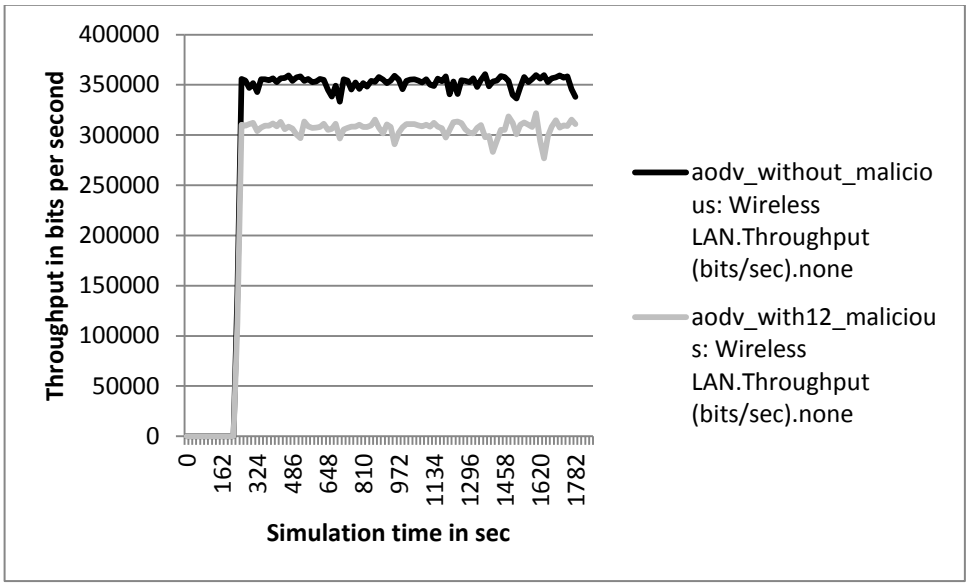


Figure 6.11 Comparison of Throughput (12 packet drop attacker)

6.4.3 Simulation environment with 24 packet drop attacker nodes

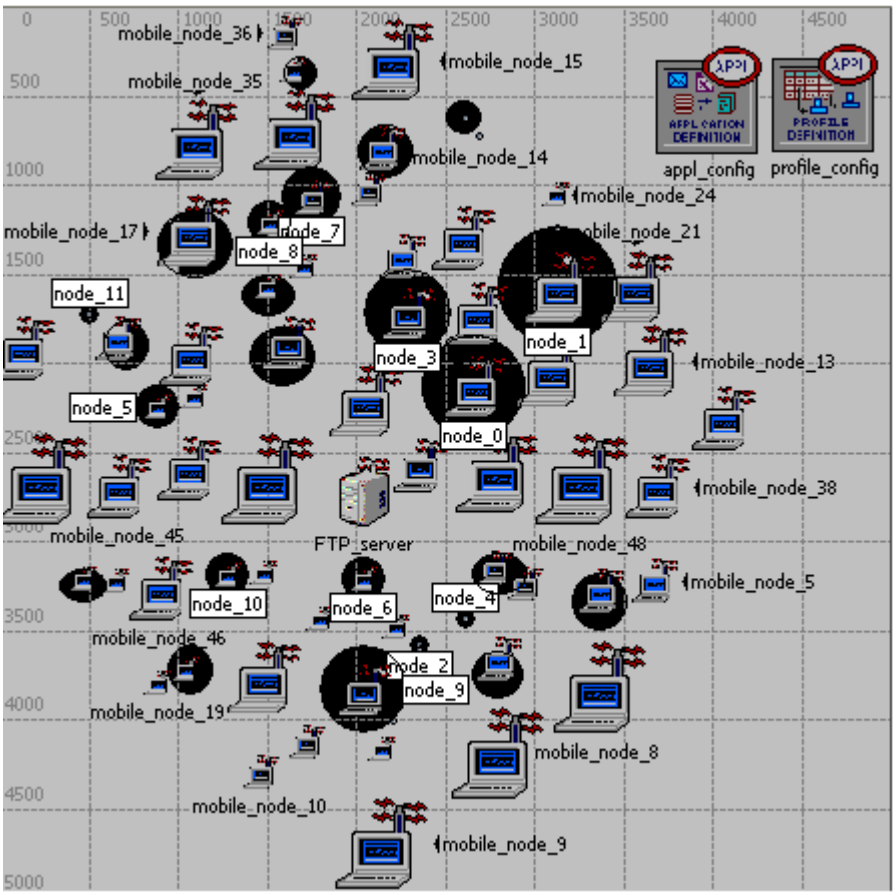


Figure 6.12 Experiment setup with 24 attackers (packet drop) nodes (image from OPNET)

After studying the effect of 6 and 12 attacker nodes out of 69 nodes in a MANET, we have added the 24 packet drop attacker nodes and create one more scenario (figure 6.12). The traffic and all other parameters are same as the previous scenarios. The comparison of throughput and route discovery time in the absence of attacker node and with 24 attacker nodes is shown in figure 6.13 and figure 6.14. From both graphs we can conclude that if we increase packet drop attacker nodes, the route discovery time is increased more compared to previous two scenarios. This is also true for throughput of the network. We have also compared the average route discovery time (2.666962079 sec) and the average throughput (264820.7 bps) obtained from the graph in figure 6.14 and figure 6.13 with the average route discovery time(0.109340915 and 0.298733205 sec) and the average throughput(308703.7 bps and 304709.4) of the same network with 6 packet drop attacker nodes and 12 packet drop attacker nodes. The network performance is drastically degraded with 24 packet drop attacker nodes. This is because in this network scenario we have used 24 packet drop attacker nodes out of 69 total wireless nodes of the network. As attacker nodes

are more and they are uniformly distributed in network, chances of attacker nodes to be a part of a route will be increased. Hence, the network performance degraded drastically.

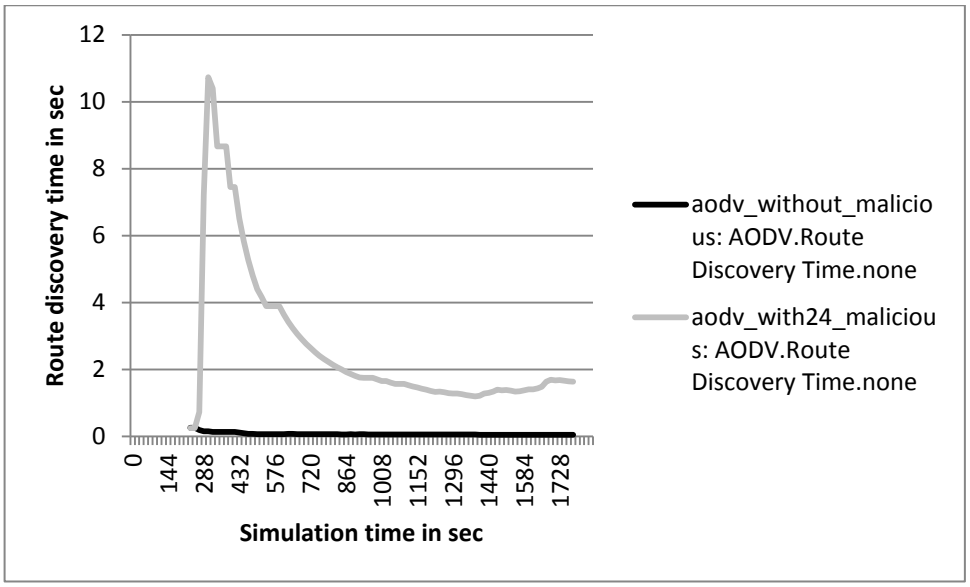


Figure 6.13 Comparison of Route Discovery Time (24 packet drop attacker)

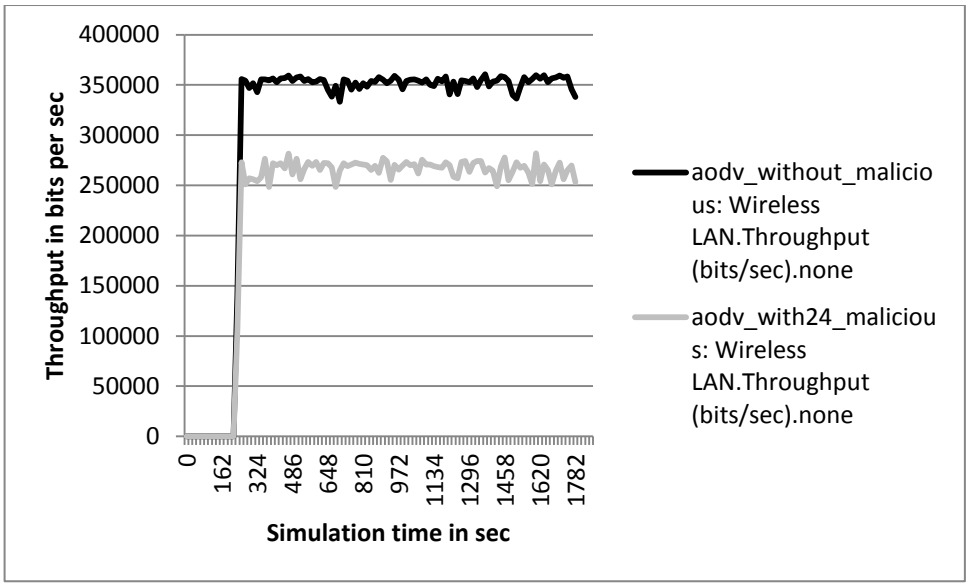


Figure 6.14 Comparison of Throughput (24 packet drop attacker)

6.4.4 Concluding Remarks

In this experiment, we have created total four scenarios. In all scenarios we have used AODV as a routing protocol on each node. One scenario has all non malicious nodes while other three scenarios are having 6, 12 and 24 packet drop attacker nodes as discussed earlier. We have compared the average of route discovery time and the average of throughput of all these four scenarios in table 6.1. The first entry in table 6.1 shows the

average of throughput and the average of route discovery time in the absence of any attacker node. Other three entries show that if we increase the attacker nodes in the network, the throughput of the network is reduced and route discovery time is increased. When we are using 6 or 12 packet drop attacker nodes out of 69 wireless nodes of the network, there are chances that some of the routes are not having any attacker node as an intermediate node. Hence, they will not affect network performance more. However, in a network scenario where we have used 24 packet drop attacker nodes out of 69 total wireless nodes of the network, which are uniformly distributed in the network, the chances of attacker nodes to be a part of an active route will be more. This may degrade the performance of the network drastically. This can be justified by the result which we have obtained. The graphs shown in figure 6.15 and 6.16 also justify our observations.

Table 6.1 Effect of packet drop attack on AODV routing and MANET

Sr No		AODV	
	Attacker nodes	Avg. Throughput (bps)	Avg. Route Discovery time(s)
1	0	350220.4	0.074528738
2	6	308703.7	0.109340915
3	12	304709.4	0.298733205
4	24	264820.7	2.666962079

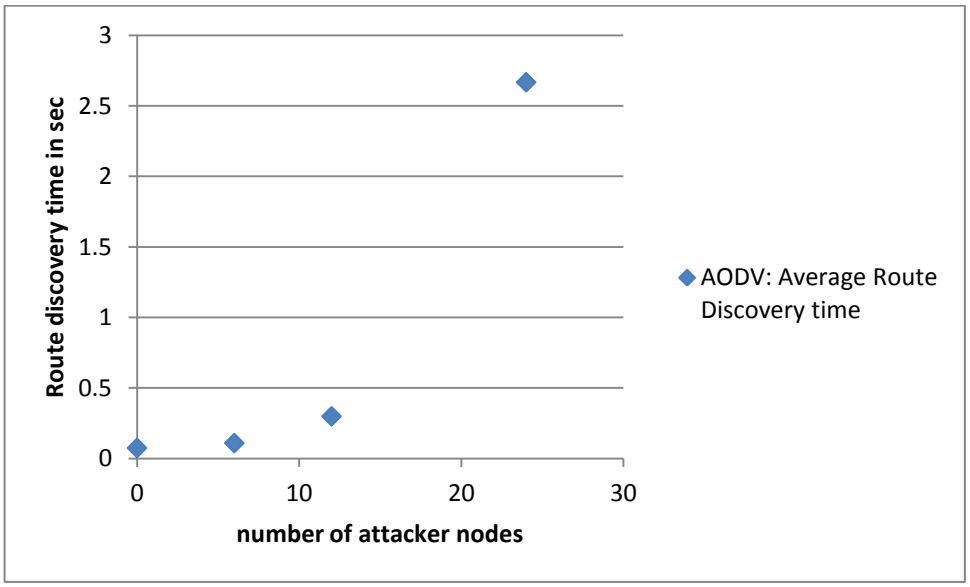


Figure 6.15 Average Route Discovery Time Vs number of attacker graph (drop attack)

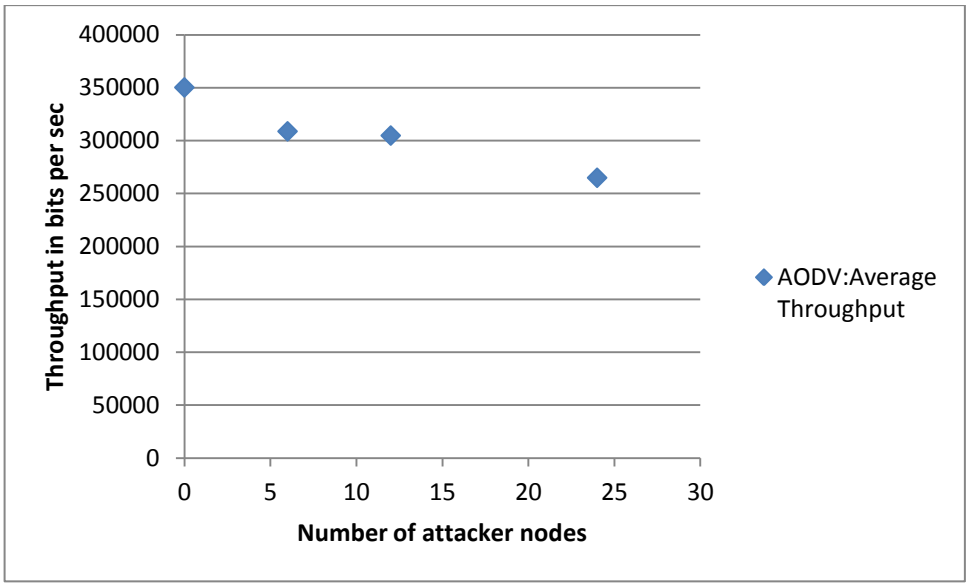


Figure 6.16 Average Throughput Vs number of attackers graph (drop attack)

6.5 Effect of packet drop and packet delay attack on AODV routing

6.5.1 Simulation environment with 3 packet drop and 3 packet delay attacker nodes

In this experiment, we have created two scenarios as shown in figure 6.4 and figure 6.17. The normal scenario (figure 6.4) contains all the standard wireless nodes with the AODV routing protocol. And the malicious scenario contains 6 attacker nodes encircled in figure 6.17. Out of 6 attacker nodes, three are packet drop attackers (black circled) and three are packet delay attackers (grey circled) with the AODV as a routing protocol.

The traffic used for simulation is TCP traffic. We have used 69 wireless nodes and one FTP server. The simulation runs for 30 minutes. All the nodes in the wireless LAN are fixed node. All nodes in the network are configured to run multiple FTP sessions. TCP traffic is generated by configuring the Standard FTP Applications (Application Config object) shown in figure 6.6.

The result obtained after the experiment is shown in figure 6.18 and figure 6.19. The throughput of both the scenarios is compared in the graph shown in figure 6.19. The figure 6.18 shows comparison of route discovery time of both the scenarios. From the graph we can conclude that, in the presence of packet drop and packet delay attacker nodes the throughput of the network is decreased and route discovery time is increased.

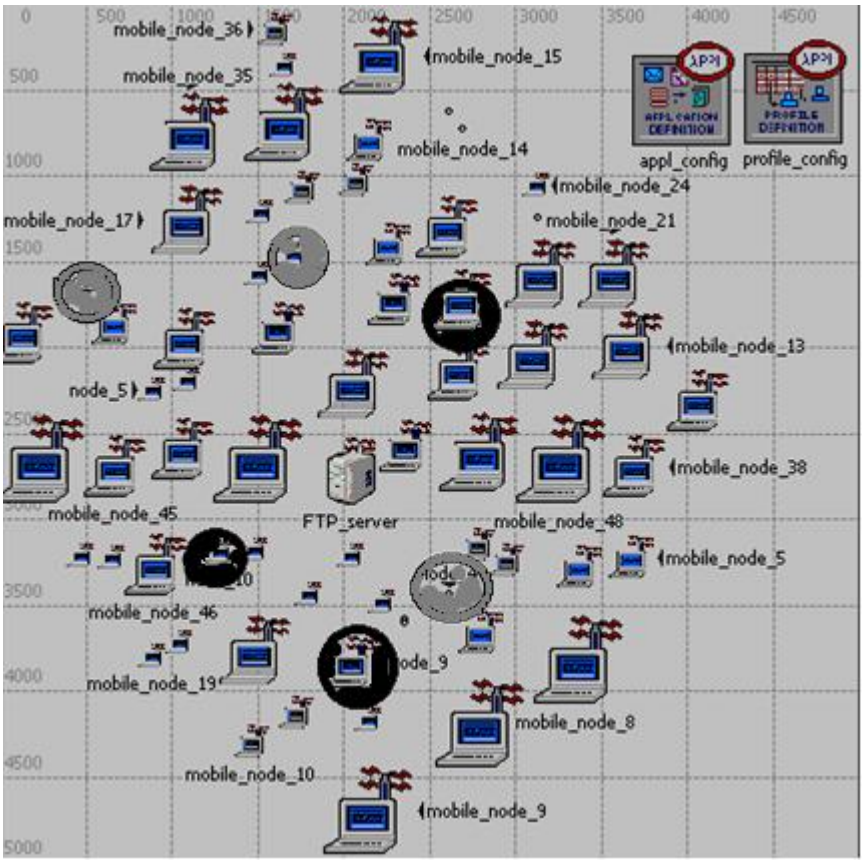


Figure 6.17 Experiment setup with 6 attackers (3 packet drop + 3 packet delay) nodes (image from OPNET)

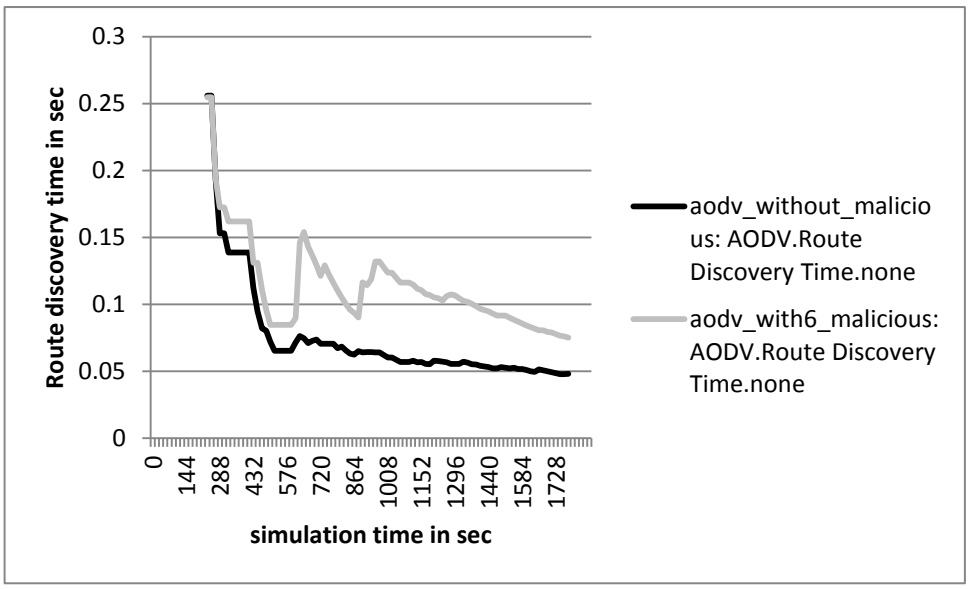


Figure 6.18 Comparison of Route Discovery Time (3 packet drop + 3 packet delay)

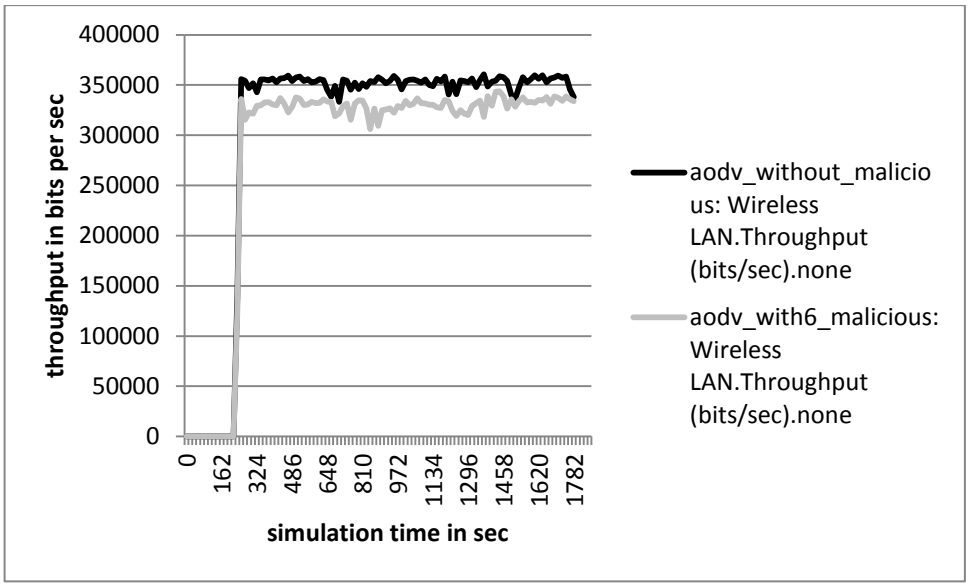


Figure 6.19 Comparison of Throughput (3 packet drop + 3 packet delay)

If we compare the average of route discovery time of this scenario (0.113924823 sec) with the average of route discovery time with 6 packet drop attacker node scenario (0.109340915 sec), we can see that the average route discovery time of packet drop and packet delay network is 0.01 sec more than network with only packet drop attacker nodes. The average of throughput with this scenario (327803.422 bps) is more compared to the average of throughput with 6 packet drop attacker nodes (308703.719 bps). The reason is, in 6 drop attacker node experiment, all 6 attackers nodes are configured to drop the packets. Hence, the receiver is not able to receive some of the packets sent by sender via the routes that include one or many of these attacker nodes. Hence, not all packets that are broadcasted during the simulation time is received by the receiver nodes. Whereas, in this experiment we have 3 nodes behaving as packet delaying nodes and the remaining 3 as drop nodes. Obviously, delaying the packet is less-hazardous event than completely dropping the packet. Hence, we see that the results are slightly better as the delaying nodes will not completely drop the packet, instead they will add some random delay before forwarding the packet, which is better (in expectation) than the previous experiment. Additionally, since having any kind of attacker node is always worse than having no attacker node, we see that the results with this experiment is lesser than the experiment with No attacker node as in that case, all nodes are behaving ideally and did not delay or drop any packet. Hence, the result of this experiment is in between the results of experiments with no attacker node and 6 packet dropping attacker nodes.

6.5.2 Simulation environment with 6 packet drop and 6 packet delay attacker nodes

After studying the effect of 6 drop and delay attacker nodes out of 69 nodes in a MANET, we have increased the attacker nodes from 6 to 12 and create one more scenario as shown in figure 6.20. In figure 6.20 encircled nodes are attacker nodes. Out of 12 attacker nodes, 6 nodes perform the packet drop attack (black circled) and 6 nodes perform the packet delay attack (grey circled). The traffic and all other parameters are same as previous scenarios.

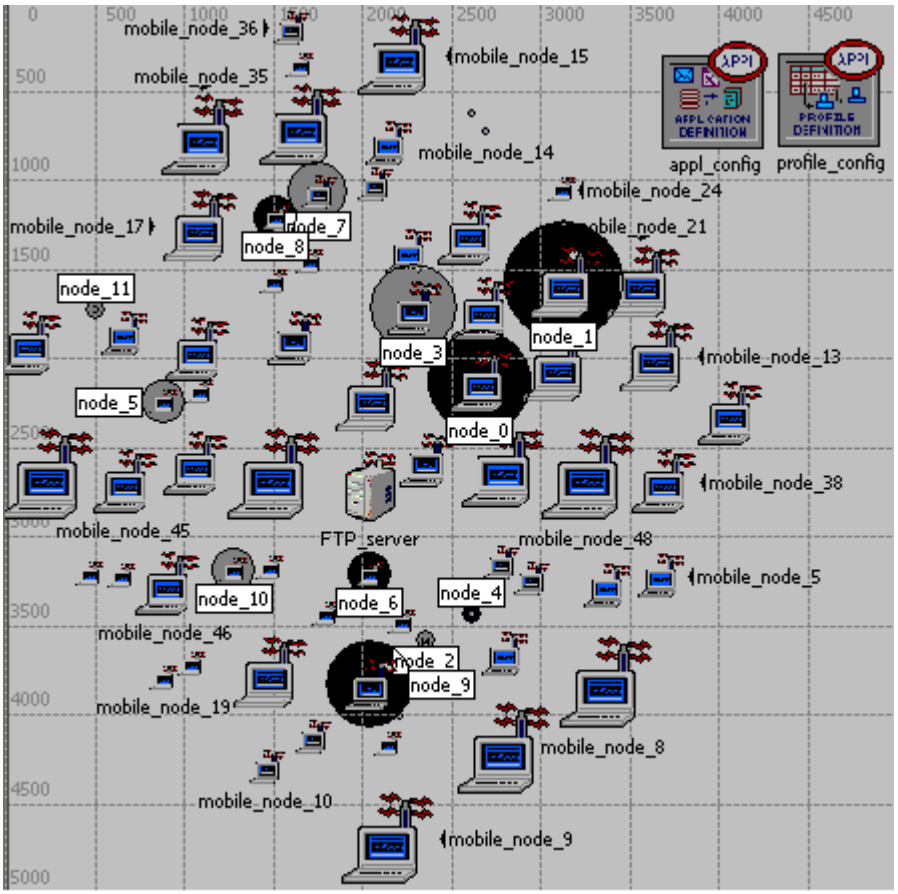


Figure 6.20 Experiment setup with 12 attackers (6 packet drop+ 6 packet delay) nodes (image from OPNET)

In this scenario we have doubled the attacker nodes compared to the last scenario (section 6.5.1), ideally the throughput should be reduced and route discovery time should be increased. The comparison of throughput and route discovery time in the absence of attacker node and with 12 delay and drop attacker nodes (6 packet drop and 6 packet delay) is shown in figure 6.22 and figure 6.21 respectively. From the graph we can see that, the presence of attacker nodes increase the route discovery time and reduced the throughput of the network.

Also, when we compared the results obtained with the last scenario (section 6.5.1) and this scenario, the average of throughput with this scenario is decreased by 6469 bps. This is because we have increased attacker nodes in this network scenario. The average of route discovery time(0.095035425 sec) with this scenario is decreased compared to the average of route discovery time(0.113924823 sec) with previous scenario. However, ideally this should be increased. This decrease in the route discovery time is due to following reason. The attacker nodes with this scenario is 6 packet drop attacker and 6 packet delay attacker which was 3 in the previous scenario. The packet drop attacker node does not drop all the incoming packets. They initially behave normally and after some time they start dropping packet coming to them. After dropping some packets, they again start behaving normally. Whereas the packet delay attacker nodes introduce delay before forwarding each packet. During the route discovery stage some of the attacker nodes may behave normally, thus the route is found without any disturbance. While sending data packets using this route if the attacker node is a part of the route it may start dropping packets and break the route. We have total 69 nodes out of which only 6 nodes are packet dropping attacker nodes. However, It is not always necessary that the route contains the attacker node as an intermediate node of a route. Thus, in this scenario more routes without any attacker nodes as an intermediate node may found and Hence, the route discovery time is reduced.

If we compare the average route discovery time of this scenario (0.095035425 sec) with the average route discovery time of a scenario with 12 packet drop attacker nodes (0.298733205 sec), we can see that with this scenario the average route discovery time is less. The reason is less packet loss due to less number of packet drop attacker nodes (6) in this scenario compared to a scenario with 12 packer drop attacker nodes. Due to the same reason the average throughput with this scenario (321633.9821 bps) is also more compared to the average throughput of the scenario (304709.4 bps) have 12 packet drop attacker nodes.

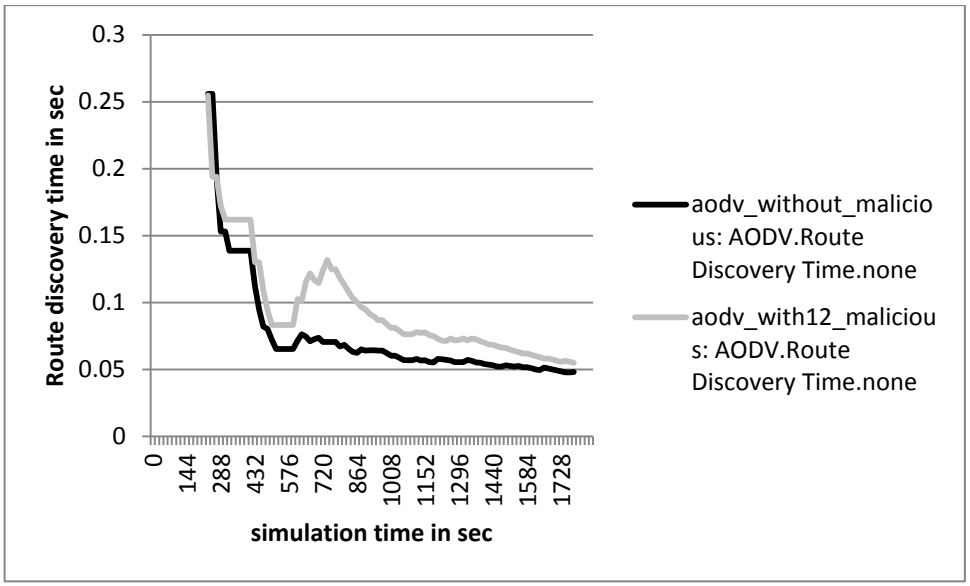


Figure 6.21 Comparison of Route Discovery Time (6 packet drop+ 6 packet delay)

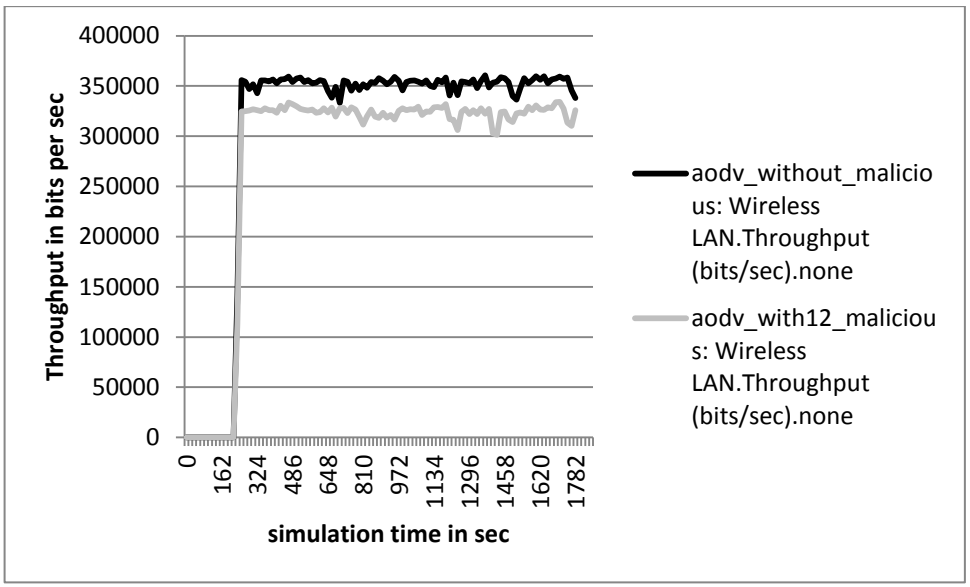


Figure 6.22 Comparison of Throughput (6 packet drop+ 6 packet delay)

6.5.3 Simulation environment with 12 packet drop and 12 packet delay attacker nodes

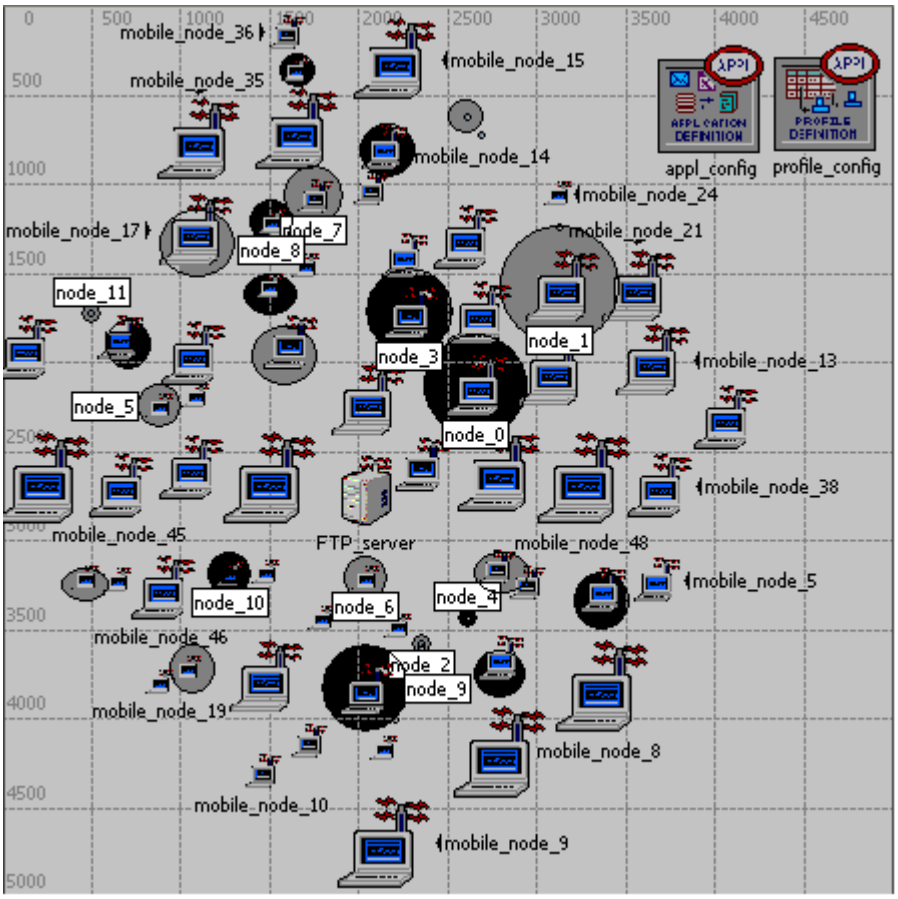


Figure 6.23 Experiment setup with 24 attackers (12 packet drop+12 packet delay) nodes (image from OPNET)

After studying the effect of 6 and 12 delay and drop attacker nodes out of 69 nodes in a MANET, we have created a new MANET network scenario as shown in figure 6.23.. We have used total 24 attacker nodes out of 69 nodes in our new MANET scenario. Out of these 24 attacker nodes, 12 nodes (black circled) implement a packet drop attack and the other 12 nodes (grey circled) perform the packet delay attack. The traffic and all other parameters are same as previous scenarios. The comparison of throughput and route discovery time in the absence of attacker node and with 24 attacker nodes is shown in figure 6.25 and figure 6.24 respectively. The figure clearly shows that in the presence of attacker node the route discovery time is increased and the throughput is decreased compared to a network without any attacker nodes.

Compared to the last two scenarios (section 6.5.1 and 6.5.2), here we have added 12 packet drop and 12 packets delay attacker nodes in the network. Hence, the route discovery time is increased and the throughput is reduced more compared to last two experiments. This is because of the addition of attacker nodes in the network. With 6 packet drop and packet delay attacker nodes the average route discovery time is (0.113924823 sec), which becomes (0.095035425 sec) with 12 packer drop and delay attacker nodes. With 24 packet delay and drop attacker nodes, the average of route discovery time becomes (0.527558508 sec), which is very large compared to last two scenarios. As attacker nodes are 27% of total nodes in the network, there are more chances of them to be a part of the route and hence affect the route discovery time and throughput of the network.

If we compare the average route discovery time (0.527558508 sec) of this scenario with a scenario which has 24 packet drop attacker nodes (2.666962079 sec), we can that the route discovery time is decreased with this scenario. The average of throughput is also increased with this scenario (300273.2874 bps) if we compare it with the average of throughput with the same network with 24 attacker nodes (264820.7 bps). The reason for this improvement is the half packet drop attacker nodes in this scenario and same number of delay attacker nodes, which are less disturbing than packet drop attacker nodes.

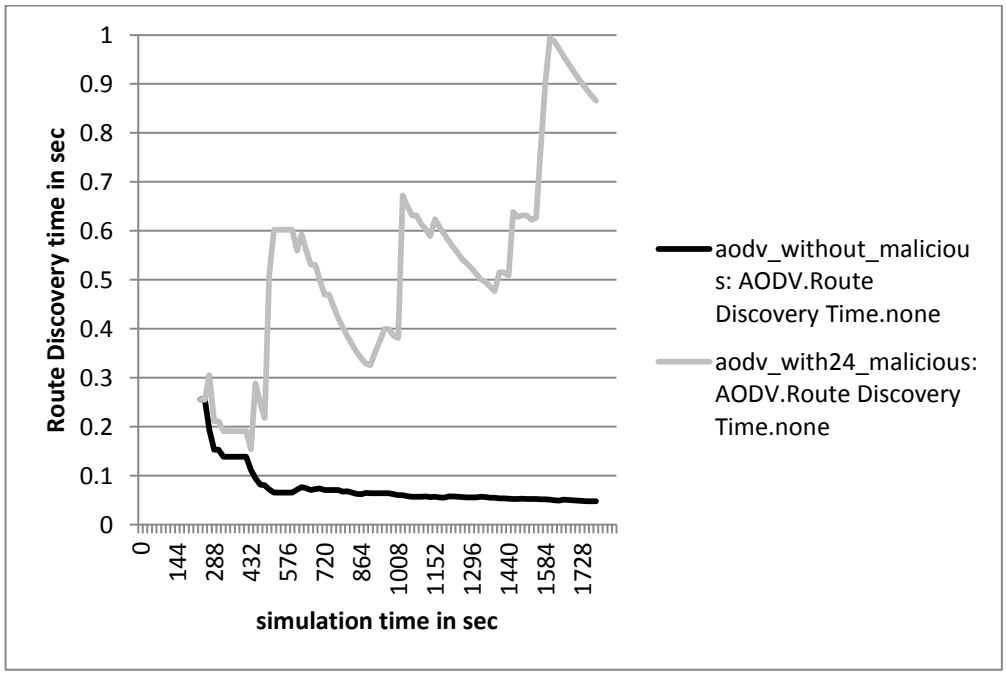


Figure 6.24 Comparison of Route Discovery Time (12 packet drop+12 packet delay)

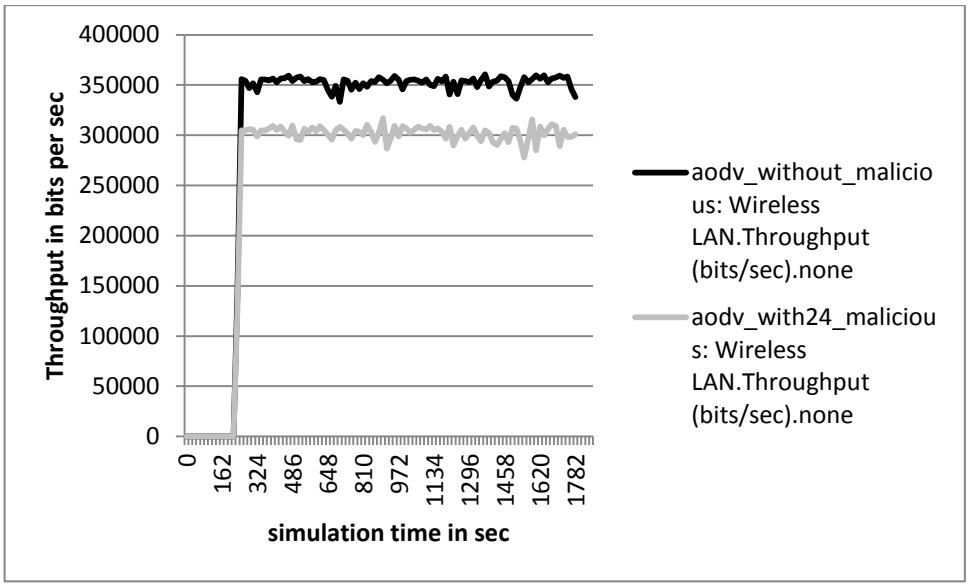


Figure 6.25 Comparison of Throughput (12 packet drop+12 packet delay)

6.5.4 Concluding Remarks

In this experiment we have created total four scenarios. In all scenarios we have used AODV as a routing protocol on each node. One scenario has all non malicious nodes while other three scenarios are having 3, 6 and 12 packet drop attacker nodes and 3, 6 and 12 packet delay attacker nodes as discussed earlier. Thus, in total they have 6, 12 and 24 attacker nodes. We have compared the average route discovery time and the average throughput of all these four scenarios in table 6.2. The first entry in table 6.2 shows the average throughput and the average route discovery time in the absence of any attacker node. Other three entries show that if we increase the attacker nodes in the network, the throughput of the network is reduced and the route discovery time is increased. This can be shown in the graphs shown in figure 6.26 and 6.27. The reduction in throughput is less compared to only packet drop attack. This is because in this experiment we have reduced packet drop attacker nodes by half and introduced packet delay attacker nodes. The packet drop attacks introduced the packet/data loss and it is more destructive than the packet delay attack.

Table 6.2 Effect of packet drop and delay attack on AODV routing and MANET

Sr No		AODV	
	Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)
1	0	350220.4036	0.074528738
2	6	327803.4227	0.113924823
3	12	321633.9821	0.095035425
4	24	300273.2874	0.527558508

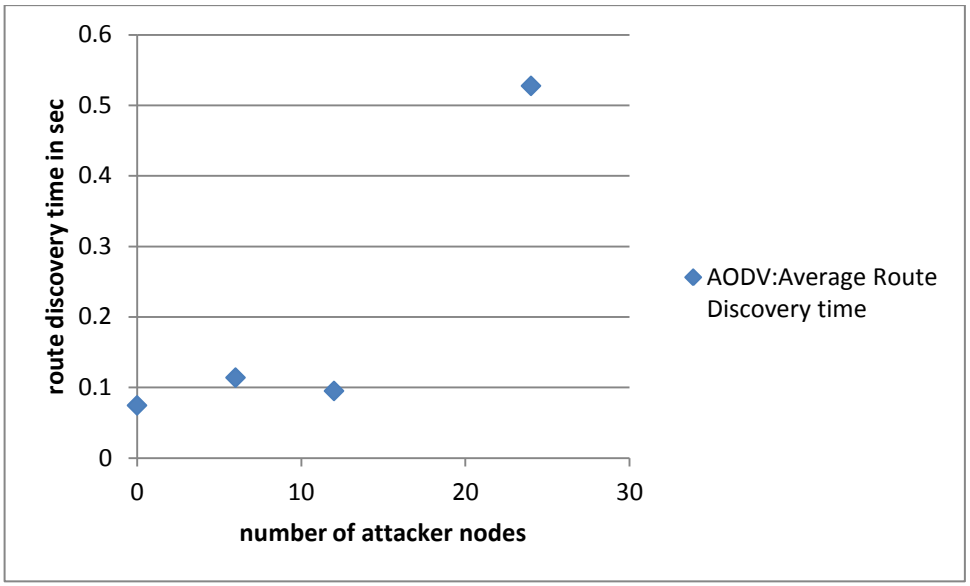


Figure 6.26 Average Route Discovery Time Vs number of attackers graph (drop+delay attack)

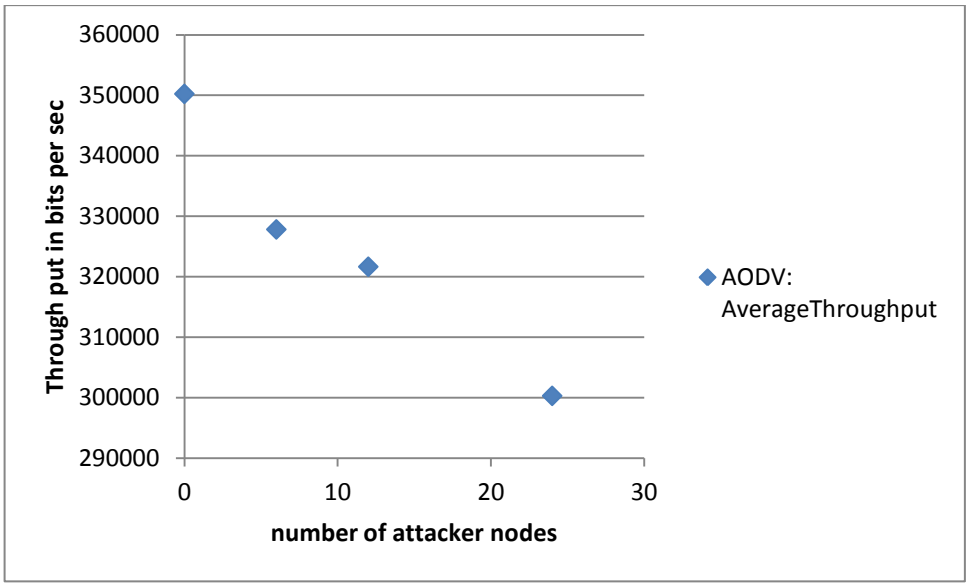


Figure 6.27 Average Throughput Vs number of attackers graph (drop+delay attack)

Also, if we compare tables 6.1 (results of only packet drop attacker nodes) with table 6.2, we can see that reduction in throughput with packet drop attack is more compared to packet drop and delay attack. This is because the packet drop attack is more destructive than packet delay attack. The route discovery time with the packet drop attacker nodes is also more compare to drop and delay attacker nodes. This is due to the packet drop attacker node will drop data as well as control packets, which affect the routing algorithm functionalities more compared to the delay attack.

6.6 Effect of mobile nodes with AODV routing

To study the effect of mobile nodes on the throughput of the network and the route discovery time of AODV routing, we have created two scenarios. The first scenario is same as the MANET scenario we have created in section 6.4.1 (figure 6.4). All the nodes of this network are fixed node with AODV as a routing protocol. The second scenario contains all standard wireless nodes with AODV routing protocol and 16 mobile nodes which move randomly as shown in the figure 6.28. In this figure, the mobile nodes are displayed using green arrows.

We have implemented this network scenario to study the effect of mobile nodes on the route discovery time of the AODV routing and the throughput of the network. We are doing this experiment to get some base results which will be compared with the results of our proposed routing protocol (Trust based Mobility Aware-AODV). In our proposed routing protocol (TMA-AODV), we have implemented a trust based routing, which detects packet drop attacks and packet delay attacks and creates a route with no or less attacker nodes. In our routing protocol, we have also created a route with less mobile nodes as an intermediate node to avoid link breaks. If in a network scenario we are keeping more mobile nodes, it would be difficult to study the effect of malicious nodes with mobile nodes. That is because the results are affected by both mobile nodes as well as malicious nodes. To reduce the effect of mobile nodes in the results, we have kept them less.

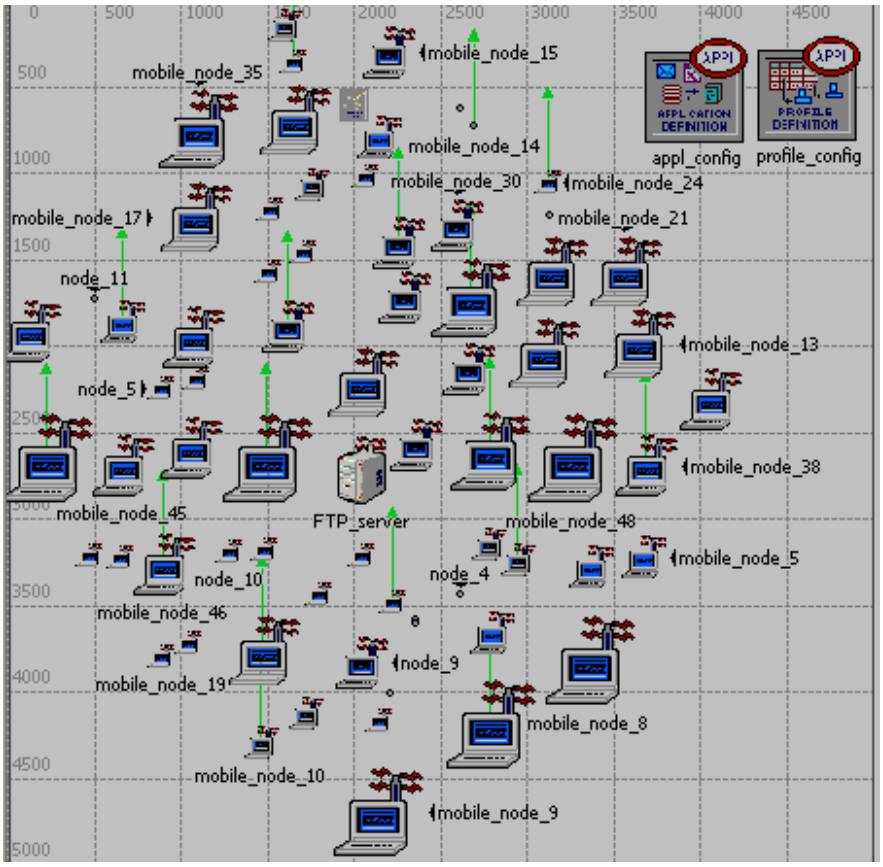


Figure 6.28 Experiment setup without attacker node and with 16 mobile nodes (image from OPNET)

The traffic used for simulation is TCP traffic. We have used 69 wireless nodes and one FTP server. Among 69 nodes, 16 nodes are mobile and they move in random directions. The simulation runs for 30 minutes. All the other nodes in the wireless LAN are fixed node. All nodes in the network are configured to run multiple FTP sessions. TCP traffic is generated by configuring the Standard FTP Applications (Application Config object) shown in figure 6.6.

The average of results obtained after running the simulation for 30 minutes shown in following table 6.3. The table clearly shows that the presence of mobile nodes reduces the average throughput of the network. This is due to the movement of mobile nodes in the network. If a node which is a part of the active route move outside the range of their next hop neighbour, the route breaks. This enforces the source node to search for a new route which adds routing overhead and due to this overhead the throughput of the network is affected. The average route discovery time of network in the presence of mobile nodes should be increased. In presence of mobile nodes in a network, routing control packets (RREQ, RREP) may drop due to movement of nodes. This leads to delay in the route formation process by routing protocol and hence increase the route discovery time. The

table 6.3 shows the comparison of the average route discovery time and the average throughput of the network with AODV routing in the absence of both attacker nodes and mobile nodes and the network with AODV routing in the presence of only mobile nodes. ↓ shows the percentage reduction in throughput and ↑ shows the percentage increase of the route discovery time in the presence of mobile nodes.

Table 6.3 Effect of mobility on route discovery time of AODV and throughput of MANET

Attacker node	AODV: Average Route Discovery time(sec)	AODV: Average Route Discovery time (sec) with mobility	AODV: Average Throughput (bits per sec)	AODV: Average Throughput (bits per sec) with mobility	Throughput (bits per sec)	Route discovery time (sec)
0	0.074528738	0.094526	350220.4	321504.213	8.2%↓	26.3%↑

6.7 Effect of packet drop and delay attack on AODV routing with mobile nodes

In the section 6.4, we have studied the effect of the packet drop attack on the route discovery time of AODV routing and the throughput of the MANET. In that experiment we have taken all wireless fixed nodes. The reason is we only wanted to study the effect of packet drop attacker nodes. If we include mobility in those network scenarios, the results may be affected due to mobility. Also, in case of packet dropping attacker, the node drops incoming packets and affect the overall network performance. If we add mobility with packet drop, they may degrade network performance badly. Due to this reason, we have not included the study of packet drop attack with mobile nodes in the network.

In this section, we have studied the effect of presence of packet drop attacker nodes, packet delay attacker nodes and mobile node in the network and how does it affect the route discovery time and throughput of the network. To avoid more disturbances in the network due to mobility, we have added mobility in 24% of network nodes in each network scenario. We have created three simulation network scenarios. In this network setup, we have kept 9%, 18% and 27% attacker nodes and 24% mobile nodes. The attacker nodes contain half packet drop attacker nodes and half packet delay attacker nodes. Ideally, in the presence of attacker nodes and the mobile nodes, the route discovery time of an AODV routing should be increased and the throughput of the network should be decreased. This performance degradation should be more than the previous experiment setup because we have added the mobile nodes in the network.

6.7.1 Simulation environment with 3 packet drop attacker nodes, 3 packet delay attacker nodes and 16 mobile nodes

In this experiment, we have created two scenarios as shown in figure 6.28 and figure 6.29. The normal scenario contains all standard wireless nodes with AODV routing protocol and 16 mobile nodes which move randomly as shown in figure 6.28. In this scenario, mobile nodes are displayed using green arrows. The malicious scenario is same as normal scenario. It contains 6 attacker nodes encircled in figure 6.29. Out of 6 attacker nodes, three are packet drop attackers (black circled) and three are packet delay attackers (grey circled) with the AODV as routing protocol.

The traffic used for simulation is TCP traffic. We have used 69 wireless nodes and one FTP server. Among 69 nodes, 16 nodes are mobile and they move in random directions. The simulation runs for 30 minutes. All the other nodes in the wireless LAN are fixed node. All nodes in the network are configured to run multiple FTP sessions. TCP traffic is generated by configuring the Standard FTP Applications (Application Config object) shown in figure 6.6.

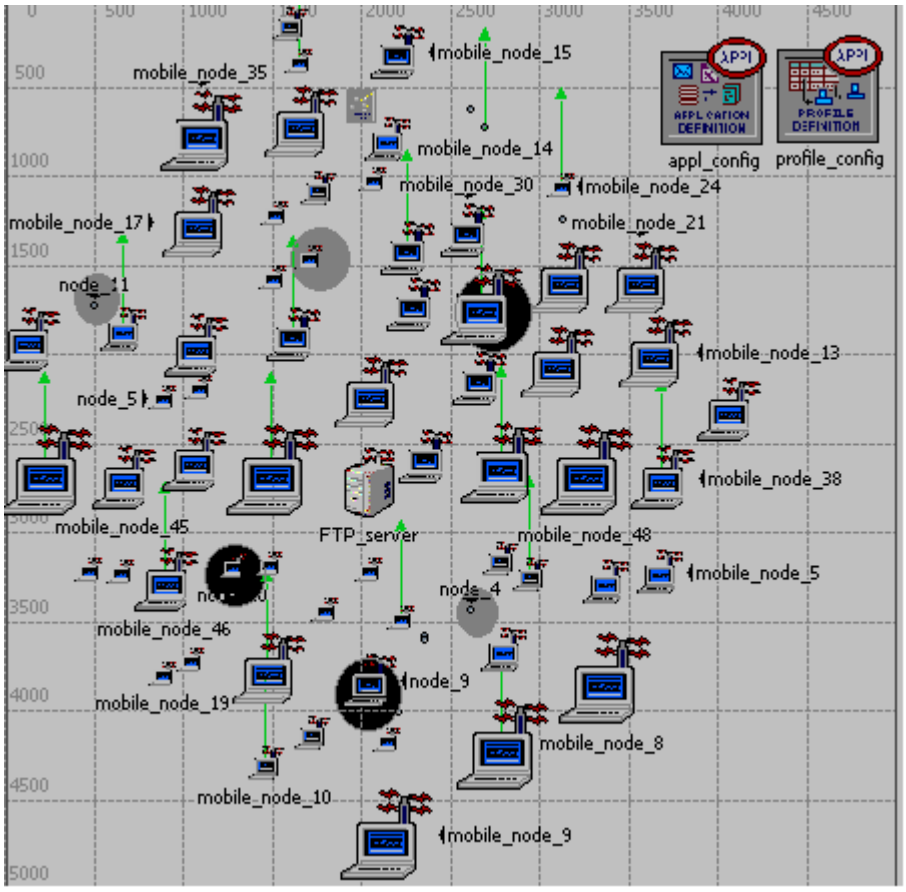


Figure 6.29 Experiment setup with 6 attacker nodes (3 packet drop + 3 packet delay+ 16 mobile nodes) (image from OPNET)

The result obtained after the experiment is shown in figure 6.30 and figure 6.31. The throughput of both scenarios is compared in the graph shown in figure 6.31. The figure 6.30 shows comparison of the route discovery time of both the scenarios. In the graph you can observe that the route discovery time of both the scenario is initially same. This is because the packet drop attacker nodes behave normally initially and then start dropping packets. Also, the initial movement of the mobile node is not affecting the network performance much, because they may take some time to move out from the region of its neighbour. After this initial period, the route discovery time is increased due to attacker nodes and mobile nodes. From the graph 6.31, we can conclude that, in the presence of packet drop attacker nodes, packet delay attacker nodes and mobile nodes the throughput of the network is decreased.

If we compare the route discovery time graph obtain with 3 packet drop and 3 packet delay attacker nodes(6.18) with the route discovery time graph obtained with this scenario(6.30), we can clearly see that in figure 6.18, route discovery time starts from a large value and gradually decreases and got steady on some value. This is because initially due to the

packet delay attacker nodes, who introduce the delay before forwarding each packet the route discovery time is large. After searching route, the same route will be used to send data packet. The packet drop attacker nodes behave normally in initial time, then they start dropping packets. When a route break due to packets drops by packet drop attacker nodes, the node from where route breaks tries to repair the route. In that scenario, as the nodes of the network are fixed the route will not be repaired or it will be repaired once the packet drop attacker node starts to behave normally. The route repair will take less time compare to rediscovery of the new route. This is because, the routing algorithm searches, route from the node where the link breaks. Hence, the route discovery time is reduced. The other reason of the decreases in the route discovery time is the absence of mobile nodes in the network. The movement of mobile nodes may disturb and hence delay, the route discovery process, when after receiving routing control packet a node moves out of the range of the sender of control packet.

In this network scenario, we have total 16 mobile nodes which are moving randomly in the network at random speed. Initially the mobile nodes may take place of any or all packet delay attacker nodes. Due to this the route discovery time may very less. After sending some data packet, the packet drop attacker node or the mobility of a node may break the route. The intermediate node tries to repair the route. However, here we have mobile nodes in the network, which may not allow doing repairs. Because once node left the location, there are less chances of its come back to the same location within a small time span. Thus, source node has to rediscover the route, which takes more time. Hence, in this network due to mobile nodes the route discovery time starts increasing.

If we compare the average throughput and the average route discovery time obtained with this experiment with the experiment which contains only packet drop and delay attacker nodes, the average throughput is reduced more (22755 bps more). The average route discovery time is also increased more (0.06 s more) compare to the experiment which contains only packet drop and delay attacker nodes. The reason for this performance degradation is the presence of 16 mobile nodes in the network. Due to the movement of these mobile nodes the routes can be frequently broken. Due to this route break, routing protocol has to search for the route again, which adds extra overhead and hence throughput is decreased. When a source node broadcast an RREQ packet for searching a route to the destination node, the intermediate nodes receive it and further broadcast it to their neighbours until the RREQ reaches to the destination node. After receiving the first RREQ

packet, the destination node will create an RREP packet and send it to the source node on the same path from where the RREQ comes. If one or more intermediate node is mobile node and after forwarding the RREQ packet if it moves and changes its location then the RREP packet may lose. This disturbs the whole routing process and hence, the presence of mobile nodes increase route discovery time of the network.

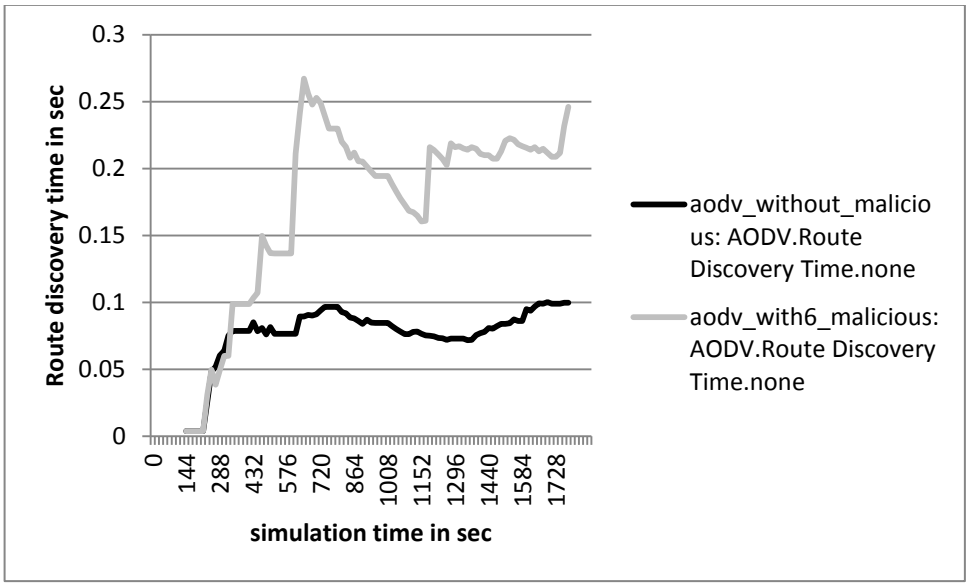


Figure 6.30 Comparison of Route Discovery Time (3 packet drop + 3 packet delay+ 16 mobile nodes)

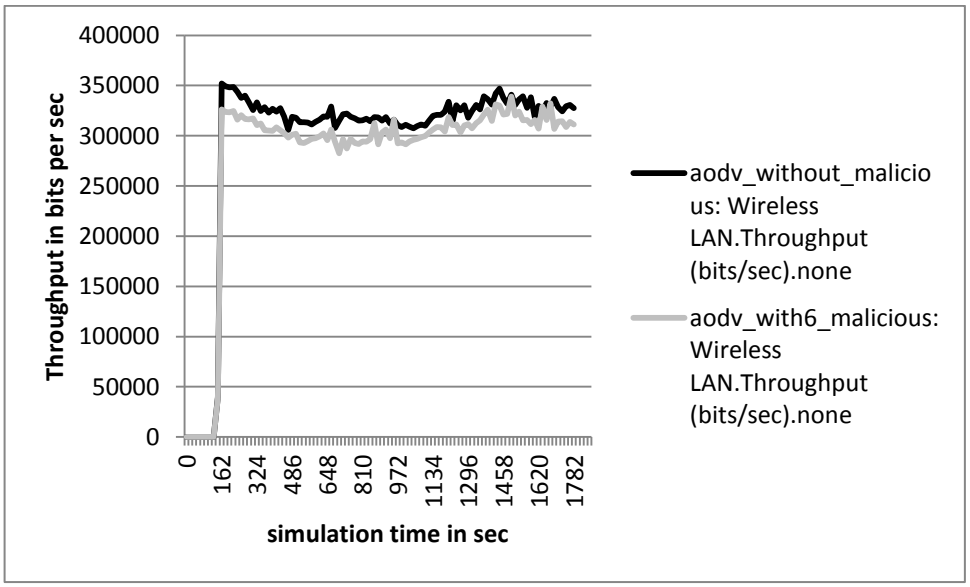


Figure 6.31 Comparison of Throughput (3 packet drop + 3 packet delay+ 16 mobile nodes)

6.7.2 Simulation environment with 6 packet drop attacker nodes, 6 packet delay attacker nodes and 16 mobile nodes

In this experimental setup, we have added 16 mobile nodes in the scenario that we have created in section 6.5.2 shown in figure 6.20. We have created this scenario with 12 attacker nodes (6 packet drop attacker nodes and 6 packet delay attacker nodes) and 16 mobile nodes. In figure 6.32 encircled nodes are attacker nodes and green arrow marked nodes are mobile nodes. Out of 12 attacker nodes, 6 nodes perform the packet drop attack (black circled) and 6 nodes perform the packet delay attack (grey circled). The traffic and all other parameters are same as previous scenarios.

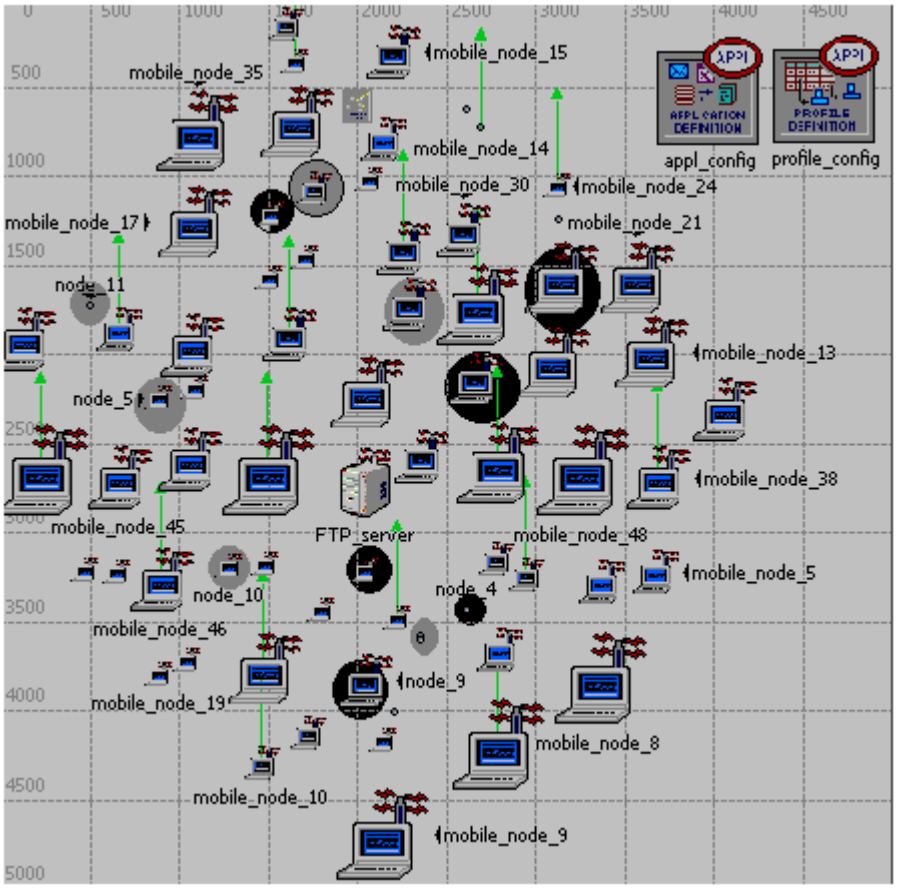


Figure 6.32 Experiment setup with 12 attacker nodes (6 packet drop+ 6 packet delay+ 16 mobile nodes) (image from OPNET)

The comparison of throughput and route discovery time in the absence of attacker node and with 6 packet drop attacker nodes, 6 packet delay attacker nodes and 16 mobile nodes are shown in figure 6.33 and figure 6.34. From the graph of throughput, shown in figure 6.34, we can conclude that the throughput is reduced in the presence of attacker nodes and

mobility. If we compare the average of this throughput with the average throughput obtained with only drop and delay attacker nodes discussed in section 6.5.2, we can say that by adding mobility throughput of the network is decreased more (28249bps more). Also the average route discovery time obtained with this experiment is increased more (0.09s more) compare to only drop and delay attacker nodes discussed in section 6.5.2. The reason for the reduction in throughput is the presence of mobile nodes in the network. The mobile node enters and exits the range of various nodes of the network due to its movement. If these mobile nodes are the part of the active route, they may break the route and hence, routing process has to discover the new route. The route discovery process may affect due to this mobile node. The mobile nodes who receive the routing control packet may not respond due to their movement and hence fail the routing process. This leads to increase of the route discovery time with the network, which has mobile nodes.

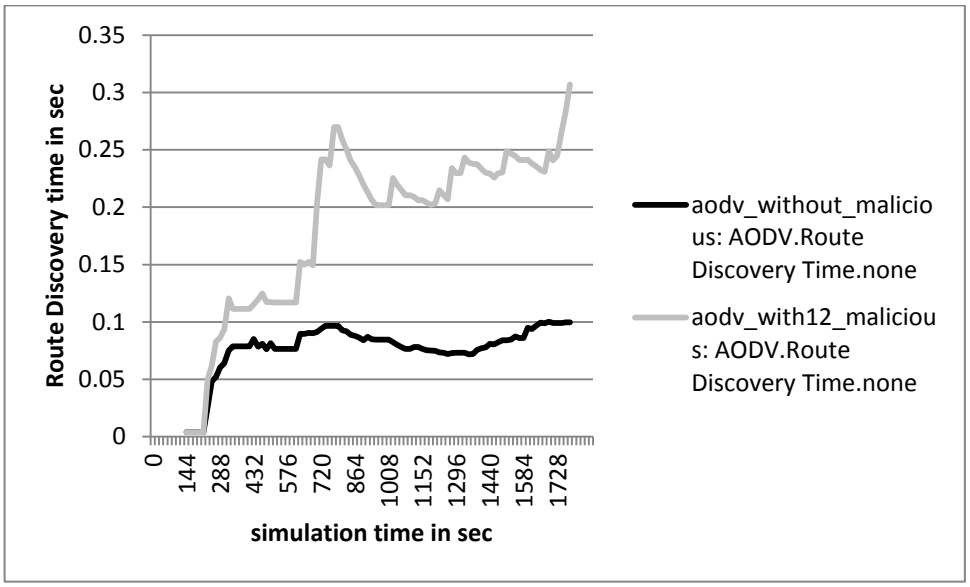


Figure 6.33 Comparison of Route Discovery Time (6 packet drop+ 6 packet delay+ 16 mobile nodes

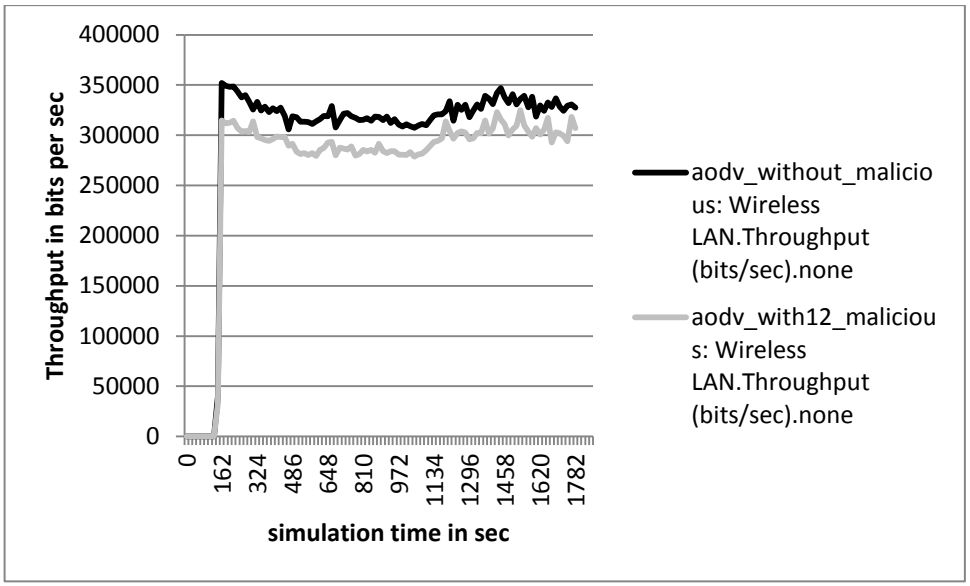


Figure 6.34 Comparison of Throughput (6 packet drop+ 6 packet delay+ 16 mobile nodes)

6.7.3 Simulation environment with 12 packet drop attacker nodes, 12 packet delay attacker nodes and 16 mobile nodes

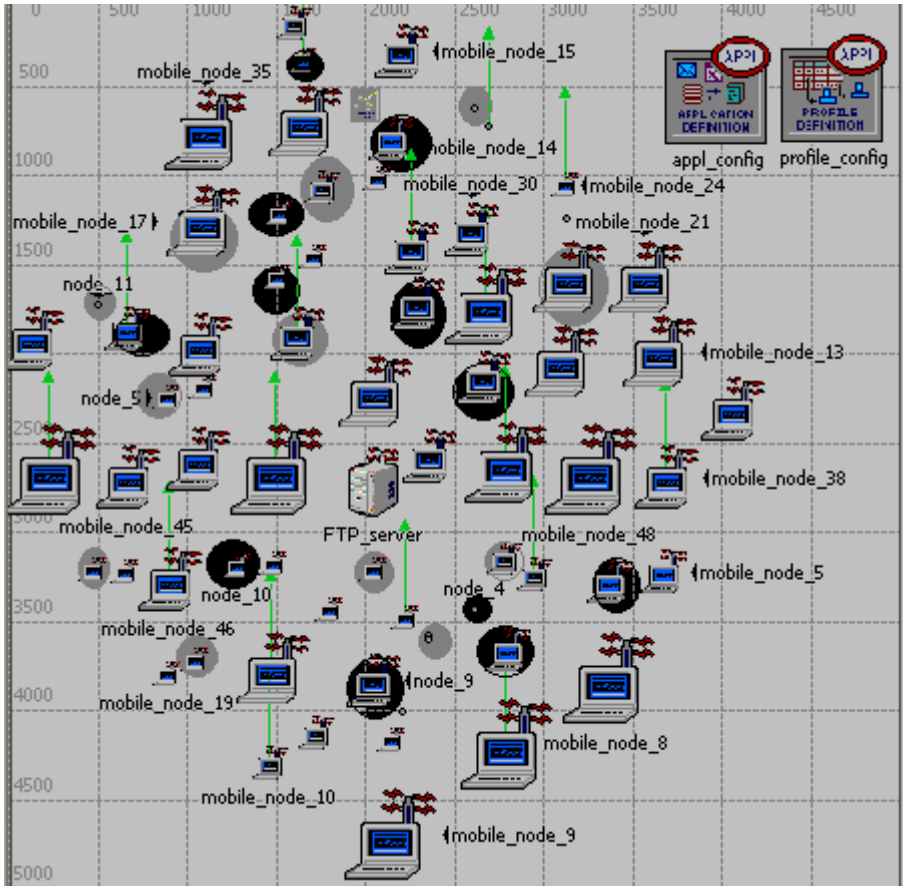


Figure 6.35 Experiment setup with 24 attacker nodes (12 packet drop+12 packet delay + 16 mobile nodes) (image from OPNET)

We have created a new MANET network scenario in which we have used total 24 attacker nodes and 16 mobile nodes out of 69 nodes. Out of these 24 attacker nodes 12 nodes (black circled) implement a packet drop attack and the other 12 nodes (grey circled) perform packet delay attack(figure 6.35). The traffic and all other parameters are same as previous scenarios. The comparison of throughput and route discovery time in the absence of attacker node and with 12 delay attacker nodes, 12 drop attacker nodes and 16 mobile nodes are shown in figure 6.37 and figure 6.36. The both the graphs shows that by adding more attacker nodes the throughput of the network is decreased and the route discovery time is increased. This is due to the presence of attacker nodes and mobile nodes. The movement of mobile nodes and the data packet drop by attacker node break the route, if they are the part of any active route. The routing process has to search for new route which adds a time delay as well as computational overhead and due to this, the throughput of the network is reduced. The attacker nodes and mobile nodes also mistreat the routing control packets which disturb the route discovery process and hence increase the route discovery time.

If we compare the average throughput and the average route discovery time obtained in experiment setup 6.5.3, we can clearly say that by adding mobile node the throughput should be decreased and the route discovery time should be increased. However, actually by adding the 16 mobile nodes, the average throughput is decreased more (18066 bps more) and the average route discovery time is decreased. In this experiment we got less (0.2s less) route discovery time compared to route discovery time obtained in experiment setup 6.5.3(Same network without mobile node). This is because in this experiment, we have added 24 attacker nodes and 16 mobile nodes. These 16 mobile nodes are not malicious. While running the simulation at some point, the mobile nodes start moving randomly. They may replace attacker nodes and form some new routes. Thus, by avoiding attacker nodes and using mobile nodes the route discovery time is decreased more compared to the scenario which has only 24 attacker nodes.

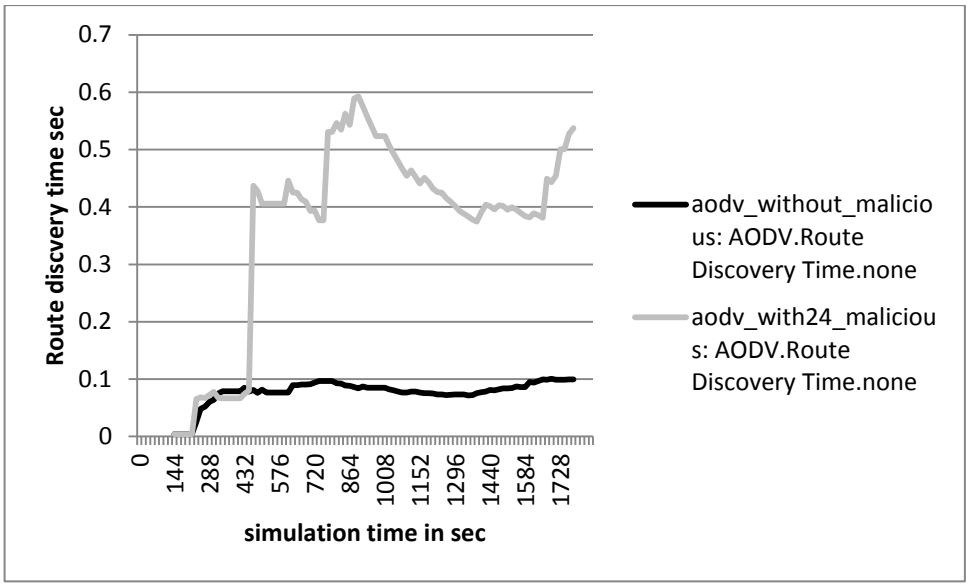


Figure 6.36 Comparison of Route Discovery Time (12 packet drop+12 packet delay nodes+ 16 mobile nodes)

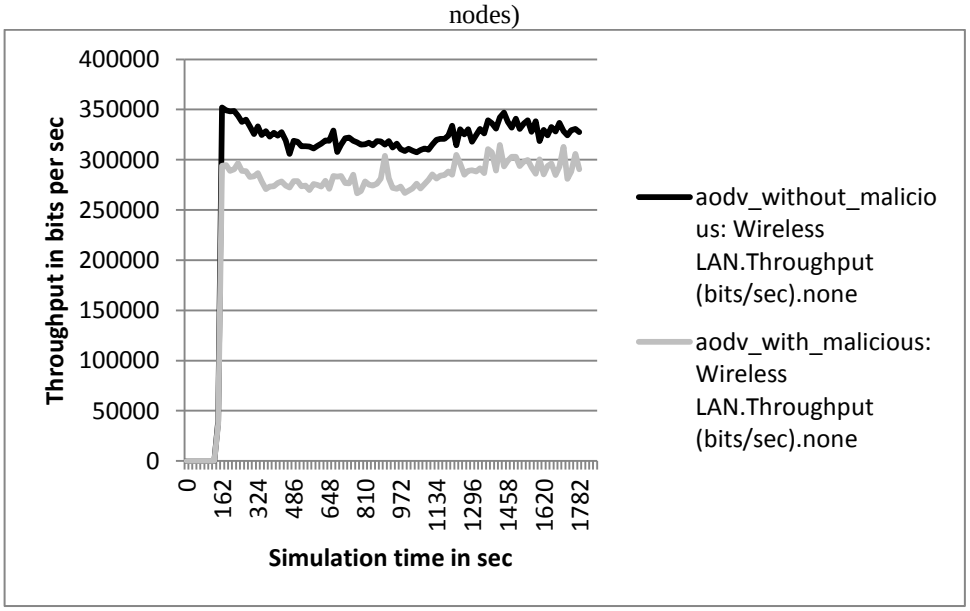


Figure 6.37 Comparison of Throughput (12 packet drop+12 packet delay nodes+ 16 mobile nodes)

6.7.4 Concluding Remarks

In this experiment, we have created total four scenarios. In all scenarios we have used AODV as a routing protocol on each node and 16 mobile nodes. One scenario has all non malicious nodes while other three scenarios are having 3, 6 and 12 packet drop attacker nodes and 3, 6 and 12 packet delay attacker nodes as discussed earlier. Thus, in total they have 6, 12 and 24 attacker nodes. We have compared the average route discovery time and the average throughput of all these four scenarios in table 6.4. The first entry in table 6.4 shows the average throughput and the average route discovery time in the absence of any

attacker node and the presence of 16 mobile nodes. Other three entries show that if we increase the attacker nodes in the network which has mobile nodes, the throughput of the network is reduced and the route discovery time is increased. This can also be shown in the graphs in figure 6.38 and 6.39. The reason for this performance degradation is due to the presence of attacker node as well as mobile nodes in the network. If with mobility, we are increasing the attacker nodes, they may gradually degrade the performance of the network.

Table 6.4 Effect of packet drop and delay attack on AODV routing and MANET in presence of mobile nodes

Sr No		AODV	
	Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)
1	0	321504.2126	0.094526
2	6	305048.2126	0.174754203
3	12	293384.5024	0.185386923
4	24	282207.2271	0.368930013

Also, if we compare tables 6.2 (results of packet drop and packet delay attacker nodes) with table 6.4, we can see that reduction in throughput with a packet drop attack and packet delay attack in the presence of mobility is more compared to packet drop and delay attack with all fixed nodes. This is because the mobility of a node is as equally destructive as a packet drop attack. The route discovery time with the packet drop attacker and packet delay attacker nodes with mobile nodes is also more compare to drop and delay attacker nodes. This is due to the movement of mobile nodes after receiving routing control packets. They may not respond as per the routing standard and fail routing process.

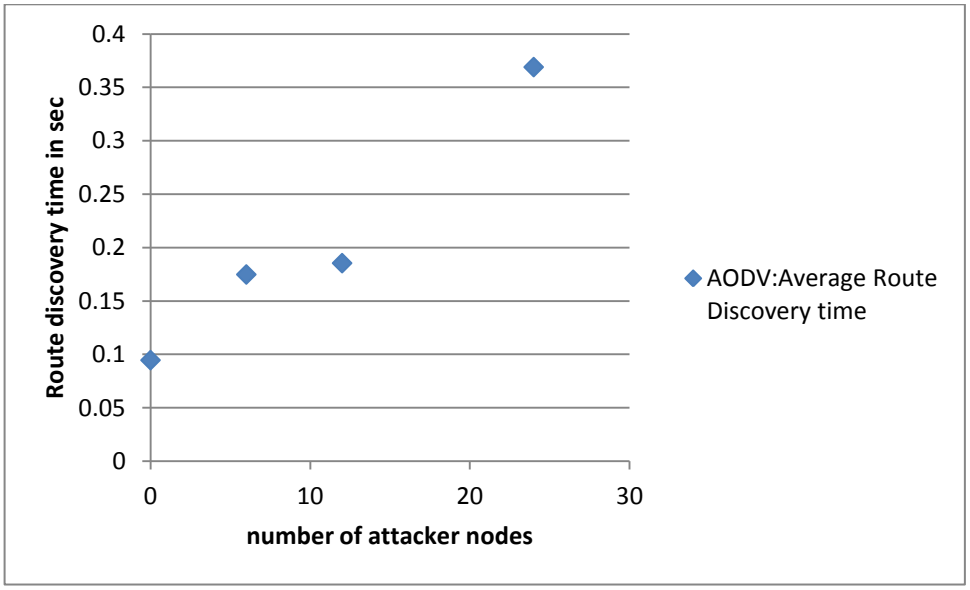


Figure 6.38 Average Route Discovery Time Vs number of attackers graph (drop+delay attack+mobility)

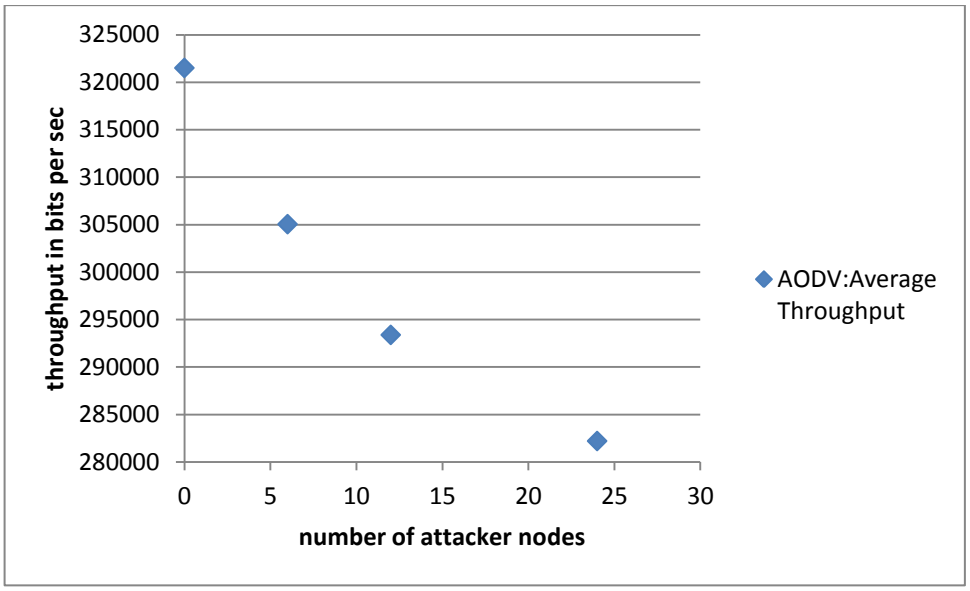


Figure 6.39 Average Throughput Vs number of attackers graph (drop+delay attack+mobility)

6.8 Summary

In this chapter we have implemented two malicious node models in OPNET. The first node model is a packet drop attacker node model, which drops the incoming packets to save its resources and not forward them further. This node model behaves normally for some time period and then starts dropping packets. After some time it again behaves normally. The second node model which we have implemented is continuously delaying all incoming packets before forwarding them further in the MANET. This node model implements a packet delay attack and thus delays all activities of the network. After implementing the packet drop and the packet delay attacker node models, we have created the MANET scenarios to check the effect of these attacker nodes on the route discovery time of the AODV routing and the throughput of the MANET. We have created three experimental setups.

In the first experiment, we have studied the effect of packet drop attacker node on the route discovery time and the throughput of the network. For this we have created a network with no attacker nodes and the networks with 9%, 18% and 27% packet drop attacker nodes. When compare the throughput of these network scenarios, we found that the throughput is reduced by 41516.68 bps (with 9% attacker nodes), 45511.03bps(with 18% attacker nodes) and 85399.7bps (with 27% attacker nodes)compared to the throughput of a network with no attacker nodes. The route discovery time is increased by 0.0348 s (with 9% attacker

nodes), 0.2242s (with 9% attacker nodes) and 2.592 s (with 9% attacker nodes) when we compare them with route discovery time of network with no attacker nodes. The reason behind the reduction in the throughput of networks with packet drop attacker nodes is the data and control packet loss by packet dropping attacker nodes. When the attacker node drops the data packets, it leads to break in existing route, which must be repaired or a new route must be searched. These both activities involve some overhead, which affects the performance of the network and hence, reduces the throughput. The route discovery time with the packet drop attacker nodes is increased due to the loss of control packets of AODV routing. When an attacker node drops the incoming control packets (RREQ, RREP, RERR), it may fail or delay the whole route discovery process and hence the route discovery time may increase.

In the second experiment, we have studied the effect of packet drop and packet delay attacker nodes on the route discovery time and the throughput of the network. For this we have created a network with no attacker nodes and the networks with 9%, 18% and 27% attacker nodes. We have added half packet drop attacker nodes and half packet delay attacker nodes in each network scenario which have the attacker nodes. When compare the average throughput of these network scenarios, we found that the average throughput is reduced by 22416.98 bps (with 9% attacker nodes), 28586.42bps(with 18% attacker nodes) and 49947.12 bps (with 27% attacker nodes)compare to the average throughput of the network with no attacker nodes. The average route discovery time is increased by 0.03939 s (with 9% attacker nodes), 0.02050s (with 9% attacker nodes) and 0.4530 s (with 9% attacker nodes) when we compare them with the average route discovery time of network with no attacker nodes. If we compare the reduction in throughput and increase in route discovery time results of this experiment with the first experiment, we can see that the reduction in throughput and increase in route discovery time results is less. This is due to the fact that we have added half packet drop attacker nodes in each scenario compare to first experiment and the other half attacker nodes are packet delay attacker nodes. The packet delay attacker node adds delay before forwarding the packet. They are not dropping any packet. Thus, they are less dangerous than packet drop attack; hence we got better results compare to the first experiment.

In the third experiment, we have studied the effect of packet drop and packet delay attacker nodes on the route discovery time and the throughput of the network in the presence of 16 mobile nodes. For this we have created a network with no attacker nodes and 16 mobile

nodes and the networks with 9%, 18% and 27% attacker nodes and 16 mobile nodes in each. We have added half packet drop attacker nodes and half packet delay attacker nodes in each network scenario which have the attacker nodes. When we are comparing the average throughput of these network scenarios, we found that the average throughput is reduced by 45172.187 bps (with 9% attacker nodes), 56835.897 bps(with 18% attacker nodes) and 68013.172 bps (with 27% attacker nodes)compare to the average throughput of a network with no attacker nodes. The average route discovery time is increased by 0.1002 s (with 9% attacker nodes), 0.1108 s (with 18% attacker nodes) and 0.2944 s (with 27% attacker nodes) when we compared them with the average route discovery time of network with no attacker nodes. The experimental setup is same as the second experiment where we added half packet drop attacker nodes and half packet delay attacker nodes. The only difference is that in this experiment we have added 16 (24%) mobile nodes which are uniformly distributed among the entire network nodes. If we compare the throughput and the route discovery time results obtained with this experiment with second experimental results, we can clearly observe that the reduction in throughput is very large. This is because we have attacker nodes as well as mobile nodes in the network. The movement of mobile node also breaks the route if it is a part of the route and leaves the range of its next hop neighbour. They may also disturb the route discovery process, if they leave the range of a node who is sending routing control packets after receiving them. Due to this reason the route discovery time is also increased.

We have also created an experiment to study the effect of only mobile nodes on route discovery time and throughput of the network. For this we have created two network scenarios. One with all non attacker nodes, but 16 mobile nodes and the other with all non attacker fixed nodes. We found 28716.18 bps reductions in the average throughput and 0.02 s increases in the average route discovery time of the network having mobile nodes compared to the network which has no mobile node. This is due to movement of mobile nodes. If a node which is a part of route move out of the range of any next hop neighbour, it breaks the route and hence throughput is reduced. The movement of mobile nodes also affects the route discovery operation by not responding, after receiving any control packet. Thus, they increase the route discovery time of AODV routing protocol.

The experiments, which we have implemented in this chapter are used to get the base results in the presence of packet drop attacker nodes, packet delay attacker nodes and mobile nodes with an AODV protocol. After implementation of our proposed routing

protocol (Trust based Mobility Aware-AODV: TMA-AODV), these results are used to compare with the results of the TMA AODV experiments. In the next chapter, we have discussed the implementation of the TMA-AODV routing protocol and shown that how these AODV routing results are compared with the results obtained with our protocol implementation - TMA-AODV.

CHAPTER 7

Implementation Of Proposed Routing Protocol And Its Analysis

7.1 Implementation of Proposed trust based routing protocol (TMA-AODV) for MANET

In the previous chapter from section 6.3 to section 6.7, we have implemented packet drop and packet delay attacker node in OPNET and shown that the introduction of malicious node degrades the performance of AODV (by increasing Route discovery time) routing as well as Wireless LAN(by decreasing throughput). In this section, we have implemented trust based mobility aware AODV (TMA-AODV) routing for MANET and show the improvement in the route discovery time and throughput. Our implemented routing protocol is an extension of the existing AODV protocol available in OPNET.

For packet monitoring, we have modified wireless node model and implemented packet monitoring code in "static void ip_dispatch_tunnel_packet_process( )" function of ip_dispatch process model. This function monitors the packet received from each destination and records observed parameters in trust table.

For modifying the AODV routing protocol in OPNET, we have to modify the process model of AODV(aodv_rte). We have created a copy of aodv_rte as my_aodv_rte and did a modification to avoid a route with the malicious node and the mobile node during the routing process. The figure 7.1 shows the my_aodv_rte process model. For implementing proposed routing protocol, we have modified three header files in the <opnet_dir>/std/include directory: my_aodv.h, my_aodv_pkt_support.h and my_aodv_ptypes.h.

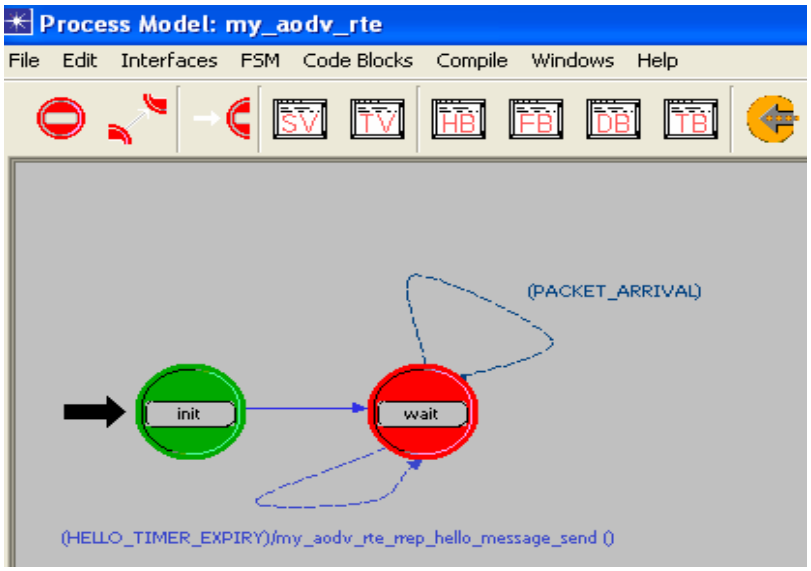


Figure 7.1 my_aodv_rte process model (from OPNET) [84][85]

The my_aodv.h contains the definition of all the constants, data structures for route, and connectivity tables. We have added two data structures and four constants in this header file as shown in figure 7.2. These data structures are used to create trust table and trust table entry in our proposed routing scheme. The constants are used when accessing trust table during routing. We have also modified the data structure in this header file which is used to create a route table entry. In this data structure we have added a field to store trust value of the route as shown in figure 7.2.

```
/* Constants to access entries in the trust table          */
#define AODVC_TRUST_ENTRY_IN_PKT 1
#define AODVC_TRUST_ENTRY_SFWD_PKT 2
#define AODVC_TRUST_ENTRY_DL_PKT 3
#define AODVC_TRUST_ENTRY_ERR_PKT 4

#define AODVC_ROUTE_TRUST_VALUE       10
/*********************************************/
/***************** ROUTE TABLE ***************/
/*********************************************/
.................
typedef struct
        {
        ...........
        ...........
        float     trust_val;
        } AodvT_Route_Entry;
.................
.................
/*********************************************/
/***************** TRUST TABLE ***************/
/*********************************************/
/* The trust table is a hash table  that stores */
/* observed value for each neighbour          */
/* Each entry in the trust table is indexed by the  node IP address.        */
typedef struct      {
        PrgT_String_Hash_Table*                 trust_table;
        IpT_Cmn_Rte_Table*                      ip_cmn_rte_table_ptr;
        IpT_Rte_Proc_Id                         aodv_protocol_id;
        AodvT_Local_Stathandles*                stat_handles_ptr;
        int                                     current_size;
        } AodvT_Trust_Table;

typedef struct      {
        IpT_Dest_Prefix                 node_prefix;
        InetT_Address                   node_addr;      // IP address of neighbour node
        int                             pin;            //number of incoming packets at neighbour
                                                        //node
        int             psfwt;          //number of packet successfully forwarded by neighbour node
        int             pdl;            //number of packet delayed at neighbour node
        int             perr;           //number of link break due to neighbour node
        } AodvT_Trust_Entry;
```

Figure 7.2 Route table related update and trust table related data structure in my_aodv.h file

The my_aodv_pkt_support.h header file contains all the definition of data structure used to create or access route request, route reply and route error packet. In this header file we have modified the data structure created for the route reply packet and route error packet. We have added a field which is used to store the trust value in the route reply and the field to store ip address of the node which is responsible for the route break in the route error (figure 7.3).

```
/* Route Reply Option       */
typedef struct
        {
        ..............
        ..............
        float                trust_val;
        } AodvT_Rrep;
/* Route Error Option */
typedef struct
        {
        ..............
        ..............
        InetT_Address       not_avail_node;
        } AodvT_Rerr;
```

Figure 7.3 Route reply and Route error related update in my_aodv_pkt_support.h file

```
AodvT_Trust_Table*                          my_aodv_trust_table_create (IpT_Cmn_Rte_Table* ,
IpT_Rte_Proc_Id ,AodvT_Local_Stathandles*);

void                                        my_aodv_trust_table_entry_create(AodvT_Trust_Table*,
InetT_Address, InetT_Subnet_Mask, int, int, int, int);

AodvT_Trust_Entry*                          my_aodv_trust_table_entry_get (AodvT_Trust_Table*,
InetT_Address);
Compcode                                    my_aodv_trust_table_entry_param_set
(AodvT_Trust_Table*, InetT_Address, int , ...);
```

Figure 7.4 Trust table related functions in my_aodv_ptypes.h file

The my_aodv_ptypes.h header file contains all function prototypes which are being used by the routing process to access the routing table. In this header file, we have added prototype of the functions required to access trust table while routing (figure 7.4) and called by external c files during routing.

Route establishment between source and destination is initiated by destination, by sending an RREP packet to the node which sends the RREQ packet. The receiver of RREP again forward the RREP packet to next predecessor until it reaches to the source node. In the new routing process model, the function "**static void my_aodv_rte_rrep_pkt_arrival_handle ( )**" process the incoming RREP packets at each node and forward the RREP packet to the next hop neighbour if the node is not a source node. We have added code to get observed parameters of next hop neighbour. After getting parameters we have called the trust calculation function to calculate trust value of next hop neighbour and added this calculated trust value in the trust value stored in RREP.

We have also modified the section of this function, where the receiver of RREP packet is the source node. After receiving each RREP, source node adds the route in its routing table

with trust value. The source node maintains a counter which is incremented by one after receiving each RREP. When this counter reaches to total number of neighbours of the source node, the source node will calculate average of trust value received in each RREP. The source node uses all the route with trust value more than average trust value for sending data packets. We consider the higher trust value as more trustworthy.

7.2 Comparison of AODV and TMA-AODV without any attacker nodes and with mobility

In this section we have compared AODV and our proposed trust based mobility aware AODV, which we have implemented in OPNET. First, we have compared AODV and TMA-AODV in the absence of any attacker node and the mobile node in the network. This comparison shows us the effect of extra code, we have added in TMA-AODV. We have studied throughput of network and route discovery time of routing protocol with AODV and TMA-AODV. Though we have added packet monitoring module on each node to record various activities of a neighbour, it adds some overhead to each node. We have also added logic to calculate the trust value before forwarding each RREP message during route establishment process and adding this trust into RREP. This also adds some overhead in routing. Due to theses both overheads, the route discovery time of routing process should be increased and throughput of the network should be decreased with TMA-AODV compared to the AODV routing protocol. We have created an experiment setup as discussed in section 7.2.1 to study the effect of TMA-AODV on throughput and route discovery time in MANET. We have also compared these results with AODV routing.

In the second case, we have compared AODV and TMA-AODV without any attacker node and in the presence of mobile nodes. In our proposed routing protocol (TMA-AODV), the protocol tries to consider less number of mobile nodes in the route during the routing process. This gives us more stable routes and more stable routes means less link breaks. This may increase throughput of the network and decrease the route discovery time compare to AODV routing. In section 7.2.2, we have set up an experiment which compares throughput and route discovery time with AODV and TMA_AODV routing protocol in MANET with 16 mobile nodes. In the ideal case, in the presence of mobile nodes, the throughput of the network should be increased and the route discovery time of routing should be reduced with TMA_AODV. In section 7.2.2, the result table justifies this.

7.2.1 AODV and TMA-AODV without any attacker nodes and with fixed nodes

After implementing TMA-AODV in OPNET, we have compared it with standard AODV routing without any malicious node and mobile node. For comparing them, we have used throughput and route discovery time parameters. For this result comparison, we have created two scenarios in OPNET as shown in figure 6.4. In one scenario, we have used AODV as a routing protocol and in second scenario; we have used TMA-AODV as a routing protocol. The traffic and other experiment setup are same as discussed in section 6.4.1. The following table 7.1 shows the result comparison. ↓ indicates decrease in value and ↑ indicate an increase in value.

Table 7.1 Comparison of AODV and TMA-AODV in absence of attack and mobility

Attacker node	AODV: Average Route Discovery time(sec)	TMA-AODV: Average Route Discovery time (sec)	AODV: Average Throughput (bits per sec)	TMA-AODV: Average Throughput (bits per sec)	Throughput	Route discovery time
0	0.074528738	0.07805756	350220.4	340092.9	3%↓	4.7% ↑

The result shows that the use of TMA-AODV without any attacker node and absence of mobility, reduces the average throughput by 3% and increase the average route discovery time by 4.7% compared to AODV. This is due to the overhead of modules, we have added in TMA-AODV to detect and avoid attacker nodes and mobile nodes.

7.2.2 AODV and TMA-AODV without any attacker nodes and with mobility

In our proposed trust based mobility aware routing protocol (TMA-AODV), we have used mobility of a node as a parameter to calculate trust value by counting link break due to each node. If a node is more mobile, we consider it less trustworthy because its link break count is high due to its movement. After implementing TMA-AODV in OPNET, we have created one more scenario which is same as figure 6.29 with TMA-AODV routing on each node. The traffic and other parameters are same as previous two scenarios. The result obtained for a third scenario (with TMA-AODV and 16 mobile nodes) is compared with AODV (with 16 mobile nodes) results as shown in table 7.2.

Table 7.2 Comparison of average throughput and average route discovery time with AODV and TMA-AODV routing

Mobile nodes	AODV: Average Route Discovery time(sec)	TMA-AODV: Average Route Discovery time (sec)	AODV: Average Throughput (bits per sec)	TMA-AODV: Average Throughput (bits per sec)	Throughput	Route discovery time
16	0.094526	0.075984588	321504.2126	33615.1159	4.67↑	19.61↓

From the table 7.2 we can conclude that with TMA-AODV routing the average route discovery time decreases compared to AODV routing in the presence of 16 mobile nodes. Ideally, the TMA-AODV gives high priority to fixed nodes over mobile node during route formation. Hence, there are less link breaks/route breaks due to mobility of nodes. However, in the actual network scenario, after searching all routes with trust value, the TMA-AODV algorithm choose the routes having no or minimum number of mobile nodes and use them. This increase the throughput of network and decrease the route discovery time compare to AODV routing. In table 7.2, the result shows that with TMA-AODV routing protocol, the average throughput of the network is increased and the average route discovery time is decreased. The increment of throughput(4.67%) and decrement of route discovery time(19.61%) with TMA-AODV is very less. This is because TMA-AODV is designed in such a way that it tries to avoid all possible mobile nodes in the route while route formation. However, sometimes the route may have one or more mobile node when protocol doesn't have any fixed node option to reach from a source node to destination. These mobile nodes may break the route due to the node movement and hence there is a delay in data packet transfer, which may affect the throughput with TMA-AODV. The other reason for less improvement in throughput is due to the overhead associated with TMA-AODV which includes trust related modules. In TMA-AODV, for searching route from source to destination, a source node has to send RREQ to all its neighbours, which is further forwarded to neighbours of receiver nodes until RREQ reaches to the destination node. The destination node will prepare and send the RREP packet with a trust value for each received RREQ. For sending RREP, the destination node uses the same path from where RREQ comes. If one or more nodes change their position after forwarding the RREQ packet, the RREP on that path may be lost and never reaches the source node. This is the reason for less decrement of route discovery time with TMA-AODV in the presence of only mobile nodes.

7.3 Effect of TMA-AODV in MANET in presence of packet drop attacker nodes

We have implemented our proposed routing protocol (TMA-AODV) in OPNET. In TMA-AODV, the trust of a node is calculated using various parameters observed for that node which is stored in trust table. The total number of packets observed for a node and total number of packets forwarded by a node is the parameters used to calculate the number of packet drop by the node (total number of packets observed – total number of packets forwarded). Thus, during TMA-AODV routing, the route which has packet drop attacker node as intermediate node has low trust value. The TMA-AODV routing find all possible routes between the same source and destination with trust value associated with each route and uses the most trust worthy routes (routes with high trust value) for sending data. Also, the TMA-AODV uses multiple routes simultaneously, thus improves the throughput and the route discovery time in the presence of packet drop attacker nodes. In this section we have compared the proposed TMA-AODV routing with the AODV routing in presence 9%, 18% and 27% of total node as packet drop attacker nodes.

7.3.1 TMA-AODV with 6 packet drop attacker nodes

We have created the experiment setup same as section 6.4.1. In experiment discussed in section 6.4.1, we have added one more scenario same as figure 6.5 which contains 6 packet drop attacker nodes and TMA-AODV as a routing protocol. The result for the throughput and the route discovery time obtained with TMA-AODV is compared with AODV without attacker nodes and AODV with packet drop attacker nodes. The figure 7.5 and figure 7.6 shows the comparison of the throughput and the route discovery time of AODV without malicious node, AODV with 6 packet drop attacker nodes and TMA-AODV with 6 packet drop attacker nodes.

The route discovery time with TMA-AODV routing in graph 7.5 is initially almost same as route discovery time with AODV with 6 attacker nodes. This is because initially TMA-AODV doesn’t have any observed values to calculate the trust value. However, with time trust values are calculated using observed values stored in trust table and hence, the

route discovery time is gradually (after 576 sec) decreased with TMA-AODV routing. Also from the graph, we can see that the route discovery time with TMA-AODV is decreased in the presence of packet drop attacker nodes. This is due to the trust based routing mechanism which we have implemented in TMA-AODV. The TMA-AODV finds all possible routes between source and destination and use the most trustworthy routes. The routes which have attacker nodes as an intermediate node must have a low trust value compare to a route which doesn't have any attacker node. Thus, with TMA-AODV, there are less chances of link break. Even if there is a link break, the TMA-AODV has the other route readily available which may use for sending data packets. The rediscovery of the route is initiated only when all the trusted routes are broken. Also the route discovery time of TMA-AODV in presence 6 packet drop attacker nodes is more than the route discovery time of AODV without any attacker node. This is because of the overhead involved in observation of neighbours and calculation of trust of a route with TMA-AODV.

The graph 7.6 also shows the improvement in the throughput with TMA-AODV compared to the AODV routing with 6 packet drop attacker nodes and reduction of the throughput with the AODV routing without any attacker nodes. The throughput with TMA-AODV is improved as it uses the trustworthy routes only. And the trustworthy route contains a minimum or no attacker nodes in the route. Thus, the communication in the network is less affected by the attacker nodes. The throughput of network with TMA-AODV and 6 packet drop attacker nodes is less compared to the throughput of the network with AODV routing and no attacker node. This reduction of throughput is due to the overhead involved in observation of neighbors and trust calculation by each node.

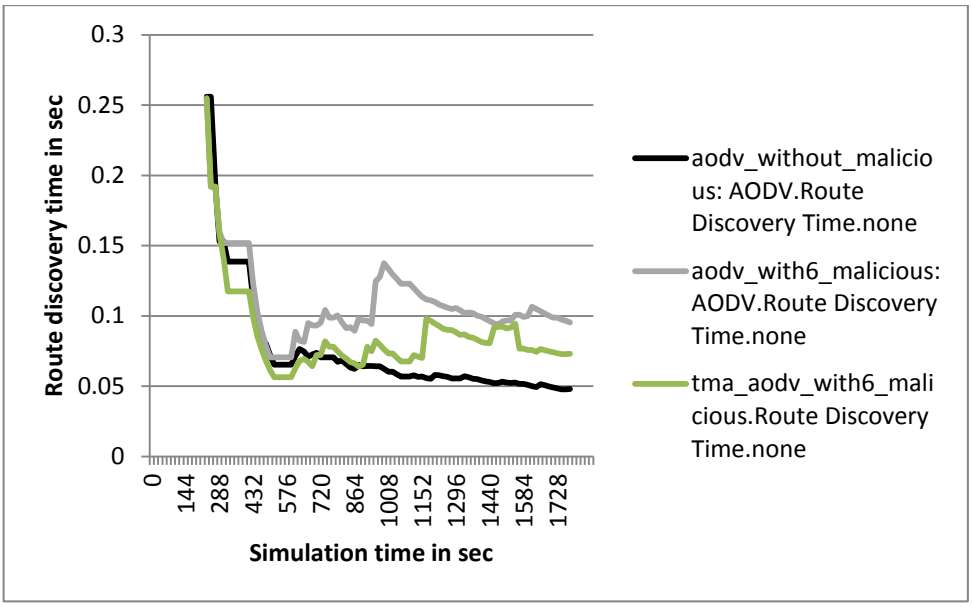


Figure 7.5 Comparison of Route Discovery Time with AODV and TMA-AODV (6 packet drop attacker)

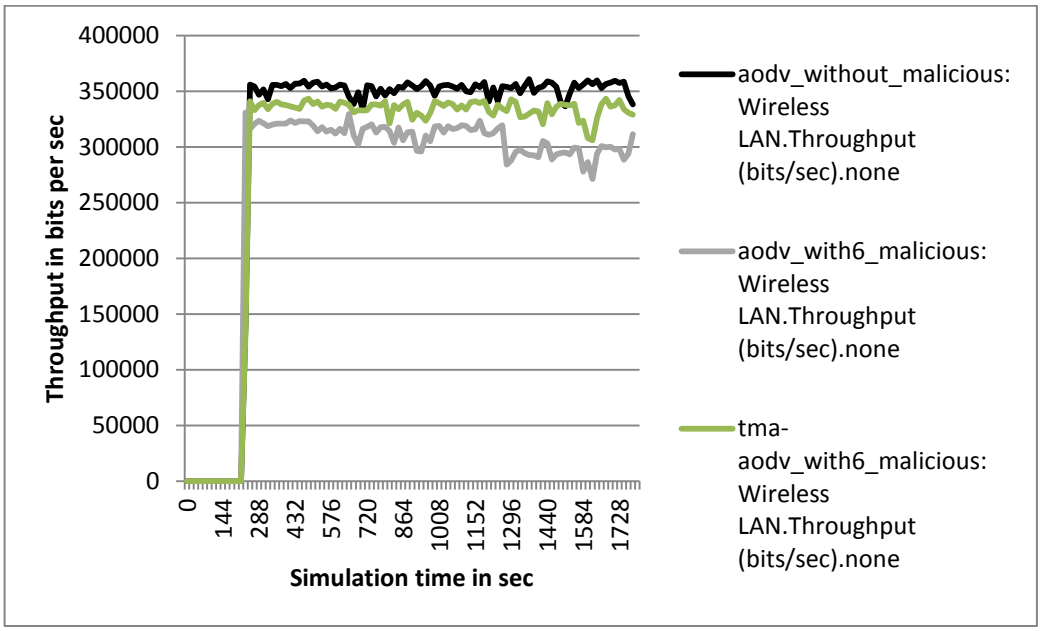


Figure 7.6 Comparison of Throughput with AODV and TMA-AODV (6 packet drop attacker)

Table 7.3 Improvement in Average throughput and Average route discovery time with TMA-AODV (6 packet drop attackers)

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)	Average Throughput (bps)	Average Route Discovery time(s)	Throughput	Route Discovery time
6	308703.7	0.109340915	332235.89	0.08583977	8%(↑)	21%(↓)

The table 7.3 is obtained from the graphs shown in figure 7.5 and figure 7.6. The values shown in table 7.3 are calculated as average of values of the graphs in figure 7.5 and 7.6. ↓

indicates decrease in value and ↑ indicate an increase in value. With the TMA-AODV routing, the throughput is increased by 8% and the route discovery time is decreased by 21% of the throughput and the route discovery time obtained with the AODV routing with 6 packet drop attacker nodes. The TMA-AODV outperforms compared to the AODV in the presence of 6 packet drop attacker nodes.

7.3.2 TMA-AODV with 12 packet drop attacker nodes

In experiment 6.4.2, we have added one more scenario same as figure 6.9 which contains 12 packet drop attacker nodes and TMA-AODV as a routing protocol. The result for the throughput and the route discovery time obtained with TMA-AODV is compared with AODV without attacker nodes and AODV with 12 packet drop attacker nodes as shown in figure 7.7 and figure 7.8. In graph 7.7, we can see that during the initial simulation time, the route discovery time with TMA-AODV is almost same as AODV with 12 attacker nodes. This is because the TMA-AODV routing needs some learning period to collect the observation about the neighbour nodes. The route discovery time with TMA-AODV start decreasing compared to AODV with 12 attacker nodes at 432 seconds as shown in graph 7.7. This is because now the TMA-AODV running on each node has enough collected observation to compute the trust and detect the attacker nodes. The throughput is also increased with TMA-AODV compare to AODV in the presence of 12 packet drop attacker nodes (figure 7.8). This is because with TMA-AODV, the packet drop attacker nodes are identified by their neighbours and they consider them less trustworthy and avoid the use of the routes having such less trustworthy nodes.

Also, the route discovery time of AODV without any attacker nodes is less than the route discovery time of TMA-AODV with 12 packet drop attacker nodes. The throughput with AODV without any attacker node is also more compared to TMA-AODV with 12 attacker nodes. This is due to the overhead involved with various modules added to implement trust modeling in AODV routing. When with TMA-AODV, we found all routes from a source node to the destination node having one or more attacker nodes, the routing protocol chose the route with less number of attacker node. In such case the communication may affect and route breaks due to attacker nodes with TMA-AODV also.

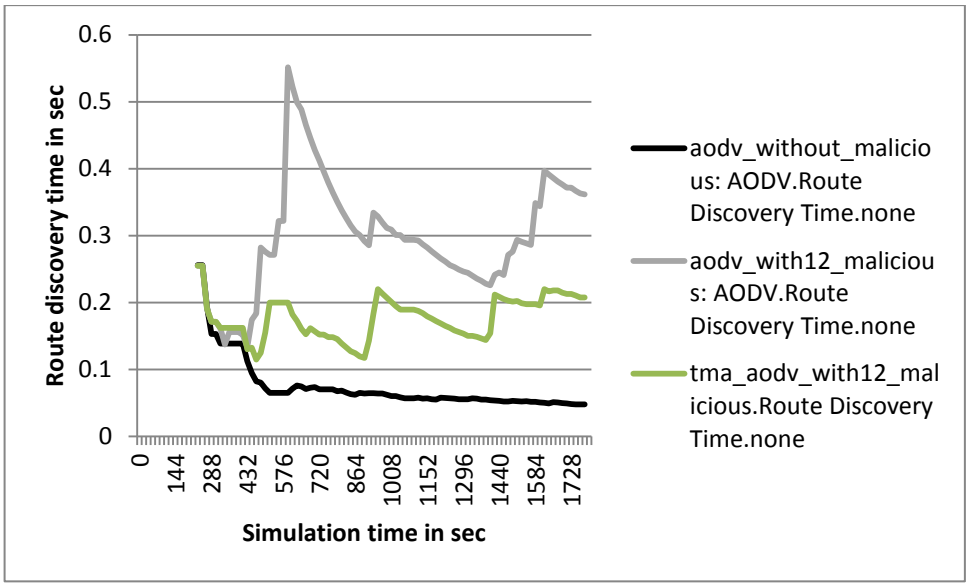


Figure 7.7 Comparison of Route Discovery Time with AODV and TMA-AODV (12 packet drop attacker)

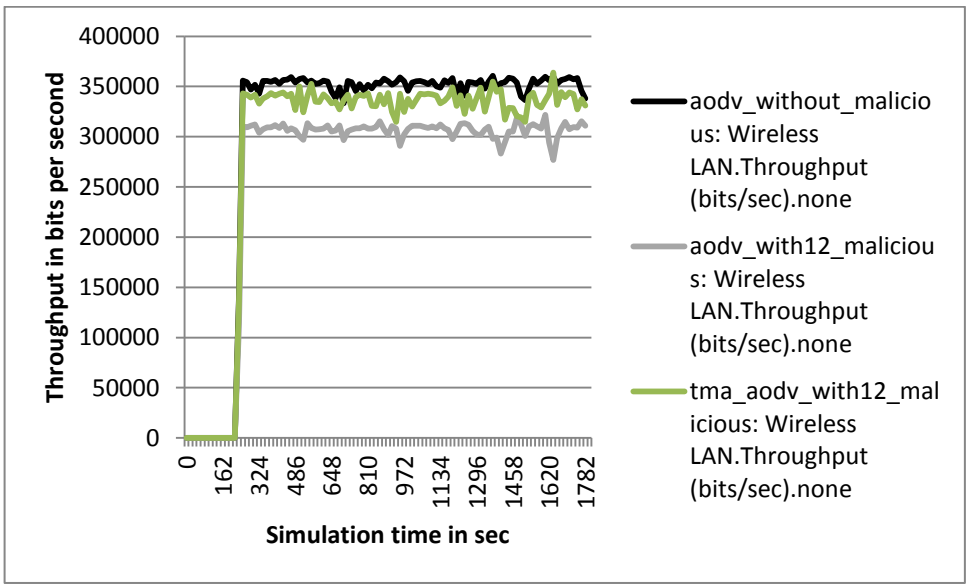


Figure 7.8 Comparison of Throughput with AODV and TMA-AODV (12 packet drop attacker)

Table 7.4 Improvement in average throughput and average route discovery time with TMA-AODV (12 packet drop attackers)

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)	Average Throughput (bps)	Average Route Discovery time(s)	Throughput	Route Discovery time
12	304709.4	0.298733205	334626.29	0.17707345	9%(↑)	40%(↓)

The table 7.4 is obtained by calculating the average of the values from the graphs shown in figure 7.7 and figure 7.8. ↓ indicates decrease in value and ↑ indicate an increase in value.

The table and graphs clearly show that the results are improved with TMA-AODV compared to AODV routing protocol in the presence of 12 packet drop attacker nodes out to 70 nodes of the MANET. With the TMA-AODV routing, the throughput is increased by 9% and the route discovery time is decreased by 40% of the throughput and the route discovery time obtained with the AODV routing with 12 packet drop attacker nodes. This result improvement is more compared to the improvement we got with the TMA-AODV routing and 6 packet drop attacker nodes. This is because with more packet drop attacker nodes more malicious activities are recorded in trust table of each node. Hence, the TMA-AODV routing can easily detect the malicious nodes and avoid them by calculating the trust value using these recorded observations.

7.3.3 TMA-AODV with 24 packet drop attacker nodes

In experiment 6.4.3, we have added one more scenario same as figure 6.12 which contains 24 packet drop attacker nodes and TMA-AODV as a routing protocol. The result for the throughput and the route discovery time obtained with TMA-AODV is compared with AODV without attacker nodes and AODV with 24 packet drop attacker nodes as shown in figure 7.9 and figure 7.10. In graph 7.9, you can see that the route discovery time with AODV routing and 24 packet drop attacker node (27% of total nodes) is very large compared to AODV without attacker node and TMA-AODV with 24 attacker nodes. This is because when 24 packet drop attacker nodes, which are uniformly distributed in network starts dropping packets, they completely disturb the routing operation of the network. With TMA-AODV the route discovery time is reduced (88%) compare to route discovery time with AODV routing with 24 packet drop attacker nodes. This is because the packet drop by 24 attacker nodes is very large and it is easily detected by the trust mechanism of TMA-AODV routing. The throughput with TMA-AODV is also improved large compared to previous two experiments (figure 6.10). The reason is more packet drop which can be easily detected by TMA-AODV and hence avoid such attacker node during route formation.

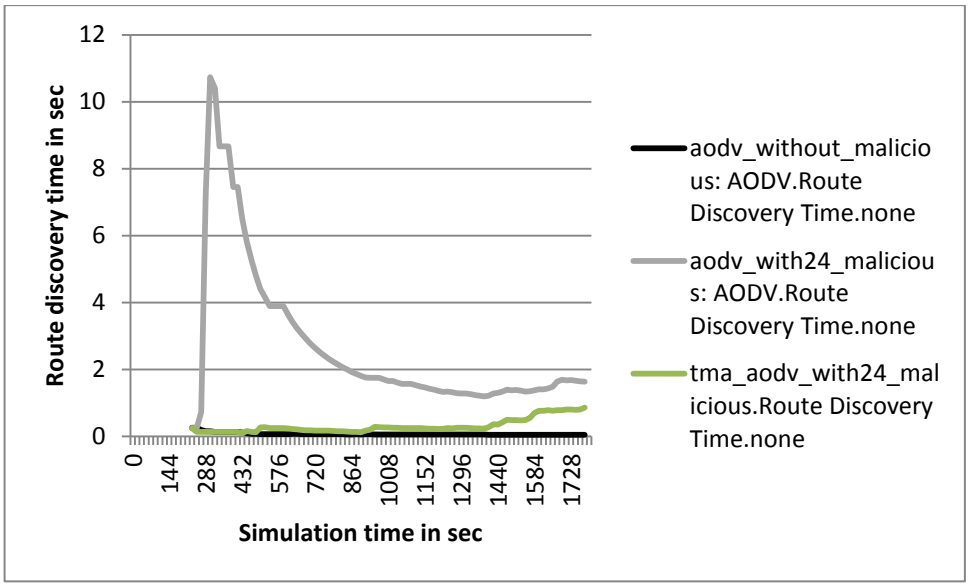


Figure 7.9 Comparison of Route Discovery Time with AODV and TMA-AODV (24 packet drop attacker)

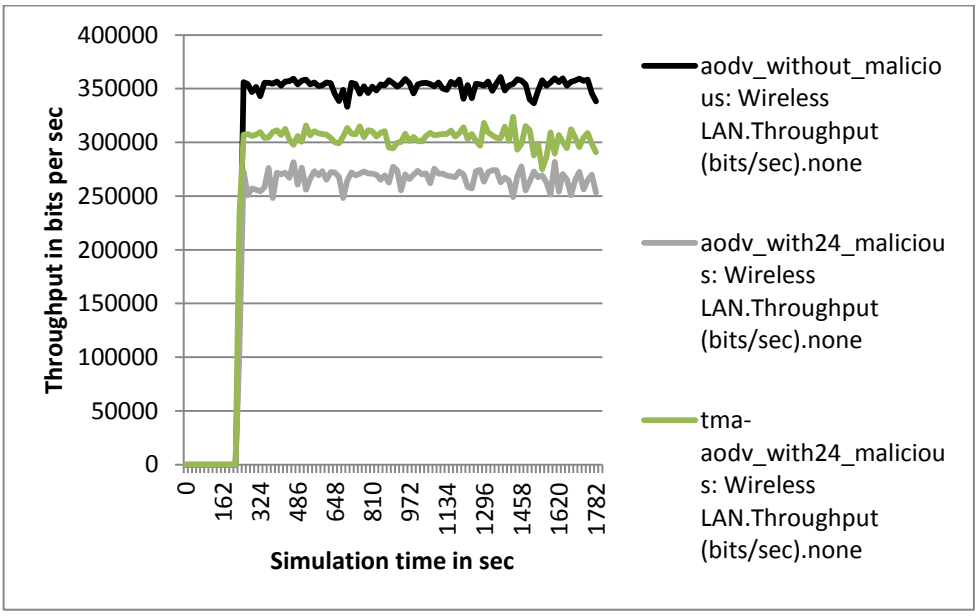


Figure 7.10 Comparison of Throughput with AODV and TMA-AODV (24 packet drop attacker)

Table 7.5 Improvement in average throughput and average route discovery time with TMA-AODV (24 packet drop attackers)

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)	Average Throughput (bps)	Average Route Discovery time(s)	Throughput	Route Discovery time
24	264820.7	2.666962079	304009.19	0.31171683	13%(↑)	88%(↓)

The table 7.5 is obtained from the graphs shown in figure 7.9 and figure 7.10. The values shown in table 7.5 are calculated as average of values of the graphs in figure 7.9 and 7.10. ↓ indicates decrease in value and ↑ indicate an increase in value. The table and graphs

clearly show that the results are improved with TMA-AODV compared to AODV routing protocol in the presence of 24 packet drop attacker nodes out to 70 nodes of the MANET. With the TMA-AODV routing, the throughput is increased by 13% and the route discovery time is decreased by 88% of the throughput and the route discovery time obtained with the AODV routing with 24 packet drop attacker nodes.

7.3.4 Concluding Remarks

The all results obtained after the experiments with packet drop attacker nodes with TMA-AODV routing is shown in table 7.6. For packet drop attack, we got following improvement with our proposed routing protocol. ↓ indicates decrease in value and ↑ indicate an increase in value.

Table 7.6 Average Result obtained when Packet drop

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)	Average Throughput (bps)	Average Route Discovery time(s)	Throughput	Route Discovery time
0	350220.4	0.074528738	340092.9	0.07805756	3%(↓)	4.7%(↑)
6	308703.7	0.109340915	332235.89	0.08583977	8%(↑)	21%(↓)
12	304709.4	0.298733205	334626.29	0.17707345	9%(↑)	40%(↓)
24	264820.7	2.666962079	304009.19	0.31171683	13%(↑)	88%(↓)

The first row of the table 7.6 shows the average throughput and the average route discovery time of the network with AODV routing and TMA-AODV routing in the absence of any attacker node and the mobile node in the network. With TMA-AODV routing the average throughput is reduced by 3% and the average route discovery time is increased by 4.7% of the average throughput and the average route discovery time obtained with AODV routing. This is because of the routing overhead with TMA-AODV, as traffic monitoring and trust modelling modules are added. Due to this overhead the results are degraded. The 2nd row onwards, the table shows the throughput and the route discovery time of the network with AODV routing and TMA-AODV routing in the presence of 6, 12 and 24 packet drop attacker node and the absence of a mobile node in the network. The table 7.6 shows that if we add the attacker node in the network, the throughput is reduced and the route discovery time is increased with AODV routing which are improved with the TMA-AODV routing.

The graphs shown in figure 7.11 and 7.12 shows the improvement in route discovery time and throughput results with our proposed routing protocol TMA-AODV compared to AODV in the presence of packet drop attacker nodes. As we increased attacker nodes the improvement in route discovery time and throughput is also increased.

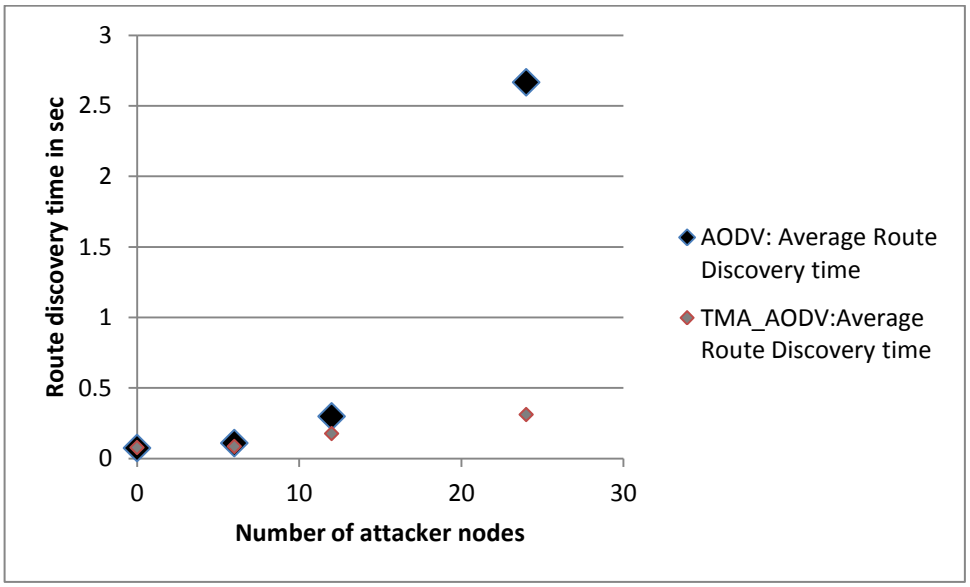


Figure 7.11 Comparison average route discovery time of AODV and TMA-AODV in presence of packet drop attack

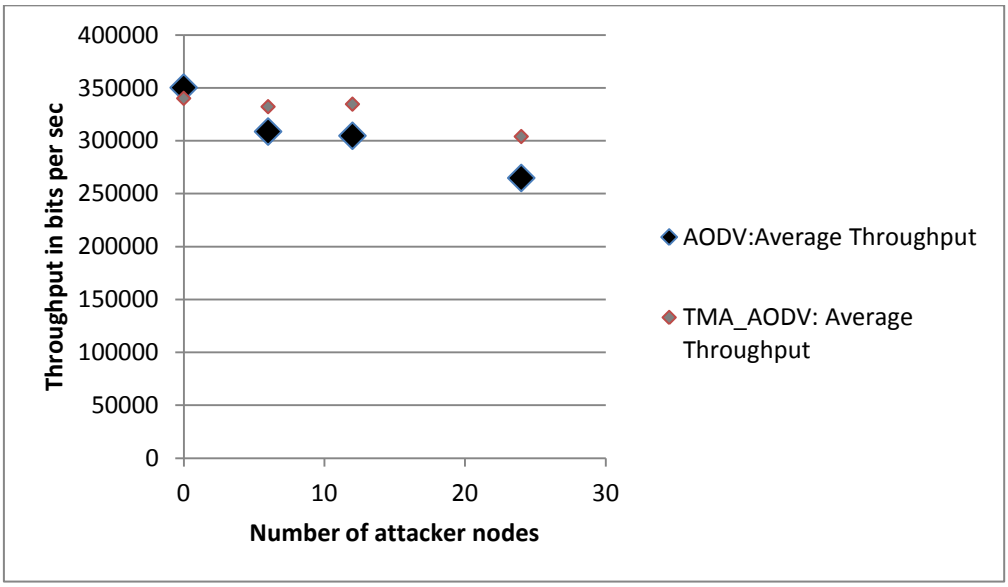


Figure 7.12 Comparison of average throughput with AODV and TMA-AODV routing in presence of packet drop attack

7.4 TMA-AODV in presence of packet drop and packet delay attacker nodes

In this section, we have tried to see the effect of our proposed routing protocol (TMA-AODV) on the route discovery time of routing and the throughput of the network in the presence of the packet drop and packet delay attacker nodes and without any mobile node. We have also compared these results, which we obtained with the AODV routing in section 6.5. The TMA-AODV calculates trust value of the node using various parameters observed for that node such as total packets observed for a node, total packets forwarded by a node, total packet delayed at a node before forwarding etc. which are stored in trust table. It uses this calculated trust value of nodes to calculate trust of each route found after the route discovery process and uses multiple trustworthy route simultaneously for sending data packets. The trust value calculated in TMA-AODV is affected by the packet dropping and the packet delaying behavior of the node. The node has less trust with such malicious behavior. Thus, TMA-AODV detects the route having such malicious nodes and avoid its usage for sending data. In this section we have compared the proposed TMA-AODV routing with the AODV routing in the presence of 9%, 18% and 27% of total node as attacker nodes. Out of total attacker nodes we have kept half packet drop attacker nodes and half packet delay attacker nodes in each scenario. As, TMA-AODV detects the malicious behavior of the nodes, the result obtained with TMA-AODV should be improved compared to results with AODV routing.

7.4.1 TMA-AODV with 3 packet drop and 3 packet delay attacker nodes

We have created the experiment setup same as section 6.5.1. In experiment 6.5.1, we have added one more scenario same as figure 6.18 which contains 6 attacker nodes (3 packer drop attackers and 3 packet delay attackers) and TMA-AODV as a routing protocol. The result for the throughput and the route discovery time obtained with TMA-AODV is compared with AODV without attacker nodes and AODV with packet drop and delay attacker nodes as shown in the figure 7.13 and figure 7.14.

In the graph, shown in figure 7.13, you can see that initially the route discovery time with TMA-AODV routing is almost same as route discovery time with AODV routing and attacker nodes. This is because the TMA-AODV doesn’t have enough observations to calculate trust value. It needs some learning period to record observation on nodes. After

that the route discovery time with TMA-AODV is reduced. For some time duration, it is even lesser than the route discovery time with AODV routing without any attacker node. This is because, in TMA-AODV, multiple routes are found and they used simultaneously to send data packet. The need for new route discovery arises only when, all the found route are not available. After that, the route discovery time with TMA-AODV is more than route discovery time with AODV without attacker nodes. This is because, the network with AODV without attacker nodes does not have any drop or delay disturbance. Though TMA-AODV finds multiple route and uses them simultaneously, they also has overhead involve with trust information gathering and trust calculation. Hence, at some point the result of AODV without attacker nodes are better than the results of TMA-AODV with attacker nodes. However the results with TMA-AODV should be better than the results with AODV in the presence of attacker nodes. This is what we have achieved in the graph 7.13.

The graph 7.14 shows the improvement in throughput with TMA-AODV in the presence of attacker nodes (delay +drop) compared to AODV routing. The throughput with TMA-AODV in the presence of 3 drop and 3 delay attacker nodes is also increased compared to the throughput obtained with AODV in the presence of same attacker nodes. This is due to TMA-AODV, which uses all the route which has no or less attacker node as an intermediate node. If a route has less attacker nodes, there are less chances of link break or delay or data loss which improves the throughput.

Table 7.7 Improvement in Average throughput and Average route discovery time with TMA-AODV (3 packet drop +3 packet delay attacker)

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)	Average Throughput (bps)	Average Route Discovery time(s)	Throughput	Route Discovery time
6	327803.4227	0.113924823	340254.8	0.074694411	4%(↑)	34%(↓)

The table 7.7 is obtained from the graphs shown in figure 7.13 and figure 7.14. The values in the tables are the average of the values shown in the graphs 7.13 and 7.14. ↓ indicates decrease in value and ↑ indicate an increase in value. The table and graphs clearly show that the results are improved with TMA-AODV compared to AODV routing protocol in the presence of 3 packet drop and 3 packet delay attacker nodes out to 69 nodes of the MANET.

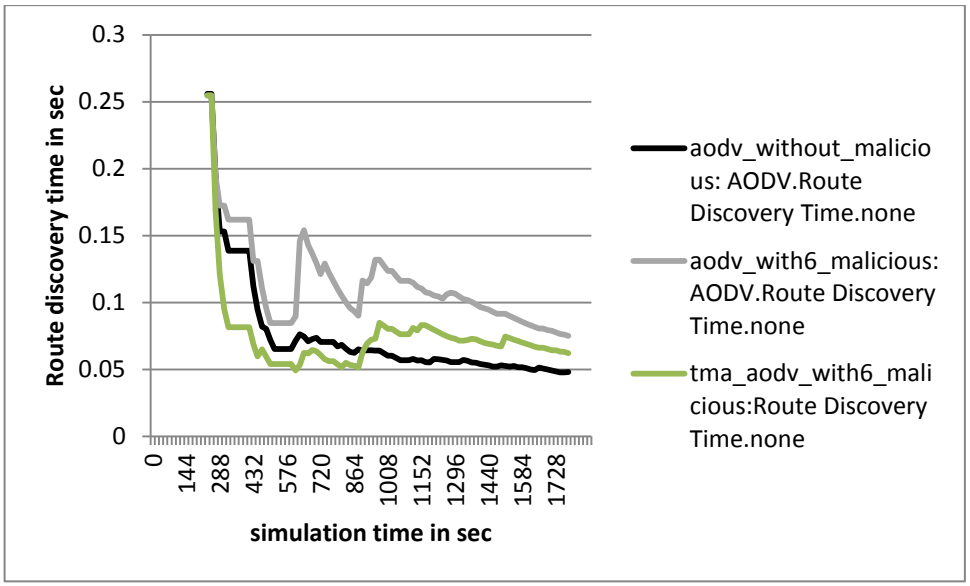


Figure 7.13 Comparison of Route Discovery Time with AODV and TMA-AODV (3 packet drop +3 packet delay attacker)

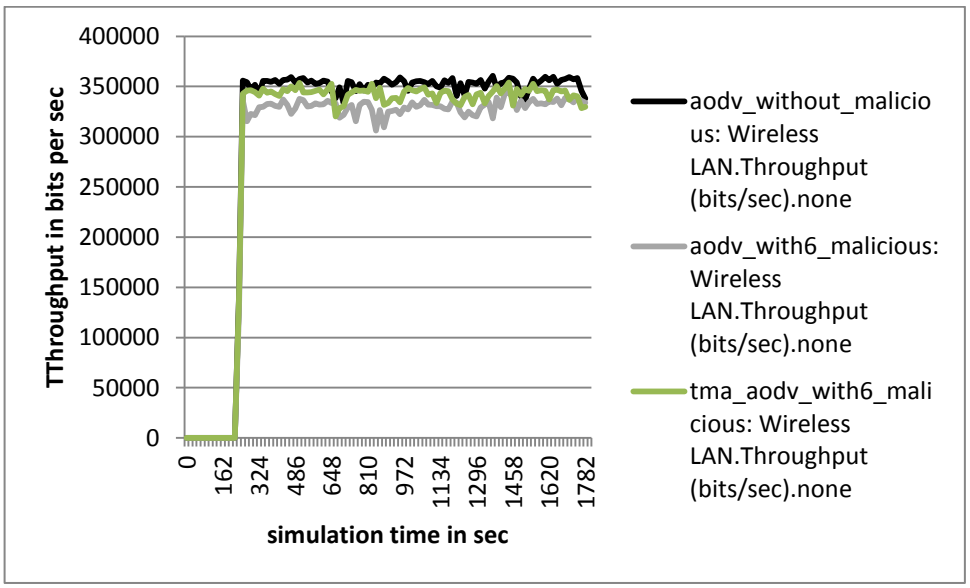


Figure 7.14 Comparison of Throughput with AODV and TMA-AODV (3 packet drop +3 packet delay attacker)

7.4.2 TMA-AODV with 6 packet drop and 6 packet delay attacker nodes

In experiment 6.5.2, we have added one more scenario same as figure 6.20 which contains 6 packet drop and 6 packet delay attacker nodes and TMA-AODV as a routing protocol. The results for the throughput and the route discovery time obtained with TMA-AODV is compared with AODV without attacker nodes and AODV with packet drop and delay attacker nodes as shown in the figure 7.15 and figure 7.16. In the graph shown in figure 7.15, you can see that initially the route discovery time with TMA-AODV routing is almost same as the route discovery time with AODV routing and attacker nodes(6 drop + 6

delay attacker nodes). This is because TMA-AODV needs learning period to record observation on nodes. After that the route discovery time with TMA-AODV is even less than route discovery time with AODV without any attacker node. This is because in TMA-AODV routing protocol, multiple routes are found and they used simultaneously to send data packet and a new route will be searched only when all routes are broken. The graph 7.16 shows the improvement in throughput with TMA-AODV in the presence of attacker nodes (6 delay + 6 drop) compare to AODV routing. The improvement is due to detection and avoidance of malicious nodes by TMA-AODV which reduce the link breaks and data loss during the communication.

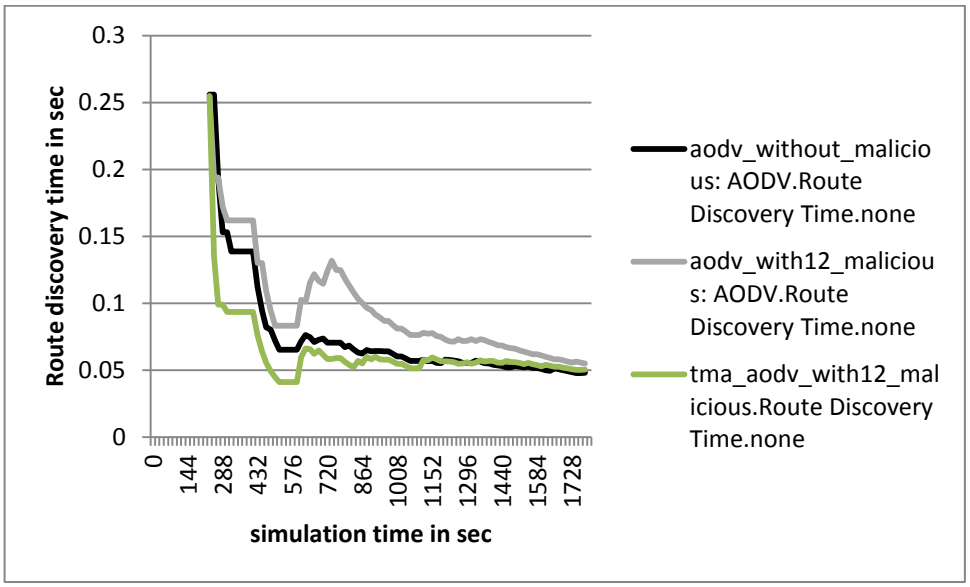


Figure 7.15 Comparison of Route Discovery Time with AODV and TMA-AODV (6 packet drop + 6 packet delay attacker)

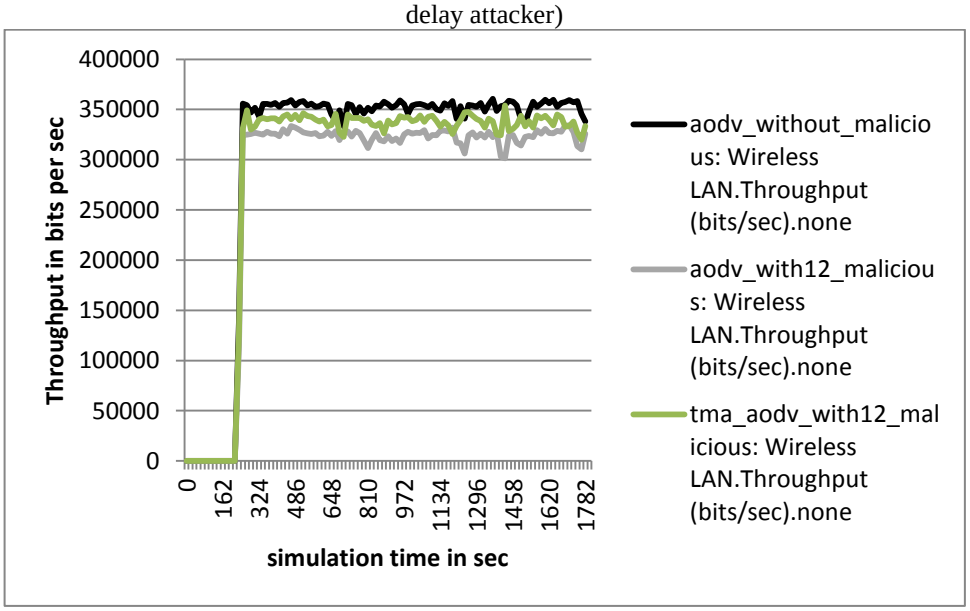


Figure 7.16 Comparison of Throughput with AODV and TMA-AODV (6 packet drop + 6 packet delay attacker)

Table 7.8 Improvement in average throughput and average route discovery time with TMA-AODV (6 packet drop + 6 packet delay attacker)

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)	Average Throughput (bps)	Average Route Discovery time(s)	Throughput	Route Discovery time
12	321633.9821	0.095035425	335520.092	0.06257037	4.2%(↑)	34.6%(↓)

The table 7.8 is obtained from the graphs by calculating average of values in graphs shown in figure 7.15 and figure 7.16. ↓ indicates decrease in value and ↑ indicate an increase in value. The table and graphs clearly show that the results are improved (route discovery time decreased by 34.6 % and throughput increased by 4.2%) with TMA-AODV compared to AODV routing protocol in the presence of 6 packet drop and 6 packet delay attacker nodes out to 69 nodes of the MANET. Ideally, if we increase attacker node, the route discovery time should be increased. If we compare the average route discovery time with AODV with 6 packet drop and 6 packet delay attacker nodes(0.095035425 sec) and 3 packet drop and 3 packet delay attacker nodes(0.113924823 sec), we can see that the average route discovery time with less attacker node is less than the average route discovery time with more attacker nodes. The reason behind that is already discussed in section 6.5.2. If we compare table 7.7 and table 7.8, we can see that the percentage improvement in the throughput and the route discovery time with TMA-AODV compared to AODV in the presence of drop and delay attacker nodes are almost same. This is due to the presence of less attacker nodes and they are uniformly distributed in the overall network. Also, the half of the attacker nodes in both network scenarios (3 and 6) are packet delay attacker nodes. The packet drop attacker nodes in both scenarios are 3 and 6 respectively. The packet drop attacker nodes are more destructive than delay attacker nodes which we have already shown in section 6.4. With less attacker node, the TMA-AODV shows less improvement. When attacker nodes are less, there are less routes containing attacker nodes. The overhead involves in recording parameters and trust calculation at each node suppresses the improvement in results due to trust based routing.

7.4.3 TMA-AODV with 12 packet drop and 12 packet delay attacker nodes

In the experiment 6.5.3, we have added one more scenario same as figure 6.23 which contains 12 packet drop and 12 packet delay attacker nodes and TMA-AODV as a routing protocol. The result for the throughput and the route discovery time obtained with TMA-AODV is compared with AODV without attacker nodes and AODV with packet drop and

delay attacker nodes shown in figure 6.25 and figure 6.26. The figure 7.18 and figure 7.17 shows the comparison of throughput and route discovery time of AODV without malicious node, AODV with 12 packet drop and 12 packet delay attacker nodes and TMA-AODV with 12 packet drop and 12 packet delay attacker nodes. In graph 7.17, the route discovery time with AODV in the presence of attacker nodes (12 drop +12 delay) is very large compared to other two scenarios (AODV without attacker and TMA-AODV with attackers). This is because of more packet drop and packet delay attacker nodes. These attacker nodes are distributed uniformly in simulation area and hence can be a part of almost each route. Thus, it disturbs each routing operation and hence route discovery time is very large. Also with TMA-AODV the route discovery time is reduced in the presence of the same attacker nodes. This is because the attacker nodes are easily detected and avoided by TMA-AODV routing. More attacker nodes mean more malicious activities which will be recorded in trust table. As more observation available, the TMA-AODV can easily detect these malicious activities. Hence, the throughput should also increase with TMA-AODV. Figure 7.18 shows the improvement in throughput with TMA-AODV in the presence of the attacker (12 drop +12 delay) nodes compare to AODV with and without attacker (12 drop +12 delay) nodes.

Table 7.9 Improvement in average throughput and average route discovery time with TMA-AODV(12 packet drop + 12 packet delay attacker)

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)	Average Throughput (bps)	Average Route Discovery time(s)	Throughput	Route Discovery time
24	300273.2874	0.527558508	315647.6424	0.253345187	5.3%(↑)	51%(↓)

The table 7.9 is obtained from the average of the values found in the graphs shown in figure 7.17 and figure 7.18. ↓ indicates decrease in value and ↑ indicate an increase in value. The table and graphs clearly show that the results are improved with TMA-AODV compared to AODV routing protocol in the presence of 12 packet drop and 12 packet delay attacker nodes out to 70 nodes of the MANET. If we compare tables 7.7 and 7.8 with the results shown in table 7.9, we can clearly see that the improvement in the route discovery time and the throughput is more here. This is because of the presence of more attacker nodes in this network scenario. With more attacker nodes, more malicious activities are recorded and detected by TMA-AODV, which helps to avoid them and also helps to improve the performance of the network.

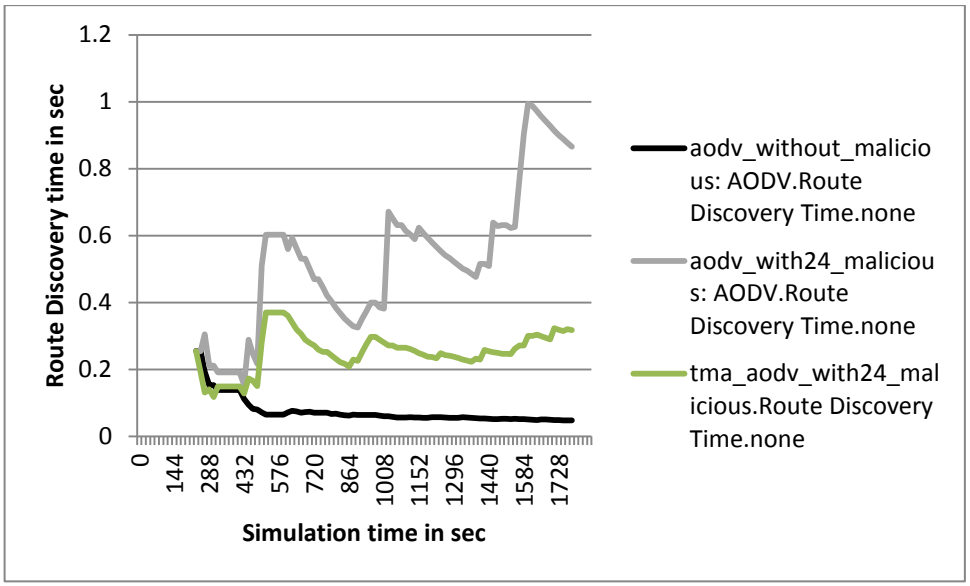


Figure 7.17 Comparison of Route Discovery Time with AODV and TMA-AODV (12 packet drop + 12 packet delay attacker)

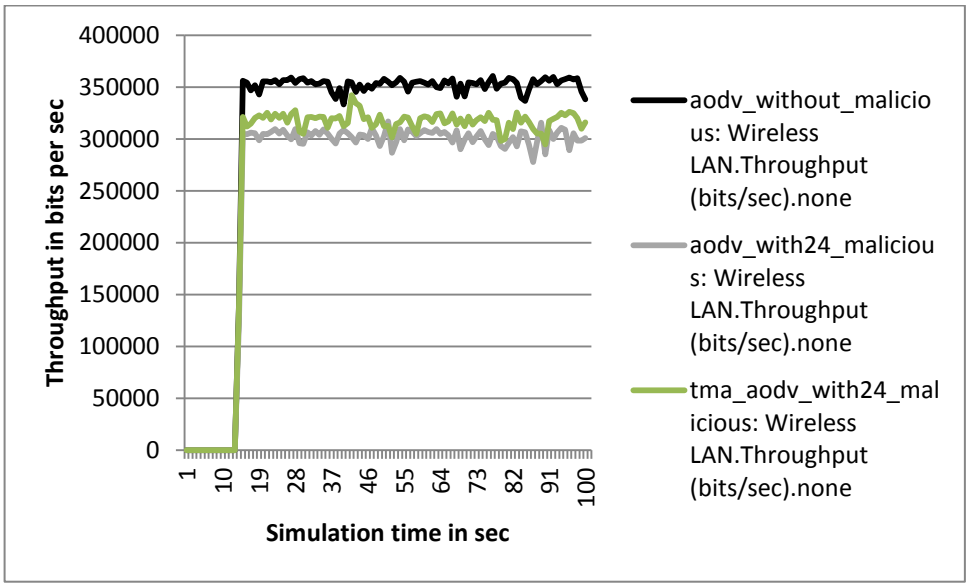


Figure 7.18 Comparison of Throughput with AODV and TMA-AODV (12 packet drop + 12 packet delay attacker)

7.4.4 Concluding Remarks

The average of all the results obtained after the experiment is shown in table 7.10. For packet drop and delay attack, we got following improvement with our proposed routing protocol(TMA-AODV). ↓ indicates decrease in value and ↑ indicate an increase in value.

Table 7.10 Average result obtained when Packet drop and delay

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)	Average Throughput (bps)	Average Route Discovery time(s)	Throughput	Route Discovery time
0	350220.4	0.074528738	340092.9	0.07805756	3%(↓)	4.7%(↑)
6	327803.4227	0.113924823	340254.8	0.074694411	4%(↑)	34%(↓)
12	321633.9821	0.095035425	335520.092	0.06257037	4.2%(↑)	34.6%(↓)
24	300273.2874	0.527558508	315647.6424	0.253345187	5.3%(↑)	51%(↓)

From the first row of the table, we can see that without attacker node the performance of the network is degraded, if we use TMA-AODV. When no attacker nodes, with TMA-AODV routing the average throughput is less and the average route discovery time is more compared to AODV, which is due to the overhead associated with TMA-AODV. From the table 7.10, we can see that, if we increase the attacker nodes in the network, the improvement in results is also more i.e. more increment in throughput and more reduction in route discovery time.

The graphs shown in figure 7.19 and 7.20 shows the improvement in route discovery time and throughput results with our proposed routing protocol TMA-AODV compared to AODV in the presence of packet drop and delay attacker nodes. As shown in figure 7.19, if we increase the attacker nodes, the route discovery time is also increased with AODV routing protocol. However, with TMA-AODV, we can see low route discovery time compare to AODV. The figure 7.20 shows reduction in throughput with AODV routing, if we increase attacker nodes. The figure also shows that the throughput is improved by using TMA-AODV routing with same attacker nodes. The reason for improvement in the route discovery time and the throughput is the trust based mechanism used in TMA-AODV which detects the malicious nodes. Also, by not using the routes which have more malicious nodes as intermediating node the TMA-AODV outperforms than AODV.

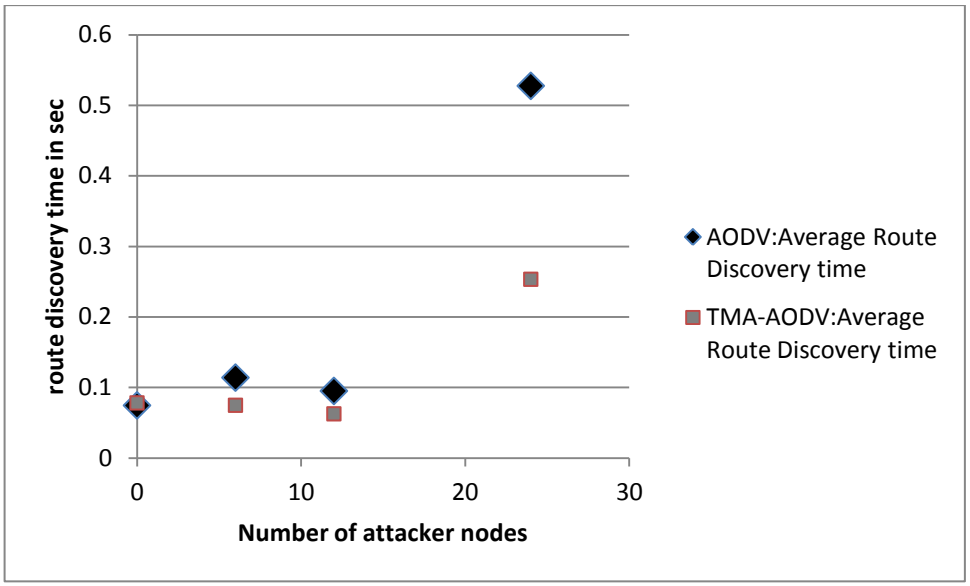


Figure 7.19 Comparison of average route discovery time of AODV and TMA-AODV in presence of packet drop and delay attack

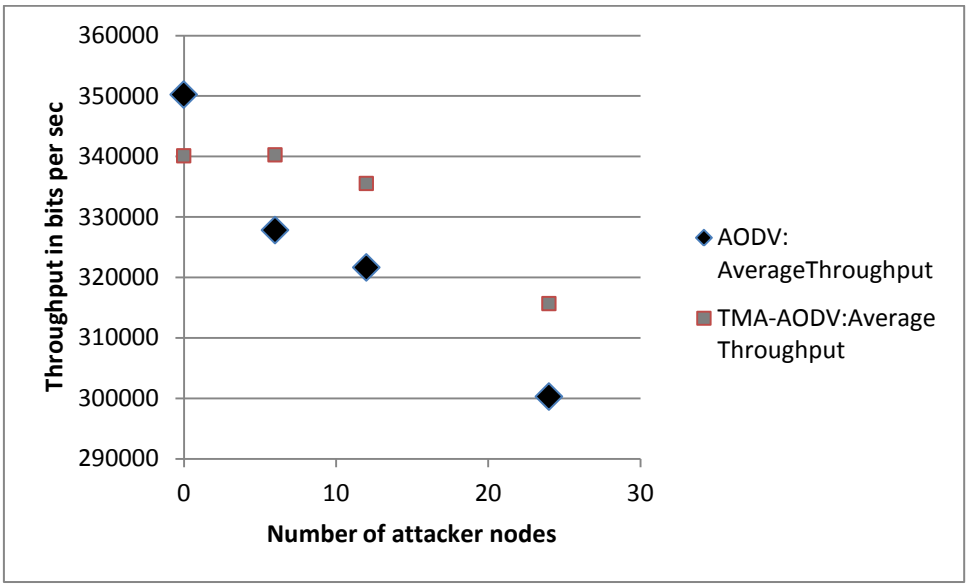


Figure 7.20 Comparison of average throughput with AODV and TMA-AODV routing in presence of packet drop and delay attack

7.5 TMA-AODV in presence of mobile nodes, packet drop and packet delay attacker nodes

The TMA-AODV routing is a trust based routing protocol. This protocol found all routes with the trust value associated with it and use the most trustworthy route for communication. For calculating trust value of a node, TMA-AODV uses various parameters like the number of packet drop by that node, the number of packets delayed by

that node and the number of link break by the node. The first two parameters are used to detect packet drop or packet delay related attacks while the third parameter is used to give us a stable route i.e. the route with less link break. During TMA-AODV routing, if the route which has a node with more mobility as an intermediate node, it has low trust value. TMA-AODV routing find all possible routes between the same source and destination with trust value associated with each route and uses the most trust worthy routes (routes with high trust value) for sending data. Also TMA-AODV uses multiple routes simultaneously, thus improves the throughput and the route discovery time in the presence of mobile nodes. To prove this claim, we have implemented an experiment comparing AODV routing and TMA-AODV routing without any attacker nodes and with 16 mobile nodes in section 7.2.2. The average route discovery time and the average throughput results with TMA-AODV routing are improved compared to AODV routing results (table 7.2). In this section we have compared the proposed TMA-AODV routing with the AODV routing in presence 9%, 18% and 27% of total node as attacker nodes and 16 mobile nodes out of 70 network nodes. The half of the attacker nodes are packet drop attacker nodes and the other half are packet delay attacker nodes in each scenario.

7.5.1 TMA-AODV with 3 packet drop and 3 packet delay attacker nodes and 16 mobile nodes

We have created the experiment setup same as section 6.7.1. In experiment 6.7.1, we have added one more scenario same as figure 6.29 which contains 6 attacker nodes (3 packer drop attackers and 3 packet delay attackers) and 16 mobile nodes with TMA-AODV as a routing protocol on each node. The result for the throughput and the route discovery time obtained with this scenario (TMA-AODV) is compared with AODV without attacker nodes and mobile nodes and AODV with packet drop attacker nodes, packet delay attacker nodes and mobile nodes are shown in the figure 7.22 and figure 7.21. The graph 7.21 shows that route discovery time with TMA-AODV is very large initially. This is due to two reasons. One reason is, as the number of attacker nodes is less and initial movement of the mobile node is also not detected due to lack of observation information. With time the nodes observe the movement and behaviors of their neighbours and use them to take a route decision. The second reason is, because TMA-AODV found multiple trusted routes from source to destination during the route discovery phase. To search multiple trusted routes more time is required compared to AODV routing. However, after finding multiple routes TMA-AODV uses them simultaneously to send data packets. Hence, with the time,

the route discovery time with TMA-AODV is reduced as shown in figure 7.21. From graph shown in figure 7.22, we can clearly see the improvement in throughput with TMA-AODV routing in the presence of attacker nodes and mobile nodes.

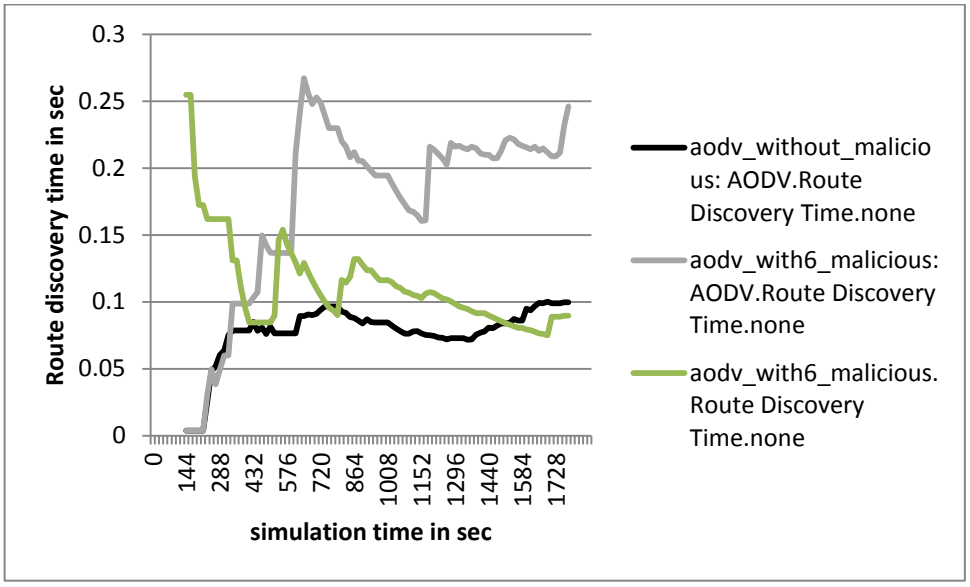


Figure 7.21 Comparison of Route Discovery Time with AODV and TMA-AODV (3 packet drop + 3 packet delay attacker and 16 mobile nodes)

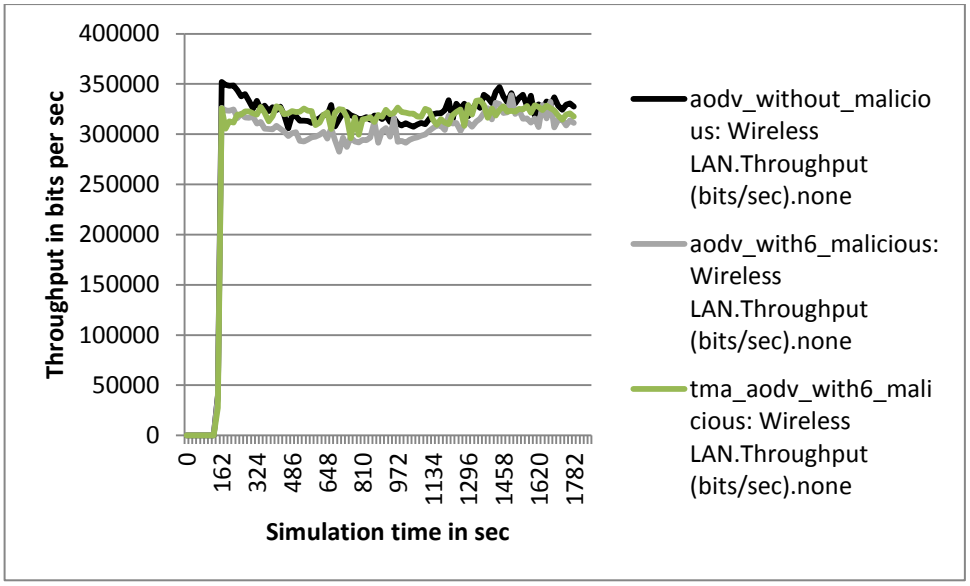


Figure 7.22 Comparison of Throughput with AODV and TMA-AODV (3 packet drop + 3 packet delay attacker and 16 mobile nodes)

Table 7.11 Improvement in Average throughput and Average route discovery time with TMA-AODV (3 packet drop + 3 packet delay attacker and 16 mobile nodes)

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)	Average Throughput (bps)	Average Route Discovery time(s)	Throughput	Route Discovery time
6	305048.2126	0.174754203	316856.768	0.112584966	10%(↑)	35%(↓)

The table 7.11 is obtained from the average of values obtained from the graphs shown in figure 7.21 and figure 7.22. ↓ indicates decrease in value and ↑ indicate an increase in value. The table and graphs clearly show that the results are improved with TMA-AODV compared to AODV routing protocol in the presence of 3 packet drop, 3 packet delay attacker nodes and 16 mobile nodes out to 70 nodes of the MANET.

7.5.2 TMA-AODV with 6 packet drop and 6 packet delay attacker nodes and 16 mobile nodes

In experiment 6.7.2, we have added one more scenario same as figure 6.32 which contains 6 packet drop attacker nodes, 6 packet delay attacker nodes and 16 mobile nodes with TMA-AODV as a routing protocol on each node. The result for the throughput and the route discovery time obtained with TMA-AODV is compared with AODV without attacker nodes and AODV with mobile nodes, packet drop and delay attacker nodes shown in figure 6.34 and figure 6.35. The figure 7.24 and figure 7.23 shows the comparison of throughput and route discovery time of AODV without malicious node, AODV with 6 packet drop, 6 packet delay attacker nodes and 16 mobile nodes and TMA-AODV with 6 packet drop, 6 packet delay attacker nodes and 16 mobile nodes. In figure 7.23, initially the route discovery time with TMA-AODV with 12 attacker nodes and 16 mobile nodes is almost same as AODV routing with 12 attacker nodes and 16 mobile nodes. This is because the initial learning phase of the trust mechanism. After some time when nodes have enough observation the attacker nodes and mobile nodes are detected and avoided by TMA-AODV and hence the route discovery time is reduced compared to AODV with attacker and mobile nodes. The figure 7.24 shows the improvement in throughput with TMA-AODV in the presence of 12 attacker and 16 mobile nodes compared to AODV routing.

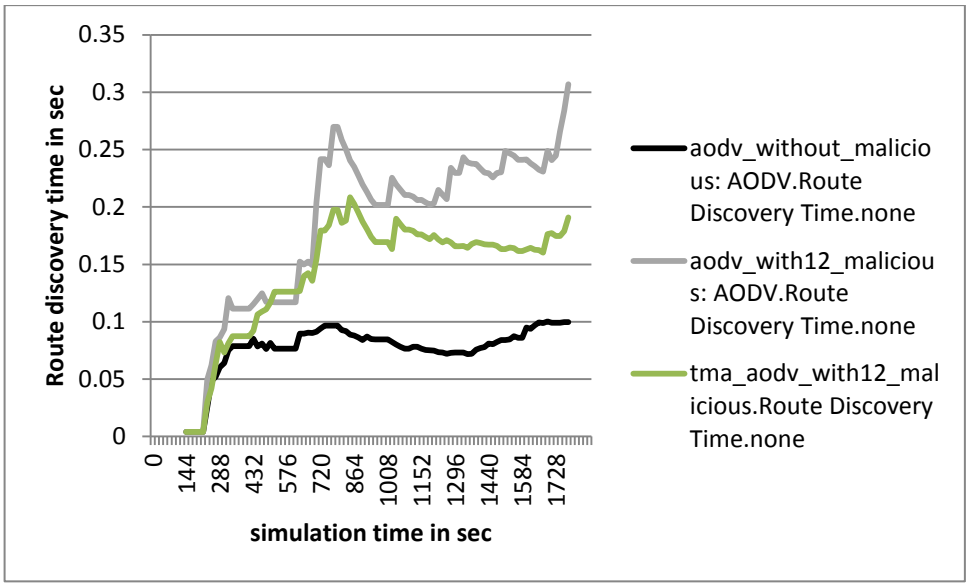


Figure 7.23 Comparison of Route Discovery Time with AODV and TMA-AODV (6 packet drop + 6 packet delay attacker and 16 mobile nodes)

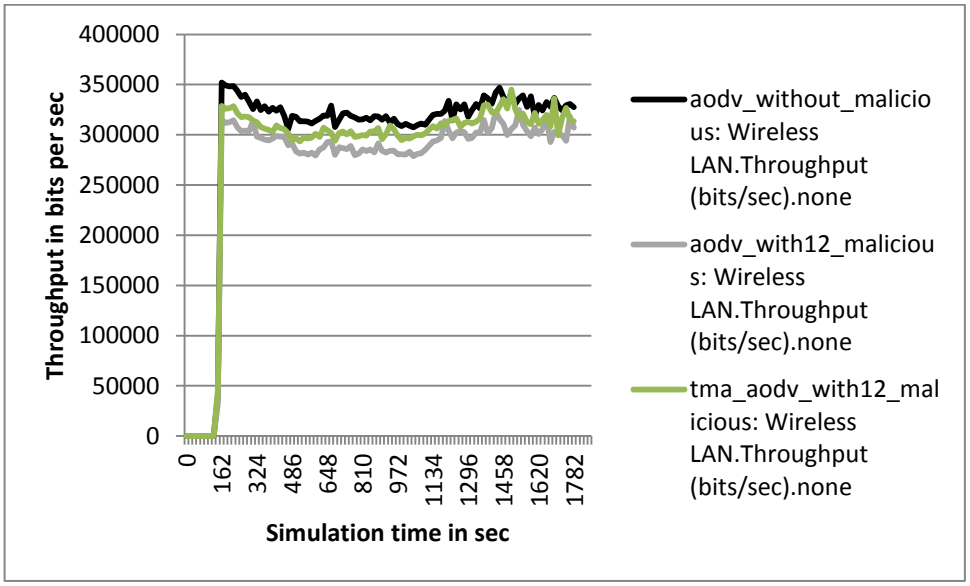


Figure 7.24 Comparison of Throughput with AODV and TMA-AODV (6 packet drop + 6 packet delay attacker and 16 mobile nodes)

Table 7.12 Improvement in average throughput and average route discovery time with TMA-AODV (3 packet drop + 3 packet delay attacker and 16 mobile nodes)

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)	Average Throughput (bps)	Average Route Discovery time(s)	Throughput	Route Discovery time
12	293384.5024	0.185386923	307228.2899	0.144419099	12%(↑)	27%(↓)

The table 7.12 is obtained by calculating the average of values from the graphs shown in figure 7.23 and figure 7.24. ↓ indicates decrease in value and ↑ indicate an increase in value. The table and graphs clearly show that the results are improved with TMA-AODV compared to AODV routing protocol in the presence of 6 packet drop, 6 packet delay attacker nodes and 16 mobile nodes out to 70 nodes of the MANET.

7.5.3 TMA-AODV with 12 packet drop and 12 packet delay attacker nodes and 16 mobile nodes

In experiment 6.7.3, we have added one more scenario same as figure 6.35 which contains 12 packet drop, 12 packet delay attacker nodes and 16 mobile nodes with TMA-AODV as a routing protocol at each node. The result for the throughput and the route discovery time obtained with TMA-AODV is compared with AODV without attacker nodes, AODV with mobile nodes, packet drop and delay attacker nodes shown in figure 6.37 and figure 6.38. The figure 7.26 and figure 7.25 shows the comparison of throughput and route discovery time of AODV without malicious node, AODV with 12 packet drop, 12 packet delay attacker nodes and 16 mobile nodes and TMA-AODV with 12 packet drop, 12 packet delay attacker nodes and 16 mobile nodes. In figure 7.25, initially the route discovery time with TMA-AODV with 24 attacker nodes and 16 mobile nodes is almost same as AODV routing with 24 attacker nodes and 16 mobile nodes. This is because of the initial learning phase of trust mechanism. After some time when nodes have enough observation the attacker nodes and mobile nodes are detected and avoided by TMA-AODV and hence the route discovery time is reduced compared to AODV with attacker and mobile nodes. The figure 7.26 shows the improvement in throughput with TMA-AODV in the presence of 24 attackers and 16 mobile nodes compare to AODV routing.

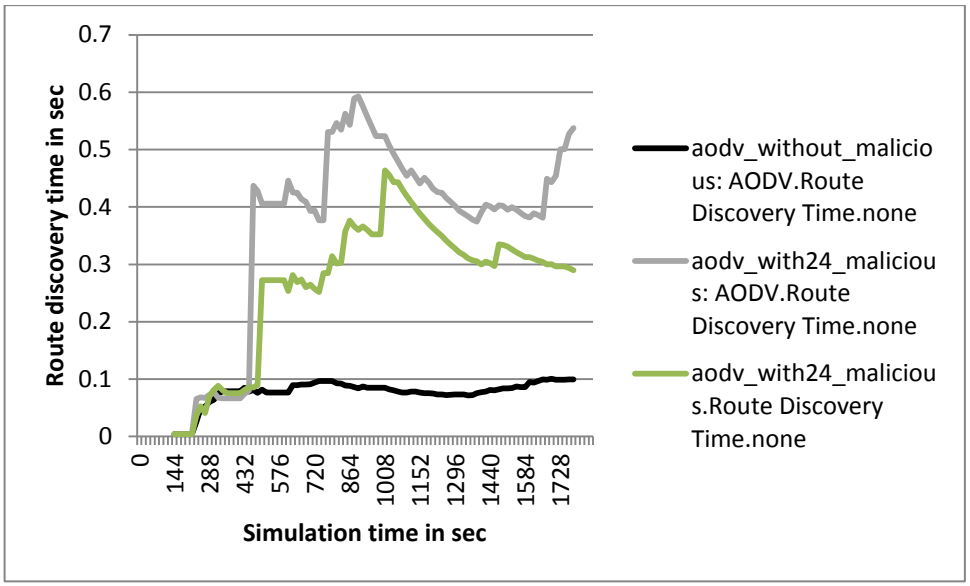


Figure 7.25 Comparison of Route Discovery Time with AODV and TMA-AODV (12 packet drop + 12 packet delay attacker and 16 mobile nodes)

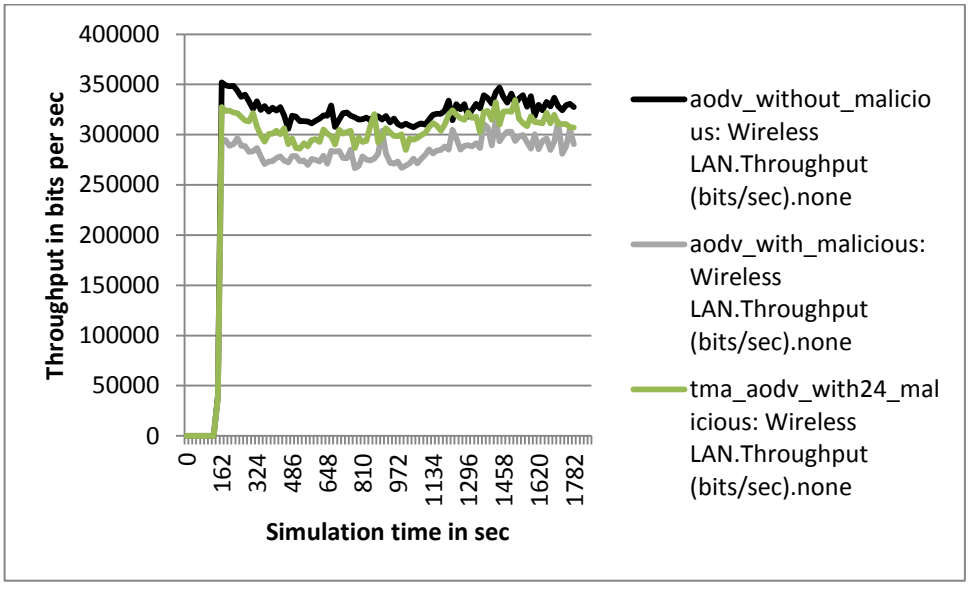


Figure 7.26 Comparison of Throughput with AODV and TMA-AODV (12 packet drop + 12 packet delay attacker and 16 mobile nodes)

Table 7.13 Improvement in average throughput and average route discovery time with TMA-AODV (12 packet drop + 12 packet delay attacker and 16 mobile nodes)

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Route Discovery time(s)	Throughput (bps)	Route Discovery time(s)	Throughput	Route Discovery time
24	282207.2271	0.368930013	304621.8164	0.267453009	14%(↑)	22%(↓)

The table 7.13 is obtained by calculating the average of values from the graphs shown in figure 7.25 and figure 7.26. ↓ indicates decrease in value and ↑ indicate an increase in value. The table and graphs clearly show that the results are improved with TMA-AODV compared to AODV routing protocol in the presence of 12 packet drop, 12 packet delay attacker nodes and 16 mobile nodes out to 70 nodes of the MANET.

7.5.4 Concluding Remarks

The all results obtained after the experiment is shown in table 7.14. For packet drop attack packet delay attack and mobility, we got following improvement with our proposed routing protocol. ↓ indicates decrease in value and ↑ indicate an increase in value.

Table 7.14 Average Result obtained when Packet drop+delay+mobility

	AODV		TMA-AODV		Improvement compares to AODV	
Attacker nodes	Average Throughput (bps)	Average Route Discovery time(s)	Average Throughput (bps)	Average Route Discovery time(s)	Throughput	Route Discovery time
0	321504.2126	0.094526	336514.1159	0.075984588	4.67%(↑)	19.61%(↓)
6	305048.2126	0.174754203	316856.768	0.112584966	10%(↑)	35%(↓)
12	293384.5024	0.185386923	307228.2899	0.144419099	12%(↑)	27%(↓)
24	282207.2271	0.368930013	304621.8164	0.267453009	14%(↑)	22%(↓)

The first row of the table 7.14 shows the average throughput and the average route discovery time of network with AODV routing and TMA-AODV routing in the absence of any attacker node and in the presence of 16 mobile nodes in the network. With TMA-AODV routing the average throughput is increased by 4.67% and average route discovery time is reduced by 19.61% of the throughput and route discovery time obtained with AODV routing. This is because the TMA-AODV routing detects the mobile nodes and avoids them in route whenever possible. The improvement in results is less because of the routing overhead with TMA-AODV, as traffic monitoring and trust modelling modules are added. The 2nd row onwards, the table shows the throughput and the route discovery time of the network with AODV routing with 16 mobile nodes and TMA-AODV routing in the presence of 6, 12 and 24 packer drop and delay attacker node and 16 mobile nodes in the network. The table 7.14 shows that if we add the attacker node in the network, the throughput is reduced and route discovery time is increased with AODV routing which is improved with TMA-AODV routing.

If you study the improvement columns of table 7.14, you can see that when attacker nodes are increased, the improvement in throughput is increased almost linearly. But the improvement of the route discovery time is reduced, when we increase the attacker nodes. This is because now in the network, we have not only attacker nodes. We have mobile nodes also. Due to movement of nodes, the route discovery process may delay, if a node moves out of the range of the sender node after receiving the routing control packets like RREQ, RREP or RERR.

The graphs shown in figure 7.27 and 7.28 shows the improvement in the average route discovery time and the average throughput results with our proposed routing protocol TMA-AODV compared to AODV in the presence of packet drop, packet delay attacker nodes and mobile nodes. You can clearly see the improvement in the average route discovery time and throughput with TMA-AODV routing protocol.

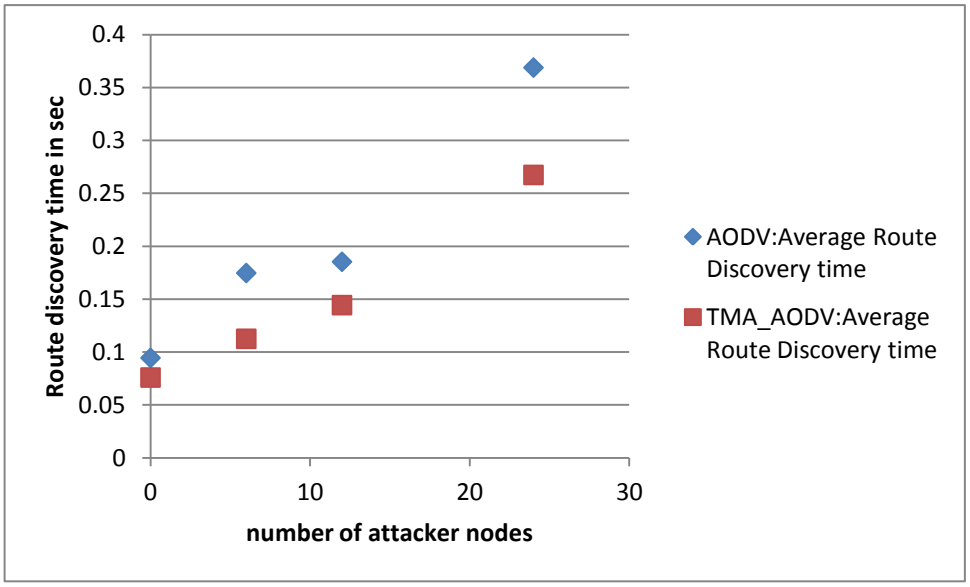


Figure 7.27 Comparison of average route discovery time of AODV and TMA-AODV in presence of mobile nodes and packet drop/delay attack

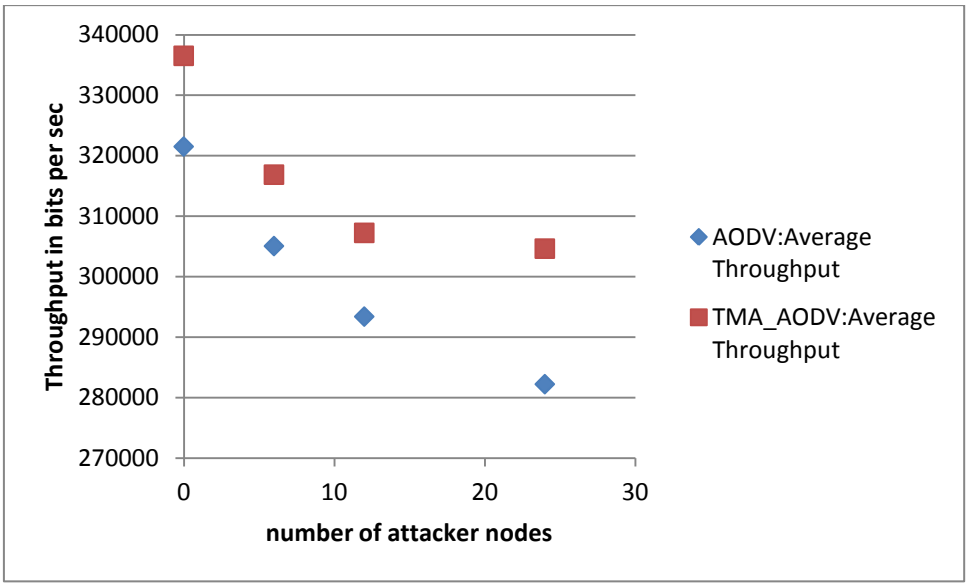


Figure 7.28 Comparison of average throughput with AODV and TMA-AODV routing in the presence of mobile nodes and packet drop/delay attacks

7.6 Conclusion

In this chapter, we have implemented our proposed trust based mobility aware routing protocol in OPNET and check whether the throughput and the route discovery time of the network is improved or not in the presence of 6, 12 and 24 attacker nodes and 16 mobile nodes. We have added one scenario in each experiment setup, we have discussed in section 6.4, 6.5 and 6.7. In these scenarios, we have added a TMA-AODV as a routing protocol, and added packet drop attacker nodes (9%, 18% and 27%), packet drop(4.5%, 9% and 13.5%) and delay(4.5%, 9% and 13.5%) attacker nodes and packet drop attacker nodes(4.5%, 9% and 13.5%) , packet delay attacker nodes(4.5%, 9% and 13.5%) and 16 mobile nodes. The average throughput and average route discovery time obtained with these scenarios are compared with results obtained with the same scenarios and AODV routing. With TMA-AODV routing the average throughput and average route discovery time is improved as shown in following tables in the presence of attacker nodes and mobility.

Table 7.15 Improvement with packet drop attack

Drop Attacker nodes	Improvement with TMA-AODV compares to AODV	
	Throughput	Route Discovery time
0	3%(↓)	4.7%(↑)
6	8%(↑)	21%(↓)
12	9%(↑)	40%(↓)
24	13%(↑)	88%(↓)

Table 7.16 Improvement with packet drop attack and packet delay attack

Drop + Delay Attacker nodes	Improvement with TMA-AODV compares to AODV	
	Throughput	Route Discovery time
0	3%(↓)	4.7%(↑)
6	4%(↑)	34%(↓)
12	4.2%(↑)	34.6%(↓)
24	5.3%(↑)	51%(↓)

Table 7.17 Improvement with packet drop attack, packer delay attack and mobility

Attacker nodes + 16 mobile nodes	Improvement with TMA-AODV compares to AODV	
	Throughput	Route Discovery time
0	4.67%(↑)	19.61%(↓)
6	10%(↑)	35%(↓)
12	12%(↑)	27%(↓)
24	14%(↑)	22%(↓)

When we compare AODV routing and TMA-AODV routing without any attacker nodes, we got a reduction in throughput and an increment in route discovery time with TMA-AODV. From this comparison, we can conclude that the TMA-AODV does not perform well in the absence of malicious nodes. This is because in TMA-AODV, we have involved packet monitoring overhead at each node. The node records the activities of all the neighbours in its trust table by monitoring traffic on them. The TMA-AODV routing also add trust calculation module during route discovery. When RREP packet is created by a node, it must have a field used to store trust value. This field is initialized to zero. Before sending RREP packet further node has to compute the trust value of the next hop neighbour, using the value recorded in a local trust table and adds it to trust value in RREP. This both activities involve overhead and hence in the absence of attacker node the performance of TMA-AODV is not good (shown in 1st row tables of 7.15 and 7.16)

We also compared AODV and TMA-AODV in the absence of any malicious node and in the presence of mobile nodes. We got improvement in the throughput and route discovery time with TMA-AODV (shown in 1st row of table 7.17). We can see that the mobility is a big factor when applied in AODV without any attacker verses TMA-AODV without any attacker. This shows that TMA-AODV works well by avoiding the routes with mobile nodes. This is significant as generally in MANET the nodes are expected to be mobile. With AODV routing in the presence of mobile nodes, we did not get good results. This is

because traditional AODV uses the first available route. In TMA-AODV, we have involved a trust value with each route which is calculated based on activities on a node and movement of that node. The TMA-AODV finds out all the possible routes from a source to the destination with the trust value involve with each. Then, it uses most trustworthy routes for sending data. If a route has more intermediate nodes which has more mobility, the route is considered less trustworthy and not used. Hence, we got better results with TMA-AODV in the presence of mobile nodes.

Additionally, from the experimental results shown in table 7.15, 7.16 and 7.17, we can conclude that the TMA-AODV is performing better with mobility, drop and delay attackers. The only thing is the overhead with TMA-AODV that is found with no attackers nodes and no mobile nodes. When we compare TMA-AODV and AODV routing without attacker nodes and with no mobile nodes in the network, the route discovery time and the throughput with AODV is better compared to TMA-AODV. This is because, with TMA AODV, we have some overhead with trust and trusted route calculation. However, it is observed that the TMA-AODV works better in the presence of mobile nodes, packet drop attacker nodes and packet delay attacker nodes.

CHAPTER 8

Conclusion And Future Enhancement

8.1 Conclusion of the thesis

Mobile Ad hoc Networks are vulnerable to soft and hard security attacks during routing, due to the unique characteristics of the MANET. The routing is also done with the help of intermediate nodes, as the special routers are not used in MANET. The nodes in a MANET can enter or leave network without interference of any administrator or controlling authority. The MANET nodes are also resource constrained. The complex and computationally expensive cryptographic solutions are not advisable for the resource constraint MANET. Also, in a mobile ad hoc network the malicious or selfish behaviour of the node which changes with time cannot be detected using the cryptographic approaches. For securing routing in MANET from attacker nodes and selfish/misbehaving nodes, the trust based scheme can be used, because it is lightweight and simple.

In this thesis, for implementing trust based routing protocol, we chose the AODV routing as a base routing protocol. This is because the AODV routing uses the important features of both the table driven (DSDV) and link state(DSR) routing scheme. The AODV routing uses the route table on each node which store route information towards each destination. In case if route information is not available in route table, the AODV routing will search for route dynamically also. Thus, this protocol works well with less mobility and high mobility in the network.

In our proposed trust based routing scheme (Trust based Mobility Aware –AODV: TMA-AODV), we have observed total packets coming to a node, total packets successfully forwarded from the node, the total links break by a node and the total packets delayed at a node for each neighbour node to calculate the trust value of the nodes as well as the trust of the route. On each node a trust table is maintained, which stores the observed parameters of each neighbour node. For recording total number of link breaks due to a node, we have modified an RERR packet of the standard AODV routing. We have appended a 32 bit field

in RERR packet which stores the node address, who is unavailable to provide service for packet transfer on active route. The RERR packet is created by a intermediate node of an active route, when the node does not find next hop neighbour of a route. The next hop node may unavailable due to its movement or failure. While creating RERR packet, the intermediate node appends an IP address of the unavailable node that is responsible for route break in it. This RERR packet then sent to the source node of the route to inform it about route break. During the whole path the nodes and their neighbours receive this RERR packet and record the link break count of the unavailable node in trust table. These recorded values are used when the route is established in response to a RREQ by sending RREP packet. We have added a 32 bit field in RREP packet, which stores the trust value of the route. When a node receives an RREP packet, it calculates the trust value of next hop neighbour, using the values stored for that neighbour node in trust table and this trust value will be added in the trust value of the RREP. Thus, when the RREP reaches at source, the route is stored in the source node's route table with the trust value associated with it.

In our proposed scheme, we have used the weighted sum model for calculating trust values from the observed parameters. A weight value is associated with each parameter which changes with time based on the event recorded for the node. If more packet drop recorded for a specific node more weight will be given to them while calculating trust. Same is for delay attack and mobility.

In our proposed routing scheme, all possible routes from source to destination are found. We have calculated the average of trust value associated with each route, which will be threshold trust value. Source node uses all routes having trust value more than the threshold value, simultaneously to send data packets. Thus, the load of sending data balanced among more than one route which increase network throughput and compensate with the overhead use to calculate trust value and monitoring network traffic. Since multiple trustworthy routes are found for same source and destination node in TMA-AODV, there is no need to search a new route for each link breakage. We need to search for the new route only when all the routes break or expire. This leads to reduced route discovery time with TMA-AODV.

The result obtained from simulator also shows improvement in throughput and reduction in route discovery time with TMA-AODV in presence of drop and delay attacker nodes and in the absence and presence of mobility.

The results show that use of TMA-AODV without any attacker node and absence of mobility, reduce the throughput by 3% and increase route discovery time by 4.7% compared to AODV. This is due to overhead of modules we have added in TMA-AODV to detect and avoid routes with attacker nodes and mobile nodes.

We have also compared the throughput and the route discovery time of AODV in the absence and the presence of mobile nodes. The results show that in the presence of 16 mobile nodes out of total 70 network nodes, the throughput is reduced by 8.2% and route discovery time is increased by 26.3%. If we use TMA-AODV instead of AODV in the same network (16 mobile nodes out of 70 nodes) the throughput is improved by 4.67% and route discovery time is reduced by 19.61%.

If we introduce 9%, 18% and 27% of total nodes attacker nodes (Drop, Drop-delay without mobility and Drop-delay with mobility) in the network, with TMA-AODV throughput is increased and route discovery time is decreased as shown in following table compare to AODV.

Table 8.1 Improvements with TMA-AODV compare to AODV

% attacker nodes	Drop		Drop Delay without mobility		Drop Delay with mobility	
	Throughput compared to AODV (% increment)	Route Discovery time compared to AODV (% decrement)	Throughput compared to AODV (% increment)	Route Discovery time compared to AODV (% decrement)	Throughput compared to AODV (% increment)	Route Discovery time compared to AODV (% decrement)
9	8	21	4	34	10	35
18	9	40	4.2	34.6	12	27
27	13	88	5.3	51	14	22

The experiment results shows that, the TMA-AODV with no attackers nodes and no mobile nodes, the throughput and route discovery time results obtained are not good compared to AODV routing due to the overhead with packet monitoring module and trust modelling module, we added in TMA-AODV. The experimental results also show that, the TMA-AODV is a better approach to use compared to AODV in a real MANET network which has frequent mobility of the nodes (and hence, higher link breaks) and the presence of attackers which aim to disrupt the network by means of dropping or delaying the data and control packets between communications.

8.2 Future enhancements

There are several ways on which the routing protocol, which we have implemented can be extended to provide more stability to the route. In a TMA-AODV, for stable route we have considered only mobile nodes. If the route has more mobile nodes, there are more chances of link break due to node movement. However, route can also be broken due to failure of node due to insufficient battery life of the node. One can include the remaining battery life of neighbour nodes as a parameter to calculate the trust value to get a more stable route.

In this thesis, we proposed a routing protocol that detects packet drop and packet delay related attacks only. If any attacker node modifies the content of packet before forwarding it, that cannot be detected by our routing protocol. One can extend our trust calculation function by adding the total number of packet modified by a node as a parameter to calculate the trust value to detect packet modification related attacks.

Our proposed routing protocol is a trust based routing protocol. According to Adun Jøsang[74], the trust based scheme which are used in network are vulnerable to various attacks like Playbooks ,Unfair ratings , Discrimination, Collusion, Proliferation, Reputation lag, Re-entry/Change of identity, etc. As a future work, one can do analysis of TMA-AODV to check how secure it is in presence of all these attacks. One can also provide the possible solutions for securing the TMA-AODV routing against these attacks.

References

1) Humayun Bakht, Lecture Slides on “History of mobile ad hoc network”, School of Computing and Mathematical Sciences, Liverpool John Moores University, 2005.
2) Jeroen Hoebeke, Ingrid Moerman, Bart Dhoedt and Piet Demeester, "An overview of mobile ad hoc networks: Applications and challenges", Journal-Communications Network-July 2004
3) http://www.home-network-help.com/wireless-network.html
4) Humayun Bakh, “Understanding mobile ad-hoc networks: WIRELESS INFRASTRUCTURE”, Computing unplugged magazine June 2004 issue.
5) Hosek, J., “Performance Analysis of MANET Routing protocols OLSR and AODV” electrorevue ISSN 1213-1539, 2 (3), 22-27,2011
6) Maghsoudlou A.,Sthilaire M. and Kunz T., “A Survey on Geographic Routing Protocols for Mobile Ad hoc Networks”, Carleton University, Systems and Computer Engineering, Technical Report SCE – 11 - 03, October 2011.
7) Atekeh Maghsoudlou, Marc St-Hilaire, and Thomas Kunz, "A Survey on Geographic Routing Protocols for Mobile Ad hoc Networks", Carleton University, Systems and Computer Engineering, Technical Report SCE-11-03, October 2011.
8) J. Macker and S. Corson, “Mobile Ad hoc Networks (MANET)”, http://www.ietf.org/charters/manet-charter.html, IETF Working Group Charter,1997
9) Alex Zinin, “Ad hoc On-Demand Distance Vector (AODV) Routing draft-ietf-manet-aodv- 10.txt” by Mobile Ad Hoc Networking Working Group, 2013.
10) http://wiki.uni.lu/secan-lab/Zone+Routing+Protocol.html
11) P. Chenna Reddy Dr. P. Chandrasekhar Reddy, “PERFORMANCE ANALYSIS OF ADHOC NETWORK ROUTING PROTOCOLS”, Academic Open Internet Journal ISSN 1311-4360,2006.
12) J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols”, Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), October 1998.
13) Fenglien Lee, “Routing in Mobile Ad hoc Networks, Mobile Ad-Hoc Networks: Protocol Design”, Prof. Xin Wang (Ed.), ISBN:978-953-307-402-3, InTech, 2011.
14) Krishna Gorantala, “Routing Protocols in MobileAd-hoc Networks”, Master thesis report, Umea University, Sweden,2006
15) Sliman KA., A. Yaklaf, Abdurrezagh S. Elmezughi, Nasser Bashir Ekreem and Adel A. M. Abosdel , "Security Routing Protocols in Ad Hoc Networks: Challenges and Solutions" proceedings of the International Conference on Recent Advances in Electrical Systems, Tunisia, 2016
16) Hosek, J., “Performance Analysis of MANET Routing protocols OLSR and AODV”, electrorevue ISSN 1213-1539, 2(3), 22-27,2011.
17) David Johnson, “Routing in Ad Hoc Networks for Mobile Hosts”, Proceedings of IEEE WMCSA1994, Santa Cruz, CA,1994.
18) Perkins and Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, Proceedings of ACM SIGCOMM94, pp234-244, London, 1994.
19) Perkins, Belding-Royer and Das, “Ad hoc On-Demand Distance Vector (AODV) Routing”, in: Network Working Group – Request for Comments 3561, available from: http://tools.ietf.org/html/rfc3561, 7/2003.
20) Johnson, Maltz and Broch, “DSR: The Dynamic Source Routing Protocol for Multi-HopWireless Ad Hoc networks”, in: Ad Hoc Networking, publisher: Edison Wesley, 2001.
21) Clausen and Jacquet, “Optimized Link State Routing Protocols (OLSR)”, in: Network Working Group – Request for Comments 3626, available from: http://tools.ietf.org/html/rfc3626, 10/2003
22) P. Jacquet, P Muhlethaler, T Clausen, A Laouiti, A Qayyum and L Viennot " Optimized Link state routing protocol for ad hoc network" Hipercom Project, France, 2010.
23) Andreas Tønnesen, "Mobile Ad-Hoc Networks", Lecture notes from andreto@olsr.org (www.olsr.org, www.unik.no), 2004.
24) H Yang, H Y. Luo, F Ye, S W. Lu, and L Zhang, "Security in mobile adhoc networks: Challenges and solutions”, IEEE Wireless Communications. 11 (1), pp. 38-47,2004.
25) Jeroen Hoebeke, Ingrid Moerman, Bart Dhoedt and Piet Demeester, "An Overview of Mobile Ad Hoc Networks: Applications and Challenges", Journal- Communications Network, July 2004.
26) Sevil Şen, John A. Clark, Juan E. Tapiador, "Security Threats in Mobile Ad Hoc Networks", University of York, UK, may 2015.

27) D. Djenouri, L. Khelladi and A.N. Badache. “A Survey of Security Issues in Mobile Ad Hoc and Sensor Networks”, Communications Surveys & Tutorials, IEEE, 2005.
28) L. Zhou and Z. J. Hass, “Securing Ad Hoc Networks”, IEEE Networks Special Issue on Network Security, November/December 1999.
29) Data Integrity, from Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Data_integrity.
30) Y. Hu, A. Perrig, and D. Johnson, “Packet leashes: A defense against wormhole attacks in wireless adhoc networks”, Technical report, Department of Computer Science, Rice University, December 2001.
31) Jaydip Sen, M. Girish Chandra, Harihara S. G., Harish Reddy and P. Balamuralidhar, "A Mechanism for Detection of Gray Hole Attack in Mobile Ad Hoc Networks", Proceedings of the 6th International Conference on Information, Communications and Signal Processing (ICICS '07), Singapore 2007
32) Hu Y. C., Perrig A. and Johnson D.B., “Rushing Attacks and Defence in Wireless Ad Hoc Network Routing Protocols”, In Proc. of the ACM Workshop on Wireless Security, 2003.
33) Karlof C. and Wagner D., “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures”, Ad Hoc Networks, pp. 293-315, 2003.
34) Jared Cordasco and Susanne Wetzel, "Cryptographic vs. Trust-based Methods for MANET Routing Security", STM 2007.
35) Hu Y. C., Perrig A. and Johnson D.B., "Ariadne: A Secure On-demand Routing Protocol for AdHoc Networks", In Proc. of the 8th International Conference on Mobile Computing and Networks, pp. 12-23, 2002.
36) K. Sanzgiri et al., "A Secure routing Protocol for Ad Hoc Networks", In Proc. of the 10th IEEE Conference on Network Protocols, 2002.
37) B. Awerbuch et al, "An On Demand Secure Routing Protocol Resilient to ByzantineFailures", In Proc. of the ACM Workshop on Wireless Security, 2002.
38) Y. Hu, D. Johnson, and A. Perrig, “Sead: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks,” IEEE WMCSA, 2002.
39) M. Zapata, and N. Asokan, “Securing Ad Hoc Routing Protocols,” ACM WiSe, 2002.
40) W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, IT-22(6):644-654, 1976.
41) M. Steiner, Tsudik G., and Waidner M., "Diffie-Hellman Key Distribution Extended to Group Communication", In Proc of the ACM Conference on Computer and Communication Security, pp. 31-37, 1996.
42) Bellovin S.M., Merritt M., "Encrypted Key Exchange:Password-based Protocols Secure Against Dictionary Attacks", In IEEE Symposium on Security and Privacy, pp. 72-84, 1992.
43) Zhou L., Haas Z.J., "Securing Ad Hoc Networks", IEEE Network 13(6), pp. 24-30, 1999.
44) Yi S., R. Kravets. "MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks", In the 2nd Annual PKI Research Workshop, 2003.
45) Capkun S, Buttyan L, Hubhaux IR "Self organized public key management for mobile adhoc network", IEEE transaction on mobile computing 2003, 52-64
46) Li Y. and Wei J., “Guidelines on Selecting Intrusion Detection Methods in MANET”, In Proc. Of Information Systems Educators Conference, 2004.
47) Tseng C. Y., Balasubramayan P., Ko C., Limprasittiporn R., Rowe J. And Levitt K., “A Specification-Based Intrusion Detection System for AODV”, In Proc. of the ACM Workshop on Security in AdHoc and Sensor Networks, 2003.
48) Huang Y. and Lee W., “A Cooperative Intrusion Detection System for Ad Hoc Networks”, In Proc. Of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, 2003.
49) Sen S. and Clark J.A., “A Grammatical Evolution Approach to Intrusion Detection on Mobile Ad Hoc Networks”, In Proc. of the 2nd ACM Conference on Wireless Network Security, pp. 95-102, 2009.
50) Marti S., Giuli T.J., Lai K.. and Baker M., “Mitigating Routing Misbehaviour in Mobile Ad Hoc Networks”, In Proc. of ACM Int. Conf. on Mobile Computing and Networking, MOBICOM, pp. 255-265, 2000.
51) Parker J., Undercoffer J., Pinkston J. and Joshi A., “On Intrusion Detection and Response for Mobile Ad Hoc Networks”, In Proc. of 23rd IEEE Int. Performance Computing and Communications Conference, 2004.
52) A A Pirzada, A Datta and C McDonald, “Trust Based Routing for ad hoc wireless networks”, International Conference on Networks, 2004.
53) A A Pirzada and C. Mcdonald , “Trust establishment in pure adhoc network”, wireless personal communication, 37(1-2), 139-168, 2006
54) Vinesh H Patel and M. A. Zaveri, “Trust based Routing in Moblie Ad hoc Networks”, Lecture Notes on Software Engineering, 3(4), 318, 2015.

55) W. T. Luke Teacy, Jigar patel, Nicholas Jennings, and Michael Luck, “TRAVOS: Trust and Reputation in the Context of Inaccurate Information Sources. Autonomous Agents and Multi-Agent Systems”, 12(2), 183-188, 2006.
56) Xiaoqi Li, M T Lyu, and J Liu, “A trust model based Routing protocol for Secure Ad hoc networks”. IEEE Aerospace Conference Proceedings, 2004.
57) Zia, Tanveer A., “Reputation based trust management in wireless sensor network”, international conference on Intelligent Sensor, Sensor network and Information Processing, 2008.
58) Sun, M Denko and Tao, “Probabilistic Trust management in pervasive computing”, international conference on embedded and ubiquitous computing, 2008.
59) Sonja Buchegger and Jans Yves Le Boudec, “A robust reputation system for mobile ad hoc networks”, EPFL-IC-LCA technical report IC/2003/50, 2003.
60) R A Shaikh, H Jameel, S Lee, S Rajput and YJ Song, “A trust management problem in distributed wireless sensor networks”, IEEE International conference on Embedded and Real time computing, 2006.
61) Riaz Ahmed Shaikh, Hassan Jameel, Brian J d'Auriol, Heejo Lee, and Sunfyoung Lee, “Group Based Trust Management Scheme for clustered wireless Sensor Networks”, IEEE transaction on Parallel and Distributed Systems, 20(11), 1698-1712, 2009.
62) Mohamed M E A Mahmooud, X Lin and X Shen, “Secure and Reliable routing protocols for heterogeneous Multihop wireless networks”, IEEE Transaction on parallel and distributed system, 1-11, 2013.
63) I R Chen, J Guo, F Bao, and J H Cho, “Trust management in mobile ad hoc network for bias minimization and application performance maximization”, Ad hoc networks, 19, 59-74, 2014.
64) Guy Guemkam, Djamel Khadraoui, Menjamin Gateau, and Zahia Guessoum, “ARMAN: Agent based Reputation for mobile adhoc networks”, Springer-Verlag Berlin Heidelberg 2013, LNAI 7879(PAAMS 2013), 122-132, 2013.
65) Gohil Bhumika, M A Zaveri, and Hemant kumar Rath, “Trust based service discovery in mobile ad hoc networks”, Lecture notes on Software engineering, 3(4), 308, 2015.
66) Datta, N Marchang and R, “Light weight trust based routing protocol for secure ad hoc networks” Information Security, 6(2), 77-83, 2012.
67) Zheng Yan and Silke Holtmanns, “Trust Modeling and Management: from Social Trust to Digital Trust”, book chapter of Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions, IGI Global, 2007
68) Tanja Azderska, “Coevolving trust mechanisms for catering user behaviour”, 6th IFIP WG 11.11, IFIPTM 2012
69) T. Grandison & M. Sloman, "A survey of trust in internet applications", IEEE Communications and Survey, 4th Quarter, 3(4), 2–16, 2000.
70) C.L.Corritore, B. Kracher, & S. Wiedenbeck, "On-line trust: concepts, evolving themes, a model" International Journal of Human-Computer Studies, Trust and Technology, 58(6), 737–758, 2003.
71) L. Mui, "Computational models of trust and reputation: agents, evolutionary games, and social networks", Doctoral dissertation, Massachusetts Institute of Technology, 2003.
72) A. Avizienis, J.C.Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing", IEEE Transactions on Dependable and Secure Computing, vol. 1, Issue 1, pp. 11–33, January 2004.
73) Z. Liu, A.W.Joy, & R. A. Thompson, "A dynamic trust model for mobile ad hoc networks", Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS 2004), pp. 80–85,May 2004.
74) Audun Jøsang, "Trust and reputation system", Purdue University, IFIPTM 2009.
75) Kannan Govindan and Prasant Mohapatra, "Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey", IEEE Communications Surveys & Tutorials, Volume: 14, Issue: 2, pages: 279 - 298, Second Quarter 2012.
76) Audun Josang, "Trust Transitivity and Conditional Belief Reasoning". IFIPTM 2012. Surat.
77) Z. N. Ulieru, "The State of the Art in Trust and Reputation Systems: A Framework for Comparison." Journal of Theoretical and Applied Electronic Commerce Research, 5(2), 97-117, 2010.
78) Audun Jøsang", The Beta Reputation System", 15th Bled Electronic Commerce Conference. Bled, Slovenia, 2002.
79) Ben-Jye Chang, Szu-Liang Kuo and Ying-Hsin Liang, "Markov Chain-Based Trust Model for Analyzing Trust Value in Distributed Multicasting Mobile Ad Hoc Networks", Asia-Pacific Services Computing Conference, 2008.

80) Abdou, Abdel Rahman, Matrawy, Ashraf, van Oorschot, and Paul, "Accurate One-Way Delay Estimation with Reduced Client-Trustworthiness", IEEE Communications Letters doi:10.1109/LCOMM.2015. 241 15 91 May 2015
81) C. Cheng, R. Riley, S. P. R. Kumar, and J. J. Garcia-Luna-Aceves, "A loop free bellman-ford routing protocol without bouncing effect ect. In Proceedings of ACM SIGCOMM '89, pages 224{237, September 1989.
82) Ehab M. ElSalamouny, "Probabilistic trust models in network security", A thesis submitted at UNIVERSITY OF SOUTHAMPTON, march 2011.
83) P. B. Velloso, R. P. Laufer, O. C. M. B. Duarte, G. Pujolle, "A trust model robust to slander attacks in ad hoc networks", IEEE International Conference on Computer Communications and Networks (ICCCN08) Workshop, 2008
84) S Anipakala, "Performance Analysis of Ad hoc On-demand Distance Vector routing (AODV) using OPNET Simulator", University of Bremen, 2004
85) OPNET 11.0 Documentation and tutorials.
86) Johnson,N.L., Kotz,S., and Balakrishnan,N, "Beta Distributions In: Continuous Univariate Distributions", 2 edn. Volume 2. Wiley (1995)
87) Kari Sentz and Scott Ferson, "Combination of Evidence in Dempster–Shafer Theory", Sandia National Laboratories, SAND 2002.
88) Gayathri Dhananjayan and Janakiraman Subbiah, "T2AR: trust-aware ad-hoc routing protocol for MANET", Springerplus PMC4937011, 5(1): 995, July, 2016.
89) S. Sridhar1, R. Baskaran, R. Anitha and R. Sankar, "Proficient and Secured Routing in MANET Based on Trust and Energy Supported AODV", Applied Mathematics and Information Science 11, No. 3, 807-817, 2017.

List of Publications

1) Kajal S Patel and Dr J S Shah, "Study the effect of packet drop attack in AODV routing and MANET and detection of such node in MANET", ICT4SD 2015 Volume 1 pp 135-142 10.1007/978-981-10-0129-1_15 ISBN(print) 978-981-10-0127-7, Springer ASIC, July 2015

2) Kajal S Patel and Dr J S Shah, "Detection and avoidance of malicious node in MANET", IEEE International Conference on Computer, Communication and Control, MGI Indore, INDIA. September 10 -12, 2015 (IEEE Xplore)

3) Kajal S Patel and Dr J S Shah, "Analysis of existing Trust based Routing schemes used in Wireless Network", International Journal of Information Security and Privacy (IJISP) IGI Global Volume 10 , Issue 2 , pg 26-40, April-June 2016 (Manuscript added in appendix B)

4) Kajal S. Patel, "Trust based Routing to avoid malicious nodes in MANET ", International Journal of Control Theory and Applications 9(21), 2016, pp. 105-110, September, 2016. (Manuscript added in appendix B)

Appendix A

In **my_aodv_route_table.c** file functions related to route table entry and route table access are available. We have created same functions for trust table. **my_aodv_route_table.c** file is a copy of **aodv_route_table.c** file, which is available in implementation of standard AODV routing in OPNET.

```
/* Function implemented for accessing  Trust table  */
static AodvT_Trust_Entry*                   my_aodv_trust_table_entry_mem_alloc (void);
static void                                 my_aodv_trust_table_entry_mem_free (AodvT_Trust_Entry*);
AodvT_Trust_Table*
my_aodv_trust_table_create (IpT_Cmn_Rte_Table* cmn_rte_table_ptr, IpT_Rte_Proc_Id proto_id,
AodvT_Local_Stathandles* local_stat_ptr)
        {
        AodvT_Trust_Table*              trust_table_ptr;
        /** Creates and allocates memory for        **/
        /** the AODV trust table                            **/
        FIN (my_aodv_trust_table_create (void));
        trust_table_ptr = (AodvT_Trust_Table*) op_prg_mem_alloc (sizeof (AodvT_Trust_Table));
        trust_table_ptr->trust_table = (PrgT_String_Hash_Table*) prg_string_hash_table_create (100, 15);
        trust_table_ptr->ip_cmn_rte_table_ptr = cmn_rte_table_ptr;
        trust_table_ptr->aodv_protocol_id = proto_id;
        trust_table_ptr->stat_handles_ptr = local_stat_ptr;
        trust_table_ptr->current_size = 0;
        FRET (trust_table_ptr);
        }
void
my_aodv_trust_table_entry_create
(AodvT_Trust_Table* trust_table_ptr, InetT_Address node_addr,InetT_Subnet_Mask subnet_mask,int
pkin,int pksfw, int pkdl, int pkerr)
        {
        AodvT_Trust_Entry*                  trust_entry_ptr;
        char                                node_addr_str [INETC_ADDR_STR_LEN];
        void*                               old_contents_ptr;
        InetT_Address*                      node_addr_ptr;
        /** Adds a new trust table entry in the trust table                         **/
        FIN (my_aodv_trust_table_entry_create (<args>));
        /* Create the node address string     */
        inet_address_print (node_addr_str, node_addr);
        node_addr_ptr = inet_address_create_dynamic (node_addr);
        /* Allocate memory for the trust entry         */
        trust_entry_ptr = my_aodv_trust_table_entry_mem_alloc ();
        trust_entry_ptr->node_prefix = ip_cmn_rte_table_dest_prefix_create (node_addr, subnet_mask);
        trust_entry_ptr->node_addr = node_addr;
        trust_entry_ptr->pin=pkin;
        trust_entry_ptr->psfwt=pksfw;
        trust_entry_ptr->pdl=pkdl;
        trust_entry_ptr->perr=pkerr;
        /* Set the trust entry for this neighbour node  in the trust table             */
   prg_string_hash_table_item_insert (trust_table_ptr->trust_table, node_addr_str, trust_entry_ptr,
&old_contents_ptr);
        /* Update the size of the trust table  */
        trust_table_ptr->current_size++;
        FOUT;
        }
AodvT_Trust_Entry*
my_aodv_trust_table_entry_get (AodvT_Trust_Table* trust_table_ptr, InetT_Address node_addr)
        {
        AodvT_Trust_Entry*                  trust_entry_ptr = OPC_NIL;
        char                                node_addr_str [INETC_ADDR_STR_LEN];
```

```
	/** Determines whether an entry exists	**/
	/** in the trust table for a node**/
	FIN (my_aodv_trust_table_entry_get (<args>));
	/* Create the node address string	*/
	inet_address_print (node_addr_str, node_addr);
	/* Get the entry for this neighbour node	*/
	trust_entry_ptr = (AodvT_Trust_Entry*) prg_string_hash_table_item_get (trust_table_ptr-
>trust_table, node_addr_str);
	if(trust_entry_ptr == OPC_NIL)
		{
		FRET (OPC_NIL);
		}
	else
		{
		FRET(trust_entry_ptr);
		}
	}
Compcode
my_aodv_trust_table_entry_param_set (AodvT_Trust_Table* trust_table_ptr, InetT_Address node_addr, int
param , ...)
	{
	AodvT_Trust_Entry*			trust_entry_ptr = OPC_NIL;
	char					node_addr_str [INETC_ADDR_STR_LEN];
	va_list					arg_list;
	/** Set the fields of the trust table entry	**/
	FIN (my_aodv_trust_table_entry_param_set (<args>));
	/* Create the node address string	*/
	inet_address_print (node_addr_str, node_addr);
	/* Get the entry for this neighbour node	*/
	trust_entry_ptr = (AodvT_Trust_Entry*) prg_string_hash_table_item_get (trust_table_ptr-
>trust_table, node_addr_str);
	if (trust_entry_ptr == OPC_NIL)
		FRET (OPC_COMPCODE_FAILURE);
   /* Initialize the list of arguments. Though a list arguments may	*/
   /* not always be passed this approach will help us identify the		*/
   /* data type of the parameters and appropriately cast it.		*/
   va_start (arg_list, param);
	/* Based on the input parameter, set*/
	/* the appropriate parameter			*/
	switch (param)
		{
		case (AODVC_TRUST_ENTRY_IN_PKT):
			{
			trust_entry_ptr->pin = va_arg (arg_list, int);
			break;
			}
		case (AODVC_TRUST_ENTRY_SFWD_PKT):
			{
			trust_entry_ptr->psfwt = va_arg (arg_list, int);
			break;
			}
		case (AODVC_TRUST_ENTRY_DL_PKT):
			{
			trust_entry_ptr->pdl = va_arg (arg_list, int);
			break;
			}
		case (AODVC_TRUST_ENTRY_ERR_PKT):
			{
			trust_entry_ptr->perr = va_arg (arg_list, int);
			break;
```

```
                }
        default :
                {
                /* Unknown input parameter          */
                FRET (OPC_COMPCODE_FAILURE);
                }
        }
    FRET (OPC_COMPCODE_SUCCESS);
    }

static AodvT_Trust_Entry*
my_aodv_trust_table_entry_mem_alloc (void)
    {
    static Pmohandle            trust_table_entry_pmh;
    AodvT_Trust_Entry*          trust_table_entry_ptr = OPC_NIL;
    static Boolean              trust_table_entry_pmh_defined = OPC_FALSE;
    /** Allocates pooled memory for a trust table entry     **/
    FIN (my_aodv_trust_table_entry_mem_alloc (void));
    if (trust_table_entry_pmh_defined == OPC_FALSE)
        {
        /* Define the pool memory handle for trust table entry          */
trust_table_entry_pmh = op_prg_pmo_define ("Route Table Entry", sizeof (AodvT_Trust_Entry), 32);
        trust_table_entry_pmh_defined = OPC_TRUE;
        }
    /* Allocate the trust table entry from the pooled memory        */
    trust_table_entry_ptr = (AodvT_Trust_Entry*) op_prg_pmo_alloc (trust_table_entry_pmh);
    FRET (trust_table_entry_ptr);
    }
static void my_aodv_trust_table_entry_mem_free (AodvT_Trust_Entry* trust_entry_ptr)
    {
    /** Frees the memory for the trust table entry          **/
    FIN (my_aodv_trust_table_entry_mem_free (<args>));
    inet_address_destroy (trust_entry_ptr->node_addr);
    ip_cmn_rte_table_dest_prefix_destroy (trust_entry_ptr->node_prefix);
    /* Free the trust entry     */
    op_prg_mem_free (trust_entry_ptr);
    FOUT;   }
```

Appendix B

Manuscript of following two papers

1) Kajal S Patel and Dr J S Shah, "Analysis of existing Trust based Routing schemes used in Wireless Network", International Journal of Information Security and Privacy (IJISP) IGI Global Volume 10 , Issue 2 , pg 26-40, April-June 2016 (Manuscript added in appendix B)

2) Kajal S. Patel, "Trust based Routing to avoid malicious nodes in MANET ", International Journal of Control Theory and Applications 9(21), 2016, pp. 105-110, September, 2016. (Manuscript added in appendix B)