A SCALABLE AND SECURE POSITION-BASED ROUTING PROTOCOL FOR AD-HOC NETWORKS

LIANA K. M. QABAJEH

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2012

A SCALABLE AND SECURE POSITION-BASED ROUTING PROTOCOL FOR AD-HOC NETWORKS

LIANA K. M. QABAJEH

THESIS SUBMITTED IN FULFILMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2012

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: LIANA K. M. QABAJEH Passport No: 2532485

Matric No: WHA080010

Name of Degree: Doctor of Philosophy

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):
A Scalable and Secure Position-Based Routing Protocol for Ad-Hoc Networks.

Field of Study: Computer Science and Information Technology, Wireless Networks

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;

(2) This Work is original;

(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;

(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;

(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;

(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date

Subscribed and solemnly declared before,

Witness's Signature Date

Name: Assoc. Prof. Dr. Miss Laiha Mat Kiah

Designation:

ABSTRACT

*Mobile Ad-Hoc NETworks (MANET*s*)* are wireless multi-hop networks formed by a set of mobile nodes in a self-organizing way without requiring already established infrastructure. Along with their traditional uses such as disaster situations and military battlefields, *MANET*s are being increasingly used in daily applications such as conferences, personal area networking and meetings.

Routing in *MANET*s is a challengeable task due to limited bandwidth of wireless links, highly dynamic topology, limited radio transmission range and limited nodes' energy. Though security issues could arise in numerous areas in *MANET*s such as physical security, key management and intrusion detection, routing is considered as one of the most difficult areas to protect against attacks. This is due to lack of centralized control, open medium, distributed cooperation, dynamic topology as well as limited capability of nodes.

In this research, we tackle security issues related to Ad-Hoc routing protocols. A new model of hierarchal and distributed routing protocol called *ARANz* has been proposed in this work. *ARAN*z aims to improve performance of the routing protocol and distribute routing load by dividing the area into zones. It seeks to achieve a high level of security and attain robustness by avoiding the single point of attack problem and solving the problem of single point of failure as a result of distributing trust among multiple certificate authority servers.

ARANz aspires to exhibit better scalability and performance by taking advantage of the restricted directional flooding position-based routing protocols. Thus, in conjunction with the chosen routing strategy, a distributed location service has been proposed. Along with the proposed protocol a misbehaviour detection system is proposed to help in identifying malicious nodes.

The performance of *ARANz* is compared to other existing routing protocols and tested using the *Global Mobile information systems Simulator (GloMoSim)*. From the results analysis, it shows that *ARANz* is highly effective in discovering and maintaining routes even with relatively high node mobility and large percentage of malicious nodes. It is also demonstrated that the proposed protocol performs efficiently in large area networks.

ABSTRAK

Rangkaian Bergerak Adhoc (Mobile Ad-Hoc Networks (MANETs)) adalah rangkaian multihop wayarles yang dibentuk oleh suatu kumpulan nod bergerak tersendiri tanpa memerlukan sebarang infrastruktur yang sedia-ada. Seiring dengan kegunaan tradisional mereka iaitu seperti situasi bencana dan medan peperangan, kegunaan MANETs semakin banyak dilihat dalam aktiviti-aktiviti harian seperti semasa persidangan, rangkaian perseorangan dan semasa mesyuarat.

Penghalaan dalam *MANET*s merupakan aktiviti yang mencabar disebabkan oleh lebar jalur terhad dari sambungan wayarlesnya, topologi yang sangat dinamik, liputan pemancar radio dan tenaga nod-nod yang terhad. Walaupun masalah-masalah keselamatan *MANET*s boleh wujud dari pelbagai sudut lain seperti keselamatan fizikal, pengurusan kunci, dan pengesanan pencerobohan, penghalaan adalah merupakan salah satu masalah yang paling sukar untuk dilindungi daripada serangan keselamatan. Ini adalah disebabkan kurangnya kawalan berpusat, media yang terbuka, kerjasama yang teragih, topologi yang dinamik serta kemampuan terhad nod-nod.

Dalam kajian ini, kami menangani masalah-masalah keselamatan yang berkaitan dengan protokol penghalaan Ad-Hoc. Sebuah model baru protokol penghalaan yang berhierarki dan teragih dipanggil *ARAN*z telah dicadangkan. *ARAN*z bertujuan untuk meningkatkan prestasi protokol penghalaan tersebut dan mengagihkan beban penghalaan dengan membahagikan kawasan ke zon-zon. Ianya berusaha untuk mencapai tahap keselamatan yang tinggi dan memperoleh ketahanan dengan mengelak dari masalah serangan titik tunggal dan menyelesaikan masalah kegagalan titik tunggal dengan cara mengagihkan kepercayaan di antara pelayan autoriti sijil yang berbilang.

*ARAN*z berhasrat untuk menunjukkan kebolehan skala, prestasi dan ketahanan yang lebih baik terhadap perubahan topologi yang kerap, dengan mengambilpakai kebaikan protokol penghalaan terhad berarah kebanjiran berasaskan-kedudukan. Dari itu, dengan strategi penghalaan yang dipilih, satu perkhidmatan lokasi teragih telah dicadangkan. Bersama dengan protokol yang dicadangkan, suatu sistem pengesanan kelakukan tidak baik telah dicadangkan untuk membantu dalam mengenalpasti nod-nod tidak baik.

Kemampuan *ARAN*z telah dibandingkan dengan protokol-protokol penghalaan lain dan telah diuji menggunakan simulator *Global Mobile information systems Simulator (GloMoSim)*. Daripada hasil analisis, ia menunjukkan bahawa *ARAN*z sangat efektif dalam mencari dan mempertahankan laluan walaupun dalam keadaan mobiliti nod yang tinggi dengan peratusan nod tidak baik yang besar. Hasil kajian juga menunjukkan bahawa protokol yang dicadangkan memberikan prestasi yang cekap dalam rangkaian luas.

ACKNOWLEDGEMENTS

First and foremost, Alhamdulillah, praise to Allah the Almighty who has bestowed upon me and given me good health and time for going through this challenging period of my life.

I would like to express my deepest appreciation to my supervisor, Dr. Miss Laiha Mat Kiah who had the foresight to accept me as her doctoral student, believed in my abilities, kept advising me to improve my knowledge and writing, and supported me at the crucial moments of my study at the University of Malaya. Her patience, kindness, understanding, encouragement, careful reading of the manuscript and especially the constructive criticisms and suggestions make the process of conducting this research run smoothly. I have benefited from her experience, advice and the intellectual freedom she afforded to me during the research period.

Very special and sincere thanks to Prof. Steven Furnell (University of Plymouth, United Kingdom), Prof. Abdul Hanan Abdullah (Technological University of Malaysia, Malaysia) and Dr. Mazliza Othman (University of Malaya, Malaysia) for examining my thesis. I wish you all the best and sincerely appreciate the time and effort you have spent.

Special thanks to Prof. Dr. Siti Salwah Binti Salim, Assoc. Prof. Dr. Ling Teck Chaw, Assoc. Prof. Dr. Abdullah Gani, Assoc. Prof. Dr. Phang Keat Keong, Dr. Rafidah Binti Md Noor, Dr. Rashid Hafeez Khokhar and Assoc. Prof. Dr. Diljit Singh (University of Malaya, Malaysia) for essential feedback, comments and suggestions to improve my research. Many thanks go to Dr. Kimaya Mittal (Santa Barbara, United States) for her help and support during my work with the *ARAN* routing protocol.

My thanks also goes to the *International Journal of Computer Science and Network*, *Malaysian Journal of Computer Science*, *Acta Polytechnica Hungarica* (*Journal of Applied Sciences*) and *Security and Communication Networks* Journals' editorial board and reviewers, for their detailed and fruitful comments over the years on various aspects of my research. I was simultaneously impressed and humbled by their expertise.

I would also like to thank the University of Malaya's staff (especially in the main library) for providing impeccable assistance and academic resources.

I am very thankful to the Islamic Development Bank for the financial support and Palestine Polytechnic University for the study leave that has enabled me to pursue my academic dreams.

Last but not least, my appreciation is also to those who were directly or indirectly involved in this research. These include my mother and father, my sisters and brothers, and my husband's family. Special thanks and apologizes to my loving husband Mohammad and my wonderful children Noor Al-Din, Saif Al-Din, Zain Al-Din and Minat Allah whom I know that I was sometimes unable to give them enough care and attention, aiming that they will understand this and forgive me. May Allah bless you all for your kindness, support, encouragement and willingness to help in completing this research.

To many others who have helped me in one way or other during my course of study but whom I may have inadvertently left out, please kindly excuse me.

This research is supported by VotF University Malaya fund no. FS132/2008C, PPP University Malaya fund no. PS091/2009A and PPP University Malaya fund no. PS410/2010B.

Having accomplished this research work, I hope that I have succeeded in adding to the field of mobile Ad-Hoc networks and now I am looking forward to having my spare time back to myself again.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

General

API	Application Programming Interface
BS	Base Station
CA	Certificate Authority
CBR	Constant Bit Rate
CK	Common Key
CPU	Central Processing Unit
DFD	Dynamic Forwarding Delay
GLONASS	GLObal NAvigation Satellite System
GLS	Grid Location Service
GPS	Global Positioning System
GUI	Graphical User Interface
IP address	Internet Protocol address
km	Kilometer
LRS	Local Reputation System
m	Meter
MAC	Message Authentication Code
MAC address	Media Access Control address
MAC layer	Media Access Control layer
MANET	Mobile Ad-Hoc NETwork
ms	Millisecond
NH	Number of Hop
OSI	Open Systems Interconnection model
OTCL	Object Tool Command Language
PAN	Personal Area Network
PDA	Personal Digital Assistant
PS	Position Server
QoS	Quality of Service
RREP	Route REPly
RREQ	Route REQuest
s	Second
SGLS	Secure Grid Location Service
TESLA	Timed Efficient Stream Loss-tolerant Authentication
TIK	Instant Key disclosure
UDP	User Datagram Protocol
VHR	Virtual Home Region

Simulation tools

GloMoSim	Global Mobile information systems Simulator
NS-2	Network Simulator 2
OPNet	OPtimized Network engineering tools
PARSEC	PARallel Simulation Environment for Complex systems

Routing protocols

AODPR	Anonymous On-Demand Position-based Routing in mobile Ad-Hoc networks
AODV	Ad-Hoc On-demand Distance Vector
ARAN	Authenticated Routing for Ad-Hoc Networks
ARANz	Zone-based Authenticated Routing for Ad-Hoc Networks
ARP	Angular Routing Protocol

BRP	Bordercast Resolution Protocol
CGSR	Clusterhead Gateway Switch Routing
DIR	Compass routing algorithms
DSDV	Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing
FSR	Fisheye State Routing
GPSR	Greedy Perimeter Stateless Routing
IARP	IntrA-zone Routing Protocol
IERP	IntEr-zone Routing Protocol
I-PBBLR	Improved Progress Position-Based BeaconLess Routing algorithm
LAR	Location-Aided Routing
LARWB	Location-Aided Routing With Backup
LASR	Location-Aided Secure Routing scheme
LSR	Location Secure Routing protocol
MDSR	Multipath Dynamic Source Routing
MFR	Most Forward within distance *R*
OLSR	Optimized Link State Routing Protocol
SAODV	Secure Ad-Hoc On-demand Distance Vector
SAR	Security-aware Ad-Hoc Routing
SEAD	Secure Efficient Ad-Hoc Distance vector routing protocol
SGF	Secure Geographic Forwarding
SLSP	Secure Link State Protocol
SPAAR	Secure Position-Aided Ad-Hoc Routing
SRP	Secure Routing Protocol
STAR	Source Tree Adaptive Routing
TORA	Temporally Ordered Routing Algorithm
WRP	Wireless Routing Protocol
ZRP	Zone Routing Protocol

Variables and notations for *ARAN* protocol

$[d]K_{A-}$	Data *d* digitally signed by node *A*
$\{d\}K_{A+}$	Data *d* encrypted with key K_{A+}
$Cert_A$	Node *A* Certificate
e	Certificate expiration time
ERR	ERRor packet identifier
IP_A	*IP* address of node *A*
K_{A-}	Private key of node *A*
K_{A+}	Public key of node *A*
K_{CA-}	Private key of the trusted *CA*
K_{CA+}	Public key of the trusted *CA*
N_A	Nonce issued by node *A*
RDP	Route Discovery Packet identifier
REP	REPly packet identifier
t	Timestamp

Variables and notations for *ARANz* protocol

$[d]K_{n-}$	Data *d* digitally signed by node *n*
$\{d\}K_{n+}$	Data *d* encrypted with key K_{n+}
$8NbrZ_Z$	Numbers and coordinates of 8-Neighbouring zones of zone *z*
$AdjL_{zs}$	*IP* address and position of adjacent *LCA* of LCA_{zs}
$AdjZ_{zs}$	Number of the zone adjacent to boundary *s* of zone *z*

ALL	All nodes existing currently in the network
ALL_z	All nodes existing currently in zone z
An	Area of the network
AN	Absent nodes, *IP* addresses and public keys of authorized nodes that were not in the network during network setup
AT	Authentication table
Az	Area of each zone
B_n	Remaining battery life time of node n
$CertLZ_z$	*LCA*s certificate of zone z
$Cert_n$	Node n certificate
CK	Common key
C_n	*CPU* power of node n
$CoorZ_z$	Coordinates of zone z
$Dist$	Distance between an intermediate node and the destination node
$Dmov$	Distance that a node moves before informing its zone *LCA*s about its new position
D_{nzs}	Distance between position of node n existing in zone z and the middle point of the zone boundary s
$Dsid$	Distance that a *LCA* is allowed to be from the zone boundary middle point prior to initiating a new *LCA* election
e	Certificate expiration time
$\xrightarrow{Fln}$	Flood packet to the entire network
$\xrightarrow{Flz}$	Flood packet to a particular zone
IP_n	*IP* address of node n
K_{n-}	Private key of node n
K_{n+}	Public key of node n
K_{NET-}	Private key of the network
K_{NET+}	Public key of the network
K_{Zz-}	Private key of zone z
K_{Zz+}	Public key of zone z
LCA	Local Certificate Authority
$LCAF_n$	Fraction of time during which node n served as a *LCA* in the last Ns time slots
$LCAsZ_z$	*IP* addresses and positions of *LCA*s in zone z
LCA_{zs}	*LCA* responsible for boundary s of zone z
Ln	Length of the network boundary
M_n	Memory capacity of node n
Nn	Number of nodes
N_n	Nonce issued by node n
Ns	Number of time slots during which the role that a node plays is recorded
Nz	Number of zones
NZ_z	Nonce issued by Zone z
PCA	Primary Certificate Authority
Pid	Packet type identifier
P_n	Position of node n
$\xrightarrow{Rdf}$	Send packet using restricted directional flooding
$\xrightarrow{Rev}$	Send packet through reverse path
$\xrightarrow{Rly}$	Relay data packet to its destination
$Role_n$	Node n current role
$Roles_n[Ns]$	An array specifying the roles (*LCA* or regular) that node n played during each of the last Ns time slots

S_n	Movement speed of node n
SR	Source route that a packet will go through
$\xrightarrow{Src}$	Send packet using source routing
t	Timestamp
Tac	Acceptance of certificate time
Tcu	Certificate update time
Tic	Information collection time
Tkc	Private key collection time
Tls	*LCA* synchronization time
Tpc	Probability collection time
Tpd	Position discovery time
TR	Nodes’ transmission range
Trd	Route discovery time
Tsl	Serving as a *LCA* time
Xc_{zc}	X-coordinate of corner c of zone z
Xm_{zs}	X-coordinate of middle point of boundary s of zone z
X_n	X-coordinate of node n
Yc_{zc}	Y-coordinate of corner c of zone z
Ym_{zs}	Y-coordinate of middle point of boundary s of zone z
Y_n	Y-coordinate of node n
$Zone_n$	Node n current zone

Variables and notations for the proposed *LCA*s election algorithm

$Bmax$	Maximum possible node battery life time
$Cmax$	Maximum node *CPU* power
$Dmax$	Maximum possible distance between a node and middle point of a zone boundary
$Mmax$	Maximum node memory capacity
$ProbL_{nzs}$	Probability of node n existing in zone z to be elected as a *LCA* of boundary s
$Smax$	Maximum possible node movement speed
Wb	Weight of node battery remaining life upon electing a new *LCA*
Wc	Weight of node *CPU* power upon electing a new *LCA*
Wd	Weight of distance between a node and middle point of a zone boundary upon electing a new *LCA*
Wf	Weight of fraction of time during which node served as a *LCA* upon electing a new *LCA*
Wm	Weight of node memory capacity upon electing a new *LCA*
Ws	Weight of node movement speed upon electing a new *LCA*

Variables and notations for the proposed misbehaviour detection system

Fd_{nm}	Number of dropped data packets by node m that it receives from node n
Fm_{nm}	Number of modified control packets sent from node m to node n
Nm	Number of packets received indicating the misbehaviour of a node so that this node is considered as compromised
Sd_{nm}	Number of delivered data packets by node m that it receives from node n
Sm_{nm}	Number of unmodified control packets sent from node m to node n
Thd	Dropping threshold
Thf	Fabrication threshold
Thm	Modification threshold

$TrstVd_{nm}$	Node *n* trust value regarding node *m* considering dropping attacks
$TrstVf_{nm}$	Node *n* trust value regarding node *m* considering fabrication attacks
$TrstVm_{nm}$	Node *n* trust value regarding node *m* considering modification attacks
TT	Trust table

Packet type identifiers for *ARANz* protocol

ACREP	Acceptance of Certificate REPly
ACREQ	Acceptance of Certificate REQuest
CLSYN	CLocks SYNchronization
CNODE	Compromised NODE
CREP	Certificate REPly
CREQ	Certificate REQuest
DATA	DATA packet
DNODE	Departing NODE
ERR	ERRor
EZONE	Empty ZONE
FALCA	Failed Adjacent *LCA*
FLCA	Failed *LCA*
FNODE	Failed NODE
MNODE	Misbehaving NODE
NALCA	New Adjacent *LCA*
NCERT	Node CERTificate
NETSET	NETwork SETup
NIN	Node INformation
NLCA	New *LCA*
NLCAE	New *LCA* Election
NNODE	New NODE
NPROB	Node PROBability
NROLE	Node ROLE
NZONE	New ZONE
PDP	Position Discovery Packet
PKPREP	Zone Private Key Part REPly
PKPREQ	Zone Private Key Part REQuest
PREP	Position REPly
RDP	Route Discovery Packet
RREP	Route REPly
SNODE	Sole NODE
UALPOS	Update Adjacent *LCA* POSition
ULPOS	Update *LCA* POSition
UNPOS	Update Node POSition

Performance metrics

AEED	Average End-to-End Delay of Data Packets
APNH	Average Path Number of Hops
ARAL	Average Route Acquisition Latency
BML	Byte Malicious Load
BNL	Byte Network Load
BRL	Byte Routing Load
CNP	Compromised Node Percentage
FEP	Fabricated Error Packets
MRP	Malicious Route Percentage
PDF	Packet Delivery Fraction

PLP	Packet Loss Percentage
PML	Packet Malicious Load
PNL	Packet Network Load
PRL	Packet Routing Load

LIST OF APPENDICES

Chapter 1

Introduction

This chapter introduces the direction of our work along with the motivation that drives us into carrying out this research. In *Section 1.1*, we introduce this work and give general overview of the thesis. Our problem statement, research significance and research objectives are discussed in *Section 1.2* through *Section 1.4*. *Sections 1.5* and *1.6* identify the proposed outcomes and research scope. Finally, in *Section 1.7* we briefly outline the main structure of the thesis.

1.1 Thesis Overview

Ad-Hoc wireless networks are self-organizing multi-hop wireless networks, where all hosts (or nodes) take part in the process of forwarding packets. Ad-Hoc networks can quickly and inexpensively be set up as needed since they do not require any fixed infrastructure, such as base stations or routers. Therefore, they are highly applicable in many fields such as emergency deployments, for instance conferences and meetings, and community networking for earthquake or other natural disasters.

A key component of Ad-Hoc wireless network is an efficient routing protocol since all nodes in the network act as routers (Prakash et al. 2011). Ad-Hoc network routing protocols are difficult to design in general. There are two main reasons for this: the highly dynamic nature of Ad-Hoc networks due to the high mobility of nodes and the need to operate efficiently with limited network bandwidth along with the limited nodes' resources, such as processing capacity, memory and battery power (energy).

The concept and structure of Ad-Hoc networks make these networks prone to security attacks via modification of routing information, fabricating false routing information and impersonating as other nodes. Security concerns arise in various areas, such as physical security, key management, routing and intrusion detection. These issues are vital in some applications, and thus, introduce different challenges which attract the attention of many researchers. This research focuses on the security of the routing protocol since non-secure routing protocols allow a variety of attacks, such as redirection of routing packets and falsifying route errors. Moreover, Ad-Hoc networks are based on collaborative routing, meaning that a node working in a malicious way may disrupt the entire network (Fernandes & Duarte 2010).

In Ad-Hoc networks, managed-open environment is the one where most research is being done today, as it is the type of environment we are most likely to see expanding in the nearest future. Such environment might be formed by peers at a conference, or students on a campus. In this environment, the possibility to use already established infrastructure (to some extent) to help secure the Ad-Hoc network is available. This means that there is an opportunity for pre-deployment or exchange of public keys, session keys, or certificates, that opens up a whole new range of strategies that use certificate servers and other similar software to provide a starting point to secure the network (Sanzgiri et al. 2005).

In general, routing protocols can be divided into two main categories: topology-based and position-based. Topology-based routing protocols represent important steps in Ad-Hoc routing research area since a route discovery process is initiated only when data packets need to be routed. However, some of these protocols (such as *DSR* (Johnson & Maltz 1996) and *AODV* (Perkins & Royer 1999)) are not scalable and exhibit security vulnerabilities, and thus, can be attacked. Even secured ones (like *SAODV* (Zapata 2002), *ARIADNE* (Hu et al. 2002) and *ARAN* (Sanzgiri et al. 2005)) have some problems including single point of attack, single point of failure, high packet and processing overhead as well as delay of route discovery process.

These problems become worse if these protocols are implemented in large networks since any request packet is broadcast to all nodes in the network, consuming bandwidth.

Therefore, reducing routing overhead becomes a key issue in achieving scalability of a routing protocol. The scalability issue in wireless multi-hop routing protocols is typically concerned with excessive routing message overhead resulted from the increased number of nodes and frequent mobility (Hong et al. 2002). In other words, to increase scalability, route discovery and maintenance should be controlled, which can be achieved by localizing control message propagation to nodes close to the destination (Abolhasan et al. 2004).

Position-based or also known as geographic Ad-Hoc routing protocols have proved to achieve better routing performance than traditional Ad-Hoc routing protocols such as *DSR* (Johnson & Maltz 1996) and *AODV* (Perkins & Royer 1999) in end-to-end throughput and network scalability (Giruka & Singhal 2005; Prakash et al. 2011). However, most of them use greedy forwarding which suffers from congestion and nodes' energy consumption due to periodic beaconing. Moreover, greedy forwarding in general may not always find the optimal route especially in sparse networks (Giruka & Singhal 2005). It is found that restricted directional flooding position-based routing protocols have better performance than the greedy ones in terms of finding the shortest path (Beijar 1998). Yet, both of them are vulnerable to some attacks as their designs were done to improve some aspects of performance and not intended for security (Kalhor et al. 2007). Although some works on security in particular were found in *SPAAR* (Carter & Yasinsac 2002), *AODPR* (Mizanur Rahman et al. 2006) and *SGF* (Song et al. 2007), they still suffer from some problems, such as the single point of failure and single point of attack, higher processing overhead of packets involved as well as the scalability problem.

Nevertheless, without online trusted servers as in wired networks, it is difficult to be acquainted with the trustworthiness of each node, thus keeping away malicious nodes from the routes (Li & Singhal 2006). The approach where one centralized server is used in Ad-Hoc network is not practical as the server may be mobile, hence it could be difficult for a node to connect to the server. Hence, it is not possible to guarantee the availability of a central resource to all nodes at any time (Fernandes & Duarte 2010). Furthermore, the server could be the operation bottleneck for position management, as it may be a selected normal Ad-Hoc node with limited memory, processing capacity and battery power. Finally, using one centralized server may result in system failure if this single node is compromised or destroyed. In order to address this problem, the position service system and the certificate authority should be distributed among a number of servers deployed in the network (Giruka & Singhal 2005; Seno et al. 2011).

As aforediscussed, it is a big concern to find a scalable, distributed and secure solution particularly for position-based routing protocol for Ad-Hoc networks. A new model of routing protocol called *ARANz* has been proposed in this work. The proposed protocol is called *ARAN*z since it adopts the authentication steps in the *Authenticated Routing for Ad-Hoc Networks (ARAN)* (Sanzgiri et al. 2005) and deals with the network as zones. *ARAN*z introduces a hierarchal distributed routing algorithm, which aims to improve performance of the routing protocol and distribute load by dividing the area into zones. Moreover, it tries to achieve robustness against nodes failure and realize a high level of security via providing a solution for the single point of failure problem and avoiding single point of attack problem. This is achieved by distributing trust among multiple *Local Certificate Authority (LCA)* servers. Each zone has multiple *LCA*s that should collaborate with each other to issue certificates for the nodes inside that zone and work as backups of each others. If a misbehaviour detection scheme is present in the network, then the security of our protocol can be improved through collaboration with this scheme. Accordingly, a misbehaviour detection system has also been proposed in this work.

Finally, *ARANz* aims to exhibit better scalability and performance by taking advantage of the idea of restricted directional flooding position-based routing protocols. By using the geographical information, the nodes forward the route requests only if their position is closer to the destination's position than their previous hop, which saves network bandwidth. Consequently, in conjunction with the chosen routing strategy, a distributed location service is proposed. Since each *LCA* in *ARANz* also acts as a position server, each node must inform the *LCA*s of its zone about its new position if it has moved (not periodically).

Due to large number of nodes and large geographical area of Ad-Hoc networks, a simulation tool is used to evaluate the performance of the new protocol. It has been decided, based on the available functionality, strong focus on wireless networks, expertise of the partners and its increasing usage, to use *Global Mobile information systems Simulator (GloMoSim)* as a simulation tool. *GloMoSim* simulator is used to study the performance of the new protocol, evaluate its effectiveness in dealing with security issues and compare it with existing routing protocols.

A comprehensive evaluation study based on a wide range of simulation scenarios is conducted. From the results, we concluded that *ARANz* is highly effective in discovering and maintaining routes even with relatively high node mobility and a large percentage of malicious nodes. It is also demonstrated that the proposed protocol performs efficiently in large area networks and maintains the minimum packet routing load in all experiments. *ARANz* has also shown its effectiveness in detecting and isolating malicious nodes performing different attacks against control and data packets.

1.2 Problem Statement

An Ad-Hoc network does not have an established infrastructure. It is a self-organizing network with no pre-deployed infrastructure and no centralized control. A routing protocol that is a fundamental part of the network infrastructure supports the delivery of packets (Joshi 2011). Routing protocols in Ad-Hoc networks are difficult to implement because they have to face the challenge of link instability, node mobility, frequently changing topology, absence of a fixed infrastructure and low transmission power (Goyal et al. 2010; Zhu et al. 2011). Also as a result of the differences in nodes' transmission capacity, some of the links existing in the network may be unidirectional, which leads to the existence of asymmetric links.

All nodes in the network act as routers, hence securing a routing protocol is necessary to defend against some attacks, including modification of various fields in the routing packets (Manikandan et al. 2011). By modifying the routing information, an attacker can cause network traffic to be dropped, redirected to a different destination or to take an extended route to the destination. An attacker may also drop data packets or fabricate error packets to affect the performance of the routing protocol. A more severe attack is the impersonation attack during which a malicious node launches many attacks and misrepresent the network topology by masquerading as a legitimate node.

Routing is considered as one of the most difficult areas to guard against attacks such as the ones mentioned above due to lack of centralized control, open medium, distributed cooperation, dynamic topology and constrained capability of nodes (Goyal et al. 2010; Dutta & Dowling 2011). For these reasons, Ad-Hoc networks routing is a difficult task to achieve efficiently, robustly and securely (Sharma & Jena 2011; Dutta & Dowling 2011).

The approach where a single centralized server is used in Ad-Hoc network is impractical because the server itself is the operation bottleneck and as it may also be mobile, it can be difficult for a node to connect to it. Also, if this single node is compromised or destroyed the entire network is handicapped. To address this concern, Ad-Hoc services must be distributed among a set of servers deployed in the network (Manikandan et al. 2011; Seno et al. 2011).

Another important issue to be considered is the routing protocol scalability. Scalability is the ability of a routing protocol to perform efficiently even if some network parameters grow to be large in value (Eriksson 2006). Typical parameters that are studied for Ad-Hoc networks are nodes mobility speed and the number of nodes existing in the network (Eriksson 2006). In order to increase scalability, the route discovery and maintenance among these nodes must be controlled. This can be attained by using position-based routing protocols to help in localizing the control message propagation to a specific region around the destination (Abolhasan et al. 2004).

Some of the existing Ad-Hoc routing protocols, such as *DSR* (Johnson & Maltz 1996) and *AODV* (Perkins & Royer 1999), are not scalable and exhibit security vulnerabilities. Even the ones that claim to be secure (such as *SAODV* (Zapata 2002), *ARIADNE* (Hu et al. 2002) and *ARAN* (Sanzgiri et al. 2005)) are proposed with a centralized trust, and so, have some disadvantages among which are reduced availability and robustness due to the compromised server problem and single point of failure. Furthermore, they have scalability problems when implemented in large networks since request packet is broadcast to all nodes in the network.

Position-based routing protocols proved to perform better than the traditional topology-based Ad-Hoc routing protocols such as *DSR* (Johnson & Maltz 1996) and *AODV* (Perkins & Royer 1999) pertaining to end-to-end throughput and network scalability (Giruka & Singhal 2005). However, most use greedy forwarding technique that suffers from congestion and nodes' energy consumption due to periodic beaconing as well as no guarantee of finding the optimal route. Furthermore, many of them, such as *GPSR* (Karp & Kung 2000) and *ARP* (Giruka & Singhal 2005), are vulnerable to some attacks as they focus more on improving performance rather than dealing with security issues.

As a result, it is our concern to address these problems. In particular, we will investigate the scalability of position-based protocols and the security issues of secure routing protocols. It is also important to improve the availability and robustness of the system by having a distributed routing in order to avoid the single point of failure, compromised server and operation bottleneck problems.

Therefore, it is the aim of this research to build a scalable, distributed and secure position-based routing protocol for Ad-Hoc networks.

1.3 Research Significance

Ad-Hoc networks are one of the most active research fields in the area of wireless networking due to their numerous applications and their special characteristics discussed in *Sections 2.2.1* and *2.2.2.* Ad-Hoc networks routing, in particular, is still considered an active and important field (Jacobsson et al. 2010) due to the rapid spread of Ad-Hoc wireless networks and variety of their applications which have put the development of an efficient and secure routing protocol as one of the important issues to study (Sharma & Jena 2011). Moreover, the existence of large Ad-Hoc networks arises the importance of designing a protocol that is able to control and restrict the forwarding of the route request packets in a scalable manner (Eriksson 2006). So, it is an important concern to develop a scalable, distributed and secure routing protocol for Ad-Hoc networks.

The new proposed protocol can be implemented for actual deployments, such as in large universities, industrial factories, large or small companies and conference events, for providing wireless communications to particular users at any time, and can be the platform for further research in the respective area.

1.4 Research Objectives

The aim of this thesis is to propose, specify and evaluate a scalable and secure routing protocol for Ad-Hoc networks.

Our main research objectives are summarized as the following:

1. To study the existing Ad-Hoc routing protocols and to investigate the main problems they suffer from.
2. To design and develop a newly improved scalable and secure position-based routing protocol for Ad-Hoc networks.
3. To solve scalability problem.
4. To distribute load and trust among multiple nodes, i.e. to solve a single point of failure and attack problem.
5. To develop a simulation for the proposed protocol and test its performance.

1.5 Expected Outcomes

The expected outcomes of this research are as follows:

1. Reducing the packet routing load of the routing protocol.
2. Reducing the effect of node failure and compromise on the protocol performance.
3. Increasing protocol efficiency in distinguishing misbehaving nodes.

1.6 Research Scope

Protocol layering is important to reduce the complexity of network design. Interaction between protocol layers in *MANET*s can meet the end-to-end performance requirements. In our model, we propose a mechanism to achieve secure routing at the network layer with the assumption that any needed information is readily available from other layers.

The existing *Authenticated Routing for Ad-Hoc Networks (ARAN)* secure routing protocol is able to defend itself against most security attacks performed by malicious nodes, such as modification and fabrication of routing packets, as well as impersonating other nodes. On the other hand, it suffers from the single point of failure problem, single point of attack problem as well as scalability problems when implemented in large networks. Hence, the goal of this thesis is to propose a newly developed routing protocol adopting the authentication steps used with the *ARAN* protocol while achieving better scalability and distributing load and trust among multiple nodes. The resulting new protocol, *ARANz*, provides a solution for scalable secure routing in the managed-open environment.

1.7 Organization of the Thesis

This chapter has started with an introduction about our research. The rest of the thesis is organized as follows:

- **Chapter 2: Literature review.**

  The subsequent chapter gives an introduction to wireless networks in general and Ad-Hoc networks in particular. This chapter also introduces the topic of routing in Ad-Hoc networks, its different techniques and categories along with discussing some of the existing routing protocols. Then, the findings from the literature review are summarized and the research direction is presented.

- **Chapter 3: Research methodology.**

  This chapter addresses our research methodology including a discussion about the different simulation environments and the reasons behind choosing the *GloMoSim* simulator.

- **Chapter 4: The proposed protocol.**

  In this chapter, the newly proposed scheme is presented. Our assumptions are explained and different protocol phases are discussed in detail. After that, a performance and security analysis of the proposed protocol is given.

- **Chapter 5: Simulations, results and performance analysis.**

  Chapter 5 addresses our simulation methodology and scenarios as well as discussing simulation results. We start by introducing the simulation methodology and explaining the scenarios and experiments carried out. The proposed protocol is evaluated and compared to other existing routing protocols considering a wide range of performance parameters and metrics. Subsequently, the obtained results are discussed.

- **Chapter 6: Discussion.**

  Chapter 6 discusses the findings of the research. This chapter presents a discussion of the studied protocols along with an analysis of the results obtained via the simulated performance evaluation.

- **Chapter 7: Conclusion and future work.**

  In this chapter we summarize the thesis and highlight main contributions of this research. Latterly, we present the drawn conclusions and suggest some of the potential future research areas.

Chapter 2

Literature Review

In the recent years, wireless networks, especially Ad-Hoc networks, have become a challenging scientific area for new fields of research (Barba et al. 2010). A routing protocol is a fundamental part of network infrastructure that supports the delivery of packets. Several routing protocols have been proposed for *Mobile Ad-Hoc NETworks (MANETs)*. In general, they can be divided into two main categories: *topology-based* and *position-based*.

This chapter is organized as follows. *Sections 2.1* and *2.2* give an introduction to wireless networks in general and Ad-Hoc networks in particular. In *Section 2.3*, we introduce routing protocols for Ad-Hoc networks. *Section 2.4* discusses the topology-based routing protocols, whereas *Section 2.5* talks about the position-based routing protocols. Finally, *Section 2.6* summarizes the findings from the literature review, justifies our interest in *ARAN* protocol to work on and discusses our research direction. In *Section 2.7 ARAN* protocol is analyzed in detail.

2.1 Introduction to Wireless Networks

In the recent years, wireless networks have become a widespread communication technology as well as a challenging scientific area for new fields of research (Barba et al. 2010). Wireless networking allows users to get anytime, anywhere access to information, communication and service by utilizing the wireless mobile technologies (Mukherjee et al. 2003). The use of wireless communication between mobile users has become increasingly popular due to recent performance advancements in computers and wireless technologies. The evolution of wireless communication technologies has reached a point that makes them popular and easy to be integrated to handheld computing devices, which have primarily been intended for personal use only.

Nowadays, a new generation of portable computers is being developed providing users with higher computational power than ever, in addition to mobility (Bur & Ersoy 2006). These advancements have led to lower prices and higher data rates, which are the two main reasons behind the widespread applications of mobile computing (Mahmoud 2005).

The history of wireless networks started in the 1970s, and they have become increasingly popular ever since. Today there are two kinds of wireless networks. The first kind and most used is the wireless network built on-top of a wired network, creating a reliable infrastructured wireless network. An example of this type is the cellular-phone networks (see Figure 2.1), where a phone connects to a *Base Station (BS)* with the best signal quality.

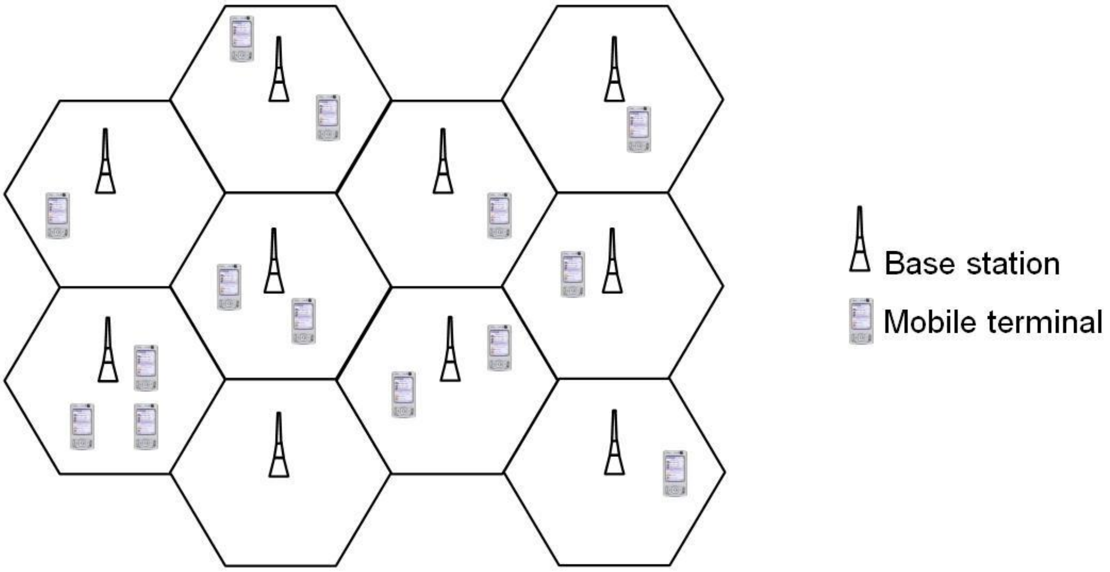


Figure 2.1: A cellular-phone network

In this network, a mobile host communicates with the network through a *BS* within its communication radius. When it goes out of range of one *BS*, it connects to a new *BS* within its range and starts communicating through it. A major problem of this approach is handoff, which handles the situation when a connection should be smoothly transferred from one *BS* to another without significant delay, packet loss or interruption of the service. Another issue is that networks based on a fixed infrastructure are limited to places having such network infrastructures (Mahmoud 2005).

The second class of wireless networks, which is the focus of this research, is to form a wireless network with no infrastructure except the participating mobile nodes. This is called an infrastructure-less network or more commonly known as an Ad-Hoc network (Mukherjee et al. 2003). The word “Ad-Hoc” can be translated as “improvised” or “not organized” which may reflect a negative meaning. In this context, this is not the case as it only describes the network situation, i.e. dynamic. All or some nodes within an Ad-Hoc network are capable of movement and expect to discover and maintain routes to other nodes located beyond their own transmission range (Mukherjee et al. 2003). Whereas cellular networks generally need a single-hop link to reach a mobile terminal, Ad-Hoc networks normally require a multi-hop path between a source and the intended destination (Mukherjee et al. 2003).

Laptops and *Personal Digital Assistants (PDA*s*)* communicating directly with each other are examples of nodes in an Ad-Hoc network. Nodes in Ad-Hoc networks are often mobile, but may also be stationary. Each node has a wireless interface and communicates with other nodes over either radio or infrared channels (Mahmoud 2005). Figure 2.2 shows a simple Ad-Hoc network.

Ad-Hoc wireless networks, mobile or static, have special resource requirements and exclusive topology features, which make them different from wired and traditional wireless networks in regards to resource management, routing, media access control and *Quality of Service (QoS)* provisioning. Besides, Ad-Hoc networks have some unique issues such as self-organization, mobility management and energy-efficient design (Cheng & Li 2008). The following sections address these issues and requirements in detail.

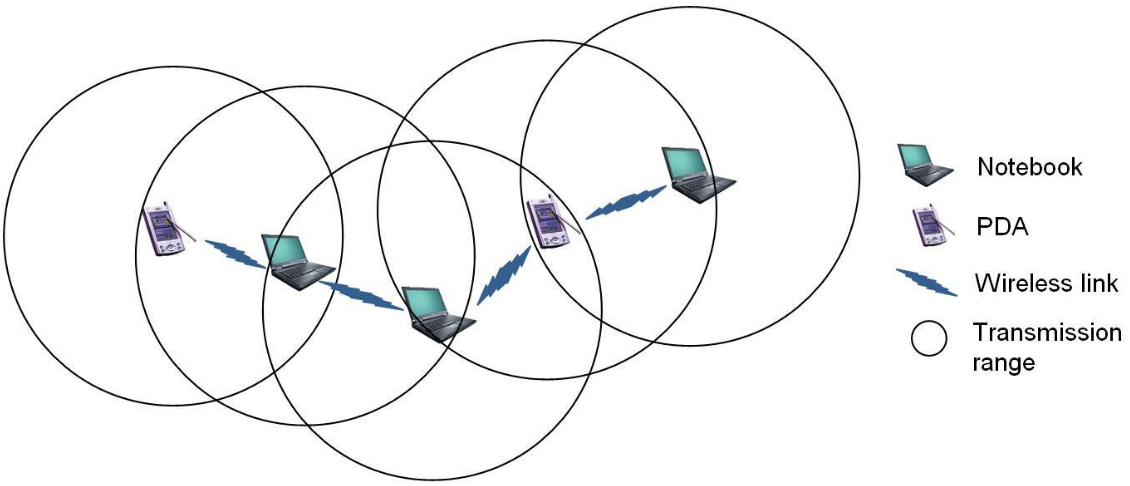


Figure 2.2: An Ad-Hoc network with five nodes

2.2 Introduction to Ad-Hoc Networks

An Ad-Hoc network is a self-organizing and self-configuring network formed when a collection of nodes, equipped with wireless interfaces, connect together and create a network by agreeing to route messages for each other. Ad-Hoc networks have no fixed infrastructure such as routers or base stations, instead, nodes co-operate to carry out basic networking functions including packet forwarding, routing and service discovery (Xenakis et al. 2010; Manikandan et al. 2011). Consequently, Ad-Hoc networks have the ability to be formed anywhere anytime, as long as the wireless nodes are willing to communicate. With this flexibility and due to the easy deployment they require, Ad-Hoc networks are receiving increasing attention (Barba et al. 2010; Joshi 2011).

A basic assumption in an Ad-Hoc network is that if two nodes aim to communicate while being outside the transmission range of each other, they are still able to communicate if other nodes in the network are willing and capable of forwarding packets among them (Mukherjee et al. 2003). Consequently, each node acts as both a host and a router. The topology of Ad-Hoc networks may change continuously as nodes move, join or leave the network. This instability in network topology requires a routing protocol to run on each node to establish and maintain routes among nodes in the network (Mahmoud 2005).

A *Mobile Ad-Hoc NETwork (MANET)* is a type of Ad-Hoc networks with rapidly changing topology due to frequent movement of the nodes. *MANET*s form a special class of dynamic multi-hop networks consisting of a set of mobile nodes that intercommunicate on shared wireless channels, where the network topology changes dynamically due to the mobility of nodes (Pirzada & McDonald 2008). Each node in a *MANET* is free to move independently in any direction, resulting in frequent changing in its links to other nodes. The highly dynamic network topology may result in unstable communication links and frequent path breaks in on-going sessions. This situation often results in frequent routes changes as well as difficulty in delivering packets to their destinations leading to high loss rate and severe performance degradation.

The successful operation of an Ad-Hoc network will be hampered if an intermediate node participating in a communication between two nodes, either moves out of range or suffers sudden failure. Thus, the energetic nature of this kind of networks resulting from the mobility and disconnection of mobile hosts, poses a number of challenges and makes the design of routing protocols for these dynamic environments a difficult task (Mukherjee et al. 2003; Barba et al. 2010; Rout et al. 2011).

Unlike fixed devices, mobile devices have limited capabilities, computing power, bandwidth and storage capacity. In addition, their architecture involves certain security problems since wireless networks have physical vulnerabilities.

Hence, the provision of secure routing to these networks faces specific vulnerabilities due to the absence of fixed infrastructure and non-reliable users that want to utilize the network resources without spending their energy in forwarding messages of other nodes (Fernandes & Duarte 2010).

Ad-Hoc networks are highly applicable in many fields. However new technologies always come with their own set of problems and challenges (Menaria et al. 2010). As such, Ad-Hoc networks have attracted many researchers in the recent years (Jacobsson et al. 2010) especially due to prominent characteristics and challenges they exhibit. The following two sections introduce some of the Ad-Hoc networks applications and summarize the challenges they face.

2.2.1 Ad-Hoc Networks Applications

Ad-Hoc wireless networks can be deployed in some situations where a wired network infrastructure is undesirable due to cost or convenience (Manikandan et al. 2011). Thus, Ad-Hoc networks have various applications, among which include (Othman 1999; Mukherjee et al. 2003; Mahmoud 2005):

- Students using laptop computers to participate in an interactive lecture,
- A group of people with laptop computers at a conference desire to exchange files and data without the need for any additional infrastructure between them,
- Meetings or conventions in which participants wish to share information in a quick manner,
- *Personal Area Networks (PANs)*, short-range localized network connecting someone's cell phones, laptops, smart watches, ear phone, belt and other wearable computers,
- Soldiers relaying information about the situation on the battle field,
- Emergency search-and-rescue operations,
- Situations where natural disasters such as earthquakes have destroyed communication infrastructures,
- Data acquisition operations in unfriendly terrain,
- In undeveloped areas,
- Train and bus station information (e.g. local traffic information),
- Weather information,
- Real time multimedia applications like Tele-medicine, Tele-commuting and collaborative environments.

From the aforementioned list, it is conspicuous that Ad-Hoc networks have numerous applications and they are increasingly involved in many life aspects. Moreover, the set of Ad-Hoc networks applications ranges from large and highly dynamic networks to small and static networks (Goyal et al. 2010).

2.2.2 Ad-Hoc Networks Characteristics and Challenges

As discussed in the previous section, Ad-Hoc networks are highly applicable in many fields. Therefore, there has been a growing interest in Ad-Hoc networks (Jacobsson et al. 2010) especially due to prominent characteristics and challenges they exhibit. These characteristics are summarized as follows (Johnson 1994; Corson et al. 1996; Johnson & Maltz 1996; Corson et al. 1999; Othman 1999; Mukherjee et al. 2003; Mahmoud 2005; Razak et al. 2009; Menaria et al. 2010; Rout et al. 2011):

- Dynamic topologies: Nodes are free to move arbitrarily. Thus, the network information, such as link-state, becomes quickly outdated due to rapid and unpredictable nodes movement and fast-changing propagation conditions. This results in recurrent network reconfigurations and control information exchanges.
- Asymmetric link characteristics: Communication between two nodes in wireless environments may not work similarly well in both directions. In other words, although node N is within the transmission range of node M, the reverse may not be true, resulting in unidirectional links.
- Multi-hop communication: Packets sent from a source may reach the desired destination in multiple hops through numerous intermediate relay nodes. However, successful operation of an Ad-Hoc network may be interrupted if an intermediate node moves out of range suddenly or switches itself off during message transfer. The situation becomes worse if there is no other path between the communicating nodes.
- Decentralized operation: Ad-Hoc network is a network type that can be rapidly deployed and that does not rely on pre-existing infrastructure, centralized

administration or standard support services. In cellular wireless networks, there are a number of centralized entities responsible for achieving the coordination among nodes. Hence, the lack of such entities in Ad-Hoc networks requires more sophisticated distributed algorithms to carry out equivalent functionality.

- Bandwidth-constrained, variable capacity links: Wireless links have significantly less capacity compared to their hardwired counterparts. Also, the realized throughput of wireless communications is often much less than a radio’s maximum transmission rate due to the effects of multiple access, fading, noise and interference conditions. The relatively low to moderate link capacities result in congestion being the normal case rather than the exception.
- Energy-constrained operation: Nodes in *MANET*s most probably rely on batteries or other exhaustible means for their energy. Hence, energy conservation is an important system design optimization criteria. One way of achieving energy conservation for these nodes is to optimize the transmission power of each node.
- Security: Mobile wireless networks are highly prone to physical security threats compared to fixed-cable networks. The increased likelihood of some attacks such as eavesdropping, spoofing and fabrication must be considered carefully while designing routing protocols for this type of networks.

The aforementioned Ad-Hoc networks characteristics and challenges create many basic assumptions and performance concerns to be considered upon designing a routing protocol for such networks. These concerns extend beyond those guiding the design of routing protocols for conventional networks with pre-configured topology and make proposing a routing protocol for Ad-Hoc networks a hard task.

2.3 Introduction to Ad-Hoc Networks Routing Protocols

The main purpose of Ad-Hoc network routing protocols is to enable transferring data packets from one point to another point in the network (Yau et al. 2007). Routing protocols act as the building blocks in Ad-Hoc networks by finding and maintaining virtual connections between the nodes to support packets delivery (Pirzada & McDonald 2008). In Ad-Hoc networks, due to the limited communication range of wireless interface, a data packet may need to be transferred via several intermediate nodes (Kadono et al. 2010). Moreover, Ad-Hoc networks do not have pre-deployed infrastructure to assist in end-to-end routing in the network. Hence, nodes in Ad-Hoc networks communicate with each other without the help of centralized access points or base stations. Each node acts as both a router and a host.

Ad-Hoc networks routing protocols and their performance are important issues due to the nature and demands of these networks (Lakshmikanth et al. 2008). Routing protocols in the Ad-Hoc networks are difficult to implement as they have to face the challenge of link instability, node mobility, frequently changing topology, absence of a fixed infrastructure and low transmission power (Goyal et al. 2010; Zhu et al. 2011). Additionally, due to differences in transmission capacity of individual nodes, some of the links among nodes may be unidirectional, leading to some asymmetric links.

Nodes mobility in these networks presents the most difficult challenge to routing protocol designers because of frequent topology changes and route invalidation, which increase the routing overhead required to re-establish routes, thus affecting the *MANET* performance. Consequently, these protocols must construct and maintain routes in dynamic networks effectively and efficiently (Lakshmikanth et al. 2008; Rout et al. 2011).

For the aforementioned reasons, routing protocols proposed for Ad-Hoc networks must be suitable for implementation in environments that may vary from the extremes of high mobility with low bandwidth to low mobility with high bandwidth (Johnson & Maltz 1996). As such, routing in Ad-Hoc networks is a particularly hard task to accomplish in an efficient and secure manner (Sharma & Jena 2011).

In *Section 2.3.1*, we look at the main categories of Ad-Hoc routing protocols. *Section 2.3.2* introduces the *Global Positioning System (GPS)* (El-Rabbany 2002; Kaplan & Hegarty 2005). *Sections 2.3.3* through *2.3.5* discuss security issues in Ad-Hoc networks routing protocols and present different security requirements and different types of attacks targeted against Ad-Hoc networks. In *Section 2.3.6,* we address different Ad-Hoc networks categories and security level needed for each category. *Section 2.3.7* introduces key management in Ad-Hoc networks. Finally, *Section 2.3.8* discusses scalability issues in Ad-Hoc networks routing protocols.

2.3.1 Classifications of Ad-Hoc Networks Routing Protocols

Multi-hop routing is the procedure to relay a message between two endpoints through a sequence of intermediate nodes. Routing protocols are designed to fulfil path discovery and maintenance of routing tables. The tradeoffs between routing strategies are quite complex since the best approach depends on many factors such as network size, mobility and data traffic (Rifa-Pous & Herrera-Joancomarti 2007). Two main categories of Ad-Hoc routing protocols are *topology-based* and *position-based*.

Topology-based routing protocols use information about the links existing in the network to perform packet forwarding. Topology-based routing protocols can be further classified into three main groups: *proactive (table-driven or periodic)* protocols, *reactive (demand-driven or source-initiated)* protocols and *hybrid (hierarchical or (reactive/proactive))* protocols.

Proactive routing protocols continuously try to enable each node to know a current route to all other nodes regardless of whether these routes are needed or not. Each node tries to keep an up-to-date topological map of the entire network. With the help of this map, route to a particular destination is known and available immediately when a data packet needs to be sent.

Proactive routing protocols are classified as either *link-state* or *distance-vector* protocols. The former type requires each forwarding node to flood the network about any change in the status of its links. Accordingly, all nodes will note the change and re-compute their routes. Hence, link-state routing protocols are known for rapid convergence, but they involve significantly high control traffic.

Distance-vector routing protocols is computationally less complex and has lower message overhead compared to link-state protocols. On the other hand, they are typically based on the distributed Bellman-Ford algorithm that is known for its slow convergence. Each node maintains a routing table to store the next hop towards each destination. Every node periodically sends its routing table to its immediate neighbours only. For each network path, the receiving nodes choose the neighbour advertising the lowest cost and add this entry into their routing tables for re-advertisement.

In proactive routings, while paths for any destination are always available, the maintenance of all paths, including unused paths, causes a significant communication overhead (Kadono et al. 2010). Consequently, proactive routing protocols are not appropriate for Ad-Hoc networks since they continuously consume power throughout the network, regardless of having network activities. Also they are not designed to track frequent topology changes.

Reactive routing protocols are suitable for use in wireless environments since a route discovery process is initiated only upon having data packets that must be routed. Discovered routes are cached until they are not used for a period of time or a link break occurs due to network topology changes. In contrast to proactive routing, reactive routing protocols do not attempt to continuously determine network connectivity. Instead, a route discovery procedure is started by flooding route discovery queries throughout the network immediately when a packet needs to be sent.

As aforementioned, proactive routing uses excess bandwidth to continuously maintain routing information. On the other hand, reactive routing involves long route request delays and inefficiently floods the entire network for route determination. The third group, *hybrid* routing protocols, aims to address these problems by combining the best properties of both approaches (Vijayakumar & Ravichandran 2011). If the destination is close to the sender, packets are routed using a proactive routing protocol. For long distance routing, a reactive protocol is used.

Position-based routing protocols use the geographical position of nodes to make routing decisions, which results in improving efficiency and performance (Prakash et al. 2011). This is achieved by reducing routing overhead through utilizing information about the nodes' positions. Thus, for any intermediate node, only its neighbouring nodes that are closer to the destination are allowed to participate in the routing process.

These protocols require that a node be able to obtain its own geographical position and the geographical position of the destination. Generally, this information is obtained via the *GPS* system and location services. An introduction to the *GPS* system is provided in the *Section 2.3.2*.

In addition to *GPS*, other systems in use or under development include the Russian *GLObal NAvigation Satellite System (GLONASS),* the Chinese *Compass navigation system* and *Galileo positioning system* of the European Union. For indoor applications *RADAR* (Bahl & Padmanabhan 2000), *SpotON* (Hightower et al. 2001), *Cricket system* (Priyantha 2005), *MoteTrack* (Lorincz & Welsh 2007) or the cricket-based location tracking system proposed in (Kim et al. 2008) can be used.

Position-based protocols are, in turn, divided into three categories: *restricted directional flooding*, *greedy forwarding* and *hierarchical* routing protocols. In *greedy forwarding*, the source node and each intermediate node select a neighbouring node that is closest to the destination as the next hop. This continues until the packet reaches its destination.

Hence, nodes periodically broadcast small packets (in a form of beacons) to enable other nodes to maintain a one-hop neighbour table. This approach is scalable since it does not require routing discovery and maintenance (Wu 2005).

However, periodic beaconing creates a lot of congestion in the network and consumes the nodes' energy (Cao & Xie 2005; Giruka & Singhal 2005). In addition, greedy forwarding protocols generally are not guaranteed to find the optimal route. They also may not find a route (even if one exists) in sparse networks if a particular node does not have any node within its transmission range that is closer to the destination than itself (Karp & Kung 2000; Wu 2005).

In *restricted directional flooding*, upon receiving the route discovery packet, each node computes its distance to the destination. The receiver will retransmit the route discovery packet to its neighbours only if it is closer to the destination than its preceding node. Sending the route discovery packet to several nodes increases the probability of finding the shortest path.

Hierarchical routing protocols aim to achieve scalability, robustness and nodes collaboration (Giordano et al. 2003). These protocols use a two level hierarchy. If the destination node is close to the source node, packets are routed using a proactive distance vector, otherwise, greedy routing is used.

Table 2.1 summarizes the different categories of Ad-Hoc routing protocols along with their advantages and disadvantages.

Table 2.1: Ad-Hoc networks routing protocols categories

Category	**Approach**	**Advantages**	**Disadvantages**
Topology-based	Proactive	Route to any destination is immediately available when a packet needs to be sent.	• Periodic control messages lead to high overhead, power and bandwidth consumption. • Not designed to track topology changes occurring at a high rate. • Scalability problem in networks with more than several hundred nodes.
	Reactive	No periodic routing packets are required.	• Long route discovery delays. • Flood the entire network for route determination (higher routing overhead than position-based protocols). • Scalability problem in networks with more than several hundred nodes.
	Hybrid	Reduced control overhead compared to pure proactive protocols, and reduced delays associated with pure reactive ones.	Inherit disadvantages of proactive protocols for large routing zone, and these of reactive ones for small routing zones.
Position-based	Greedy	Scalable since they do not need routing discovery and maintenance.	• Periodic beacons lead to network congestion and nodes energy consumption. • Low probability of finding the shortest path. • May fail to find a path at all, even if one exists, especially in sparse networks.
	Restricted directional flooding	High probability of finding the shortest path.	Several nodes manage the route request message (higher routing overhead than greedy, but less than that of topology-based protocols).
	Hierarchical	Reduce control overhead compared to proactive protocols, and eliminate disadvantages associated with beacons used in greedy ones.	Inherit disadvantages of proactive protocols for large routing zone, and those of greedy ones for small routing zones.

2.3.2 Introduction to Global Positioning System (*GPS*)

The *Global Positioning System (GPS)* (El-Rabbany 2002; Kaplan & Hegarty 2005; Myers et al. 2006) is a satellite-based navigation system that provides reliable positioning, navigation and timing services. It was developed by the United States

Department of Defense. *GPS* was originally intended for military applications, but in the 1980s, the government made the system freely available for civilian use.

GPS consists of three major segments: *space segment*, *control segment* and *user segment*. The United States Air Force develops, maintains and operates the space and control segments. *GPS* satellites broadcast signals from space, and a *GPS* receiver uses these signals to calculate its location and the current time. The *space segment* consists of 24-32 satellites orbiting the earth at altitudes of approximately eleven thousand miles. *GPS* satellites are powered by solar energy and have backup batteries onboard to keep them running in the event of a solar eclipse. They are constantly moving, making two complete orbits a day. These satellites travel at speeds of roughly seven thousand miles per hour (El-Rabbany 2002; Kaplan & Hegarty 2005).

The *control segment* is composed of control stations, ground antennas and monitor stations. The *user segment* is composed of hundreds of thousands of United States and allied military users of the secure *GPS* precise positioning service, in addition to tens of millions of civil, commercial and scientific users of the standard positioning service. Figure 2.3 shows *GPS*'s three major segments.

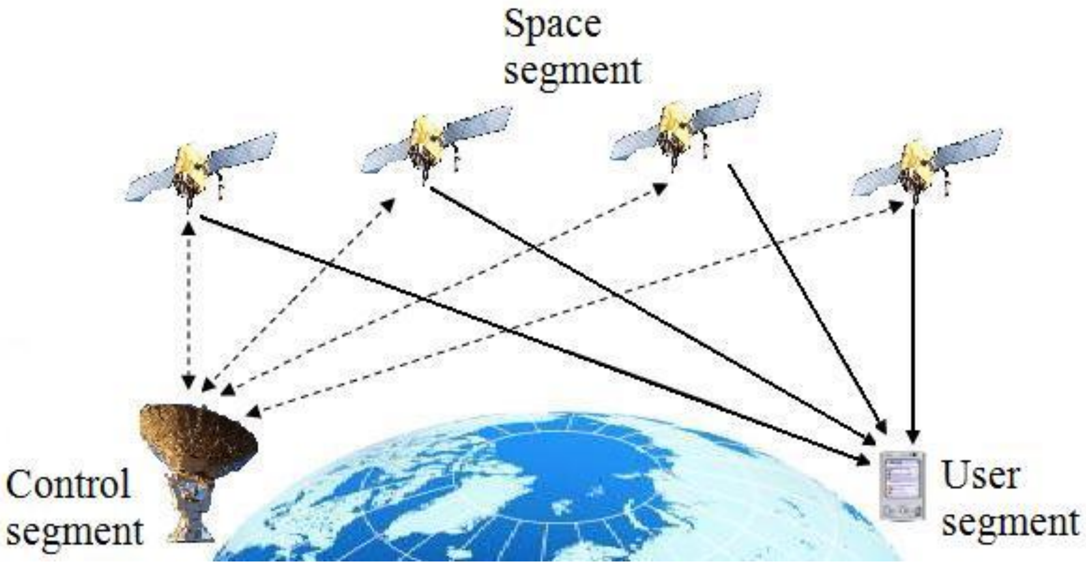


Figure 2.3: *GPS* segments

GPS satellites transmit three pieces of information, the satellite's number, its position in space and the time the information is sent. These signals are picked up by the *GPS* receiver, which uses this information to calculate the distance between itself and the *GPS* satellites. *GPS* receivers take this information and use triangulation to calculate the user's exact location. Essentially, the *GPS* receiver compares the time a signal was transmitted by a satellite with the time it was received. The time difference tells the *GPS* receiver how far away the satellite is. With signals from three or more satellites, a *GPS* receiver can triangulate its 2-Dimensional position on the ground (i.e. longitude and latitude) from the known position of the satellites. With four or more satellites, a *GPS* receiver can pinpoint the user's 3-Dimensional position (i.e. longitude, latitude and altitude). In addition, a *GPS* receiver can provide data on user's speed and direction of travel (El-Rabbany 2002; Kaplan & Hegarty 2005; Myers et al. 2006).

Since anyone, anywhere in the world, equipped with a *GPS* receiver can access the free real-time *GPS* services, *GPS* is used in numerous applications including geographic information system data collection, surveying and mapping. Moreover, as *GPS* units are becoming smaller and less expensive, there are an expanding number of applications for *GPS* (El-Rabbany 2002; Kaplan & Hegarty 2005; Myers et al. 2006). In transportation applications, *GPS* assists pilots and drivers in identifying their locations and avoiding collisions. Farmers can use *GPS* to guide equipment and control accurate distribution of fertilizers and other chemicals. *GPS* has become a widely deployed and useful tool for commerce, scientific uses, tracking and surveillance. Actually, *GPS* helps farmers, surveyors, geologists and countless others perform their work more efficiently, safely, economically and accurately (El-Rabbany 2002; Kaplan & Hegarty 2005; Myers et al. 2006).

2.3.3 Security Issues in Ad-Hoc Networks Routing Protocols

Mobile Ad-Hoc networks have no clear line of defence as they could be accessible to both legitimate network users and malicious attackers (Khokhar et al. 2008). Nodes in Ad-Hoc routing protocols exchange information with each other about the network topology, constructing a virtual view of the network topology to allow routing of data packet. This information allows them to create, delete and update routes between the nodes in the network. On the other hand, this capability can pose as security weak point in Ad-Hoc networks since a compromised node might give wrong information to redirect traffic or suspend it. Thus, this information has to be protected to avoid malicious nodes disrupting the network (Rifa-Pous & Herrera-Joancomarti 2007). Additionally, since Ad-Hoc networks are based on collaborative routing, a node working in a malicious way may disrupt the entire network operation (Fernandes & Duarte 2010).

Ad-Hoc network security, in particular routing protocols security, has attracted significant attention (Li et al. 2007; Dutta & Dowling 2011). Securing Ad-Hoc routing faces many challenges, especially that each user brings to the network his/her own mobile unit without any centralized control as that found in a traditional network. Accordingly, securing Ad-Hoc routing faces difficulties that do not exist in wired networks as well as infrastructure-based wireless networks. These difficulties make establishing trust among nodes virtually impossible. Among these difficulties are the wireless medium itself and its physical vulnerability, lack of centralized control and permanent trust infrastructure, restricted power and resources, cooperation of nodes, highly dynamic topology, short-lived connectivity and availability, implicit trust relationship between neighbours and other problems associated with wireless communication (Rifa-Pous & Herrera-Joancomarti 2007; Pirzada & McDonald 2008; Joshi 2011).

For the aforementioned reasons, the design of a secure routing protocol is really a difficult task taking into account the diversity of Ad-Hoc network applications and their security requirements. As a result of this variety, it is difficult to have a general solution to protect against different threats and attacks targeted toward Ad-Hoc routing protocols. Thus, many protocols have been proposed focusing on different parts of the problem using various mechanisms and cryptographic techniques to countermeasure different attacks (Khokhar et al. 2008; Joshi 2011). In the following sections we present different security requirements and different types of attacks targeted against Ad-Hoc networks. Also we address different Ad-Hoc networks categories and security level needed for each category.

2.3.4 Ad-Hoc Networks Security Requirements

To ensure the security of Ad-Hoc network, a number of requirements need to be satisfied. Although Ad-Hoc network requires the same security requirements needed for other types of wireless and wired networks, such security requirements need to be addressed in a specific way that suits the Ad-Hoc environment (Razak et al. 2008; Goyal et al. 2010). These requirements are summarized as *availability*, *confidentiality*, *integrity*, *authentication* and *non-repudiation* (Zhou & Haas 1999; Murthy & Manoj 2004; Mahmoud 2005; Chimphlee et al. 2007; Razak et al. 2008; Jhaveri et al. 2010):

- Availability: The network should remain operational and available to send and receive messages at any time. It should be robust to tolerate link failure and survive despite attacks. Hence, availability assures that the resources needed to be accessed are accessible to authorized parties in the ways and at the times they are needed.
- Confidentiality: Provides secrecy to sensitive data being sent over the network, i.e. the contents of every message can be understood only by its source and destination. Although an intruder may intercept the data being sent, he should not be able to

derive any useful information from it. This is especially important in military combat operations where strategic and tactical information is exchanged.

- Integrity: Assures that messages being sent over the network are not corrupted by intentional or accidental modification. Possible attacks that may compromise the integrity property are malicious attacks on the network or radio signal failures.
- Authentication: Ensures the identity of nodes existing in the network, i.e. assure that they are who they claim to be. If the authentication fails to work properly, it may be possible for any node to masquerade as a particular node and then be able to send and receive sensitive information privy only to authorized nodes.
- Non-repudiation: Guarantees that neither sender nor receiver can deny that he has sent or received the message.

Recently, as privacy has emerged as an important security issue, there are many researches about anonymous Ad-Hoc routing protocol. By providing *anonymity*, Ad-Hoc routing keeps the network information against an adversary who wants to collect information for illegal act (Paik et al. 2008). The anonymity in an Ad-Hoc routing implies that the identity of node, route path information and location information must be veiled, not only from an adversary, but also from other valid nodes (Mizanur Rahman et al. 2006; Paik et al. 2008; Goyal et al. 2010).

Different Ad-Hoc network applications have different security requirements to be taken into account. For instance, it is worth nothing that the security level required in a network established among a group of soldiers for tactical operation is much higher than that anticipated for a network among students in a class. Moreover, there are many threats and attacks targeted toward routing protocols and it is difficult to have a general solution to protect routing protocol against them all. As a result of this variety, many protocols have been proposed focusing on different parts of the problem (Goyal et al. 2010; Manikandan et al. 2011). In the following two sections we discuss different types of attacks targeted against Ad-Hoc networks and address different Ad-Hoc networks categories and security level needed for each category.

2.3.5 Attacks against Ad-Hoc Networks Routing Protocols

Ad-Hoc routing is a very fundamental operation on an Ad-Hoc network, and hence it has been a main target for an attacker to disrupt an Ad-Hoc network (Park et al. 2007). For example, the attacker may snoop and interpret the data exchanged in the network that violates the requirement of confidentiality discussed in the previous section. As such, before the development of a security measure to secure mobile Ad-Hoc networks, it is important to study the variety of attacks that might be targeted against such networks (Razak et al. 2008).

Two kinds of attacks can be launched against Ad-Hoc networks (as well as wired networks and infrastructure-based wireless networks) (Razak et al. 2004; Murthy & Manoj 2004; Li et al. 2007; Rifa-Pous & Herrera-Joancomarti 2007; Mandala et al. 2008; Pirzada & McDonald 2008; Goyal et al. 2010; Manikandan et al. 2011):

- Passive attacks: The attacker does not interfere with the routing protocol. It merely eavesdrops on the routing traffic and endeavours to extract any valuable information such as node hierarchy and network topology. For example, if a route to a particular node is requested more often than to other nodes, the attacker might infer that the node is important for the functioning of the network and that disabling it could bring the entire network down. Detection of passive attacks is very difficult since the operation of the network itself is not affected. One way of overcoming such problems is using a powerful encryption technique to encrypt the data being transmitted, making it impossible for the attacker to obtain any useful information from the data overheard.
- Active attacks: The attacker does not only eavesdrop on the network activities, but also consumes some of its energy in order to perform the attack. Nodes that perform

active attacks with the aim of disrupting other nodes and the network are considered to be malicious. In active attacks, malicious nodes can disturb the correct functioning of a routing protocol by *modifying* routing information, *fabricating* false routing information and *impersonating* as other nodes.

Modification attacks are normally targeted against the integrity of routing computation. By modifying routing information, an attacker can cause network traffic to be dropped, redirected to a different destination or to take an extended route to the destination. A more skillful modification attack is to create a tunnel or wormhole in the network between two colluding malicious nodes connected through a private network connection. This exploit allows the colluding attackers to short-circuit the normal flow of routing messages.

Fabrication attacks are performed by generating deceptive routing messages. The rushing attack is a classic example of a fabrication attack, where an attacker rapidly spreads routing messages all over the network so that nodes drop legitimate routing messages by evaluating them as duplicates. Fabrication attacks are difficult to recognize as they are received as legitimate routing packets.

During *impersonation attacks*, a malicious node launches many attacks and misrepresents the network topology by masquerading as another legitimate node through spoofing. This occurs when a malicious node fakes its identity by altering its *Media Access Control (MAC)* address or *Internet Protocol (IP)* address in order to change the perspective of a benevolent node regarding the network.

In addition, several attacks are possible in the forwarding operation. Data packets could be dropped, replayed or redirected. Data packets forwarding attacks can be launched even when a secure routing protocol is being used (Zouridaki et al. 2007). A secure routing protocol aims to establish a route devoid of unauthorized nodes between the source and destination nodes. Once a route is established, nodes on the path are supposed to forward packets to the right next hop. However, during the data transmission phase an authorized node may misroute or replay packets. Authorized nodes also could drop all data packets passing through themselves, which is known as black hole attack. They also may perform grey-hole attack, i.e. drop some of data packets that traverse these nodes (Zouridaki et al. 2007).

2.3.6 Main Categories of Ad-Hoc Networks

In order to be able to provide solutions to the security issues in Ad-Hoc networks, it must be recognized first that there are different kinds of these networks. For example, Ad-Hoc networks can be implemented in a disaster area or a military war-zone. Others could be found among students in a campus or even among users that are unknown previously, such as that established among users on a highway.

Different types of networks place different demands on the infrastructure to determine what means are available to improve security. Ad-Hoc networks are divided into three categories (Sanzgiri et al. 2005): *open*, *managed-open* and *managed-hostile*. These classes are defined since it is difficult to construct a single secure Ad-Hoc routing protocol to suit the needs of various heterogeneous wireless applications. These types differ in both the level of security needed and the opportunity of exchanging some security parameters before deploying the network.

In *open* environments, participating nodes are not linked by any organizational relationship, and network security mechanisms do not rely on any existing trust relationship among the participating nodes. Nodes in an open environment are not necessarily known beforehand. Consequently, any central authority system that requires former knowledge of the nodes participating in the network is not suitable for such environments. This scenario may exist for users walking through an urban environment or driving on a highway. Usually this is not a very common environment.

In managed environments, nodes are controlled by an organization and an in-advance trust relationship between the nodes can be derived from the already existing trust relationship of the organization. The *managed-open* environment is probably the one where most research is being carried out today as it is the type of environment expected to spread out in the nearest future. In this type of environment, there is a possibility to use already established infrastructure to help secure the Ad-Hoc network. This opens up a whole new range of strategies that use certificate servers and other similar software to provide a basis for security in the network. Such Ad-Hoc networks could be formed by students on a campus or peers at a conference.

The *managed-hostile* is the classic Ad-Hoc environment, described as nodes in a military war-zone or a disaster area. In a managed-hostile environment, security is the primary goal, even information such as location of participating nodes is considered very sensitive information. In such environments, security is considered to be much more important than performance, and accordingly, the security measures can be made a bit more extreme. The distinguishing security threat of the managed-hostile environments is that every node is susceptible to physical capture and equipment taking over. Hence, a hostile entity can pose as a friendly entity at a compromised node. Therefore, the exposure of nodes' locations from the routing protocol messages is not desirable since this may give adversaries the opportunity to eradicate trusted users.

Table 2.2 summarizes different categories of Ad-Hoc networks.

Table 2.2: Ad-Hoc networks categories

Category	**Properties**	**Example**
Open	• Network security mechanisms cannot rely on any existing trust relationship among the nodes. • Not a very common environment.	• Users walking through an urban environment. • Users driving on a highway.
Managed-open	• Participating nodes are controlled by an organization. • There is a possibility to use already established infrastructure to help in securing the network. • The one where most research is being done.	• Peers at a conference. • Students on a campus.
Managed-hostile	Security is considered to be more important than performance.	Nodes in a military war-zone.

2.3.7 Key Management in Ad-Hoc Networks

Keys are needed for the authentication and encryption of the transmitted messages. The primary goals of a key management system are to distribute keys to nodes securely and to ensure that periodic (or on-demand) key updates occur regularly. As a result of the lack of infrastructure, key management in Ad-Hoc networks is considered a challenging task (Mahmudul Islam et al. 2008; Dutta & Dowling 2011).

Cryptographic schemes, such as digital signatures, are often employed to protect both routing information and data. Public key infrastructure can be used for easy distribution and certification of keys. In public key infrastructure, each node has a public/private key pair. Public keys are distributed to all nodes, while the private key is kept by the individual node. A trusted third party, known as *Certificate Authority (CA)*, is normally used for key management. A *CA* has its own public/private key pair, with the public key made known to all nodes. The *CA* certifies a particular node by combining the public key and identity of that node and signing them with its own private key. The trusted *CA* has to stay online to reflect the current bindings as the bindings could change overtime. A public key must be revoked if the owner node is no longer trusted or moves outside the network (Mahmudul Islam et al. 2008; Murthy & Manoj 2004).

A single key management service for an Ad-Hoc network is possibly not a good design choice because if a *CA* is down or becomes unavailable, all nodes will not be able to obtain the current public keys of other nodes to establish secure connections. Furthermore, if a *CA* is compromised, the attacker will be capable of signing erroneous certificates, thus disrupting the entire network functionality. As such, a *CA* consisting of a single node is exposed to a single point of failure and compromise problems, which reduces the system's robustness and availability (Murthy & Manoj 2004; Mahmudul Islam et al. 2008).

2.3.8 Scalability Issues in Ad-Hoc Networks Routing Protocols

As the scale of Ad-Hoc networks continues to grow, one of the most critical design issues of a routing protocol is its applicability in large-scale deployments, i.e. the protocol scalability (Papavassiliou et al. 2002; Lee et al. 2003; Gerla 2005). Many proposed routing protocols for Ad-Hoc networks (such as DSR (Johnson & Maltz 1996) and AODV (Perkins & Royer 1999)) are designed with small network and flat topology in mind (Inn 2006). These protocols can scale reasonably well to dozens of nodes because their focus is mainly on performance in relatively small networks, and less on scalability (Eriksson 2006). However, the widespread of mobile devices and the deployment of large-scale Ad-Hoc networks for military, rescue and commercial applications, that may consist of hundreds or possibly thousands of nodes, raise the scalability issue of routing protocols (Hong et al. 2002; Eriksson 2006; Inn 2006). Hence, the development of large-scale Ad-Hoc networks has drawn a lot of attention and the scalability of Ad-Hoc networks has been the subject of extensive research (Kwak et al. 2004).

The scalability of a routing protocol is a measure of its ability to support the increase of one or more network parameters (such as network size, network density, mobility rate and data generation rate) without degrading the network performance (Santivanez et al.

2002; Eriksson 2006; Inn 2006). Designing a reliable and scalable routing protocol for Ad-Hoc networks is a challenging task, especially due to the continuous change in the network topology (Nagar et al. 2011). Moreover, the absolute protocol scalability (Santivanez et al. 2002) is very hard to be defined in mobile environments. Therefore, in some researches, such as (Arpacioglu et al. 2003), the weak scalability notion is adopted. Weak scalability refers to the comparison of the performance metrics of interest with respect to a specified range of the network parameters. In literature, the performance metrics in Ad-Hoc routing protocols include the packet delivery ratio, the delay performance and the routing overhead. Meanwhile, typical network parameters of interest include the number of nodes, network density, network size, mobility rate and data generation rate (Hong et al. 2002; Inn 2006; Schleich 2010).

Clustering algorithms and hierarchical routing are proposed in Ad-Hoc networks as attractive approaches to improve routing protocol scalability (Inn 2006; Abrougui et al. 2011; Yang & Bao 2011). A clustering algorithm is usually used to divide the network into smaller sub-groups. In general, clustering can provide scalability and reduce signaling traffic (Bettstertter & Konig 2002; Hong et al. 2002). For example, if a flat structure is used in a large network, routing tables and location updates would grow to a huge size. Therefore, partitioning the network into multiple clusters can limit the size of routing tables (Bettstertter & Konig 2002; Eriksson 2006). Moreover, detailed topology information for a particular cluster is only exchanged among local cluster members whereas aggregated information is propagated between neighboring clusters in a higher hierarchical level (Bettstertter & Konig 2002).

Additionally, a scalable and efficient solution must avoid concentrating responsibility

at any individual node and keep the necessary state to be maintained at each node as small as possible. (Eriksson 2006). Distributing load among multiple nodes improves performance and scalability of the routing protocol. It also helps in achieving robustness and solving the single point of failure problem.

Because of the multi-hop nature of Ad-Hoc networks and due to the scarce bandwidth, the scalability of these networks is directly related to the used routing protocol (Hong et al. 2002; Kwak et al. 2004; Abrougui et al. 2011). For example, global broadcast of control packets may generate overhead and consume most of the bandwidth, causing scalability problems in large-scale networks (Hong et al. 2002; Eriksson 2006; Inn 2006). Thus, reducing routing control overhead becomes a key issue in achieving routing scalability (Santivanez et al. 2001; Hong et al. 2002). In proactive routing, for example, the routing protocol periodically broadcasts routing information throughout the network, so that, every node keeps routing information about every other node, leading to lack of scalability (Naumov & Gross 2005; Eriksson 2006). The common characteristic among all topology-based routing protocols is that performance degrades as network density increases, leading to a scalability problem (Al-Rabayah & Malaney 2011)

Position-based routing protocols, on the other hand, are an attractive scalable alternative (Koutsonikolas et al. 2010). In position-based routing, geographical location information is used to localize the control message propagation and to help the routing layer scale to support very large networks (Hong et al. 2002; Abolhasan et al. 2004; Eriksson 2006). Position-based routing is scalable to large networks, since it uses only knowledge of the source and the destination locations and is independent of network topology and size (Wang & Ravishankar 2009).

2.4 Topology-Based Routing Protocols

Topology-based routing protocols use information about links that exist in the network to perform packet forwarding (Prakash et al. 2011). They are, in turn, divided into three categories: *proactive (table-driven)* protocols, *reactive (demand-driven)* protocols and *hybrid (hierarchical)* protocols. These categories are discussed in the following three sections. After that, the work done to secure topology-based routing protocols is addressed. Finally, a summary of the discussed topology-based routing protocols is conducted.

2.4.1 Proactive Routing Protocols

Numerous proactive routing protocols have been proposed for Ad-Hoc networks such as *Destination-Sequenced Distance-Vector (DSDV)* (Perkins & Bhagwat 1994), *Wireless Routing Protocol (WRP)* (Murthy & Garcia-Luna-Aceves 1996), *Clusterhead Gateway Switch Routing (CGSR)* (Chiang et al. 1997), *Source Tree Adaptive Routing (STAR)* (Garcia-Luna-Aceves & Spohn 1999), *Fisheye State Routing (FSR)* (Pei et al. 2000) and *Optimized Link State Routing Protocol (OLSR)* (Jacquet et al. 2003).

Proactive routing protocols periodically broadcast control messages in an attempt to have each node always knows a current route to all destinations and remove local routing entries if they time out. Hence, the periodically broadcast control messages contribute to overhead (Lin 2004). We observe that proactive routing protocols may be inappropriate for Ad-Hoc wireless networks as they repeatedly consume power throughout the network, regardless of the existence of network activity. Also they are not designed to track rapid topology changes (Beijar 1998).

2.4.2 Reactive Routing Protocols

Reactive routing protocols are deemed more appropriate for wireless environments since they initiate a route discovery process only when the route is needed (Beijar 1998). Many Ad-Hoc routing protocols that use reactive route determination have been developed. The *Dynamic Source Routing (DSR)* (Johnson & Maltz 1996), *Ad-Hoc On-demand Distance Vector (AODV)* (Perkins & Royer 1999), *Multipath Dynamic Source Routing (MDSR)* (Nasipuri & Das 1999) and *Temporally Ordered Routing Algorithm (TORA)* (Park & Corson 2001) are typical on-demand routing protocols.

Similar to *DSR*, *AODV* is a reactive protocol, i.e. both use flooding to detect routes on-demand. The query packet in *AODV* has a number-of-hop field which is incremented by each intermediate node. In *DSR*, a list of intermediate nodes addresses is stored in the query packet (Lin 2004). So, *DSR* protocol performs source routing with the addresses obtained from the query packet while *AODV* uses next hop information stored on the nodes involved in the route (Beijar 1998).

One advantage of reactive routing protocols is that no periodic routing packets are required. On the other hand, they may have poor performance in terms of control overhead in networks with high mobility and heavy traffic loads. Scalability is said to be another disadvantage because they rely on blind broadcast to discover routes (Lin 2004). Broadcasting routing packets to the entire network leads to congestion and large routing overhead as well as affecting the protocol's performance due to dropping data packets.

2.4.3 Hybrid Routing Protocols

As seen in the previous two sections, proactive routing uses surplus bandwidth to maintain routing information while reactive routing involves long route request delays. Additionally, reactive routing propagates route request packets to the entire network, resulting in higher control overhead. Hybrid routing protocols aim to address these problems by combining the best properties of both approaches (Beijar 1998).

Zone Routing Protocol (ZRP) (Beijar 1998) is an example of hybrid routing protocols. *ZRP* maintains an up-to-date topological map of a zone centered on each node separately. The routing zone has a radius *r* expressed in hops. The zone of a particular node includes the nodes whose distance is at most *r* hops from the corresponding node. Within the zone, routes are immediately available. For destinations outside the zone, *ZRP* employs a route discovery procedure that benefits from the local routing information of the zones.

ZRP is composed of three sub-protocols: the *IntrA-zone Routing Protocol (IARP),* the *IntEr-zone Routing Protocol (IERP)* and the *Bordercast Resolution Protocol (BRP)*. *IARP* is a limited scope proactive routing protocol while *IERP* is the reactive routing component of *ZRP*. *IARP* maintains routing information for nodes that are within the routing zone of the node. Correspondingly, *IERP* offers enhanced route discovery and route maintenance services based on the local connectivity monitored by *IARP*. To reduce traffic when global route discovery is needed, *ZRP* uses bordercasting instead of broadcasting packets. Bordercasting utilizes the topology information provided by *IARP* of each node to direct query request to the border of the zone. The bordercast packet delivery service is provided by the *BRP*. *BRP* uses a map of an extended routing zone to construct bordercast trees for the query packets.

The advantage of *ZRP* protocol is that it considerably reduces the amount of communication overhead when compared to pure proactive protocols. It also reduces delays associated with pure reactive protocols by discovering routes more rapidly (Vijayakumar & Ravichandran 2011). On the other hand, for large values of routing zone, *ZRP* behaves like a proactive protocol while for small values it could behave like a pure reactive protocol (Abolhasan et al. 2004). In general, topology-based routing protocols are considered not to scale in networks with more than several hundred nodes (Cao & Xie 2005).

2.4.4 Securing Topology-Based Routing Protocols

We note that none of the Ad-Hoc routing protocols mentioned above defined their security requirements and that they inherently trust all participants. Obviously, this could result in security vulnerabilities and exposures that could easily allow routing attacks, such as dropping or modifying the routing messages (Gera et al. 2011).

Since then, many works were done on secure routing protocols such as *Security-aware Ad-Hoc Routing (SAR)* (Yi et al. 2001), *Secure Efficient Ad-Hoc Distance vector* *routing protocol (SEAD)* (Hu et al. 2002), *Secure Link State Protocol (SLSP)* (Campbell et al. 2002), *Secure Routing Protocol (SRP)* (Papadimitratos & Haas 2002), *Secure Ad-Hoc On-demand Distance Vector (SAODV)* (Zapata 2002), *ARIADNE* (Hu et al. 2002) and *Authenticated Routing for Ad-Hoc Networks (ARAN)* (Sanzgiri et al. 2005). We look at this in detail in the following sections.

2.4.4.1 Secure Ad-Hoc On-demand Distance Vector (*SAODV*)

As the name indicates, *SAODV* is a secure extension of *AODV*. The main objective of *SAODV* is to ensure the integrity, authentication and non-repudiation of *AODV* routing information. *SAODV* uses two mechanisms to secure messages which are digital signatures to authenticate the non-mutable fields of the packets, in addition to hash chains to secure the hop count information which is the only mutable information in the messages.

Although *SAODV* is an extension of *AODV* and their performance characteristics are similar, the known problems of *AODV*, such as the lack of scalability, become a greater problem in *SAODV*. It is well-known that increasing the mobility in *AODV* results in increasing packet overhead. This becomes a more serious problem for *SAODV* since the processing of each packet requires some extra processing time because of its use of asymmetric cryptography. This accordingly may affect the performance of the low computational resource nodes, even without the presence of malicious nodes in the network (Fonseca & Festag 2006).

2.4.4.2 *ARIADNE* Protocol

ARIADNE improves *DSR* protocol by involving security functions using symmetric cryptography. It uses *Timed Efficient Stream Loss-tolerant Authentication (TESLA)* (Canetti et al. 2001), the broadcast message authentication scheme. *TESLA* assumes a maximum time synchronization error between all nodes participating in the protocol and uses the hash chain for the nodes in the network to authenticate packets from other nodes. *ARIADNE* has a good merit of requiring the source nodes to authenticate all intermediate nodes in the routing path using the *TESLA Message Authentication Code (MAC)* in the route reply packet.

On the other hand, *ARIADNE* has some drawbacks. The first one is that *ARIADNE* assumes the time synchronization between the nodes and maximum transmission delay of the links. These assumptions are somewhat impractical and impose unrealistic constraints upon Ad-Hoc networks (Pirzada & McDonald 2008). The second weak point of this protocol is the use of hash chain. In *ARIADNE*, all nodes must know the hash chain values of other nodes. These hash chain values must be updated after the last value of the hash chain is used. For updating hash value, nodes must perform the public key operation, which also decreases the efficiency merits of *ARIADNE* (Lee et al. 2003).

In regard to performance, it is worth noting that every intermediate node increases the signaling messages length (route request and route reply) which could result in large signaling packets for long routes to occur (Lee et al. 2003). Additionally, the time delayed key disclosure increases the end-to-end delay of route discovery processes. Both issues negatively impact the packet delivery ratio especially for vastly mobile scenarios (Fonseca & Festag 2006). Finally, as other topology-based routing protocols,

ARIADNE suffers from scalability problems due to blind broadcast of route discovery packets.

2.4.4.3 Authenticated Routing for Ad-Hoc Networks (*ARAN*)

A particular protocol of interest is *Authenticated Routing for Ad-Hoc Networks (ARAN)*. Effectively, *ARAN* is similar to *AODV*, but provides authentication of route discovery, setup and maintenance. The main intention of *ARAN* is to protect routing packets against attacks from malicious nodes in a managed-open environment where no network infrastructure is pre-deployed. *ARAN* requires some security coordination before deploying the nodes. It requires the use of a trusted *Certificate Authority (CA)* server whose public key is known by all valid nodes. Before entering the Ad-Hoc network, each node requests a certificate from this *CA*. *ARAN* uses cryptographic certificates to prevent most of the security attacks that Ad-Hoc routing protocols face. It introduces authentication, message integrity and non-repudiation as part of a minimal security policy for the Ad-Hoc environment.

ARAN consists of a preliminary certification process followed by a route instantiation process. Route discovery in *ARAN* is accomplished by broadcasting a *Route Discovery Packet (RDP)* from a source node, which is replied to by a unicast *REPly (REP)* packet that is launched from the destination, and sent back along the reverse path to the source. Routing messages are authenticated end-to-end and only authorized nodes participate at each hop from source to destination, as well as on the reverse path from the destination to the source. Hence, every node that forwards a request or a reply must also sign it so that the following node can check the validity of the previous one.

ARAN requires that nodes keep one routing table entry per source-destination pair that is currently active. This is certainly more costly than per-destination entries in non-secure Ad-Hoc routing protocols. Although there is a greater performance cost to *ARAN* compared to *AODV*, the increase in cost is minimal and outweighed by the increased security. Compared to basic *AODV*, *ARAN* prevents a number of attacks including altering routing messages, misrepresenting node’s identity (spoofing attack) and injecting into the network routing messages that have been captured previously (replay attack). Moreover, simulation results in (Sanzgiri et al. 2005) show that *ARAN* has a good performance equivalent to *AODV* in discovering and maintaining routes. Also, computational delays associated with *ARAN* protocol are comparable to the mandatory authentication delays required by *TESLA* that is used with *ARIADNE*.

On the other hand, besides scalability problem with the number of nodes (which is inherited from *AODV*) *ARAN* incurs more packet overhead and higher latency in route discovery due to signing each packet. Finally, *ARAN* uses one certificate server which leads to an extreme need to keep this server uncompromised, having a centralized certificate authority in physically insecure environments forms a single point of compromise and capture reducing protocol's availability and robustness against attacks (Pirzada & McDonald 2008).

2.4.5 Summary of the Discussed Topology-Based Routing Protocols

Table 2.3 summarizes the discussed non-secure topology-based routing protocols whereas Table 2.4 summarizes the secured ones.

Table 2.3: Non-secure topology-based routing protocols

Protocol **Criterion**	***DSR*** **(Johnson & Maltz 1996)**	***AODV*** **(Perkins & Royer 1999)**	***ZRP*** **(Beijar 1998)**
Approach	Reactive	Reactive	Hybrid
Main idea/ contribution	Initiates a route discovery process only when the route is needed.	Initiates a route discovery process only when the route is needed.	Aims to combine best properties of both proactive and reactive approaches.
Proposal	Performs source routing with the list of intermediate nodes addresses obtained from the query packet.	Uses next hop information stored in the nodes of the route with the least number-of-hop field.	Packets are routed using a proactive routing protocol if the destination is close to sender and a reactive one in long distances.
Scalability	Low	Low	Low
Advantages	No periodic routing packets are required.	No periodic routing packets are required.	Reduced control overhead compared to pure proactive protocols and reduced delays associated with pure reactive ones.
Disadvantages	• Uses blind broadcast to discover routes, which increases the control overhead. • Long route request delays. • Source routing may result in large signaling packets. • May have security vulnerabilities.	• Relies on blind broadcast to discover routes resulting in increasing the control overhead. • Long route request delays. • May have security vulnerabilities.	• Inherits proactive protocols disadvantages for large routing zone and those of reactive ones for small routing zones. • May have security vulnerabilities.

Proactive routing protocols try to discover and maintain a complete set of routes for the network's lifetime (Yau et al. 2007). Hence they are excluded as we believe that they are less suitable for Ad-Hoc networks than their reactive counterparts since they constantly consume power throughout the network, regardless of the presence of network activity.

It is clear that none of the Ad-Hoc routing protocols listed in Table 2.3 define its security requirements and that they inherently trust all participants. This could result in security vulnerabilities that could easily allow routing attacks including modification and fabrication of routing packets and impersonation of other nodes. Even the secured protocols listed in Table 2.4 concentrate on a part of the problem and most of them suffer from a single point of compromise and failure as well as scalability problem. In general, topology-based protocols are considered to not scale in networks with more than several hundred nodes (Cao & Xie 2005).

Table 2.4: Secured topology-based routing protocols

Protocol **Criterion**	***SAODV*** **(Zapata 2002)**	***ARIADNE*** **(Hu et al. 2002)**	***ARAN*** **(Sanzgiri et al. 2005)**
Approach	Reactive	Reactive	Reactive
Secure extension for	*AODV*	*DSR*	*AODV*
Basic security mechanism	Digital signatures and hash chains.	Symmetric cryptography primitives, hash functions and timestamps.	Certificates and timestamps.
Synchronization	No	Yes	No
Central trust	Certificate Authority	Key Distribution Center	Certificate Authority
Main idea/ contribution	Providing integrity, authentication and non-repudiation of *AODV* routing information.	Extending *DSR* by security functions using symmetric cryptography.	Protecting routing packets against attacks from malicious nodes in managed-open environments.

Table 2.4: Secured topology-based routing protocols (continued)

Protocol Criterion	***SAODV*** **(Zapata 2002)**	***ARIADNE*** **(Hu et al. 2002)**	***ARAN*** **(Sanzgiri et al. 2005)**
Proposal	Uses digital signatures to authenticate the non-mutable fields of messages, and hash chains to secure the hop count information which is the only mutable information in the packets.	Uses *TESLA* as the authentication protocol and the hash chain to authenticate packets of other nodes.	• Provides authentication of route discovery, setup and maintenance. • Uses cryptographic certificates to prevent most security attacks that face Ad-Hoc routing protocols. • Routing messages are authenticated at each hop.
Scalability	Low	Low	Low
Authentication	Yes	Yes	Yes
Confidentiality	No	No	No
Integrity	Yes	Yes	Yes
Non-repudiation	Yes	No	Yes
Advantages	Secure on-demand routing.	Secure on-demand routing.	Robust against most security attacks.
Disadvantages	• Single point of compromise and failure. • Increased packet overhead and route discovery delay, compared to original *AODV*, due to signing each packet.	• Impractical and unrealistic assumptions as assuming the time synchronization between the nodes and maximum transmission delay of the links. • All nodes must know the hash chain values of each other. • May have large signaling packets for long routes. • Increased route discovery delay compared to *DSR*, *SAODV* and *ARAN*.	• Single point of compromise and failure. • Increased packet overhead and route discovery delay, compared to original *AODV*, due to signing each packet.

2.5 Position-Based Routing Protocols

Recently, research has shown that position-based routing protocols exhibit better scalability and performance (Cao & Xie 2005; Giruka & Singhal 2005). Position-based routing protocols use the geographical position of nodes to make routing decisions that results in improved efficiency and performance (Prakash et al. 2011). Although each of these protocols employs different techniques, their basic idea is similar, that is, only nodes resulting in forward progress toward the destination are supposed to be involved in the route discovery process, causing a decrease in the overall routing overhead (Carter & Yasinsac 2002).

These protocols require a node to be aware of its own geographical position and to be able to obtain the geographical position of the destination. Generally, this information is obtained via *GPS* and location services (Carter & Yasinsac 2002). There are different kinds of position-based routing protocols and they are categorized into three main groups: *greedy forwarding*, r*estricted directional flooding* and *hierarchical routing protocols*. These groups are discussed in the following three sections. Afterward, the work done to secure position-based routing is highlighted. Lastly, a summary of the discussed position-based routing protocols is presented.

2.5.1 Greedy Forwarding

Most position-based protocols, such as *Greedy Perimeter Stateless Routing (GPSR)* (Karp & Kung 2000), use greedy forwarding to route packets from a source to the destination. In greedy forwarding, a source node selects a neighbouring node that is closest to the destination as the next hop. Similarly, each intermediate node selects a next hop node until the packet reaches the destination. In order to enable the nodes to do this, nodes periodically broadcast small packets (called beacons) to announce their positions and enable other nodes to maintain a one-hop neighbour table.

Such an approach is scalable since it does not need routing discovery and maintenance (Wu 2005). However, periodic beaconing creates a lot of congestion in the network and consumes the nodes’ energy (Cao & Xie 2005; Giruka & Singhal 2005). In addition, *GPSR* uses link-layer feedback from *Media Access Control (MAC)* layer to route packets and such feedbacks are not available in most *MAC* layer protocols (Giruka &

Singhal 2005). Finally, Greedy forwarding in general may not always find the optimal route (Wu 2005). *GPSR*, for example, works well in dense networks, but in sparse networks, greedy forwarding fails due to voids (regions without nodes) (Karp & Kung 2000).

Another scalable position-based routing protocol is *Angular Routing Protocol (ARP)* (Giruka & Singhal 2005). In *ARP*, nodes emit a hello packet on a need-basis (non-periodic) at a rate proportional to their speeds. These hello packets enable each node to maintain a one-hop neighbour table. *ARP* uses geographic forwarding to route packets to the destination. If the geographic forwarding fails, it uses an angle-based forwarding scheme to circumvent voids in sparse networks.

Some other position-based routing protocols, such as *Most Forward within distance R (MFR)* (Takagi & Kleinrock 1984), try to minimize the number of hops by selecting the node with the largest progress from the neighbours, where progress is defined as the projection of the distance of the next hop from the sender on the straight line between the sender and the destination. Others, such as *compass routing algorithms (DIR)* (Kranakis et al. 1999) is based on forwarding the packet to the immediate neighbour that minimizes the angle between the node itself, the previous node and the destination node.

All the aforementioned position-based routing protocols use forwarding strategies based on distance, progress or direction. *Improved Progress Position-Based BeaconLess Routing algorithm (I-PBBLR)* (Cao & Xie 2005) combines the traditional progress with the direction metric to form the improved progress definition. Moreover, it eliminates the drawbacks associated with beaconing by using a beaconless protocol. In *I-PBBLR*, if a source node has a data packet to be sent, it initially determines the position of the destination node and stores these geographical coordinates along with its own current position in the data packet header.

Intermediate nodes replace the preceding node’s position in the header with their own current position before forwarding the packet. Since a node does not have knowledge of neighbouring nodes, it simply transmits the packet to all its immediate neighbours. Nodes located within the forwarding area of the relaying node, apply *Dynamic Forwarding Delay (DFD)* before relaying the packet, while nodes outside this area drop the received packet. The value of the *DFD* depends on the relative position coordinates of the current, previous and destination nodes. Ultimately, the node that computes the shortest *DFD* forwards the packet first and other nodes existing in the forwarding area detect the further relaying of the packet and cancel their scheduled transmission of the same packet.

The simulation results in (Cao & Xie 2005) show that position-based beaconless routing using the improved progress decreases the overhead and increases the delivery rate compared to the traditional progress.

2.5.2 Restricted Directional Flooding

Location-Aided Routing (LAR) (Ko & Vaidya 2000) is an example of restricted directional flooding routing protocols, within which the sender broadcasts the packet to all single-hop neighbours towards the destination. In order to find the shortest path in the network level, instead of selecting a single node as the next hop, several nodes are selected for managing the route request message. In *LAR*’s approach, the node that receives the route request message compares its distance to the destination with the distance of the previous hop to the destination. If the receiver node is closer to the destination, it retransmits the route request message, otherwise, it will drop the message. Using restricted directional flooding helps *LAR* in exhibiting high scalability and performance.

Each intermediate node puts its *IP* address in the header of the request packet before retransmitting it. Therefore, the route through which the route request message is passed will be saved in the header of the message, causing the size of the message to increase (Kalhor et al. 2007). In *LAR*, if the discovered route breaks for any reason, the route discovery process will have to start again. This problem is solved by *Location-Aided Routing With Backup (LARWB)* (Kalhor et al. 2007) where another route is selected as a backup route, which is then used when a breakage occurs in the primary route.

2.5.3 Hierarchical Routing Protocols

TERMINODES (Blazevic et al. 2001) is an example of a hierarchical routing protocol. *TERMINODES* presents a two-level hierarchy within which, if the destination is close to the sender (in number of hops), packets are routed based on a proactive distance vector. Otherwise, greedy routing is used for longer distances. *TERMINODES* addresses three main objectives, which are scalability (in terms of both the number of nodes and the geographical coverage), robustness and collaboration of the involved nodes (Giordano et al. 2003).

2.5.4 Securing Position-Based Routing Protocols

All the above mentioned position-based routing protocols are vulnerable to some attacks, as they focus on improving performance while disregarding security issues (Mizanur Rahman et al. 2006). For the past years, limited work has yet been done to address the security issues of position-based routing protocols. Examples of these are *Secure Position-Aided Ad-Hoc Routing (SPAAR)* (Carter & Yasinsac 2002), *Anonymous On-Demand Position-based Routing in mobile Ad-Hoc networks (AODPR)* (Mizanur Rahman et al. 2006), *Secure Geographic Forwarding (SGF)* (Song et al. 2007), *Location-Aided Secure Routing Scheme (LASR)* (Lee et al. 2007; Lee et al. 2008) and *Location Secure Routing Protocol (LSR)* (Xu & Cai 2010). The consequent sections discuss these protocols in detail.

2.5.4.1 Secure Position-Aided Ad-Hoc Routing (*SPAAR*)

SPAAR uses information about nodes' positions to assist in improving the efficiency and security of mobile Ad-Hoc networks. It is designed to be deployed in managed-hostile environments where security is a primary concern. It uses geographical information to make forwarding decisions, resulting in a significant reduction in the number of routing messages. *SPAAR* provides the necessary requirements to secure routing in a high-risk environment, i.e. authentication, non-repudiation, confidentiality and integrity. It makes use of asymmetric cryptography to guard the network against malicious nodes and attempts to minimize the probable damage from attacks launched by compromised nodes (Fonseca & Festag 2006).

Two of the well-known attacks are the invisible node attack and wormhole attack. In the invisible node attack, a malicious node may forward a packet without appending its address to the address field of that packet. On the other hand, the wormhole attack involves the cooperation between two malicious nodes sharing a private communication. One attacker captures routing packets at one point of the network and tunnels these packets to another point. The new attacker then selectively injects the tunnelled traffic back into the network. *SPAAR* prevents both the invisible node attack and the wormhole attack by allowing the nodes to accept routing messages only from one-hop neighbours.

To participate in *SPAAR*, each node needs a public/private key pair, a certificate binding its identity to its public key (signed by a *Certificate Authority (CA)* server) as well as the public key of that *CA*. Additionally, each node keeps two keys for each neighbour. The first is the neighbour's public key that is obtained from its certificate and used to encrypt some routing messages such as *Route REPly (RREP)*. The second is the neighbour's group decryption key that is used to decrypt some routing messages such as *Route REQuest (RREQ)* to verify that the sender is a one-hop neighbour.

Route instantiation is triggered by the source through broadcasting a *RREQ* that is encrypted with its group encryption key. *SPAAR* uses a *RREQ* sequence number that is incremented each time a node initiates a *RREQ* and used to prevent replays of *RREP* packets. Nodes receiving *RREQ* packets decrypt it with the appropriate group decryption key to verify that the sender of the packet is a one-hop neighbour. Intermediate node tests if it or any of its neighbours is closer to destination. If so, it will encrypt the *RREQ* with its group encryption key, forward the *RREQ* and record the address of the predecessor neighbour. If not, the *RREQ* packet is discarded. This process is repeated until the *RREQ* packet reaches the destination.

Upon getting a *RREQ*, the destination constructs a *RREP* packet signed with its private key and encrypted with the public key of the neighbour it received the *RREQ* from. The *RREP* packet is sent through the reverse path of the *RREQ*, being verified at each hop. When an intermediate node receives a *RREP*, it decrypts the *RREP* with its private key and verify the signature with the public key of the neighbour node they received the *RREP* from. Then, they sign the *RREP* and encrypt it with the public key of the subsequent node in the reverse path. Upon receiving the *RREP*, and after successful decryption and signature verification, the source node begins sending data.

The fact that *SPAAR* makes use of geographic routing helps in reducing the overall overhead of routing packets. *SPAAR* also provides a high security level against malicious as well as compromised nodes. However, it requires double the processing time since it uses asymmetric cryptography, not only for end-to-end communication, but also for hop-to-hop communications (Fonseca & Festag 2006). In large area networks the probability of having long routes will increase, and since each node spends time in signing and encrypting the messages, the probability of node movements and route breakage will increase. In addition, *SPAAR* has a centralized trust, and so, suffers from the compromised server problem and the single point of failure. Hence, *SPAAR* is considered to have a medium scalability.

2.5.4.2 Anonymous On-Demand Position-Based Routing in Mobile Ad-Hoc Networks (*AODPR*)

Due to the dynamic topology, infrastructure-less and broadcast nature of *MANET*s, communications in these networks are vulnerable to malicious traffic analysis. As a subsequent step, an attacker may determine a target node and conducts an intensive attack against it, called target-oriented attack (Mizanur Rahman et al. 2006). *AODPR* keeps routing nodes anonymous, thereby preventing possible traffic analysis. A time variant temporary identifier is computed from the time and the position of a node in an attempt to keep the node anonymous.

AODPR uses the concept of *Virtual Home Regions (VHR)* which is a geographical region around a fixed center. In this scheme, each node stays in one of the *VHR*s. Nodes within a *VHR* obtain their own geographic position through *GPS* and report their position information to the *Position Servers (PS*s*)*. *PS*s are trusted Ad-Hoc nodes distributed in the network. The *PS* keeps the position information of the nodes securely. Upon joining the network, a node registers with the *PS* and gets a *Common Key (CK)* and a public/private key pair from the *PS*.

When a node wants to get position information of other nodes, it first sends a signed request and authenticates itself to the *PS*. Accordingly, *PS* provides it with the required position information, public key of the destination and other needed information. Then, the source estimates the minimum *Number of Hop (NH)* that the route request packet travels to find a route from the source to the destination. Each intermediate node decrements *NH* by one and compares the updated *NH* with the minimum number of hop which route request packet travels to find a route from this node to the destination (*NH*'). If *NH*' is less than or equal to *NH*, then the intermediate node forwards the packet to its neighbours and keeps the needed route information, else it discards the packet. Both *NH*' and *NH* are calculated depending on the distance from the node to the destination and the radius of the maximum radio range coverage of each node.

To improve the security of their protocol, the destination's position is encrypted with *CK* on the route request phase, hence there is no position information exposure to nodes outside the intended network. After authenticating the sources, the destination sends a route reply and keeps the route information to itself. When it receives the route reply and has authenticated the destination, the source begins sending the data encrypted by the destination's public key. If the source receives a fail packet, then it tries again with a new larger estimated *NH*.

AODPR is robust against the wormhole attack in which an attacker records a packet in one point of the network and sends it to another location by constructing a tunnel and later retransmits the packet to the network under the attacker's control. Therefore, the packet might travel a long distance before finding the route from source to destination. In *AODPR*, the source as well as intermediate nodes wait for a limited time (depending on the estimated *NH*) to get response. If the attacker response exceeds the limited time, then he cannot be a forwarder within a routing path. Hence, wormhole attack is not effective in *AODPR*.

Even if the *AODPR* is applicable to any node density in a network, ensures the anonymity of both route and nodes, and robust against the target-oriented attack (Mizanur Rahman et al. 2006), it suffers from many problems. Many fields, such as *NH* and destination's position sent by *PS*s, are encrypted using a common key. If this key is compromised, a large percentage of the communication in the whole network will be compromised. Moreover, *AODPR* suffers from two problems inherited from the *VHR* approach. First, nodes may be hashed to a distant *VHR* than that they are currently residing in, leading to increased communication and time complexity, as well as problems if the *VHR* of a node cannot be reached. Second, since an Ad-Hoc network is dynamic, it might be difficult to guarantee that at least one position server will be present in a given *VHR* due to regions not including nodes. For all these reasons, *AODPR*'s scalability is considered as medium.

2.5.4.3 Secure Geographic Forwarding (*SGF*)

In (Song et al. 2007) the *SGF* mechanism has been proposed. It provides source authentication, neighbour authentication and message integrity via its use of both the shared key and the *Instant Key disclosure (TIK)* protocol (Hu et al. 2003). By combining *SGF* with the *Grid Location Service (GLS)* (Li et al. 2000), the authors have proposed the *Secure Grid Location Service (SGLS)* where any receiver is able to verify the correctness of location messages. In this paper also, a *Local Reputation System (LRS)* has been proposed aiming to detect and isolate compromised as well as selfish users.

SGF mechanism incorporates both the hashed *Message Authentication Code (MAC)* (Krawczyk et al. 1997) and the *Timed Efficient Stream Loss-tolerant Authentication (TESLA)* (Perrig et al. 2004) with *TIK* protocol. The *MAC* is computed over the non-mutable part (e.g. location information of a destination) of unicast messages with the pair-wise shared secret key between the source and the destination nodes.

Instead of introducing overhead by signing the destination's location information of all data and control messages, the authors proposed the use of a reputation system, *LRS*, for classifying nodes as good or bad, as well as detecting and isolating message tampering and dropping attackers. The *TIK* protocol with tight time synchronization is used to authenticate a preceding forwarding node in order to prevent malicious users from joining a path and to avoid a message replay attack, i.e. re-sending recorded old valid control messages. Finally, when the destination receives a message, it is capable of verifying the authenticity of the message by comparing the received *MAC* to the *MAC* value that it computes over the received message with the secret key it shares with the source node.

In combination with *SGF*, the secure location service, *SGLS*, is proposed by combining *SGF* with the *GLS* so that any receiver can verify the correctness of location messages. The original *GLS* is a distributed location service in which each node maintains information about locations of specific subsets of the nodes based on the nodes' identifiers. *GLS* divides the *MANET* area into a hierarchy of squares. Each node periodically broadcasts the list of neighbours it has. Consequently, each node involved in the network can keep a table of immediate neighbours as well as each neighbour's neighbours. Each node enlists nodes with identities close to its own identity to serve as its location servers by sending location update messages.

The general concept of the proposed *SGF* can generally be applied to any unicast message of *GLS*, such as location query and location reply. So, the one-hop neighbour's location information can be verified by the use of a location verification technique (Capkun & Hubaux 2005) and the *TIK* protocol can be used for neighbour authentication. *TESLA* broadcast authentication method is used to verify the location information of the two-hop neighbouring nodes.

Unlike other messages, the location update message has no assigned destination address field in it. Thus, it is unfeasible to provide source authentication with a symmetric secret key. Hence, a public key infrastructure is assumed in the *MANET* under consideration. Each node stores the public key of the trusted *Certificate Authority (CA)* and signs the location update message with its private key.

Although there are several forwarding strategies, such as those discussed in *Section 2.5.1*, they all forward a given message to a single optimal neighbouring node based on their optimization criterion. Therefore, *SGF* can be applied to any of these forwarding strategies without any modification.

The simulation results in (Song et al. 2007) show that *SGLS* can operate efficiently by using effective cryptographic mechanisms. Results also demonstrate that *LRS* effectively extracts message dropping attackers from the network. On the other hand, the simulations illustrate that the average end-to-end delay for *SGLS* is a little higher than that of *GLS*, and that *SGLS*'s routing overhead is significantly higher than that of *GLS*. This increase in *SGLS*'s routing overhead obviously results from the larger size of routing control messages due to digital signatures and *MAC*s (Song et al. 2007).

Generally, systems using reputation system along with the a cryptography scheme in order to defend against both compromised and malicious nodes do not scale well as they have to track the reputation of all nodes, which may require huge tables of information that are difficult to manage and to keep up-to-date (Fonseca & Festag 2006). Moreover, *SGF* assumes the existence of pair-wise shared secret keys between the nodes that is difficult to implement in large area networks (Fonseca & Festag 2006). Additionally, *SGF* uses greedy forwarding that is not guaranteed to find an optimal path and suffers from congestion and high nodes' energy consumption due to periodic beaconing (Giruka & Singhal 2005). Finally, *SGF* assumes all nodes have tightly synchronized clocks which is somewhat impractical for Ad-Hoc networks, but is possible by using *GPS* (Fonseca & Festag 2006). Consequently, these problems result in limiting the scalability of *SGF*.

2.5.4.4 Location-Aided Secure Routing Scheme (*LASR*)

In (Lee et al. 2007) and (Lee et al. 2008), the *LASR* protocol has been proposed aiming to reduce the number of intermediate nodes in routing paths and to guarantee secure route establishment. The authors assume that every node knows the current positional information of all other nodes existing in a routing domain. In a try to minimize the number of hops, each sender node selects as its next hop the neighbour with the largest progress on the projection-line between the sender and the destination nodes. In *LASR*, the progress is referred to as the shadow-line.

Upon starting the route discovery, the sending node calculates the shadow-line values of its neighbours. This is possible as the sending node can obtain its position and is aware of the positional information of its neighbours and the intended destination node. Later, the neighbours' shadow-line values are used by the sending node to detect malicious behaviour of any neighbour. To establish a reliable and secure routing path, three types of *Route REQuest (RREQ)* packet are defined in *LASR*: inquiry *RREQ* (I-type *RREQ*), candidate *RREQ* (C-type RREQ) and decision *RREQ* (D-type *RREQ*).

As a first step, the sending node requests the shadow-line of its neighbours by broadcasting an I-type *RREQ* packet including the positional information of itself and the destination node. Each neighbour, upon receiving an I-type *RREQ* packet, calculates its shadow-line and sends it back to the sending node using a C-type *RREQ* packet. The sending node, after receiving a C-type *RREQ* packets from its neighbours, temporarily selects a node with the largest shadow-line as a next intermediate node along a routing path. Now, the sending node compares the shadow-line value calculated by itself with the value received from this neighbour. If the two values are not identical, the number of malicious behaviours of the neighbouring node is increased by one. If identical, the sending node additionally examines whether the number of malicious behaviours (actions) of the selected node is less than the maximum permissible value. If the two conditions are satisfied, the selected node is determined as the next hop node. Otherwise, if either the two shadow-line values are not identical or the number of neighbour's malicious behaviours is larger than the maximum value, this neighbour is excluded from being selected as an intermediate node, and the neighbour with the second-largest shadow-line is selected as the next intermediate node. By repeating this process, a reliable and trusted next hop node is selected. The following step involves the sending node initiating a D-type *RREQ* packet and broadcasting this packet to its neighbours. The selected neighbour will eventually perceive itself as the next hop node and look for a next intermediate node that is the nearest to the destination and can guarantee safety.

After the establishment of the route, if any intermediate node on the routing path detects the breakage of a link between itself and its next hop, it carries out a route re-establishment to the destination node. Instead of generating a route error message to the source, the intermediate node rebroadcasts a *RREQ* to its neighbours seeking for another next hop closer to the destination and providing safety (reliability and security). The use of a local repair strategy helps in reducing control overhead and latency to find another path from the source to the destination, especially that *LASR* uses three types of *RREQ* packet.

Simulated performance evaluation of *LASR* shows that the average packet delivery success ratio of *LASR* decreases as mobility increases. The reason behind the relatively lower performance in the case of high mobility is link disconnections. Higher link disconnections lead to dropping some data packets, resulting in a reduced delivery success ratio. Generally, *LASR* and *LAR* have similar performance in terms of packet delivery success ratio.

The average routing overhead of *LASR* is higher than that for *LAR* and the gap increases as the mobile nodes move rapidly. This increase in *LASR*'s routing overhead is expected due the use of three types of *RREQ* packet to exclude malicious nodes from the route establishment process. Hence, *LASR*'s higher control message overhead is the trade off for its network security and reliability.

Results also show that the used paths in *LASR* are shorter than those used in *LAR*. *LAR* chooses the routing path with the first arrived *RREQ* packet to the destination. On the other hand, the sending node in *LASR* chooses the nearest neighbour to the destination as the relay node. This strategy helps in reducing the number of hops in the used routes.

In summary, *LASR* has proven that it is able to prevent malicious nodes from being included in the route establishment process. Moreover, it is able to discover the shortest paths even in high mobility scenarios. However, *LASR* uses greedy forwarding, and hence, exhibits the disadvantages associated with it including not guaranteeing to find the optimal route especially in sparse networks (Giruka & Singhal 2005). Moreover, the cost of *LASR*'s high security and reliability is the increased routing overhead due to the use of three types of *RREQ* packet. Furthermore, assuming that all nodes are aware of the positions of all other nodes in the network requires each node to periodically broadcast its position to other nodes. Periodic broadcast results in extra overhead and consumes the nodes' energy. Also, the compromise of a single node results in the positional information of all nodes in the network to be exposed. Thus, *LASR* is considered to have medium scalability.

2.5.4.5 Location Secure Routing Protocol (*LSR*)

Position-based routing protocols rely on nodes' location information for path discovery and construction, which requires participating nodes to expose their locations to other nodes. The location information included in a packet can be used to identify the sender of the packet. Moreover, recognizing a spatial region containing a set of nodes persuades an adversary to physically locate and destroy these nodes (Xu & Cai 2010).

The main idea of *LSR* protocol (Xu & Cai 2010) is that a node can report a cloaking region as its location, instead of disclosing its exact location. A cloaking region of a node is a spatial region that contains its current position (Ghinita et al. 2010). Using cloaking regions helps in achieving both privacy and safety protection. *Privacy protection* means preventing a location from being linked to some specific node. *Safety protection* ensures that the disclosed location information cannot be used to identify any spatial region with a node density that exceeds a pre-defined threshold. Ensuring safety protection is important since a spatial region with denser nodes is more attractive for an adversary to locate the nodes and destroy them (Xu & Cai 2010).

In (Xu & Cai 2010), authors assume that the adversary has access to the communications among the networking nodes (it can be one of these nodes) and that it knows the nodes' location information which they disclose during packet delivery. The adversary is interested in the location refinement attack, which is to derive more accurate location information than reported. The key to prevent such attack is to ensure that data packets are forwarded only through safe links. A network link is considered as a safe link if the packet delivery through the link does not allow an adversary to refine the sender and receiver location resolution. To verify if a link is safe or not, a node receiving a packet needs to check if the cloaking region it discloses is completely covered by the sender's transmission range.

In *LSR*, a packet is routed using greedy routing whenever possible, and if not, it is detoured along some faces in the network connectivity graph. When a node receives a data packet, it checks if the link is safe and if it is closer to the destination than the sender. If either condition is not satisfied, the packet is dropped. Otherwise, a node waits for a certain period of time, during which if the node recognizes that the same packet has been forwarded by another node, it drops the packet. If not, it forwards the packet. The length of the waiting period is proportional to the distance between the intermediate node and the destination node. In other words, a node closer to the destination waits a shorter period and has a higher probability to forward the packet. This strategy does not require nodes to periodically advertise their latest location to neighbours (as it is the case in most greedy forwarding protocols, like *GPSR* (Karp & Kung 2000)), and therefore, *LSR* has lower overhead and high scalability.

The forwarding node also sends an acknowledgement packet back to the sender with a transmitting power that is ensured to cover the sender's cloaking circle. If the sender node does not receive any acknowledgement, this is an indication that there is no safe link to any other node closer to the destination. In this case, the packet reaches a dead-end in greedy routing mode and the packet forwarding switches to face routing mode. In face routing mode, the packet delivery is detoured around the dead-end until a closer next-hop is found.

LSR's performance has been evaluated via simulation. For comparison purpose, another approach referred to as *native* has been implemented. In this scheme, a node forwards a received packet as long as it is closer to the destination than its previous hop. Unlike *LSR*, nodes in this scheme forward a data packet without verifying link safety and it is, therefore, exposed to location refinement attack.

Simulation results show that *LSR* has a lower delivery rate compared to *native* and that *LSR*'s delivery rate decreases as the size of the cloaking circles increases. In *LSR*, a packet is forwarded only via safe links. When a cloaking circle is larger, the chance of having it totally covered by a node's transmission range reduces. Thus, when the size of cloaking circles increases, the number of safe links in the network reduces, making it more difficult to find a safe route. On the other hand, *native*'s high data delivery rate is achieved at the expense that the locations of nodes are known more accurately to adversary nodes.

The delivery rate of *native* is not affected by network density. This stable delivery rate is due to the fact that the next hop in *native* is always selected as the closest neighbour to the destination, whether the selected link is save or not. On the other hand, the delivery rate of *LSR* increases as the network becomes denser. In dense networks, nodes are closer to each other, resulting in more safe links. Consequently, the chance of finding a safe route from a source to a destination increases.

Upon studying the impact of transmitting power, the results show that the delivery rate of *LSR* increases with increasing the transmitting power. A larger transmission range covers more nodes' cloaking circles, i.e. generates more safe links. More safe links in a network makes it easier to find a safe path between the source and destination nodes.

As a summary, nodes in *LSR* report cloaking regions as their locations and the routing paths are constructed using safe links only. As such, *LSR* can work with inaccurate locations and it is robust against the location refinement attack. Sending data packets using only safe links increases *LSR*'s security on one hand and results in lower delivery rate on the other hand. *LSR* suffers low delivery rate especially in the cases of large cloaking circles, small transmission range and sparse networks. Finally, the used greedy forwarding technique eliminates the periodic advertisement of the nodes' latest locations, resulting in high scalability. However, this technique requires all nodes located within the transmission range of the relaying node to check if the link is safe and if they are closer to the destination even if they are not going to forward the packet.

2.5.5 Summary of the Discussed Position-Based Routing Protocols

Table 2.5 summarizes the discussed non-secure position-based routing protocols whereas Table 2.6 compares the secure ones. Most position-based protocols use greedy forwarding that requires periodic beaconing, creating a lot of congestion in the network and consuming the nodes' energy as well as having a low probability of finding the shortest path due to sending one copy of the packets. They are also vulnerable to some attacks as they focus on improving performance while disregarding security issues (Mizanur Rahman et al. 2006). Even secured protocols have increased processing and packet overhead, and some of them have a centralized trust, and so, suffer from the compromised server problem and the single point of failure.

Table 2.5: Non-secure position-based routing protocols

Protocol Criterion	***GPSR*** **(Karp & Kung 2000)**	***ARP*** **(Giruka & Singhal 2005)**	***MFR*** **(Takagi & Kleinrock 1984)**
Approach	Greedy	Greedy	Greedy
Main idea/ contribution	Increasing scalability and performance.	Circumventing voids in sparse networks.	Minimizing the number of hops by utilizing the progress concept.
Proposal	Each intermediate node forwards the packet to the closest neighbour to destination.	If the geographic forwarding fails, it uses an angle-based forwarding scheme.	Each intermediate node forwards the packet to the neighbour with the largest progress.
Scalability	High	High	High
Advantages	Does not need route discovery and maintenance.	Does not need route discovery and maintenance.	Does not need route discovery and maintenance.
Disadvantages	• Uses *MAC* layer feedback, which may not be available in most *MAC* layer protocols. • Periodic beacons lead to network congestion and nodes' energy consumption. • Low probability of finding the shortest path. • May fail to find a path at all in sparse networks. • May have security vulnerabilities.	• Hello packet lead to network congestion and nodes' energy consumption, however, the problem is not as bad as that in *GPSR* since nodes emit a hello packet at a rate proportional to their speeds (non-periodic). • May have security vulnerabilities.	• Periodic beacons lead to network congestion and high nodes' energy consumption. • Low probability of finding the shortest path. • May fail to find a path at all, even if one exists, especially in sparse networks. • May have security vulnerabilities.

Table 2.5: Non-secure position-based routing protocols (continued)

Protocol Criterion	***I-PBBLR*** **(Cao & Xie 2005)**	***LAR*** **(Ko & Vaidya 2000)**	***TERMINODES*** **(Blazevic et al. 2001)**
Approach	Greedy	Restricted directional flooding	Hierarchical
Main idea/ contribution	Eliminates the beaconing drawbacks.	Increases the probability of finding the shortest path by hiring several nodes to manage the route request.	Achieving scalability, robustness and nodes collaboration using a two-level hierarchy.
Proposal	• Uses a beaconless protocol utilizing *DFD*. • Combines the traditional progress with the direction metric to form the improved progress definition.	Intermediate node broadcasts packet to its neighbours only if it is closer to the destination than its predecessor.	Packets are routed base on a proactive distance vector if the destination is close to the sender, greedy routing is used in longer distances.
Scalability	High	High	High
Advantages	Reduced overhead and increased delivery rate compared to the traditional progress.	High probability of finding the shortest path.	Reduces control overhead compared to proactive protocols, and eliminates beacons' disadvantages used with greedy ones.
Disadvantages	• All nodes located within the forwarding area of the relaying node replace the predecessor's position with their current position and compute the *DFD* even if they will not forward the packet. • Low probability of finding the shortest path. • May fail to find a path in sparse networks. • May have security vulnerabilities.	• Several nodes manage the route request message resulting in higher routing overhead compared to greedy protocols. • If the discovered route breaks for any reason, the route discovery process has to be started again. • May have large signaling packets due to source routing. • May have security vulnerabilities.	• Inherits disadvantages of proactive protocols for large routing zone, and those of greedy ones for small routing zones. • May have security vulnerabilities.

Table 2.6: Secured position-based routing protocols

Protocol Criterion	***SPAAR*** **(Carter & Yasinsac 2002)**	***AODPR*** **(Mizanur Rahman et al. 2006)**	***SGF*** **(Song et al. 2007)**
Approach	Restricted directional flooding	Restricted directional flooding	Greedy
Basic security mechanism	Certificates and timestamps.	Both symmetric and asymmetric cryptography and hash functions.	Both symmetric and asymmetric cryptography and hashed *MAC* algorithm.
Synchronization	No	Yes	Yes
Central trust	Certificate Authority	Key Distribution Center	Certificate Authority
Main idea/ contribution	Uses cryptographic certificates to protect routing packets in managed-hostile environments.	Keeps routing nodes anonymous to prevent possible traffic analysis and target-oriented attack.	Provides source authentication, neighbour authentication and message integrity.
Proposal	• Intermediate node checks whether it or any of its neighbours is closer to the destination, and if so, it will encrypt the *RREQ* with its group encryption key so that recipients can decrypt it with the appropriate group decryption key and verify that the sender is a one-hop neighbour. • Intermediate node signs the *RREP* with its private key and encrypts it with the public key of the neighbour it received the *RREQ* from to enable this node to decrypt the *RREP* with its private key and verify the signature with the public key of the neighbour node it received the *RREP* from.	• Uses *VHR*s, nodes' positions are reported to *PS*s. • Each intermediate node decides to rebroadcast the route request packet or not depending on the distance from the node to the destination and the radius of the maximum radio range coverage of each node. • Destination's position is encrypted with *CK* in the route request phase. • After authenticating the source, the destination replies with a route reply. • Upon receiving the route reply and authenticating the destination, source begins sending the data encrypted by the destination's public key.	• Uses a reputation system to detect and isolate message tampering and dropping attackers as well as a secure location service to verify the correctness of location messages. • Incorporates *MAC* and *TESLA* with *TIK* protocol. • The *MAC* is computed over the destination's location with the pair-wise shared secret key between the source and destination to enable the destination to verify the authenticity of the message. • The *TIK* protocol is used to authenticate the predecessor and *TESLA* is used to verify the location information of two-hop neighbouring nodes.

Table 2.6: Secured position-based routing protocols (continued)

Protocol **Criterion**	***SPAAR*** **(Carter & Yasinsac 2002)**	***AODPR*** **(Mizanur Rahman et al. 2006)**	***SGF*** **(Song et al. 2007)**
Scalability	Medium	Medium	Medium
Authentication	Yes	Yes	Yes
Confidentiality	Yes	Yes	No
Integrity	Yes	No	Yes
Non-repudiation	Yes	No	No
Advantages	• Provides a high security level against malicious and compromised nodes in a high-risk environment. • Has a high probability of finding the shortest path. • Robust against invisible node and wormhole attacks.	• Applicable to any node density in a network. • Ensures the anonymity of both route and nodes. • Has a high probability of finding the shortest path. • Robust against target-oriented and wormhole attacks.	• Effectively detects and isolates message dropping attackers from the network. • Robust against the replay attack.
Disadvantages	• Requires high processing time, since it uses asymmetric cryptography, not only for end-to-end communication, but also for hop-to-hop communications. • Single point of compromise and failure.	• Large percentage of network's communication will be compromised if the *CK* is compromised. • Nodes may be hashed to a distant *VHR* leading to increased communication and time complexity, unreachable *VHR*s and problems due to void regions.	• Assuming pair-wise shared secret keys, assuming tightly synchronized nodes' clocks and tracking reputation of all nodes result in having scalability problems. • Large routing packets with digital signatures and *MAC*s. • Has low probability of finding the shortest path.

Table 2.6: Secured position-based routing protocols (continued)

Protocol Criterion	***LASR*** **(Lee et al. 2008)**	***LSR*** **(Xu & Cai 2010)**
Approach	Greedy	Greedy
Main idea/ contribution	Minimizes the number of intermediate nodes on a routing path and tries to achieve route establishment with security.	Uses cloaking region to reduce nodes' location resolution to achieve a desired level of protection, either privacy or safety.
Proposal	• Assumes that every node knows the position of all other nodes. • The sending node requests the shadow-line of neighbours by broadcasting I-type *RREQ* packet. • Each neighbour, upon receiving I-type *RREQ* packet calculates its shadow-line and sends it via C-type *RREQ* packet. • The shadow-line values are used by the sending node to detect malicious behaviour of its neighbours. • The sending node looks for a next intermediate node which is the nearest to the destination and can guarantee safety.	• Nodes reveal their locations only as cloaking circles. • Data packets are delivered only through safe links. To verify if a link is safe, a node receiving a packet checks if the cloaking region it discloses is completely covered by the sender's transmission range. • A packet is routed using greedy routing whenever possible, otherwise, it is detoured along some faces in the network connectivity graph.
Scalability	Medium	High
Advantages	• Effectively detects misbehaving nodes trying to send wrong shadow-line values and excludes them from the route establishment process. • The ability to discover shortest paths even in high mobility scenarios.	• Trying to achieve privacy and safety protection. • Can work with inaccurate location and construct a routing path using only safe links. • Robust against location refinement attack. • Does not require nodes to periodically advertise their locations.
Disadvantages	• Increased routing overhead due to suggesting the three types of *RREQ* packet. • Periodic position information broadcast results in extra overhead and consumes the nodes' energy. • The compromise of a single node results in the expose of the position of all nodes in the network. • Uses greedy forwarding, hence, it is not guaranteed to find the optimal route especially in sparse networks.	• All nodes located within the transmission range of the relaying node check if the link is safe and if they are closer to the destination even if they will not forward the packet. • Low probability of finding the shortest path. • Has a lower delivery rate compared to *native*, and the delivery rate decreases as the size of the cloaking circles increases or the transmitting power decreases. • The delivery rate of *LSR* decreases in sparse networks.

2.6 Discussion and Research Direction

Our findings from the conducted literature review are summarized as follows:

- Proactive topology-based routing protocols are less appropriate for Ad-Hoc wireless networks as they continually consume power throughout the network, regardless of having network activity, and have problems in networks with high-rate changing topologies.
- Reactive topology-based routing protocols are deemed more appropriate for wireless environments because they initiate a route discovery process only when there are data packets need to be routed. On the other hand, they suffer from scalability problems since they rely on blind broadcast to discover routes and request packet is propagated to all nodes in the network.
- In general, topology-based routing protocols are considered to not scale in networks with more than several hundred nodes. Moreover, a large percentage of them (such as *DSR* (Johnson & Maltz 1996) and *AODV* (Perkins & Royer 1999)), did not define their security requirements and inherently trust all participants (Gera et al. 2011). Even secured ones (such as *SAODV* (Zapata 2002), *ARIADNE* (Hu et al. 2002) and *ARAN* (Sanzgiri et al. 2005)) have some problems such as single point of attack, single point of failure, increased packet processing time and delay of route discovery process.
- On the other hand, position-based routing protocols use the geographical position of nodes to make routing decisions, which results in improving efficiency and performance. However, most of them (such as *GPSR* (Karp & Kung 2000), *ARP* (Giruka & Singhal 2005) and *MFR* (Takagi & Kleinrock 1984)) use greedy forwarding which suffers from congestion and nodes' energy consumption due to periodic beaconing. It is also not guaranteed to find the optimal route.

- It is found that restricted directional flooding position-based routing protocols (like *LAR* (Ko & Vaidya 2000) and *LARWB* (Kalhor et al. 2007)) have better performance than greedy ones in terms of finding the shortest path. However, both of them are vulnerable to some attacks as they focus on improving performance while disregarding security issues. Even the secure ones (as *SPAAR* (Carter & Yasinsac 2002), *AODPR* (Mizanur Rahman et al. 2006) and *SGF* (Song et al. 2007)) suffer from some problems, such as the single point of failure and attack, increased packet and processing overhead as well as scalability problems.
- Without online trusted servers, it is difficult to determine the trustworthiness of each node and exclude malicious nodes from the routes. Furthermore, the approach where one centralized server is used in the Ad-Hoc network is not practical because of the server mobility, operation bottleneck, system single point of failure and single point of attack. In order to address this problem, the position service system and the certificate authority should be distributed among multiple servers.

As a result, it is important to develop a scalable, distributed, secure and position-based routing protocol for Ad-Hoc networks. To the best of our knowledge, there are few secure and position-based routing protocols, such as *AODPR* and *SGF*. Though they suffer from many problems, *AODPR*, for example, uses a common key. If this key is compromised, a large percentage of the communication in the whole network will be compromised. Moreover, it suffers from the increased communication and time complexity if the nodes are hashed to a distant *VHR*, as well as if the *VHR* of a node cannot be reached. Additionally, due to nodes movement, it may be difficult to ensure that at least one position server exists in a specified Ad-Hoc network.

SGF, on the other hand, suffers from high average end-to-end delay and packet overhead. Moreover, *SGF* assumes the existence of pair-wise shared secret keys between the nodes which is difficult to implement in large area networks. Another drawback is that *SGF* assumes all nodes have tightly synchronized clocks, which is somewhat impractical for Ad-Hoc networks. Finally, it uses the greedy forwarding that is not guaranteed to find the optimal path.

A particular protocol of interest is *Authenticated Routing for Ad-Hoc Networks (ARAN)* (Sanzgiri et al. 2005). *ARAN* provides authenticated route discovery, setup and maintenance. Moreover, it uses cryptographic certificates to prevent and detect most of the Ad-Hoc routing protocols security attacks and satisfies the majority of security requirements introduced in earlier sections. However, it has a scalability problem with increasing the number of nodes, and causes high packet overhead and latency during route discovery due to the signing of each packet and broadcasting the route discovery packet to the entire network. Finally, *ARAN* uses one certificate server leading to a single point of failure and compromise.

As a result of this comparison, we ended up choosing to work with the *ARAN* protocol as the secure routing protocol and to work on finding solutions to the problems it suffers from. In the following section, *ARAN* protocol is studied in detail considering the above mentioned security requirements.

2.7 Detailed Discussion of the *ARAN* Protocol

In this section, the *Authenticated Routing for Ad-Hoc Networks (ARAN)* protocol (Sanzgiri et al. 2005) is analyzed. In the following five sections, the details of *ARAN* protocol's different phases are presented. After that, a security analysis of *ARAN* is provided in *Section 2.7.6*.

2.7.1 Introduction and Assumptions

ARAN protocol is classified as a secure reactive routing protocol. *ARAN* provides authenticated route discovery, setup and maintenance. The main objectives of *ARAN* are to protect routing packets against attacks from malicious nodes in a managed-open environment. It requires the use of a trusted *Certificate Authority (CA)* server whose public key is known by all valid nodes. *ARAN* uses cryptographic certificates to prevent most of the security attacks that Ad-Hoc routing protocols face and detect erratic behaviour. This protocol introduces authentication, message integrity and non-repudiation as part of a minimal security policy for the Ad-Hoc environment.

ARAN consists of a preliminary certification process followed by a route instantiation process that guarantees end-to-end authentication. Before proceeding further, let us define the following variables and notations that are used with *ARAN* protocol.

Table 2.7: Variables and notations for *ARAN*

Notation	**Description**
K_{A+}	Public key of node *A*
K_{A-}	Private key of node *A*
K_{CA+}	Public key of the trusted *CA*
K_{CA-}	Private key of the trusted *CA*
$\{d\}K_{A+}$	Data *d* encrypted with key K_{A+}
$[d]K_{A-}$	Data *d* digitally signed by node *A*
$Cert_A$	Node *A* Certificate
N_A	Nonce issued by node *A*
IP_A	*IP* address of node *A*
t	Timestamp
e	Certificate expiration time
RDP	Route Discovery Packet identifier
REP	REPly packet identifier
ERR	ERRor packet identifier

2.7.2 Certification of Authorized Nodes

ARAN requires the use of a trusted *CA*, whose public key (K_{CA+}) is known to all valid nodes. In managed-open environments, keys are generated and exchanged in advance through an existing relationship between *CA* and each node. Before entering the Ad-Hoc network, each node must request a certificate from *CA*. Each node receives one certificate after securely authenticating its identity to *CA*. Nodes use these certificates to authenticate themselves to other nodes during the exchange of routing messages. A node *A*, for example, receives a certificate from *CA* as follows:

$$CA \rightarrow A\text{: } Cert_A = [IP_A, K_{A+}, t, e]\ K_{CA-}$$

The certificate contains the *IP* address of *A* (IP_A), the public key of *A* (K_{A+}), a timestamp *t* of when the certificate was created and a time *e* at which the certificate expires. These variables are concatenated by the *CA* and signed using its private key (K_{CA-}). All nodes maintain fresh certificates with the trusted server.

2.7.3 Authenticated Route Discovery

The goal of end-to-end authentication is to make sure that a secure path between the source node *A* and the destination node *X* can be established. A source node *A* commences route instantiation to a destination *X* by broadcasting to its neighbours a *Route Discovery Packet (RDP)*:

$$A \rightarrow \text{Broadcast: } [RDP, IP_X, N_A]\, K_{A-}, Cert_A$$

The *RDP* packet includes a packet type identifier (*RDP*), the *IP* address of the destination (IP_x) and a nonce N_A. These are signed with *A*'s private key (K_{A-}), and *A*'s certificate ($Cert_A$) is appended to the message. The purpose of the nonce is to uniquely identify a *RDP* packet initiated by a said source. Each time *A* performs route discovery, it monotonically increases the nonce value.

Each intermediate node records the neighbour from which it received the *RDP* packet. The receiving node uses *A*'s public key, which is extracted from *A*'s certificate, to validate the signature and verify that *A*'s certificate is valid. Nodes do not forward messages for which they have already seen the (N_A, IP_A) tuple. It then signs the content of the message, appends its own certificate and forwards the message to each of its neighbours. This signature prevents spoofing attacks that may alter the route or form loops. Let *A*'s neighbour be *B* and that *B* consequently rebroadcasts:

$$B \rightarrow \text{Broadcast: } [[RDP, IP_X, N_A]\, K_{A-}]\, K_{B-}, Cert_A, Cert_B$$

Upon receiving the *RDP* packet, *B*'s neighbour *C* validates the signatures for both *A* (the *RDP* initiator) and *B* (the neighbour it received the *RDP* from) using the certificates in the *RDP* packet. Node *C* then rebroadcasts the *RDP* to its neighbours after recording its predecessor, removing *B*'s signature, signing the contents of the message and appending its own certificate:

$$C \rightarrow \text{Broadcast: } [[RDP, IP_X, N_A]\, K_{A-}]\, K_{C-}, Cert_A, Cert_C$$

2.7.4 Authenticated Route Setup

At this stage, the source trusts the destination to choose the return path. Eventually, the *RDP* packets are received by the destination node, *X*, which replies to the first *RDP* that it receives with a particular (source, nonce) pair. There is no guarantee that the first *RDP* received travelled along the shortest path from the source, but at least it is the one of the least delay. The destination returns a *REPly packet (REP)* back along the reverse path to the source. Assume that the first node that receives the *REP* sent by *X* is node *D*:

$$X \rightarrow D\text{: } [REP, IP_A, N_A]\, K_{X-}, Cert_X$$

The *REP* packet includes a packet type identifier (*REP*), the *IP* address of *A* (IP_A) and the nonce sent by *A*. All *REP*s are signed by the sender and its certificate is appended to the message. In this case, the packet is signed by *X* using K_{X-}, and the certificate of *X* ($Cert_X$) is appended. Intermediate nodes receiving the *REP* forward the packet back to the predecessor from which they received the original *RDP*. Each node along the reverse path back to the source signs the *REP* and appends its own certificate before forwarding the *REP* to the following hop. Let *D*'s next hop to the source be node *C*. Node *D* will unicast the following *REP* packet:

$$D \rightarrow C\text{: } [[REP, IP_A, N_A]\, K_{X-}]\, K_{D-}, Cert_X, Cert_D$$

Now *C* validates *D*'s signature, removes the signature and certificate and then signs the contents of the message and appends its own certificate before unicasting the *RDP* packet to the next node, *B*:

$$C \rightarrow B\text{: } [[REP, IP_A, N_A]\, K_{X-}]\, K_{C-}, Cert_X, Cert_C$$

All nodes check the signature of their preceding hop as the *REP* is returned to the source. This avoids attacks where malicious nodes instantiate routes by impersonation and replay of *X*'s message.

Upon receiving the *REP* packet the source verifies that the correct nonce is returned by the destination and validates the destination's signature. Only the destination is legible to reply to a *RDP* packet, i.e. other nodes that already have paths to the destination are not allowed to reply to *RDP* packets. Whereas other protocols allow this optimization, removing it also removes several possible exploits and reduces the reply traffic towards the source.

2.7.5 Route Maintenance

ARAN is an on-demand protocol. When there is no active traffic on an existing route during route's lifetime, the route is deactivated in the route table. Data which is received on an inactive route causes nodes to generate an *ERRor message (ERR)*. *ERR* packets are also used to report links in active routes that are broken due to nodes movement. All *ERR* messages must be signed by the upstream node that discovers the broken link. For a route between source *A* and destination *X*, a node *C* generates the *ERR* message for its neighbour *B* as follows:

$$C \rightarrow B\text{: } [ERR, IP_A, IP_X, N_C]\ K_{C-}, Cert_C$$

This message is sent through the path toward the source without alteration. A nonce is used to ensure that the *ERR* message is fresh.

2.7.6 *ARAN* Security Analysis

In this section, the security requirements satisfied by *ARAN* protocol are discussed and an analysis of its robustness in the presence of the different attacks introduced in earlier sections is given.

ARAN protocol introduces authentication, message integrity and non-repudiation as part of a minimal security policy for the Ad-Hoc environment. Since all *ARAN*'s packets have to be signed, a node cannot participate in routing without authorization from the *CA*. This access control, therefore, relies on the security of the *CA*, the authorization mechanisms employed by the *CA*, the strength of the issued certificates and the applied certificate revocation mechanism. Finally, this *CA* may also be a single point of failure and attack, therefore, it is a big concern to keep this *CA* uncompromised. The centralized *CA* in *ARAN* protocol results in lower availability since the compromise of this *CA* affects the security of the entire network.

The robustness of *ARAN* protocol in the presence of some passive and active attacks can be summarized as follows (Sanzgiri et al. 2005):

- Passive attacks: Detection of passive attacks is very difficult since the operation of the network itself is not affected. Encryption techniques used with *ARAN* make it impossible for the attacker to obtain any useful information from the packets overheard, i.e. it prevents passive attacks.
- Active attacks: *ARAN* protocol is robust against most active attacks, as discussed in the following points:
    - Spoofed route signaling: *RDP*s are signed with the source's private key and contain its certificate. Similarly, *REP*s include the destination's certificate and signature, ensuring that only the destination can respond to a particular *RDP* packet. This mechanism prevents impersonation attacks where either the source or destination is spoofed.
    - Fabricated routing messages: *ARAN* does not prevent generating false routing messages, but it offers a deterrent by ensuring non-repudiation since all routing messages must contain the sender's certificate and signature. Therefore, a node

    that injects false messages into the network may be prevented from participating in route discovery processes in the future.

    - Alteration of routing messages: *ARAN* identifies that all fields in *RDP* and *REP* packets remain unaltered between source and destination. Since both packet types are signed by the initiating node, any modifications in transit would be detected and the modified packet would be accordingly discarded. Thus, modification attacks are prevented in *ARAN*.

- Forwarding attacks: The authors in (Sanzgiri et al. 2005) have not detailed a specific method of secure forwarding. They suggested many opportunities, such as using the cryptographic material available to *ARAN*, but this would add overhead to the cost of data transmission. Another suggestion to protect data packets was to use the route reply process to instantiate shared keys between neighbours and to use these keys as the basis for a pair-wise *MAC*. This enforces that only certificate owners are able to forward data. It does not prevent certificate holders from replay attacks, but in any protocol, authorized participants can effectively attack the system by flooding the network with valid data packets for routes they create. End-to-end integrity can be ensured by the shared key derivable from the two peers' public keys.

Table 2.8 summarizes the security requirements satisfied by *ARAN* protocol while Table 2.9 gives a summary of the different attacks that the *ARAN* protocol defends against (Sanzgiri et al. 2005).

Table 2.8: Security requirements satisfied by *ARAN* protocol

Requirement	**Satisfied**
Availability	Low
Authentication	Yes
Confidentiality	No
Integrity	Yes
Non-repudiation	Yes

Table 2.9: Robustness of *ARAN* against existing attacks

Type	**Attack**	**Robust against**
Passive attacks	Eavesdropping	Prevented via the use of encryption techniques.
Active attacks	Impersonation	Yes
	Fabrication	No, but provides non-repudiation.
	Modification	Yes
Forwarding attacks	Modification	Yes
	Dropping	No

As a summary, *ARAN* provides authenticated route discovery, setup and maintenance. It also uses cryptographic certificates to prevent most of the security attacks against Ad-Hoc routing protocols and satisfies most of their security requirements. However, it has a scalability problem with increasing the number of nodes, and causes high packet overhead and latency during route discovery due to the route discovery packet broadcast to the entire network and the signing of each packet. Finally, *ARAN* assumes one certificate server existing in the network, leading to a single point of failure and compromise. As a result, it is our interest in this work to find solutions to the problems that *ARAN* protocol suffers from.

2.8 Chapter Summary

This chapter presented a detailed study about Ad-Hoc networks and some existing routing protocols. *Sections 2.1* and *2.2* introduce wireless networks in general and Ad-Hoc networks in particular. In *Section 2.3*, an introduction to routing protocols for Ad-Hoc networks has been given. *Sections 2.4 and 2.5* look at the topology-based and position-based routing protocols respectively. In brief, *Section 2.6* highlighted the advantages and disadvantages of the discussed categories and presented a justification of adopting the *ARAN* protocol authentication steps in our model. In *Section 2.7 ARAN* protocol different phases were presented and a security analysis of *ARAN* was conducted.

The next chapter addresses our research methodology, including a discussion about the different simulation environments and the reasons behind choosing the *GloMoSim* simulator.

Chapter 3

Research Methodology

In this chapter, the general steps of our research methodology are outlined. After that, the most popular Ad-Hoc networks simulation packages are discussed in *Section 3.2*. Lastly, justifications for the chosen simulation tool are presented in *Section 3.3*.

3.1 Research Methodology Phases

Our methodology is divided into four phases:

1. Literature review phase.
2. Modelling phase.
3. Development phase.
4. Testing and evaluation phase.

These phases are discussed in the following four sections.

3.1.1 Literature Review Phase

In this phase, a review and analysis of the latest researches and publications related to the Ad-Hoc networks field are carried out, especially those concentrating on routing protocols in this type of networks. Hence, many topology-based routing protocols, position-based routing protocols as well as secure routing protocols are studied and their strengths and weaknesses are specified.

3.1.2 Modelling Phase

After exploring the existing literature and considering the analyzed results, the specifications of the new routing protocol are proposed. The newly developed protocol is proposed taking into consideration the important problems of the existing protocols, such as scalability, robustness, security and single point of failure and attack. During the modelling phase, all the details of the new protocol are specified, including the assumptions, different types of nodes existing in the network, different types of packets exchanged as well as techniques used for forwarding these packets.

3.1.3 Development Phase

Due to Ad-Hoc networks high cost, experimentation and performance evaluation of a new protocol is mostly achievable through simulation. Moreover, the construction of real testbeds for any pre-defined scenario is usually an expensive or even impossible task if factors like mobility, testing area and the number of nodes are taken into account. Furthermore, most measurements are not repeatable and require high efforts. As a result, simulations are needed to bypass these problems (Schilling 2005).

Consequently, after finalizing the new protocol’s assumptions and details, the *Global Mobile Information System Simulator (GloMoSim)* (Zeng et al. 1998) is used for the development of a simulation of the new protocol. *Section 3.2* gives a review of several simulation tools and *Section 3.3* explains the reasons for choosing *GloMoSim* as our simulation tool.

3.1.4 Testing and Evaluation Phase

In this phase, extensive simulation runs and tests are carried out to study the performance of the proposed protocol, compare it to the existing ones and determine the best values of the protocol parameters. Experiment results are used to revise the details of the new protocol until satisfying results are obtained.

In order to have a comprehensive comparison between our protocol and the existing ones, experiments must consider different node densities and mobility speeds as well as different area sizes. Also, many performance metrics can be tested including packet delivery fraction, packet and byte routing load, average path number of hops and average route acquisition latency.

3.2 Simulation Tools Overview

Numerous tools were proposed for Ad-Hoc networks simulation, among which include *OPNet* (Desbrandes et al. 1993), *NS-2* (McCanne & Floyd 1997), *GloMoSim* (Zeng et al. 1998), *pdns* (Riley et al. 1999), *OMNet++* (Imre et al. 2001), *GTNets* (Riley 2003), *DIANEmu* (Klein 2003), *Jane* (Frey et al. 2003; Lehnert et al. 2004) and *SWANS* (Barr 2004).

These tools differ in their simulation capabilities, environments, set of parameters to play with as well as scalability. Some are dedicated to *MANET*s simulation, such as *Jane* and *SWANS*, while some others resulted as extensions of wired network simulators (such as *NS-2*) and general-purpose discrete-event simulation engines (such as *Maisie* (Bagrodia & Liao 1994) and *PARSEC* (Bagrodia et al. 1998)).

In this section, we look at the three most popular simulation tools according to (Hogie et al. 2006). These simulation tools are *Network Simulator 2 (NS-2)* (McCanne & Floyd 1997), *Global Mobile information systems Simulator (GloMoSim)* (Zeng et al. 1998) and *OPtimized Network engineering tools (OPNet)* (Desbrandes et al. 1993). In the next section we will explain our selection strategy.

3.2.1 Network Simulator 2 (*NS-2*)

NS-2 is developed at the *Information Sciences Institute* and is supported by the *Defence Advanced Research Projects Agency* and *National Science Foundation*. *NS-2* is a discrete-event network simulator organized according to the *Open Systems Interconnection model (OSI)* (Wetteroth 2001) and was initially intended to simulate wired networks (Hogie et al. 2006). After that, the 802.11 *MAC* layer and important routing protocols needed in *MANET*s have been added to it (Schilling 2005).

The core of *NS-2* is a huge piece of code written with *C++* language due to its quickness and object-oriented support. To ease the use of *NS-2*, it appears to the user as an *Object Tool Command Language (OTCL)* interpreter. It reads scenarios files written in *OTCL* and produces a trace file in its own format. This trace needs to be processed by user scripts or converted and rendered using the network animator *NAM* (Estrin et al. 1999). *NAM* permits to visualize the output, provides packet-level animation and provides a *Graphical User Interface (GUI)* to design and debug network protocols. The combination of the two languages offers an interesting compromise between performance and ease of use, though this increases the complexity of the simulator and results in a steep learning curve for *NS-2* and difficulty in debugging (Cavin et al. 2002).

NS-2 is an open-source simulator, making it interesting on the one hand, but on the other hand, there are some negative aspects that come along with it. *NS-2* suffers from its lack of modularity and its inherent complexity. Indeed, adding components/protocols or modifying already existing ones is not a simple task as it should be (Schilling 2005; Hogie et al. 2006).

Learning *NS-2* needs a long period of time due to the lack of documentation in the source code and the usage of two programming languages. For a long time, *NS-2* has few good documentation. The situation has been changed recently when several users presented their experience online in the form of tutorials or example-driven documentations. Another well-known weakness of *NS-2* is its high consumption of computational resources. A harmful result is that *NS-2* lacks scalability, which slows down the simulation of large networks. Usually, *NS-2* is used to simulate scenarios consisting of no more than a few hundreds of nodes (Schilling 2005; Hogie et al. 2006).

3.2.2 Global Mobile Information Systems Simulator (*GloMoSim*)

GloMoSim is a scalable simulation environment for wireless and wired network systems that was developed at the *University of California*. *GloMoSim* is aimed at stimulating models that may contain up to hundred thousands of mobile nodes with a rational execution time (Bajaj et al. 1999). It is the second most popular wireless network simulator. *GloMoSim* is written in the parallel discrete-event simulation capability provided by a *C*-based parallel simulation language, *PARallel Simulation Environment for Complex systems (PARSEC)* (Bagrodia et al. 1998), and hence benefits from the *PARSEC* ability to run on shared-memory symmetric processor computers.

GloMoSim respects the *OSI* standard and has been developed using languages, libraries and frameworks specialized to discrete-event simulation. These middleware technologies classically focus on performance, concurrency and distribution (Hogie et al. 2006). Standard *Application Programming Interfaces (APIs)* are used between the different layers. This allows the rapid integration of models developed at different layers by different users (Cavin et al. 2002). Two versions of the simulation tool exist: the academic research version, which is dedicated for academic uses only, and a commercial version, which is distributed as the *QualNet* software package.

GloMoSim uses parallelism that refers to the simultaneous execution of multiple instructions of the same program. Parallelism is used to speed up simulations and allow *GloMoSim* to model networks involving tens of thousands of stations (Hogie et al. 2006). The parallelization technique used by *GloMoSim* is to split the network into different subnetworks, each of them being simulated by different processor. The network is partitioned in such a way that the number of nodes simulated by each partition is the same.

The source code is written primarily in *C* language and the *PARSEC* compiler is used to create executable files. For the development of custom protocols in *GloMoSim*, some familiarity with *PARSEC* is required. Most protocol developers write purely *C* language code with some *PARSEC* functions for time management. *PARSEC* code is used generally in the *GloMoSim* kernel, but it is not necessary for the programmers to know and understand how the kernel works (Christiansen et al. 2003).

3.2.3 Optimized Network Engineering Tools (*OPNet*)

OPNet is a discrete-event network simulator first proposed by the *Massachusetts Institute of Technology* in 1986. *OPNet* is written in *C++* and it is a well-established and professional commercial suite for network simulation. It is actually the most widely used commercial simulation environment (Hogie et al. 2006). One of the most interesting features of *OPNet* is its ability to execute and monitor several scenarios in a concurrent manner (Hogie et al. 2006).

OPNet comes along with a large number of pre-defined functions, protocols, devices and behaviours, which makes it a powerful program just from the start up and without much effort. Additionally, the opportunity to implement new algorithms is given. Also, several tools and editors are provided. The aim is to make use of the numerous existing components that are part of *OPNET* in order to decrease the developers' effort, shrink the implementation time and reduce the number of errors. A *GUI* interface is provided and a lot of documentation comes along with it (Schilling 2005).

Nevertheless, *OPNET* is not an open-source software, and therefore, users and companies need to purchase licenses. Hence, the cost of the software could discourage many developers since open-source solutions are available (Schilling 2005). Additionally, the main disadvantage is its relative complexity to model a given system. The time required to learn it and achieve the modelling of a system can be very long, especially for new developments (Christiansen et al. 2003). Furthermore, it is reported that the *OpNet* simulator is quite memory consuming and that it is difficult to modify the library models (Christiansen et al. 2003; Schilling 2005).

3.3 Discussion and Simulator Choice

When choosing a simulation package, the question of which routing protocol to be simulated and which simulations to be conducted are of great importance. Moreover, the number of nodes targeted also determines the choice of the simulation tool. Sequential simulators should not be expected to run more than one thousand nodes (Hogie et al. 2006). If larger scales are needed, then parallel simulators are a wise choice. So in this section, the reasons behind the decision of using the *Glomosim* simulation package in the experimental work are stated. Table 3.1 summarizes the properties of the three discussed simulation tools.

Table 3.1: Ad-Hoc simulators comparison

Simulator **Criterion**	***NS-2*** **(McCanne & Floyd 1997)**	***Glomosim*** **(Zeng et al. 1998)**	***OPNet*** **(Desbrandes et al. 1993)**
Parallelism	No	Yes	Yes
Interface	*C++/OTCL*	*PARSEC* (*C*-based)	*C*
Popularity	High	Medium	Low
License	Open source	Open source	Commercial
Required time to learn	Long	Moderate	Long
Scalability	Moderate	High	High

Since *OpNet* is a commercial tool, it is expensive. In addition, it suffers from many disadvantages such as complexity, time required to learn it, memory consumption and difficulty to modify the library models. The most elaborate tools are *GloMoSim* and *NS-2*. Both, the latter and the academic research version of the former are available freely on the Internet. Still, we can conclude the following points from the previous section:

- It is quite a complex task to install *NS-2* and have it work right. Even after installing it, it is difficult to learn and use especially due to the fact that it uses two languages *C++* for data and *OTcl* for control. *GloMoSim*, on the other hand, is built as a set of libraries. The libraries are built in a *C*-based discrete event simulation language (*PARSEC*). Even the new protocols are developed using *C* language, *C* language is more admired for most programmers compared to *OTcl*.
- *NS-2* does not work well for large topologies. It can be used for the simulation of hundreds of nodes. *GloMoSim*, on the other hand, is a scalable simulator that is able to model networks made of tens of thousands of nodes.

Last but not least, the source code of *ARAN* protocol which is implemented under the *Glomosim* package is found online. Thus, having this code implemented under *Glomosim*, there is no need to reinvent the wheel and rewrite the protocol in *NS-2*.

It is decided, based on the available functionality, strong focus on wireless networks, scalability, expertise of the partners, its increasing usage to simulate wireless networks and having the *ARAN* protocol source code developed under it to use *GloMoSim* as a simulation tool. An introduction to *GloMoSim* simulator is provided in *Appendix B.*

3.4 Chapter Summary

In this chapter, the general steps of our research methodology have been outlined. After that, the different simulation packages those are used in mobile Ad-Hoc networks were discussed. Then, justifications for the choice of the *Glomosim* simulation environment to be used to conduct all experimental parts of this thesis work have been presented.

In *Chapter 4* we look at the details of the newly proposed scheme.

Chapter 4

The Proposed Protocol

In this chapter, we propose a new routing model called *ARANz*. The proposed protocol is called *ARANz* because it adopts the authentication steps used with the *Authenticated Routing for Ad-Hoc Networks (ARAN)* protocol and deals with the network as zones. *Section 4.1* gives an introduction to the new protocol along with the important assumptions. *Sections 4.2* and *4.3* discuss the network setup and maintenance, respectively. In *Section 4.4*, the proposed location service is presented. After that, the authenticated route discovery, setup and maintenance are explained in *Section 4.5*. Then, *Section 4.6* presents the data transmission phase. Afterward, the proposed misbehaviour detection system is discussed in *Section 4.7*. Finally, a performance and security analysis of *ARANz* is given in *Section 4.8*.

4.1 Introduction to the *ARANz* protocol

This section gives a general description of *ARANz. Section 4.1.1* gives an overview of the new protocol while *Section 4.1.2* states the important assumptions. *Section 4.1.3* describes the notation used. In *Sections 4.1.4* and *4.1.5,* the different types of packets exchanged as well as the different types of nodes existing in the network are presented. After that, the used keys and certificates are explained in *Sections 4.1.6* and *4.1.7*, respectively. Finally, *Section 4.1.8* presents the different techniques used for forwarding the packets in *ARANz*.

4.1.1 An Overview

ARANz, simply like *ARAN*, uses cryptographic certificates to prevent most of the attacks against Ad-Hoc routing protocols (among which include impersonation of other nodes and modification of packets) and detect erratic behaviour (including the use of invalid certificates and improperly signed packets). Additionally, *ARANz* introduces a hierarchal distributed routing algorithm, which aims to distribute load and improve performance, robustness and scalability of the *ARAN* routing protocol.

To achieve robustness and load distribution, as well as solving the single point of failure and attack problems, *ARAN*z divides the network area into zones and distributes trust among multiple *Local Certificate Authority (LCA)* servers. As opposed to *ARAN* protocol that depends on a centralized trust, single *Certificate Authority (CA)*, each zone in *ARAN*z has multiple *LCA*s. *LCA*s of a particular zone collaborate with each other to issue certificates for nodes inside that zone and work as backups for each others. If a misbehaviour detection scheme is present on the network, then the security of this protocol can be improved through collaboration with this scheme. Consequently, in conjunction of the proposed protocol, a misbehaviour detection scheme is proposed in this work.

Moreover, *ARAN*z aims to exhibit better scalability and performance by taking advantage of the idea of restricted directional flooding position-based routing protocols. Hence, unlike *ARAN* that broadcasts the route request packet to the entire network, whenever a source node implementing *ARANz* needs to communicate with a specified destination node, it obtains the position of the destination node from the *LCA*s of its current zone. After that, the route discovery packet is sent towards the destination using restricted directional flooding. This helps in reducing the overall overhead and saving the network bandwidth. As such, *LCA*s also serves as position servers, and each node is supposed to notify the *LCA*s of its zone about its new position if it has moved (not periodically) at a rate proportional to its speed.

The use of restricted directional flooding requires that each node should be able to obtain its own geographical position (via *GPS*, for example) and the geographical position of the destination through the proposed location service. By utilizing the geographical information, nodes forward the route requests only if their positions are closer to the destination's position than their predecessors, which saves network bandwidth.

In *ARANz* all *LCA*s in the network are supposed to have synchronized clocks to ensure protocol correctness, e.g. to avoid a situation where two nodes in different zones are issued certificates at the same time with different timestamps. Regular nodes, on the other hand, use timestamp included in their certificates to be aware of the system's time and check the validity of other nodes' certificates.

The following table summarizes the main differences between *ARAN* and *ARANz* protocols.

Table 4.1: Differences between *ARAN* and *ARANz* protocols

Protocol **Criterion**	***ARAN***	***ARANz***
Type	Topology-based (reactive)	Position-based (restricted directional flooding)
Route discovery sending mechanism	Route discovery packets are flooded to all nodes in the network.	Intermediate nodes broadcast route discovery packet only if they are closer to the destination than the previous hop.
Centralized trust	Yes (single *CA*)	No (multiple *LCA*s in each zone)
Load distribution	No	Yes
Scalability	Medium	High
Packet overhead	High	Medium
Synchronization	No	Yes
Extra hardware	No	Yes (*GPS* receivers)

ARANz consists mainly of six phases which are *network setup*, *network maintenance*, *location service*, *route instantiation and maintenance*, *data transmission* and *misbehaviour detection*. *Network setup* includes certifying trusted nodes, dividing an area into zones and electing initial certificate authority servers. The *Network maintenance* phase ensures the maintenance of the network structure, taking into consideration issues like updating nodes' certificates, *LCA*s synchronization, movement of nodes in and out the network as well as corrupted and destroyed nodes.

Whenever a node has data to send to a particular destination, it obtains the destination's position before beginning the route discovery process. The *Location service* phase enables the source to obtain the destination's position by communicating with *LCA*s in its zone. After getting the destination's position, *route instantiation and maintenance* phase is initiated. The source begins route discovery to the destination by sending a *Route Discovery Packet (RDP)*. This is done using restricted directional flooding towards the destination node. Upon receiving the first *RDP*, the destination unicasts a *Route REPly (RREP)* packet back along the reverse path to the source to setup the route.

After finishing route discovery and setup, the source begins *data transmission* to the destination node. In order to maintain the selected route, nodes in *ARANz* keep track of whether routes are active or not, and use *ERRor* (*ERR)* packets to report links in active routes that are broken due to node movement. Our protocol collaborates with a *misbehaviour detection system* to help in identifying malicious nodes in order to exclude them from future communications.

Since each node, in *ARANz*, by the end of the network setup phase has its own certificate, these certificates can be used to apply the authentication steps used with *ARAN* protocol. Hence the source of any packet and all intermediate nodes sign the packet using their private keys and append their certificates to the packets. Also, each intermediate node, as well as the destination, validates the previous node's signature using the previous node's public key, which is extracted from its certificate. Thus, it is assured that packets sent during the route discovery are authenticated end-to-end and only authorized nodes participate at each hop between source and destination. Consequently, as in *ARAN*, data packets exchanged between nodes are not signed and do not have attached certificates. Hence, each node simply relays data packets as is to its successor in the route obtained during the route instantiation process.

Table 4.2 summarizes the different phases in *ARANz* protocol, whereas Figure 4.1 shows the general system flowchart. *Appendix A* includes the pseudocode of most parts of *ARANz* protocol.

Table 4.2: *ARANz* protocol different phases

Phase	**Description**	**Started**	**Ended**
Network setup (*Section 4.2*)	Includes certification process, dividing area into zones, electing initial certificate authority servers and informing each node about the initial role it will play in the network.	By the commencement of the network's lifetime.	When the network structure is settled and each node becomes aware of its initial role.
Network maintenance (*Section 4.3*)	Ensures maintenance of the network structure, taking into consideration some issues like updating nodes' certificates, needed synchronization, as well as nodes movement, corrupting and distortion.	After finishing the network setup phase.	By the end of the network's lifetime.
Location service (*Section 4.4*)	Enables source to obtain destination's position by communicating with *LCA*s in its zone.	Whenever a node has data to send to a particular destination.	Upon obtaining the destination's position.
Route instantiation and maintenance (*Section 4.5*)	Includes sending a *RDP* using restricted directional flooding from source to destination, unicasting a *RREP* from the destination along the reverse path to the source as well as maintaining the selected route using *ERR* packets to report links in active routes that are broken due to node movement.	After getting the destination's position.	Upon completing sending the data.
Data transmission (*Section 4.6*)	Relaying data packets through the route obtained during the route instantiation process until reaching the destination.	After instantiating the first route to the destination.	Upon completing sending the data.
Misbehaviour detection system (*Section 4.7*)	Helps in identifying malicious nodes and excluding them from future communications.	After finishing the network setup phase.	By the end of the network's lifetime.

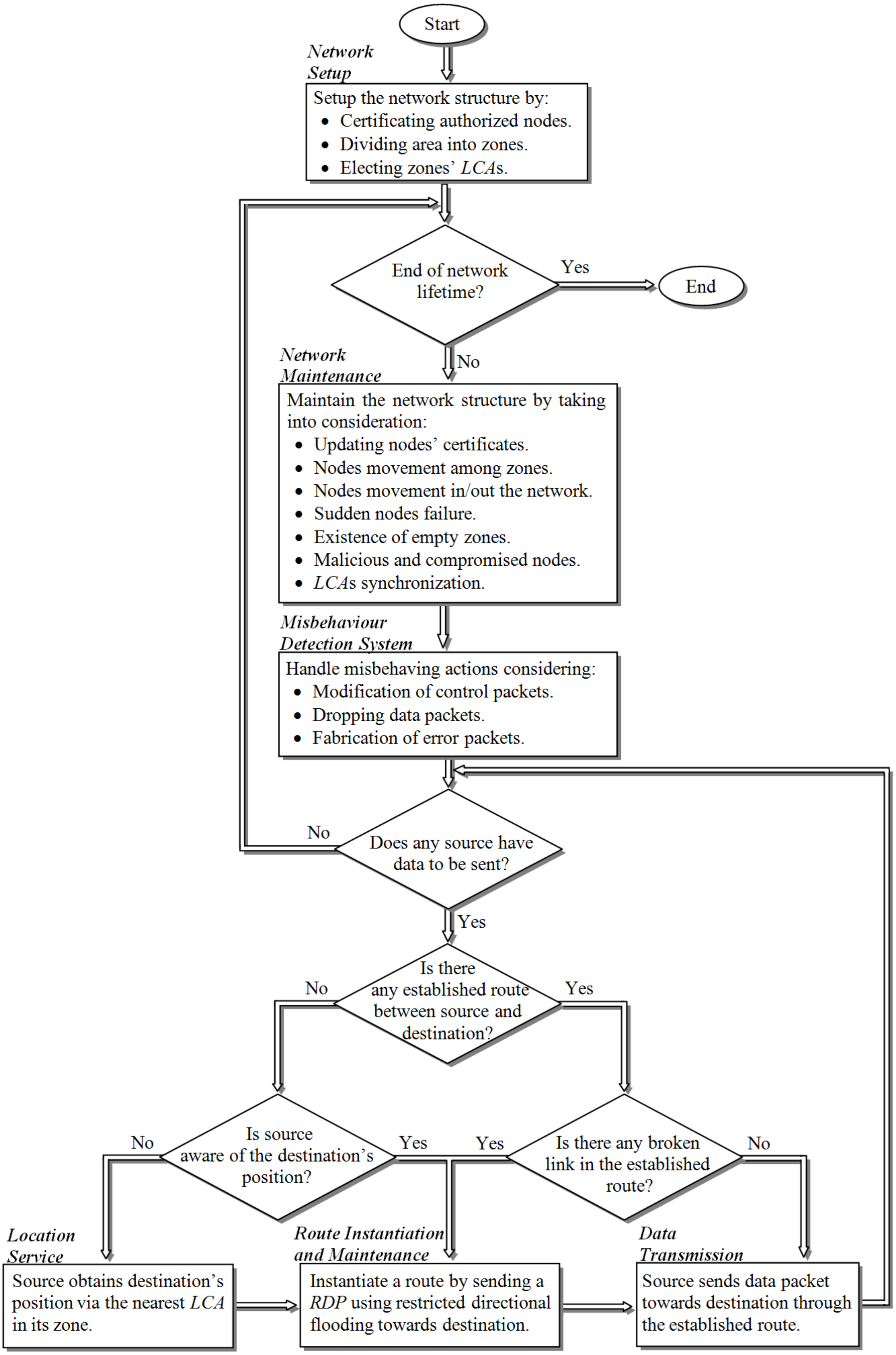

Start
Network Setup
Setup the network structure by:
• Certificating authorized nodes.
• Dividing area into zones.
• Electing zones' LCAs.
End of network lifetime?
Yes
End
No
Network Maintenance
Maintain the network structure by taking into consideration:
• Updating nodes' certificates.
• Nodes movement among zones.
• Nodes movement in/out the network.
• Sudden nodes failure.
• Existence of empty zones.
• Malicious and compromised nodes.
• LCAs synchronization.
Misbehaviour Detection System
Handle misbehaving actions considering:
• Modification of control packets.
• Dropping data packets.
• Fabrication of error packets.
No
Does any source have data to be sent?
Yes
No
Is there any established route between source and destination?
Yes
No
Is source aware of the destination's position?
Yes
Yes
Is there any broken link in the established route?
No
Location Service
Source obtains destination's position via the nearest LCA in its zone.
Route Instantiation and Maintenance
Instantiate a route by sending a RDP using restricted directional flooding towards destination.
Data Transmission
Source sends data packet towards destination through the established route.

Figure 4.1: System flowchart

4.1.2 Important Assumptions

We assume *Nn* cooperative trusted nodes in a managed-open environment. These nodes are aware of their positions (equipped with *GPS* receivers, for example) and have equal transmission range *TR*. These nodes are distributed randomly in a network of area $An = (Ln \times Ln)$, where *Ln* is the boundary length of the network. This area is divided into multiple zones with multiple *LCA*s assigned for each zone. In our implementation environment, we consider to use square-shape zones with four *LCA*s. We note that this choice may seem a bit rigid for now, but we believe that this will be the starting point for future work in regards to the implementation of other zone shapes with a dynamic number of *LCA*s. Hence, the network area is divided into *Nz* zones, where *Nz* is a square of an integer number and the area of each zone is $Az = (Ln \times Ln)/Nz$. Nodes communicate among each other via restricted directional flooding and adopting the authentication steps as in *ARAN* protocol. Similar to *AODV* and *ARAN*, *ARANz* assumes symmetric (bi-directional) links.

One trusted node is chosen to have the software needed to begin the network setup, divide the area into zones and elect the initial *LCA*s. This node is called the *Primary Certificate Authority (PCA)* server and possesses the private key of the network (K_{NET-}). *PCA* is chosen prior to starting network deployment, this is possible since we are dealing with managed-open environments.

Each trusted node *n* willing to participate in the network has a private/public key pair (K_{n-}/K_{n+}), the public key of the network (K_{NET+}) and a common key (*CK*) which is used for encryption and decryption of packets sent by non-*PCA* nodes in the network setup phase. In managed-open environments, keys are generated and exchanged in advance through an existing relationship between *PCA* and each trusted node.

4.1.3 Notations

Before proceeding further, let us define the notations that are used in *ARANz*. Tables 4.3 through 4.5 give the notations used for the description of the protocol, including those used for the proposed *LCA*s election algorithm and misbehaviour detection system.

Table 4.3: Variables and notations for *ARANz*

Notation	**Description**
[d]Kn-	Data d digitally signed by node n
{d}Kn+	Data d encrypted with key Kn+
8NbrZZ	Numbers and coordinates of 8-Neighbouring zones of zone z
AdjLzs	IP address and position of adjacent LCA of LCAzs
AdjZzs	Number of the zone adjacent to boundary s of zone z
ALL	All nodes existing currently in the network
ALL_z	All nodes existing currently in zone *z*
An	Area of the network
AN	Absent nodes, *IP* addresses and public keys of authorized nodes that were not in the network during network setup
AT	Authentication table
Az	Area of each zone
B_n	Remaining battery life time of node *n*
CertLZz	LCAs certificate of zone z
$Cert_n$	Node *n* certificate
CK	Common key
C_n	*CPU* power of node *n*
CoorZz	Coordinates of zone z
Dist	Distance between an intermediate node and the destination node
Dmov	Distance that a node moves before informing its zone LCAs about its new position
Dnzs	Distance between position of node n existing in zone z and the middle point of the zone boundary s
Dsid	Distance that a LCA is allowed to be from the zone boundary middle point prior to initiating a new LCA election
e	Certificate expiration time
$\xrightarrow{Fln}$	Flood packet to the entire network
$\xrightarrow{Flz}$	Flood packet to a particular zone
IP_n	*IP* address of node *n*
Kn-	Private key of node n
Kn+	Public key of node n
K_{NET-}	Private key of the network
K_{NET+}	Public key of the network
KZz-	Private key of zone z
KZz+	Public key of zone z
LCA	Local Certificate Authority
LCAFn	Fraction of time during which node n served as a LCA in the last Ns time slots

Table 4.3: Variables and notations for *ARANz* (continued)

Notation	**Description**
LCAsZz	IP addresses and positions of LCAs in zone z
LCAzs	LCA responsible for boundary s of zone z
Ln	Length of the network side
M_n	Memory capacity of node *n*
Nn	Number of nodes
Nn	Nonce issued by node n
Ns	Number of time slots during which the role that a node plays is recorded
Nz	Number of zones
NZz	Nonce issued by Zone z
PCA	Primary Certificate Authority
Pid	Packet type identifier
P_n	Position of node *n*
$\xrightarrow{Rdf}$	Send packet using restricted directional flooding
$\xrightarrow{Rev}$	Send packet through reverse path
$\xrightarrow{Rly}$	Relay data packet to its destination
Rolen	Node n current role
Rolesn[Ns]	An array specifying the roles (LCA or regular) that node n played during each of the last *Ns* time slots
S_n	Movement speed of node *n*
SR	Source route that a packet will go through
$\xrightarrow{Src}$	Send packet using source routing
t	Timestamp
Tac	Acceptance of certificate time
Tcu	Certificate update time
Tic	Information collection time
Tkc	Private key collection time
Tls	LCA synchronization time
Tpc	Probability collection time
Tpd	Position discovery time
TR	Nodes’ transmission range
Trd	Route discovery time
Tsl	Serving as a LCA time
Xczc	X-coordinate of corner c of zone z
Xmzs	X-coordinate of middle point of boundary s of zone z
X_n	X-coordinate of node *n*
Yczc	Y-coordinate of corner c of zone z
Ymzs	Y-coordinate of middle point of boundary s of zone z
Y_n	Y-coordinate of node *n*
Zonen	Node n current zone

Table 4.4: Variables and notations for the proposed *LCA*s election algorithm

Notation	Description
Bmax	Maximum possible node battery life time
Cmax	Maximum node *CPU* power
Dmax	Maximum possible distance between a node and middle point of a zone boundary
Mmax	Maximum node memory capacity
$ProbL_{nzs}$	Probability of node *n* existing in zone *z* to be elected as a *LCA* of boundary *s*
Smax	Maximum possible node movement speed
Wb	Weight of node battery remaining life upon electing a new *LCA*
Wc	Weight of node *CPU* power upon electing a new *LCA*
Wd	Weight of distance between a node and middle point of a zone boundary upon electing a new *LCA*
Wf	Weight of fraction of time during which node served as a *LCA* upon electing a new *LCA*
Wm	Weight of node memory capacity upon electing a new *LCA*
Ws	Weight of node movement speed upon electing a new *LCA*

Table 4.5: Variables and notations for the proposed misbehaviour detection system

Notation	Description
Fd_{nm}	Number of dropped data packets by node *m* that it receives from node *n*
Fm_{nm}	Number of modified control packets sent from node *m* to node *n*
Nm	Number of packets received indicating the misbehaviour of a node so that this node is considered as compromised
Sd_{nm}	Number of delivered data packets by node *m* that it receives from node *n*
Sm_{nm}	Number of unmodified control packets sent from node *m* to node *n*
Thd	Dropping threshold
Thf	Fabrication threshold
Thm	Modification threshold
$TrstVd_{nm}$	Node *n* trust value regarding node *m* considering dropping attacks
$TrstVf_{nm}$	Node *n* trust value regarding node *m* considering fabrication attacks
$TrstVm_{nm}$	Node *n* trust value regarding node *m* considering modification attacks
TT	Trust table

4.1.4 Types of Packets

The packets exchanged between different nodes are classified into two main classes, *control* packets and *data* packets. *Control* packets are all packets exchanged among nodes during the network setup, network maintenance, location service, route instantiation and maintenance as well as misbehaviour detection phases. On the other hand, *data* packets are the packets relayed to the destination node during data transmission phase. Data packets are sent through the path selected in the route instantiation phase. Table 4.6 summarizes different types of packets and Table 4.7 gives notations for packet type identifiers (*Pid*s) used in different phases.

Table 4.6: Types of packets in *ARANz*

Packet	**Description**
Control packets	All packets exchanged between nodes during the network setup, network maintenance, location service, route instantiation and maintenance as well as misbehaviour detection phases.
Data packets	Packets relayed to the destination node during data transmission phase (after setting up the route).

Table 4.7: Packet type identifiers for *ARANz*

Phase	***Pid***	**Stand for**
Network setup	*NETSET*	NETwork SETup
	NIN	Node INformation
	NROLE	Node ROLE
Network maintenance	*CREQ*	Certificate REQuest
	CREP	Certificate REPly
	ACREQ	Acceptance of Certificate REQuest
	ACREP	Acceptance of Certificate REPly
	NCERT	Node CERTificate
	UNPOS	Update Node POSition
	DNODE	Departing NODE
	NNODE	New NODE
	NZONE	New ZONE
	ULPOS	Update *LCA* POSition
	UALPOS	Update Adjacent *LCA* POSition
	NLCAE	New *LCA* Election
	FLCA	Failed *LCA*
	FALCA	Failed Adjacent *LCA*
	FNODE	Failed NODE
	NPROB	Node PROBability
	NLCA	New *LCA*
	NALCA	New Adjacent *LCA*
	EZONE	Empty ZONE
	PKPREQ	Zone Private Key Part REQuest
	PKPREP	Zone Private Key Part REPly
	SNODE	Sole NODE
	CLSYN	CLocks SYNchronization
Location service	*PDP*	Position Discovery Packet
	PREP	Position REPly
Route instantiation and maintenance	*RDP*	Route Discovery Packet
	RREP	Route REPly
	ERR	ERRor
Data transmission	*DATA*	DATA packet
Misbehaviour detection system	*MNODE*	Misbehaving NODE
	CNODE	Compromised NODE

4.1.5 Classification of Nodes

During the network's lifetime there are different types of nodes with different states that a node possibly be in, this section explains these types and states. In this section, we simply mention keys and certificates owned by different nodes. A detailed explanation of these keys and certificates is given in the following two sections.

During the network setup phase, nodes are classified into *Primary Certificate Authority (PCA)* server and non-*PCA* nodes. *PCA* is a previously chosen node that has the software needed to start the network setup. Non-*PCA* nodes are all other trusted nodes that are allowed to participate in the network. Trusted nodes are characterized during the network setup phase by owning a private/public key pair. Non-*PCA* nodes wait for a packet from the *PCA* node informing them about starting the network setup phase, accordingly, they provide it with information about themselves to help it in dividing the area and electing *Local Certificate Authority (LCA)* servers. Non-*PCA* nodes have a private/public key pair (K_{n-}/K_{n+}), the public key of the network (K_{NET+}) and a common key (*CK*). In addition to these keys, *PCA* has the private key of the network (K_{NET-}) to be used as evidence that it is really the *PCA*.

After finishing the network setup phase, each trusted node may be a *LCA* server or a regular (non-*LCA*) node. Each zone in *ARANz* has multiple *LCA*s. Each *LCA* of a particular zone works as a position server for nodes residing currently in that zone, collaborates with other *LCA*s to issue certificates for those nodes and works as backup of other *LCA*s in its zone. In addition to keys owned in the network setup phase ((K_{n-}/K_{n+}), K_{NET+} and *CK*), each trusted node, by the end of this phase, owns a node certificate ($Cert_n$). Therefore, after the network setup, trusted nodes own a private/public key pair as well as an unexpired node certificate. Additionally, all trusted nodes have the public key of the zone (K_{Zz+}) where they are residing currently. Moreover, *LCA*s have the private key of their zone (K_{Zz-}) as well as a zone *LCA* certificate ($CertLZ_z$). A zone private/public key pair is used for encrypting and decrypting the zone *LCA* certificate that is used to assure that a node is actually a *LCA* for a specific zone.

Different nodes can be in one of five states, initiating sending a data packet or a control packet (source), receiving a packet for which it is the intended destination (destination), receiving a packet destined to another node (intermediate), forwarding a packet towards its destination (forwarding) as well as being idle.

A non-*PCA* node can be a source, destination, intermediate or forwarding node of a control packet or simply be idle. *PCA*, on the other hand, can be in any of these states except being an intermediate or forwarding node. During the setup phase, *PCA* is always a source of a packet sent to non-*PCA* nodes or a destination of a packet sent from non-*PCA* nodes, but never be an intermediate node.

Both *LCA*s and regular nodes can participate in network maintenance, location service, route instantiation and maintenance, data transmission as well as misbehaviour detection phases, whether as a source, destination, intermediate or forwarding node. The different types of nodes existing in the network and the different states a node can be in are given in Table 4.8 and Table 4.9, respectively.

Table 4.8: Types of nodes in *ARANz*

Phase	**Node**	**Description**	**Keys/certificates owned**	**Possible states**
Network setup	Primary Certificate Authority (*PCA*)	Has the software needed to begin the network setup, divide area into zones and elect the initial *LCA*s.	• Network private/public key (K_{NET-}/K_{NET+}). • Private/public key pair (K_{n-}/K_{n+}). • Common key (*CK*).	• Source or destination of a control packet. • Idle.
	Non-*PCA*	All trusted nodes that are allowed to participate in the network other than the *PCA*.	• Public key of the network (K_{NET+}). • Private/public key pair (K_{n-}/K_{n+}). • Common key (*CK*).	• Source, destination, intermediate or forwarder of a control packet. • Idle.

Table 4.8: Types of nodes in *ARANz* (continued)

Phase	Node	Description	Keys/Certificates Owned	Possible States
After finishing the network setup	Local Certificate Authority (*LCA*)	A trusted node that is elected to keep information about and participate in certifying the trusted nodes (whether *LCA* or regular) in a particular zone.	• Zone *LCA*s certificate. • Node certificate ($Cert_n$). • Network private/public key (K_{NET-}/K_{NET+}). • Zone private/public key (K_{Zz-}/K_{Zz+}). • Private/public key pair (K_{n-}/K_{n+}). • Common key (*CK*).	• Source, destination, intermediate or forwarder of a control packet (in the network maintenance, location service, route instantiation and maintenance, and misbehaviour detection phases) or a data packet (in the data transmission phase). • Idle.
	Regular (non-*LCA*)	• All trusted nodes (other than the *LCA*s) which have been certified by the *PCA*. • Each regular node is required to inform its zone *LCA*s about its up-to-date position and ask them to update its certificate.	• Node certificate ($Cert_n$). • Public key of the network (K_{NET+}). • Public key of the zone (K_{Zz+}). • Private/public key pair (K_{n-}/K_{n+}). • Common key (*CK*).	

Table 4.9: States of nodes in *ARANz*

State	Description
Source	A trusted node that initiates sending a packet to a particular destination whether in the same zone or in a different one.
Destination	A trusted node that receives a packet for which it is the intended destination.
Intermediate	A trusted node that receives a packet distinct to another node.
Forwarding	A trusted intermediate node that participates in forwarding a packet towards its destination.
Idle	A trusted node that is not sending nor receiving any packet.

4.1.6 Keys

At the beginning of the network setup phase, *PCA* has the network's private key (K_{NET-}). All trusted nodes which will participate in the network have a private/public key pair (K_{n-}/K_{n+}), the public key of the network (K_{NET+}) and a common key (*CK*). Messages sent from the *PCA* to other nodes during the network setup phase are signed using the K_{NET-} to ensure that the *PCA* is truly the node that sent the message. Also, if these messages include private information, this information is encrypted using the destination's public key (K_{n+}) to ensure that the corresponding node is the single node that is able to decrypt this critical and important information. On the other hand, packets sent from non-*PCA* nodes are encrypted and decrypted using *CK* to enable other nodes to ensure that these packets are forwarded only by authorized nodes.

All control packets sent after the network setup phase are authenticated end-to-end and only authorized nodes participate at each hop between source and destination. The source of any packet signs the packet using its private key and appends its node certificate to the packet. If the source of a packet is a *LCA* server, it also includes its zone *LCA* certificate within the packet to enable the destination to make sure that the source *LCA* has a valid certificate for a particular zone. Moreover, if there is a private data to be sent between source and destination, it may also be encrypted using the public key of the destination to ensure its privacy and prevent other trusted nodes from reading the data itself. Each node along the path validates the previous node's signature (using the previous node's public key, which is extracted from its certificate), removes the previous node's certificate and signature, signs the original contents of the packet and appends its own certificate.

The network's private/public key pair (K_{NET-}/K_{NET+}) is used for encrypting and decrypting the nodes' certificates that are issued by the *LCA*s of a particular zone to nodes residing in that zone. On the other hand, a zone's private/public key pair (K_{Zz-}/K_{Zz+}) is used for encrypting and decrypting the zone's *LCA* certificate that is used to assure that a particular node is actually a *LCA* server for that zone. *LCA*s of a particular zone collaborate with each other to issue these certificates. Thus, the public key of a zone is owned by nodes residing in that zone and the private key is owned by *LCA*s of that particular zone.

The following table summarizes the keys involved in *ARANz*.

Table 4.10: Different keys used with *ARANz*

Key	**Owned by**	**Used for**
Common key (*CK*)	All nodes.	Encrypting and decrypting packets sent by non-*PCA* nodes in the network setup phase.
Node private/public key pairs (K_{n-}/K_{n+})	Each particular node *n*.	• Encrypting and decrypting control packets sent by a specific node after the network setup phase. • Destination's public key may be used for encrypting data packets to ensure data privacy and confidentiality.
Network private/public key pair (K_{NET-}/ K_{NET+})	• Public key is owned by all nodes. • Private key is owned by *PCA* and all *LCA*s.	• Encrypting and decrypting packets sent by *PCA* in the network setup phase. • Encrypting and decrypting nodes' certificates.
Zone private/public key pairs (K_{Zz-}/ K_{Zz+})	• Public key of a particular zone is owned by nodes residing in that zone. • Private key is owned by *LCA*s of that zone.	Encrypting and decrypting a particular zone *LCA*s certificate.

4.1.7 Types of Certificates

ARANz uses two types of certificates, node certificates and zone *LCA* certificates. Node *n* uses its node certificate ($Cert_n$) to authenticate itself to other nodes during exchanging control packets after the network setup is accomplished. On the other hand, zone *LCA* certificate of zone *z* ($CertLZ_z$) is used by the *LCA*s responsible for zone *z* as a proof that they are really *LCA*s of that particular zone.

Each regular node *n* has exactly one node certificate as follows:

$$Cert_n = [IP_n, K_{n+}, t, e]\ K_{NET-}$$

This certificate contains the *IP* address of *n* (IP_n), the public key of *n* (K_{n+}), a timestamp (*t*) of when the certificate was created and a time (*e*) at which the certificate expires. These variables are concatenated and signed with K_{NET-}. Nodes use the node certificates to authenticate themselves to other nodes upon exchanging control packets during the network maintenance, location service, misbehaviour detection as well as route instantiation and maintenance phases.

Each *LCA* owns two certificates, a node certificate and a zone *LCA* certificate. The node certificate is exactly the same as discussed in the previous paragraph. The zone *LCA* certificate binds the zone's number to its public key and is used by *LCA*s as a proof that they are *LCA*s of that particular zone. These certificates are used between *LCA*s of different zones, and between *LCA*s and nodes in their zones, during the exchange of packets related to network maintenance, location service and misbehaviour detection.

The zone *LCA* certificate contains the zone number (z), zone public key (K_{Zz+}), certificate creation timestamp (t) and certificate expire time (e). These certificates are signed using the zone private key. Consider if node n has been chosen to be a *LCA* server for zone z where n resides, it will own the following certificates:

$$Cert_n = [IP_n, K_{n+}, t, e]\ K_{NET-}$$

$$CertLZ_z = [z, K_{Zz+}, t, e]\ K_{Zz-}$$

Table 4.11 summarizes the different types of certificates used with our protocol.

Table 4.11: Different types of certificates used with *ARANz*

Certificate	**Issued by**	**Issued to**	**Used for**
Node certificate ($Cert_n$)	*LCA*s of the zone where node n resides.	Each trusted node n	Nodes authentication during network maintenance, location service, misbehaviour detection , and route instantiation and maintenance.
Zone *LCA*s certificate ($CertLZ_z$)	*LCA*s of the zone where the *LCA* resides.	*LCA*s	*LCA*s verification during network maintenance , misbehaviour detection and location service.

Both node certificate and zone *LCA* certificate are issued for the first time by the *PCA* during the network setup phase and have to be updated periodically via collaboration of *LCA*s of the zone where the node resides currently. Network private/public key pair is used for encrypting and decrypting a node's certificate to facilitate node movement between different zones. On the other hand, a particular zone private/public key pair is used for encrypting and decrypting the zone's *LCA* certificate. Thus, it is guaranteed that a zone's *LCA* certificate is really issued by *LCA*s of that zone since they are the exclusive nodes owning the private key of that particular zone.

4.1.8 Forwarding Techniques

There are seven techniques that are used with *ARANz* to forward packets, *network flooding*, *zone flooding*, *restricted directional flooding*, *source routing*, *LCA flooding*, *reverse path* and *data packets relaying*. In *network flooding*, a packet is sent to all nodes existing currently in the network. Thus, each node upon the receipt of a packet continues broadcasting the packet to all its neighbours. The general structure of a packet initiated by source *S* and targeted to all nodes residing currently in the network is:

$$S \xrightarrow{Fln} ALL\text{: } [Pid, \ldots]\ K_{S\text{-}},\ Cert_S$$

Where *Pid* is the packet type identifier. Note that the packet is signed using the source's private key and the source's certificate is attached to the packet if it is sent after the network setup phase. On the other hand, if the packet is sent during the network setup phase, it is signed using $K_{NET\text{-}}$ if it is initiated by *PCA*, or encrypted using the *CK* if the initiating node is a non-*PCA*.

In *zone flooding*, a packet is sent to all nodes residing currently inside a particular zone. Hence, when a node receives a packet, it processes and continues broadcasting the packet to its neighbours only if it exists currently in the specified zone. The general structure of a packet initiated by source *S* and targeted to all nodes residing in zone *z* is:

$$S \xrightarrow{Flz} ALL_z\text{: } [Pid, z, \ldots]\ K_{S\text{-}},\ Cert_S$$

When *restricted directional flooding* is used, each intermediate node continues broadcasting the packet only if it is closer to the destination than its previous hop. The general structure of a packet initiated by source *S* and sent using restricted directional flooding towards destination *D* is:

$$S \xrightarrow{Rdf} D\text{: } [Pid, IP_D, \ldots]\ K_{S\text{-}}, Cert_S$$

In *source routing* the source node chooses the route (*SR*) that the packet is supposed to go through until it reaches the intended destination and includes this route in the message itself. Each node along the path checks its next hop in the source route and forwards the packet to that node. Hence, the general structure of a packet sent using source routing technique is:

$$S \xrightarrow{Src} D\text{: } [Pid, SR, \ldots]\ K_{S\text{-}}, Cert_S$$

In *LCA flooding* a packet is targeted to all *LCA*s in the network. When a *LCA* receives the first copy of a packet initiated by one of the *LCA*s in the same zone, it processes the packet and continues sending it to the adjacent *LCA* in the neighbouring zone. This packet is sent using source routing if the adjacent *LCA* can be reached in one-hop or using restricted directional flooding if the adjacent *LCA* is not within the transmission range of the first *LCA*. Each *LCA* in the neighbouring zones checks whether it has received the packet from other *LCA*s in its zone, and if so the packet is dropped. Otherwise, the adjacent *LCA* will send multiple unicast packets using source routing to other *LCA*s in its zone. This process continues until the packet reaches all *LCA*s in the network. Thus, a packet targeted to all *LCA*s in the network is forwarded from *LCA* to another *LCA* either by source routing or by restricted directional flooding.

In the *reverse path* technique, reply packets are sent through reverse paths of their corresponding request packets. The following is the general structure of a packet sent through the reverse path assuming that node *I* is the intermediate node from which node *D* has received the first request packet initiated by node *S*:

$$D \xrightarrow{Rev} S\text{: } [Pid, IP_S, IP_I, \ldots]\ K_{D\text{-}}, Cert_D$$

Intermediate nodes continue sending the reply packet back through the reverse path to the source by unicasting packet to the predecessor from which they received the original request.

After finishing the route discovery and setup, the source begins sending the data to the destination. As in *ARAN*, once the route reply reaches the originator, it is guaranteed that the route found is authentic, hence, data packets exchanged between nodes after a route has been set up are not signed nor have attached certificates. Accordingly, each node simply *relays data packets* without any modification to its successor in the route obtained during the route instantiation process.

$$S \xrightarrow{Rly} D\text{: } [DATA, IP_S, IP_D, \ldots]$$

Table 4.12 summarizes the different techniques used to forward packets in *ARANz*, while Table 4.13 shows the strategies for sending packets during different phases of *ARANz* considering the source and destination nodes of the packets.

Table 4.12: Packets forwarding techniques used with *ARANz*

Forwarding technique	**Notation**	**Description**	**General packet structure**
Network flooding	$S \xrightarrow{Fln} ALL$	• Flooding packet to all nodes currently in the network. • Any node continues broadcasting the packet upon receiving it.	$[Pid, \ldots]K_{S\text{-}}, Cert_S$
Zone flooding	$S \xrightarrow{Flz} ALL_z$	• Flooding packet to all nodes existing currently in zone *z*. • Only nodes residing currently in zone *z* process and continue broadcasting the packet.	$[Pid, z, \ldots]K_{S\text{-}}, Cert_S$
Restricted directional flooding	$S \xrightarrow{Rdf} D$	• Sending packet using restricted directional flooding. • Intermediate node continues broadcasting the packet if it is closer to *D* than its predecessor.	$[Pid, IP_D, \ldots]K_{S\text{-}}, Cert_S$
Source routing	$S \xrightarrow{Src} D$	• Sending packet using source routing. • Each node along the path forwards packet to its next hop in *SR*.	$[Pid, SR, \ldots]K_{S\text{-}}, Cert_S$

Table 4.12: Packets forwarding techniques used with *ARANz* (continued)

Forwarding technique	**Notation**	**Description**	**General packet structure**
LCA flooding	$S \xrightarrow{Src} D$ or $S \xrightarrow{Rdf} D$	• Flooding packet to all *LCA*s in the network. • Each *LCA* upon receiving a packet from the same zone sends it to adjacent *LCA* (if it has not yet received the packet from that *LCA*). If the adjacent *LCA* is reachable in one-hop the packet is sent using source routing, else restricted directional flooding is used. • Each *LCA* in the neighbouring zones sends multiple source routing packets to other *LCA*s in its zone (if it has not yet received the packet from one of them).	$[Pid, SR, \ldots]K_{S-}, Cert_S$ or $[Pid, IP_D, \ldots]K_{S-}, Cert_S$
Reverse path	$D \xrightarrow{Rev} S$	• Sending packet through reverse path. • Intermediate node sends the reply packet to the predecessor from which it received the request.	$[Pid, IP_S, IP_I, \ldots]K_{D-}, Cert_D$
Data packet relaying	$S \xrightarrow{Rly} D$	• Relaying data packet to its destination. • Intermediate node simply relays data packets as is to its successor in the route obtained during the route construction process.	$[DATA, IP_S, IP_D, \ldots]$

Table 4.13: Strategies used for sending different packets in *ARANz*

Phase	**Packet sent**		**Packet sending strategy**
	From	**To**	
Network setup	*PCA*	Non-*PCA*	First packet is sent using network flooding. After that, source routing is used since *PCA*, at this stage, is acquainted with the positions of all other nodes.
	Non-*PCA*	*PCA*	Reverse path towards *PCA*.
Network maintenance as well as location service	Regular	*LCA* in the same zone	Restricted directional flooding.
	LCA	*LCA* in a different zone	Source routing containing only the destination's *IP* address (one-hop unicast) if destination node is within the transmission range of the source node. Otherwise, restricted directional flooding is used.
	LCA	Node in the same zone	Source routing if the destination is a single node. Otherwise, zone flooding is used.

Table 4.13: Strategies used for sending different packets in *ARANz* (continued)

Route instantiation and maintenance	Source	Destination	Restricted directional flooding.
	Destination/ intermediate	Source	Reverse path towards the source.
Data transmission	Source	Destination	Data packets relaying.
Misbehaviour detection system	Regular	*LCA* in the same zone	A packet indicating the misbehaviour of a node is sent using restricted directional flooding.
	LCA	*ALL*	A packet informing that a particular node is compromised is sent using network flooding.

The sole packet that is flooded to the entire network during the network setup phase is the first packet initiated by *PCA* to inform nodes currently in the network about starting the setup phase and collecting information about them. Packets sent from non- *PCA* nodes to *PCA* are sent through the reverse paths of the packets they receive from the *PCA*. After that, source routing is used to send packets from *PCA* to other nodes since *PCA*, at this stage, is acquainted with the position of all nodes in the network.

After finishing the network setup phase, sending packets from the regular nodes to *LCA*s of their zones is done using restricted directional flooding, since each node within that zone is aware of the positions of the *LCA*s. Restricted directional flooding is also used for communications among adjacent *LCA*s in neighbouring zones (if they are not reachable within one-hop). Source routing is used to send packets among *LCA*s of the same zone and from *LCA*s to regular nodes in their zones, since these *LCA*s are aware of positions of all nodes in their zone.

Communication during the route instantiation between the source and the destination nodes (whether they are in the same zone or in different zones) is done using restricted directional flooding. To circumvent voids (regions without nodes) in sparse networks, if restricted directional flooding of a request fails after two attempts, the packet is broadcast to the entire network. In route maintenance, error packets are forwarded along the reverse path towards the source.

As mentioned earlier, reply packets are sent through reverse paths of their corresponding request packets, and data packets are simply relayed to the next successor in the route obtained during the route instantiation process.

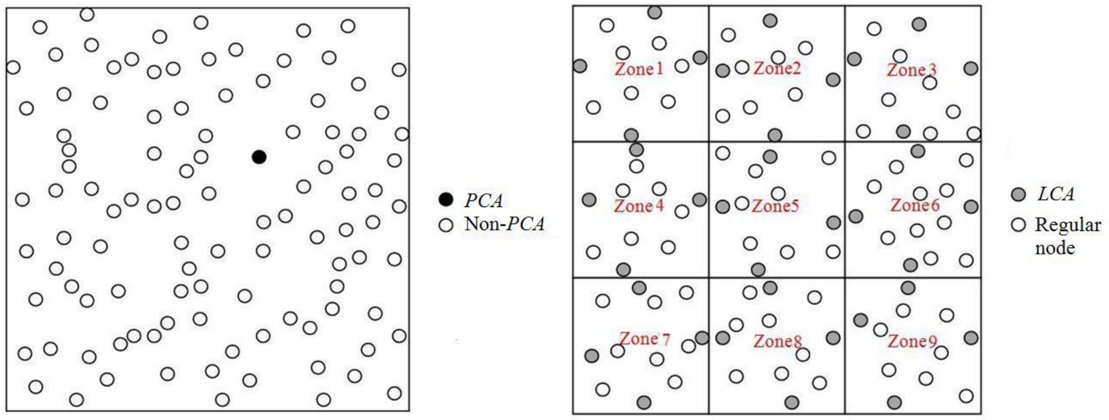
Two types of packets are sent during the misbehaviour detection system phase. The first packet is sent from nodes to report the misbehaviour of a particular node using restricted directional flooding towards the *LCA*s of the corresponding zone. The other packet is broadcast by the *LCA* servers to all nodes in the network to inform them that a particular node is compromised and should be excluded from the future routes.

4.2 Network Setup

In this section, the needed steps to deploy the network are explained. In the beginning, some initial communications must take place to enable the *PCA* to collect information about authorized nodes currently found in the network area. This information collection is an essential step to enable the *PCA* to proceed in dividing the area into zones, electing *LCA*s for each zone, certifying authorized nodes and informing nodes about the initial roles that they will play.

In the beginning, the needed initial communications between *PCA* and other trusted non-*PCA* nodes are discussed in *Section 4.2.1*. After that, dividing the area into different zones, electing the *LCA*s and the certification process are explained in *Sections 4.2.2, 4.2.3* and *4.2.4,* respectively. Next, *Section 4.2.5* addresses how to notify nodes of the initial role that each node will play, *LCA* or regular node. Finally, this phase is summarized in *Section 4.2.6*.

The network structure at the beginning as well as by the end of the network setup phase are shown in Figure 4.2, assuming that the area is divided into nine zones.



(a) At the beginning of the setup phase. (b) By the end of the setup phase.

Figure 4.2: Network structure before and after the network setup phase

4.2.1 Information Collection and Node Authentication

PCA starts the network setup by broadcasting a *NETwork SETup (NETSET)* packet. The purpose of this packet is to notify nodes of the initiation of the network setup phase and to collect information about nodes currently in the network. Suppose that node *P* has been chosen to play the role of *PCA*. It broadcasts the following *NETSET* packet:

$$PCA \xrightarrow{Fln} ALL\text{: } [NETSET, IP_P]\ K_{NET-}$$

The *NETSET* packet contains the packet identifier (*NETSET*) and the source's *IP* address (IP_P). This packet is signed using K_{NET-} to enable nodes to make sure that the *PCA* is really the node that sent the packet. Upon receiving the first *NETSET* packet, each node *n* records the *IP* address of the previous node, continues broadcasting the packet and replies with a *Node INformation (NIN)* packet to the *PCA*. The *NIN* packet contains the packet identifier (*NIN*) and the *IP* address of the source node (IP_n) along with the node's position (P_n), speed (S_n), battery remaining life time (B_n), *Central Processing Unit (CPU)* power (C_n) and memory (M_n). For example, node *A* will send the following packet to *PCA* (node *P*), assuming that node *B* is the intermediate node from which node *A* received the first *NETSET* packet:

$$A \xrightarrow{Rev} PCA\text{: } (NIN, IP_A, IP_P, IP_B, [P_A, S_A, B_A, C_A, M_A]\ K_{A-})\ CK,$$

where P_A is a pair of the X-coordinate (X_A) and Y-coordinate (Y_A) of node *A*. The *NIN* packets are encrypted and decrypted using *CK* to enable other nodes to ensure that this packet is forwarded only by authorized nodes. After encrypting the *NIN* packet, it is sent through the reverse path until it reaches the *PCA*. Each node upon receiving a *NIN* packet tries to decrypt it using *CK* to ensure that its previous node is trusted and proceeds to process the packet, otherwise, the packet is dropped. The node's information contained in the *NIN* packet is signed using the node's private key (K_{A-}) so that the *PCA* can be sure that the node that sent the packet is really the node claiming that, and to ensure the node's privacy by assuring that *PCA* is the sole node that is able to read this private information.

The *PCA* has an authentication table (*AT*) that includes a tuple for each node. This table contains the *MAC* address and/or the *IP* address for each node that intends to participate in the network along with its public key. This is possible as we are dealing with managed-open environment. This table is used by the *PCA* to authenticate the nodes' identities before allowing them to be part of the network.

Upon receiving a *NIN* packet from a specific node, the *PCA* will authenticate the node's identity. This authentication is done using the *AT* stored in the *PCA*, by searching for the tuple corresponding to the *IP* address of the node and trying to decrypt the node's information using the node's public key stored along with its *IP* address. If the node is trusted, it will be allowed to participate in the network and obtain the needed certificate(s), else the *NIN* packet is dropped.

After receiving the *NIN* packets from the authorized nodes currently in the network (waiting for a pre-defined time (*Tic*)), the *PCA* proceeds to divide the network into multiple virtual zones, assigning four *LCA*s in each zone (one for each side) and certifying the authorized nodes. These steps are explained in detail in the following sections.

4.2.2 Area Division

The network area is divided into multiple equal-size square-shape zones. The Coordinates ($CoorZ_z$) of each zone z are defined by both X-coordinate and Y-coordinate of its four corners, corner 1 (Xc_{z1}, Yc_{z1}), corner 2 (Xc_{z2}, Yc_{z2}), corner 3 (Xc_{z3}, Yc_{z3}) and corner 4 (Xc_{z4}, Yc_{z4}). The coordinates ($CoorZ_1$) of zone 1 for example are defined as:

$$CoorZ_1 = ((Xc_{11}, Yc_{11}), (Xc_{12}, Yc_{12}), (Xc_{13}, Yc_{13}), (Xc_{14}, Yc_{14}))$$

Figure 4.3 shows the network structure after dividing the area into nine zones illustrating corners of zone 1 and sides of zone 3.

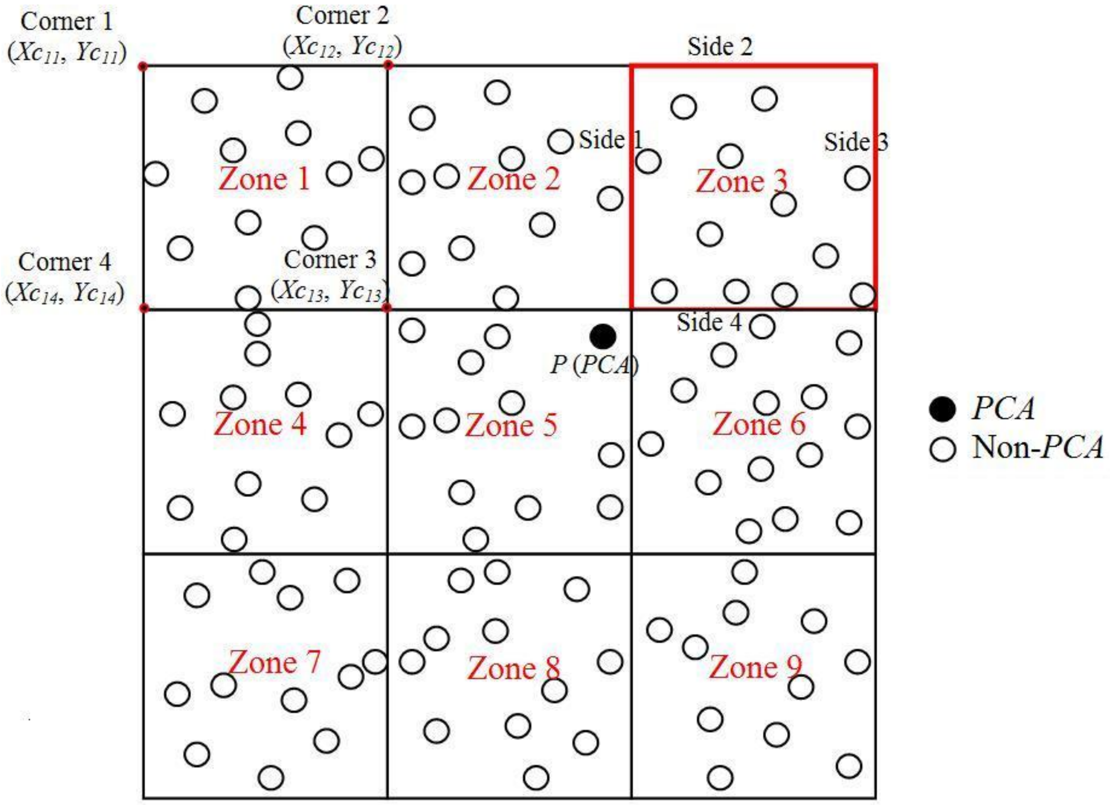


Figure 4.3: Network structure after dividing the area into zones

The *PCA* should take into consideration that if size of the zones is chosen to be small (*TR*×*TR*, for example), the overhead inside each zone is considerably reduced. Because all nodes inside a specific zone are within the transmission range of each other, all communications between *LCA*s and regular nodes, as well as among *LCA*s themselves, in a zone will take place using one-hop only. However, small zones lead to increased communications among *LCA*s in different zones because most communications become external. Additionally, having more *LCA*s increases the number of nodes with critical information (such as network private key), and hence, should be kept uncompromised.

On the other hand, if the zone size is chosen to be large, then position packets sent among *LCA*s of different zones are reduced since most communications will be local. Also, the number of *LCA*s having critical information is reduced. However, internal overhead increases since communication inside a zone is carried out using restricted directional flooding instead of one-hop communication.

As a result, *PCA* ought to take these points into consideration when dividing an area into zones. Accordingly, it has been decided to determine the most suitable zone size (or most suitable number of zones) through simulations (*Section 5.2.4*).

4.2.3 *LCA*s Election

After dividing an area into zones, the *PCA* will begin the process of electing *LCA*s. The *LCA*s are chosen to be near the zones' boundaries to make communication between *LCA*s of different zones easier and faster. Upon electing *LCA*s, each node *n* inside a zone *z* is assigned a weight representing its probability of being a *LCA* to a particular zone boundary *s*. The most important points in selecting *LCA*s are the distance between the node and the middle point of the zone boundary that the *LCA* will be responsible for (D_{nzs}), node's speed (S_n) and battery remaining life time (B_n). Choosing a *LCA* that is close to the middle point of the zone boundary and moving with a low speed increases the probability that the communication between *LCA*s of different zones will be done in one hop, which helps in protecting important packets. Choosing *LCA*s with low movement speed also increases the probability that the elected *LCA* will stay longer in the zone, and so there is no need to re-elect a new *LCA* within a short period of time. Moreover, choosing a node with high battery remaining life time reduces the likelihood of having its battery energy drained, i.e. reduces the probability of electing a new *LCA* and transferring important and secure information it possesses.

Another two important factors that must be considered when electing a *LCA* are the *CPU* processing power (C_n) and memory (M_n) of the nodes. *LCA*s with high *CPU* processing power and large memory significantly affect network performance since these *LCA*s could be the operation bottleneck for the position management scheme.

The *PCA* uses *NIN* packets it receives to calculate the probability of each node inside a specific zone to be elected as a *LCA* for a particular boundary. The probability ($ProbL_{nzs}$) of node *n* in zone *z* to be elected as a *LCA* of a particular boundary *s* is given as:

$$ProbL_{nzs} = Wd \times (1 - \frac{D_{nzs}}{Dmax}) + Ws \times (1 - \frac{S_n}{Smax}) + Wb \times (\frac{B_n}{Bmax}) +$$

$$Wc \times (\frac{C_n}{Cmax}) + Wm \times (\frac{M_n}{Mmax})$$

Where:

Wd: weight of distance between a node and middle point of a zone boundary,

Ws: weight of node movement speed,

Wb: weight of node battery remaining life,

Wc: weight of node *CPU* power,

Wm: weight of node memory capacity,

Dmax: maximum possible distance between a node and middle point of a zone boundary,

Smax: maximum possible node movement speed,

Bmax: maximum possible battery life time,

Cmax: maximum *CPU* power found in the market,

Mmax: maximum memory capacity exists in the market.

Values of the weights *Wd*, *Ws*, *Wb*, *Wc* and *Wm* are chosen as follows since we believe that speed, distance and battery lifetime are the most important when selecting the *LCA*.

$$Wd = Ws = 0.25$$

$$Wb = 0.2$$

$$Wc = Wm = 0.15$$

Distance D_{nzs} between node's position $P_n = (X_n, Y_n)$ and middle point (Xm_{zs}, Ym_{zs}) of boundary *s* in zone *z* is given as:

$$D_{nzs} = \sqrt{(X_n - Xm_{zs})^2 + (Y_n - Ym_{zs})^2}$$

Dmax is calculated once as distance between the middle point of a zone boundary and an opposite zone corner. Referring to Figure 4.4, *Dmax* can be calculated, for example, as the distance between the middle point (Xm_{13}, Ym_{13}) of boundary 3 in zone 1 and one of the zone corners in front of it ((Xc_{11}, Yc_{11}) or (Xc_{14}, Yc_{14})).

$$Dmax = \sqrt{(Xc_{11} - Xm_{13})^2 + (Yc_{11} - Ym_{13})^2}$$

Smax is a pre-defined value that depends on the environment where the protocol is deployed. *Bmax*, *Cmax* and *Mmax* are pre-defined values that depend on the current technology found in the market.

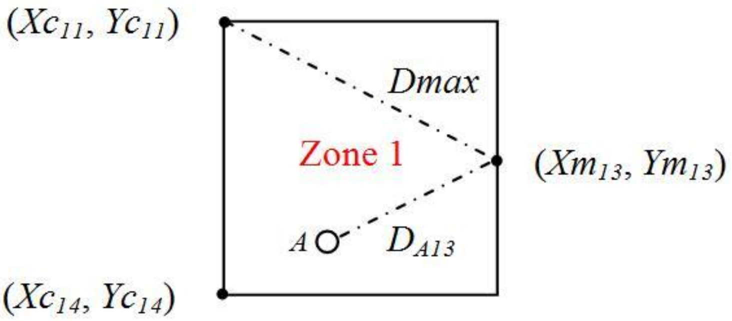


Figure 4.4: The maximum possible distance between the middle point of a zone boundary and a node inside the zone

Finally, the direction of node movement must also be considered, a node that is close to the zone boundary and moves outside the zone should not be chosen as a *LCA* since this will cause a new *LCA* election process within a short period of time. Figure 4.5 shows the network structure after electing the initial *LCA*s.

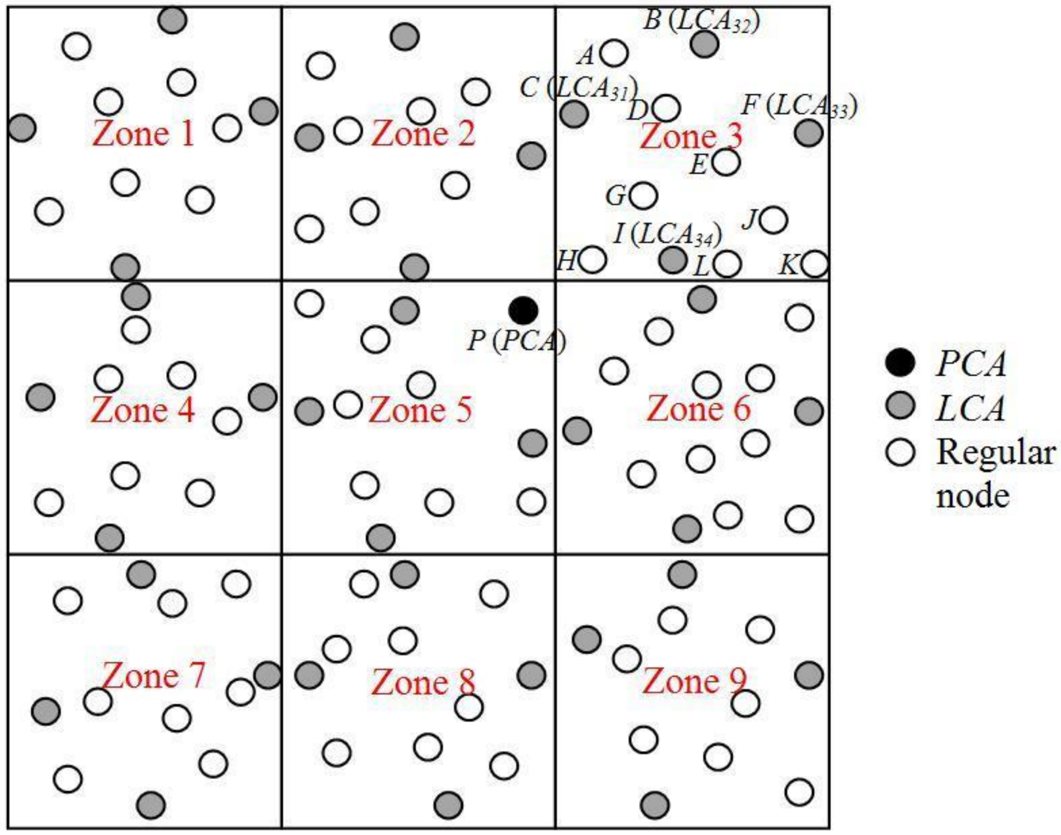


Figure 4.5: Network structure after electing the initial *LCA*s

4.2.4 Nodes Certification

In this section, we discuss the certification stage. All authorized nodes are issued node certificates. A node uses this certificate to authenticate itself to other nodes during the exchange of network maintenance, misbehaviour detection, position and routing packets. Referring to Figure 4.5, node *E*, for example, receives the following certificate from *PCA*:

$$Cert_E = [IP_E, K_{E+}, t, e]\ K_{NET-}$$

This certificate contains the *IP* address (IP_E) and public key (K_{E+}) of node *E*, in addition to a timestamp (t) at which the certificate was created and a time (e) of when the certificate will expire. These variables are concatenated and signed with K_{NET-}.

After issuing a certificate to a node, the corresponding tuple in the *AT* is completed by adding a timestamp and a certificate expiration date. This table (after being sent to corresponding *LCA*s) is used to update nodes' certificates. It is also used upon receiving a position request packet, *LCA* checks whether the destination of the route is local or external one, in order to send a position reply packet to the source or send position discovery packet to adjacent zone respectively.

In addition to its node certificate, each *LCA* receives a zone *LCA* certificate. This certificate is used between *LCA*s of different zones and between *LCA*s and nodes in their zones during the exchange of network maintenance and position packets. Node *I* (LCA_{34}), for example, will receive the following zone *LCA* certificate:

$$CertLZ_3 = [3, K_{Z3+}, t, e]\ \mathrm{K}_{\mathrm{Z3-}}$$

The zone *LCA* certificate binds the zone number, 3, to its public key, and contains the zone number, zone public key (K_{Z3+}), timestamp of when the certificate was created (*t*) and certificate expiration time (*e*). These certificates are signed using the zone private key and used by *LCA*s as a proof that they are *LCA*s of the specified zone.

4.2.5 Roles Notification

After dividing the area into zones and electing the initial *LCA*s, the *PCA* sends a unicast *Node ROLE* (*NROLE*) message to each participant node. These messages enable each node to know its preliminary role in the network (*LCA* or regular node). Source routing is used to send these messages because the *PCA* knows the position of all nodes in the network. Thus, the *PCA* chooses the route that each *NROLE* message will go through until it reaches the intended node and includes this route in the message itself.

Referring to Figure 4.5, *PCA* (node *P*) sends a unicast *NROLE* message to each regular node (non-*LCA*) *n*. This message contains the packet identifier (*NROLE*), the *IP* address of the source node (IP_P), the source route (*SR*) that the packet will pass through, the initial role that the node will play ($Role_n$), the node’s certificate ($Cert_n$), the zone number where it resides currently ($z{=}Zone_n$), the identities and positions of *LCA*s in its zone ($LCAsZ_z$), and the public key to use in this zone (K_{Zz+}). Hence, the general structure of the *NROLE* message sent to a regular node *n* residing in zone *z* is:

$$PCA \xrightarrow{Src} n\text{: } [NROLE, IP_P, SR, \{Role_n, Cert_n, z, LCAsZ_z, K_{Zz+}\}\ K_{n+}]\ K_{NET-}$$

Accordingly, the *PCA* sends the following *NROLE* message to node *E*, for example:

$$PCA \xrightarrow{Src} E\text{: [}NROLE, IP_P, (IP_H, IP_G, IP_E), \{\text{‘R’}, Cert_E, 3, LCAsZ_3, K_{Z3+}\}\ K_{E+}]\ K_{NET-}$$

Where ‘R’ indicates that *E* is a regular node, ‘3’ is the current zone of node *E*, and $LCAsZ_3$ = ((IP_C, P_C), (IP_B, P_B), (IP_F, P_F), (IP_I, P_I)) is the information about the *LCA*s responsible for Zone 3.

The *NROLE* messages are signed using K_{NET-} to ensure that the *PCA* is the node that sent the message. Also, private information is encrypted using the node’s public key (K_{n+}) to ensure that the corresponding node is the sole node that is able to decrypt this critical and important information.

The *PCA* also sends a unicast *NROLE* message to each node *n* that will play the role of a *LCA* for a particular zone boundary. This *NROLE* message contains the packet identifier (*NROLE*), the *IP* address of the source node (IP_P), the source route that the packet will pass through (*SR*), node’s role ($Role_n$), network private key (K_{NET-}), node’s certificate ($Cert_n$), number of that *LCA* in its zone (*s*), the number (*z*), coordinates ($CoorZ_z$) of the zone it is responsible for, zone *LCA*s certificate ($CertLZ_z$), numbers and coordinates of the 8-neighbouring zones ($8NbrZ_z$) of this zone, private/public key pair to be used in this zone (K_{Zz-}/K_{Zz+}), identities and positions of other *LCA*s in this zone ($LCAsZ_z$) as well as the identity and position of its adjacent *LCA* in the neighbouring zone ($AdjL_{zs}$).

The *NROLE* message also contains the number (l=$AdjZ_{zs}$), public key (K_{Zl+}) and part of the private key (K_{Zl-}) of the immediate neighbouring zone in order to be used when this neighbouring zone becomes empty. Moreover, it includes an authentication table (*AT*) that contains a tuple (*IP* address (IP_m), public key (K_{m+}), certificate issuing timestamp (*t*), certificate expiration time (*e*) and position (P_m)) for each node *m* that currently resides in this zone. Finally, this message includes the absent nodes (*AN*) list that contains the *IP* addresses and public keys of authorized nodes that were not in the network during network setup. Sending this list to all *LCA*s in the network enables absent nodes to join the network from any zone at any time. The general structure of the *NROLE* message sent to node *n* that will play the role of *LCA* responsible for boundary *s* in zone *z* is:

$PCA \xrightarrow{Src} n$: [$NROLE$, IP_P, SR, {$Role_n$, K_{NET-}, $Cert_n$, s, z, $CoorZ_z$, $CertLZ_z$, $8NbrZ_z$, K_{Zz-}, K_{Zz+}, $LCAsZ_z$, $AdjL_{zs}$, l, K_{Zl+}, part of K_{Zl-}, AT, AN} K_{n+}] K_{NET-}

Referring to Figure 4.5, node *I* will receive the following *NROLE* message:

$PCA \xrightarrow{Src} I$: [$NROLE$, IP_P, (IP_H, IP_I), {*‘L’*, K_{NET-}, $Cert_I$, 4, 3, $CoorZ_3$, $CertLZ_3$, $8NbrZ_3$, K_{Z3-}, K_{Z3+}, $LCAsZ_3$, (IP_X, P_X), 6, K_{Z6+}, part of K_{Z6-}, AT, AN} K_{I+}] K_{NET-}

Where *‘L’* indicates that *I* is a *LCA* node, $8NbrZ_3$ = (2, $CoorZ_2$, 5, $CoorZ_5$, 6, $CoorZ_6$) and $LCAsZ_3$ = ((IP_C, P_C), (IP_B, P_B), (IP_F, P_F), (IP_I, P_I)). Referring to Figure 4.5, AT= ((IP_A, K_{A+}, t, e, P_A), (IP_B, K_{B+}, t, e, P_B), …, (IP_L, K_{L+}, t, e, P_L)). Finally, suppose that both *U* and *Z* were not in the network during the network setup phase, then AN = ((IP_U, K_{U+}), (IP_Z, K_{Z+})).

4.2.6 Network Setup Phase Summary

As an essential step, the network setup phase starts with primary communications between *PCA* and other authorized non-*PCA* nodes currently found in the network. During these communications, *PCA* collects information about other nodes to assist it in certifying authorized nodes, dividing the area into zones, electing *LCA*s for each zone as well as informing nodes about their initial role that each is supposed to play (*LCA* or regular node). In Table 4.14, the packets exchanged during the network setup phase of *ARANz* are summarized. The pseudocode of this phase is provided in *Section A.1* in *Appendix A.*

Table 4.14: Packets sent during the network setup phase of *ARANz*

Pid	**Stand for**	**Description**	**From**	**To**
NETSET	NETwork SETup	• Sent to notify nodes currently in the network of initiating the network setup phase and collecting information about these nodes. • Signed using K_{NET-} so that nodes can make sure that the *PCA* is actually the node that has sent the packet.	*PCA*	All non-*PCA*
NIN	Node INformation	• Contains information about the source node such as position, speed, battery remaining life time, *CPU* power and memory. • Encrypted and decrypted using *CK* to ensure that this packet is forwarded by authorized nodes only. • Sent through the reverse path of the *NETSET* packet until reaching *PCA*.	All non-*PCA*	*PCA*
NROLE	Node ROLE	• A particular message is unicast to each participant node using source routing, containing the initial role (*LCA* or regular node) that this node will play.	*PCA*	All non-*PCA*

4.3 Network Maintenance

This section explains the needed communications among nodes to maintain the network structure. After the network setup phase, nodes can update their certificates, move freely in the network, move in and out the network as well as becoming corrupted or even destroyed. Our protocol has to be able to cope with these issues.

Before proceeding further let us highlight some points. Since by the end of the network setup phase each node owns its node certificate, this certificate can be used to apply the *ARAN* protocol authentication steps. The source node signs the packet using its private key and appends its node certificate to the packet. If the source of a packet is a *LCA*, it includes its zone *LCA* certificate in the packet to enable the destination to make sure that the source *LCA* has a valid certificate for its zone. Each node along the path validates the previous node’s signature (using the previous node’s public key, which is extracted from its certificate), removes the previous node’s certificate and signature, signs the original contents of the packet and appends its own certificate.

Restricted directional flooding is used for sending packets from regular nodes to the *LCA*s of their zones as well as for communications among adjacent *LCA*s in neighbouring zones (if they are not reachable within one hop). On the other hand, source routing is used for sending packets from the *LCA*s to nodes in their zones (including other *LCA*s), since these *LCA*s are aware of the positions of these nodes. By default, reply packets are sent through reverse paths of their corresponding request packets.

Section 4.3.1 discusses certificates update process while *Section 4.3.2* addresses node mobility. *Section 4.3.3* describes the steps that take place upon node failure. In *Section 4.3.4*, the coping mechanism for empty zones is presented. Finally, *Section 4.1.5* discusses the required *LCA*s synchronization.

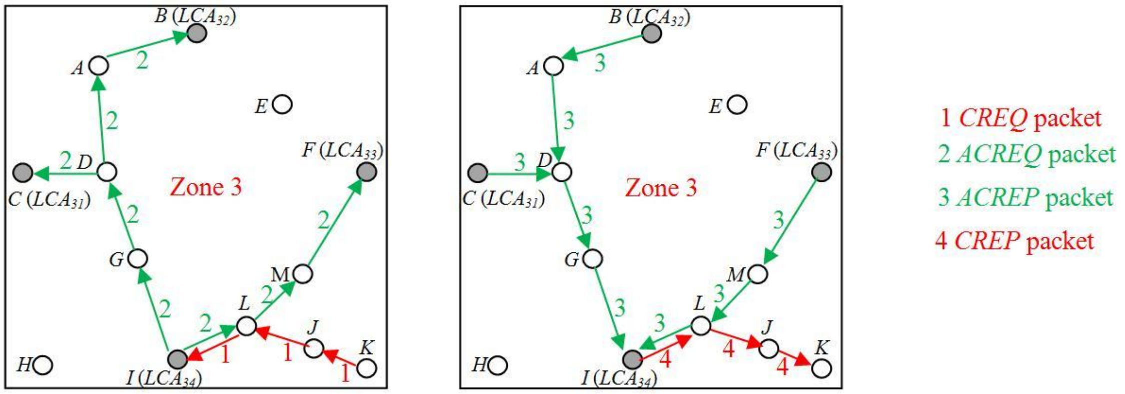
4.3.1 Certificates Update

At the end of the network setup phase, the area will have been divided into multiple zones. Each zone has four trusted *LCA*s located at the zone's boundaries and the positions of these *LCA*s are known to all nodes inside this zone. All nodes in a zone must maintain valid certificates with these *LCA*s. This is done by sending a *Certificate REQuest (CREQ)* packet periodically to any one of these *LCA*s, however, each node may update its certificate with the nearest *LCA* to itself to reduce overhead. This *CREQ* packet is sent using restricted directional flooding. Considering Figure 4.6, node *K* for example, will send the following *CREQ* packet to LCA_{34} (node *I*):

$$K \xrightarrow{Rdf} I\text{: } [CREQ, IP_I, N_K]\ K_{K\text{-}}, Cert_K$$

The *CREQ* packet includes a packet type identifier (*CREQ*), the *IP* address of the nearest *LCA* (IP_I) and the node's nonce (N_K). The packet is signed using the node's private key ($K_{K\text{-}}$) and the node's certificate ($Cert_K$) is appended to the packet to enable other nodes to validate the signature and verify that *K*'s certificate has not expired.

The purpose of the node's nonce is to uniquely identify a packet coming from a particular source. Each time *K* performs certificate request, it monotonically increases this nonce. Hence, a given (IP_n, N_n) pair is used to check whether this *CREQ* is processed previously or not.



(a) Certificate request packets sent. (b) Certificate reply packets sent.

Figure 4.6: Node *K* certificate update

The first node that receives the *CREQ* packet sets up a reverse path back to the source by recording its *IP* address. This is in anticipation of receiving a certificate reply packet to be sent back to the source. The receiving node uses *K*'s public key, which it extracts from *K*'s certificate, to validate the signature and verify that *K*'s certificate has not expired. The receiving node also checks the (IP_K, N_K) tuple to verify that it has not already processed this *CREQ*, nodes do not forward packets with already-seen tuples. The receiving node adds a new field (*Dist*) indicating the distance from itself to LCA_{34}, signs the content of the packet, appends its own certificate and continues broadcasting the packet to its neighbours. Let *J* be a neighbour that has received the *CREQ* sent by *K*. Node *J* subsequently rebroadcasts the *CREQ* packet to its one-hop neighbours.

$$J \text{ broadcasts: } [[CREQ, IP_I, N_K]\, K_{K\text{-}}, Dist]\, K_{J\text{-}}, Cert_K, \ Cert_J$$

Upon receiving the *CREQ* packet and using the certificates attached to it, *J*'s neighbour *L* validates the signatures for both *K* (the *CREQ* packet's initiator) and *J* (the neighbour it received the *CREQ* from). Node *L* now compares the recorded distance (*Dist*) to the distance between itself and LCA_{34} (P_I is not included in the packet since it is known to all nodes in zone 3). If *L* is closer to LCA_{34}, it continues broadcasting the packet after changing the distance value in the packet, else the packet is dropped. If node *L* decided to rebroadcast the packet, it removes *J*'s certificate and signature, records *J* as its predecessor, signs the content of the packet originally broadcast by *K* and appends its own certificate. Then *L* rebroadcasts the new *CREQ* packet:

$$L \text{ broadcasts: } [[CREQ, IP_I, N_K]\ K_{K-}, Dist]\ K_{L-}, Cert_K, Cert_L$$

Each intermediate node along the path repeats the same steps as *L* until the packet reaches LCA_{34}, which replies to the first *CREQ* that it receives for a source and a specified nonce. The intended *LCA*, upon receiving a *CREQ* packet, communicates with other *LCA*s in its zone to ask them whether to update the certificate or not. This is done by sending a packet to each *LCA* asking for *Acceptance of the Certificate REQuest (ACREQ)*. For example, LCA_{34} (node *I*) sends the following *ACREQ* packet to LCA_{32} (node *B*):

$$LCA_{34} \xrightarrow{Src} LCA_{32}\text{: } [ACREQ, (IP_G, IP_D, IP_A, IP_B), IP_K, N_K, CertLZ_3]\ K_{I-}, Cert_I$$

ACREQ packet is sent using source routing and includes a packet type identifier (*ACREQ*), source routing (IP_G, IP_D, IP_A, IP_B) towards LCA_{32}, zone certificate that LCA_{34} has ($CertLZ_3$) in addition to the *IP* address (IP_K) and nonce (N_K) of the node requesting the certificate. As with other packets, the sending *LCA* signs the *ACREQ* packet with its private key (K_{L-}) and appends its node certificate ($Cert_I$).

Each intermediate node in the path specified in the source routing sends the packet to its next hop in the route. Before proceeding in sending the packet, the intermediate node validates the signatures for both the *ACREQ* initiator and the neighbour it received the *ACREQ* from, removes previous hop certificate and signature, records its predecessor, signs the packet and appends its own certificate. $CertLZ_3$ is used by the destination *LCA* (LCA_{32}) to ensure that the *LCA* that sent the *ACREQ* (LCA_{34}) has a fresh zone *LCA* certificate. Figure 4.6 (a) shows sending the required certificate request packets (*CREQ* and *ACREQ*) to update *K*'s certificate.

If the node requesting the certificate is a well-behaving node, other *LCA*s in that zone will reply to the *ACREQ* packet they receive by sending an *Acceptance of Certificate REPly (ACREP)*. *ACREP* packets are sent through the reverse paths of their corresponding *ACREQ* packets. For example, LCA_{32} (node *B*) will send the following *ACREP* packet:

$$LCA_{32} \xrightarrow{Rev} LCA_{34}: [ACREP, IP_I, IP_A, IP_K, N_K, CertLZ_3]\ K_{B-}, Cert_B$$

This *ACREP* packet contains a packet type identifier (*ACREP*), the *IP* address of the *ACREQ* packet initiator (IP_I), the *IP* address of the intermediate node from which node *B* has received the *ACREQ* packet (IP_A), the zone certificate that LCA_{32} possesses ($CertLZ_3$) in addition to the *IP* address (IP_K) and nonce (N_K) of the node requesting the certificate. As with other packets, the sending *LCA* signs the *ACREP* packet with its private key (K_{B-}) and appends its node certificate ($Cert_B$).

LCA_{34} waits a pre-defined time (*Tac*) to collect *ACREP* packets from the *LCA*s of its zone. LCA_{34} is allowed to issue a certificate to the requesting node only if it receives *ACREP* packets from the majority of the *LCA*s of that zone (signed by their private keys). This technique helps in increasing robustness and security of the protocol by avoiding a single point of failure and attack, i.e. if one server fails or is compromised, the other three servers are still able to issue valid certificates to trusted nodes. Nodes failure and compromise are discussed in detail in *Sections 4.3.3* and *4.7.3*.

In the case that there are no failed or compromised servers, LCA_{34} will be able to issue a fresh certificate for *K* after receiving at least three *ACREP* packets (one of them is from itself). Then, LCA_{34} will unicast a *Certificate REPly (CREP)* packet back along the reverse path to the certificate requestor. Let the first node that receives the *CREP* sent by LCA_{34} be node *L*:

$$LCA_{34} \xrightarrow{Rev} K\text{: } [CREP, IP_K, IP_L, N_K, Cert_K, CertLZ_3]\ K_{I\text{-}}, Cert_I$$

This *CREP* packet includes a packet type identifier (*CREP*), the *IP* address (IP_K) of the original *CREQ* packet initiator, the *IP* address of the intermediate node from which node *I* received the *CREQ* packet (IP_L), the zone certificate that LCA_{34} possesses ($CertLZ_3$), the nonce sent by *K* (N_K) and finally the certificate issued for node *K* ($Cert_K$). $CertLZ_3$ is used by node *K* to ensure that the *LCA* issuing the certificate is really a *LCA* for its zone and has a fresh zone *LCA* certificate. As with other packets, the sending *LCA* signs the *CREP* packet with its private key ($K_{I\text{-}}$) and appends its node certificate ($Cert_I$).

Nodes receiving the *CREP* packet send it back to the predecessor from which they receive the original *CREQ*. Each node along the reverse path back to the source signs the *CREP* and appends its own certificate before forwarding the *CREP* to the next hop. Let *L*'s next hop to *K* be node *J*, then *L* will send the following:

L unicasts: $[[CREP, IP_K, IP_L, N_K, Cert_K, CertLZ_3]\ K_{I\text{-}}, IP_J]K_{L\text{-}}, Cert_I, Cert_L$

Before unicasting the *CREP* packet to *K*, node *J* validates *L*'s signature on the received packet, removes the signature and certificate, signs the content of the packet and appends its own certificate.

J unicasts: $[[CREP, IP_K, IP_L, N_K, Cert_K, CertLZ_3]\ K_{I\text{-}}, IP_K]\ K_{J\text{-}}, Cert_I, Cert_J$

Figure 4.6 (b) shows sending the needed certificate reply packets (*ACREP* and *CREP*) through the reverse paths.

*LCA*s inside a specific zone carry identical information about nodes in their zone to provide information backup and avoid a single point of failure. In order to achieve this, each *LCA* upon issuing a certificate should unicast a *Node CERTificate (NCERT)* packet to other *LCA*s containing the new issued certificate. For example, LCA_{34} (node *I*) will send the following packet to LCA_{32} (node *B*) using source routing:

$$LCA_{34}\xrightarrow{Src}LCA_{32}\text{: [}NCERT, (IP_G, IP_D, IP_B), IP_K, N_K, Cert_K, CertLZ_3\text{] }K_{I\text{-}}, Cert_I$$

*LCA*s also must maintain fresh node and zone *LCA* certificates. Hence, each *LCA* must periodically unicast an *ACREQ* packet to other *LCA*s in its zone. Upon receiving the required *ACREP* packets, the corresponding *LCA* will be issued both node and zone *LCA* certificates.

4.3.2 Node Mobility

In order to bring this model very close, if not identical, to the implemented real life practical systems, each node (including *LCA*s) can move freely in the network and depart to another zone without causing problems. This discussion considers two cases, when the moving node is a regular node or a *LCA* server.

4.3.2.1 Regular Node Mobility

When a regular node has moved a pre-defined distance (*Dmov*) from its last known position, it includes its new position in an *Update Node POSition (UNPOS)* packet sent to the nearest *LCA* in its zone. This *UNPOS* packet is signed using the node's private key and sent using restricted directional flooding. Considering Figure 4.6, node *K*, for example, sends the following *UNPOS* packet to LCA_{34} (node *I*):

$$K\xrightarrow{Rdf}I\text{: [}UNPOS, IP_I, N_K, P_K\text{] }K_{K\text{-}}, Cert_K$$

The *UNPOS* packet includes a packet type identifier (*UNPOS*), the *IP* address of the nearest *LCA* (IP_I), the sending node's nonce (N_K) and the node's new position (P_K). The packet is signed by the node's private key ($K_{K\text{-}}$) and the node's certificate ($Cert_K$) is appended to the packet to enable other nodes to validate the signature and verify that *K*'s certificate has not expired. Each time node *K* updates its position, it monotonically increases its nonce. Hence, a given (IP_n, N_n) pair is used to check whether this *UNPOS* has been processed previously or not.

The nearest *LCA*, in turn, sends the node's new position in an *UNPOS* packet to other *LCA*s in its zone using source routing. This helps *LCA*s to keep track of up-to-date positions of nodes inside the zone and enables them to discover the departure of nodes to the neighbouring zones. For example, LCA_{34} (node *I*) sends the following packet to LCA_{32} (node *B*):

$$LCA_{34} \xrightarrow{Src} LCA_{32}: [UNPOS, (IP_G, IP_D, IP_A, IP_B), IP_K, N_K, P_K, CertLZ_3]\ K_{I-}, Cert_I$$

UNPOS packet is sent using source routing and includes a packet type identifier (*UNPOS*), source routing (IP_G, IP_D, IP_A, IP_B) towards LCA_{32}, zone certificate that LCA_{34} has ($CertLZ_3$) in addition to the *IP* address (IP_K), nonce (N_K) and new position (P_K) of the moving node *K*. The sending *LCA* signs the *UNPOS* packet with its private key (K_{I-}) and appends its node certificate ($Cert_I$).

When a node leaves to one of the immediate 4-neighbouring zones, the *LCA*s of the departed zone remove the departing node's information from their tables and the nearest *LCA* to the new zone sends a *Departing NODE (DNODE)* packet to its adjacent *LCA*. This packet indicates that the departing node is trusted and includes its position. *DNODE* packets are sent using one-hop unicast if the adjacent *LCA* is within the transmission range of the *LCA* in the departed zone, else restricted directional flooding is used. Suppose that node *R* in Figure 4.7 leaves Zone 5 to Zone 6 (moves from position P_R to P'_R). It is clear that LCA_{61} (node *W*) is one-hop from LCA_{53} (node *V*), hence the following one-hop unicast packet is sent:

$$LCA_{53} \xrightarrow{Src} LCA_{61}: [DNODE, (IP_W), 5, 6, IP_R, P'_R, Cert_R, NZ_5, CertLZ_5]\ K_{V-}, Cert_V$$

The purpose of the zone nonce (NZ_5) is to uniquely identify a packet initiated by *LCA*s of Zone 5. Each time a *LCA* sends *DNODE* packet, it monotonically increases the zone nonce. Hence, a given (z, NZ_z) pair is used to check whether a *DNODE* packet initiated from zone *z* has been processed previously or not.

As a subsequent step, the *LCA* in the new zone sends a *New ZONE (NZONE)* packet to the departing node, containing the number and public key of the new zone, in addition to the *IP* addresses and positions of *LCA*s responsible for that zone. This *LCA* also sends multiple *New NODE (NNODE)* packets to other *LCA*s in its zone informing them about the new node. Both *NZONE* and *NNODE* packets are sent using source routing.

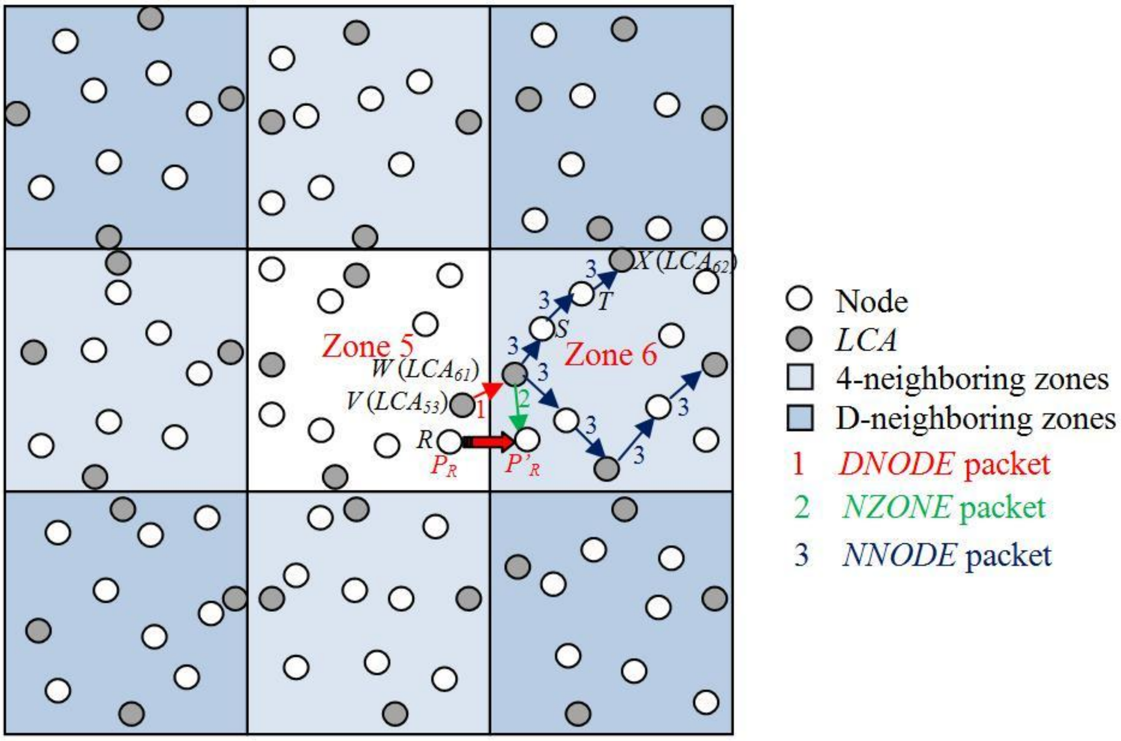


Figure 4.7: Movement of node *R* from zone 5 to a 4-neighbouring zone

In our example, LCA_{61} will send the following unicast *NZONE* packet to node *R*:

$$LCA_{61} \xrightarrow{Src} R\text{: } [NZONE,\ (IP_R),\ 6,\ K_{Z6+},\ LCAsZ6,\ NZ_6,\ CertLZ_6]\ K_{W-},\ Cert_W$$

Also the following *NNODE* packet is sent to LCA_{62} (node *X*) for example:

$$LCA_{61} \xrightarrow{Src} LCA_{62}\text{: } [NNODE,\ (IP_S,\ IP_T,\ IP_X),\ 6,\ IP_R,\ P_R,\ Cert_R,\ NZ_6,\ CertLZ_6]\ K_{W-},\ Cert_W$$

Now, let us suppose that the moving node leaves to one of the diagonal D-neighbouring zones. In this case, the *LCA* in the departed zone sends a *DNODE* packet to the adjacent *LCA* in the immediate neighbouring zone to indicate that this node is trusted. This *LCA*, in turn, resends the packet to the *LCA* adjacent to the new D-neighbouring zone. The latest will resend this packet to the adjacent *LCA* in its immediate neighbouring zone.

Suppose that node *R* in Figure 4.8 has left Zone 5 to Zone 9 (moved from position P_R to P'_R), the following unicast packets are sent:

$$LCA_{53} \xrightarrow{Src} LCA_{61}: [DNODE, (IP_W), 5, 9, IP_R, P'_R, Cert_R, NZ_5, CertLZ_5]\ K_{V-}, Cert_V$$

$$LCA_{61} \xrightarrow{Src} LCA_{64}: [DNODE, (IP_U, IP_Y), 5, 9, IP_R, P'_R, Cert_R, NZ_5, CertLZ_6]\ K_{W-}, Cert_W$$

$$LCA_{64} \xrightarrow{Src} LCA_{92}: [DNODE, (IP_Q), 5, 9, IP_R, P'_R, Cert_R, NZ_5, CertLZ_6]\ K_{Y-}, Cert_Y$$

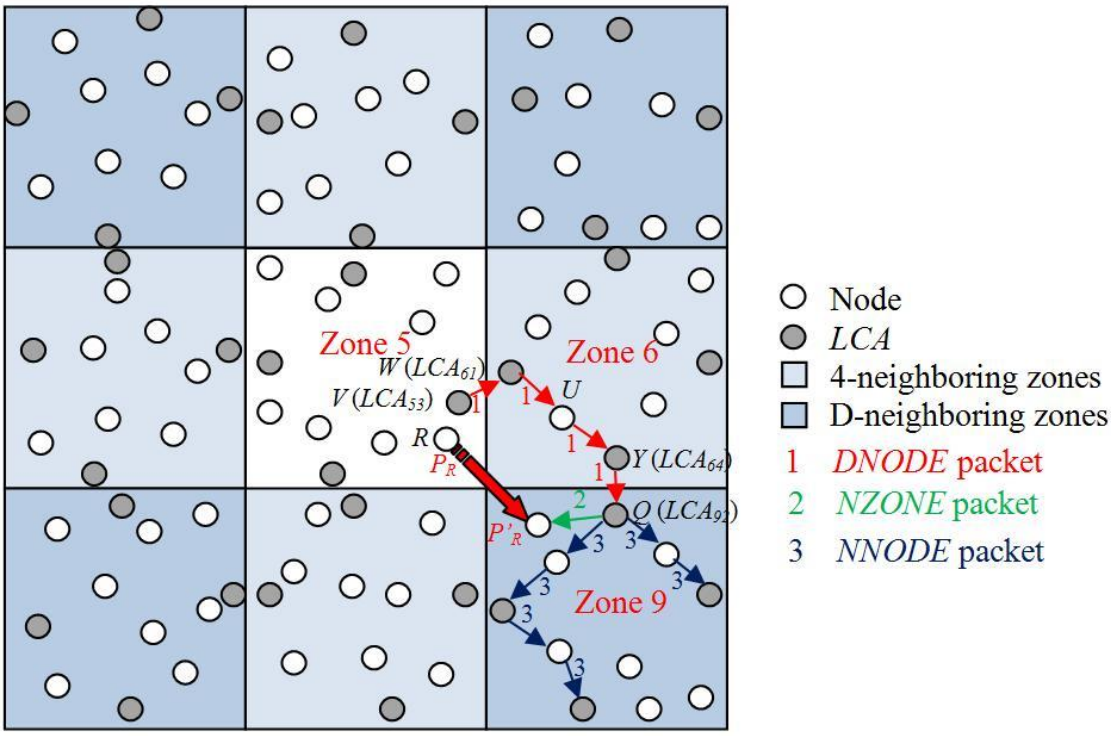


Figure 4.8: Movement of node *R* from zone 5 to a D-neighbouring zone

The first and third packets are sent using one-hop unicast since the destination is within the transmission range of the source. The second one is sent using source routing since the source and the destination are *LCA*s in the same zone. Now, *LCA* in the neighbouring zone that receives the *DNODE* packet bears the responsibility of sending a *NZONE* packet to the departing node. This *LCA* also sends multiple unicast *NNODE* packets to other *LCA*s in its zone telling them about the newly entered node.

4.3.2.2 Mobility of *LCA* Nodes

If a *LCA* has moved a pre-defined distance (*Dmov*) from its last known position, it must broadcast its position to nodes inside its zone, including other *LCA*s. It is also required to send its new position to its adjacent *LCA* in the neighbouring zone. Suppose that

LCA_{61} (node *W*) has moved *Dmov* distance from its last known position and that LCA_{53} (node *V*) is within the transmission range of LCA_{61}. LCA_{61} sends the following *Update LCA POSition (ULPOS)* and *Update Adjacent LCA POSition (UALPOS)* packets to nodes in its zone and its adjacent *LCA* (LCA_{53}) respectively:

$$LCA_{61} \xrightarrow{Flz} ALL_6\text{: } [ULPOS, 6, 1, P_W, NZ_6, CertLZ_6]\ K_{W-}, Cert_W$$

$$LCA_{61} \xrightarrow{Src} LCA_{53}\text{: } [UALPOS, (IP_V), 6, 1, P_W, NZ_6, CertLZ_6]\ K_{W-}, Cert_W$$

Where ‘6’ and ‘1’ mean that *LCA* number 1 in Zone 6 is the *LCA* that has moved to the specified position P_W. Hence, any node outside Zone 6 discards this packet upon receiving it.

Moreover, a *LCA* may decide to leave its zone, or its distance from the middle point of the zone boundary may become higher than a pre-defined distance (*Dsid*). In these two cases, a new *LCA* election operation is required. LCA_{61}, for example, upon deciding to leave its zone, sends the following *New LCA Election (NLCAE)* packet to nodes in its zone:

$$LCA_{61} \xrightarrow{Flz} ALL_6\text{: } [NLCAE, 6, 1, NZ_6, CertLZ_6]\ K_{W-}, Cert_W$$

Where ‘6’ and ‘1’ indicate that the elected *LCA* will replace *LCA* 1 in Zone 6. Thus, nodes in the neighbouring zones drop this packet upon receiving it. To help preserve resources of the nodes selected to play the role of *LCA*s, *LCA* is also able to send a *NLCAE* packet if it has served as a *LCA* for a pre-defined time (*Tsl*).

Upon electing the new *LCA*, the same points that were discussed in *Section 4.2.3* are considered. However, each node in the corresponding zone calculates its probability by itself to reduce load on the leaving *LCA*. Moreover, in an attempt to ensure fairness and load balancing, nodes that have recently played the role of a *LCA* are exempted. Hence, each node *n* should calculate fraction of time ($LCAF_n$) during which it serves as a *LCA* within the last *Ns* time slots. For each node to be able to calculate $LCAF_n$, it must keep track of a binary array, $Roles_n$ [*Ns*], containing *Ns* elements where:

$$Roles_n[i] = \begin{cases} 1\text{, if node } n \text{ has played the role of a } LCA \text{ during the last } i^{th} \text{ time slot} \\ 0\text{, if node } n \text{ has played the role of a regular node during the last } i^{th} \text{ time slot} \end{cases}$$

So $LCAF_n$ is calculated as follows:

$$LCAF_n = \frac{\sum_{i=1}^{Ns} Roles_n[i]}{Ns}$$

Upon electing a new *LCA*, each node calculates its probability ($ProbL_{nzs}$) as follows:

$$ProbL_{nzs} = Wd \times (1 - \frac{D_{nzs}}{Dmax}) + Ws \times (1 - \frac{S_n}{Smax}) + Wb \times (\frac{B_n}{Bmax}) +$$

$$Wc \times (\frac{C_n}{Cmax}) + Wm \times (\frac{M_n}{Mmax}) + Wf \times (LCAF_n)$$

Where *Wf* is the weight of the time fraction that node *n* served as a *LCA* in the last *Ns* time slots. Values of the weighting factors *Wd, Ws, Wb, Wc, Wm* and *Wf* are chosen as follows:

$$Wd = Ws = 0.25$$

$$Wb = 0.2$$

$$Wc = Wm = Wf = 0.1$$

These values are chosen since we believe that the speed, distance and battery lifetime are the most important when selecting the *LCA*.

Then each node sends its calculated probability through reverse path to the leaving *LCA*. Suppose that node *U* has received the *NLCAE* packet immediately from LCA_{61} (within one-hop), it sends the following *Node PROBability (NPROB)* packet to LCA_{61}:

$$U \xrightarrow{Rev} LCA_{61}\text{: } [NPROB, IP_W, 6, 1, NZ_6, ProbL_{U61}]\ K_U, Cert_U$$

The leaving *LCA* waits a pre-defined time (*Tpc*) to collect *NPROB* packets from nodes existing in its zone. After that, the leaving *LCA* selects the node with the highest probability of serving as the new *LCA*. Then, the leaving *LCA* broadcasts the following

New LCA (NLCA) packet so that all nodes inside that zone become aware of the address and position of the new *LCA* (node *S* for example):

$$LCA_{61} \xrightarrow{Flz} ALL_{6}: [NLCA, 6, 1, IP_{S}, P_{S}, NZ_{6}, CertLZ_{6}]\ K_{W-}, Cert_{W}$$

This information is also sent to adjacent *LCA* in the neighbouring zone (LCA_{53} (node *V*)) through a *New Adjacent LCA (NALCA)* packet:

$$LCA_{61} \xrightarrow{Src} LCA_{53}: [NALCA, (IP_{V}), 6, 1, IP_{S}, P_{S}, NZ_{6}, CertLZ_{6}]\ K_{W-}, Cert_{W}$$

Now the leaving *LCA* should transfer to the new *LCA* the needed information (similar to that included in the *NROLE* message sent from *PCA* to *LCA* nodes during network setup phase discussed in *Section 4.2.5*).

4.3.3 Node Failure

In this discussion, we consider the failure of both *LCA*s and regular nodes. This section discusses the sudden failure of the nodes. If a *LCA* knew, for example, that its battery power will run out in a specific time period, its response will be as in the case of deciding to leave its zone (as discussed in *Section 4.3.2*). The steps discussed in this section may be implemented also in the case of nodes movement outside the network boundaries.

4.3.3.1 *LCA* Node Failure

Sudden failure of a *LCA* (or moving outside the network boundaries) can be discovered from the periodic unicasting between the *LCA*s in a specific zone. As discussed in *Section 4.3.1* each *LCA* is supposed to update its *LCA* zone and node certificates with the cooperation of other *LCA*s in its zone. Consequently, if other *LCA*s do not receive the *ACREQ* packet in a pre-determined time (*Tcu*), they conclude that this *LCA* has a problem. So, one of these *LCA*s takes the responsibility of electing a new *LCA*. This *LCA* is called the voluntary *LCA* and can be chosen as the *LCA* with index equals to index of the failed *LCA* plus one (for example). Supposing that LCA_{61} had a problem, LCA_{62} (node *X*) will broadcast a *Failed LCA (FLCA)* packet to all nodes in the zone informing them about the failure and asking them to calculate their probability to replace the failed *LCA*.

$$LCA_{62} \xrightarrow{Flz} \text{ALL}_6\text{: [}FLCA\text{, 6, 1, }NZ_6, CertLZ_6\text{] }K_{X\text{-}}, Cert_X$$

Another packet is also sent using restricted directional flooding to the *LCA* adjacent to the failed one (LCA_{53} in our example). The *Failed Adjacent LCA (FALCA)* packet contains the same information as the *FLCA* packet in addition to the *IP* address of LCA_{53}.

Until the election of a new *LCA*, LCA_{61} is removed temporarily from $LCAsZ_6$. LCA_{62} also bears the responsibilities of LCA_{61}, such as renewing certificates of nodes near the failed LCA_{61}, initiating *ACREP* and sending important packets to the *LCA* adjacent to LCA_{61} in the neighbouring zone.

After receiving the probability packets from all nodes in the zone, LCA_{62} elects a new *LCA* to replace LCA_{61} by choosing the node with the highest probability. Suppose that the elected *LCA* is node *U*, LCA_{62} will broadcast the following *New LCA (NLCA)* packet so that all nodes inside Zone 6 know the address and position of the new *LCA*:

$$LCA_{62} \xrightarrow{Flz} \text{ALL}_6\text{: [}NLCA\text{, 6, 1, }IP_U, P_U, NZ_6, CertLZ_6\text{] }K_{X\text{-}}, Cert_X$$

This information is also sent to LCA_{61}'s adjacent *LCA* in the neighbouring zone through a *New Adjacent LCA (NALCA)* packet using restricted directional flooding.

Now LCA_{62} starts transferring all needed information to the new LCA_{61}. This information is similar to the information included in the *NROLE* message sent from the *PCA* to *LCA* nodes during the network setup phase (refer to *Section 4.2.5*).

After that, if the failed LCA_{61} has been repaired, it rejoins the network as a regular node. To enable this node to rejoin the network from any zone, the node's *IP* address and public key are sent to all *LCA*s in the network using *LCA* flooding. Hence, each *LCA* in Zone 6 sends a *Failed NODE (FNODE)* packet to its adjacent *LCA* in the neighbouring zone. The adjacent *LCA,* in turn, sends it to *LCA*s in its zone. This process continues until the *IP* address and public key of the failed node reaches all *LCA*s in the network.

Our protocol's robustness against node failure is expected to be higher, compared to original *ARAN* protocol. In *ARANz,* the failure of one or more *LCA*s does not prevent other nodes from updating their certificates since other *LCA*s in the zone will discover the failure and replace the failed *LCA*. Upon the sudden failure of the four *LCA*s in a particular zone, only nodes inside that zone will be incapable of updating their certificates. This situation is still better than that in *ARAN* protocol, where a *CA* failure prevents all nodes from updating their certificates and participating in the network activities.

4.3.3.2 Regular Node Failure

Regular node failure is also discovered from the periodic node certificate update. If a *LCA* has in its authentication table an expired node certificate and does not receive a *CREQ* packet within a pre-defined period of time (*Tcu*), it will conclude that this node has a problem. Then, the *LCA* that issued the last certificate for the failed node will send a *FNODE* packet like the one sent in the case of *LCA* sudden failure.

4.3.4 Empty Zones

Due to node movement, some zones may become empty. This section discusses network maintenance in case of empty zones. When many nodes leave a particular zone, the last four nodes remaining in that zone are its four *LCA*s. Whenever one of these *LCA*s desires to leave the zone, it must transfer its responsibilities to one of the other *LCA*s. This continues until the last node in the zone (that plays the role of the four *LCA*s) decides to leave the zone. Before departing from the zone, this node sends an *Empty ZONE (EZONE)* packet to its adjacent *LCA* in the zone it is leaving to. This packet informs the *LCA* of the new zone that this node is the last node leaving the zone. This

EZONE packet is also forwarded to *LCA*s in the 8-neighbouring zones (4-neighbouring zones and D-neighbouring zones) of the empty zone.

Now let us assume that a node leaves a particular zone and enters the empty zone. The *LCA* of the departed zone knows that this zone is empty, so it sends a zone *Private Key Part REQuest (PKPREQ)* packets to the four adjacent *LCA*s in the immediate 4-neighbouring zones of the empty zone. These packets are sent to ask the *LCA*s to send the empty zone private key's parts that they possess. These parts are sent via zone *Private Key Part REPly (PKPREP)* packets.

The sending *LCA* waits a pre-defined time (*Tkc*) to collect *PKPREP* packets from other *LCA*s. Upon receiving the key's parts this *LCA* combines these parts and sends a *Sole NODE (SNODE)* packet to the newly entered node notifying it that it is the only node in the zone and providing it with the needed information (similar to that included in *NROLE* message sent from *PCA* to *LCA*s during network setup phase discussed in *Section 4.2.5*).

The newly entered node issues itself the needed certificates and plays the role of the four *LCA*s until other nodes enter the zone. For example, if another node enters this zone, both nodes will play the role of two *LCA*s according to their positions.

4.3.5 *LCA* Synchronization

All *LCA*s in the network should have synchronized clocks to ensure the correctness of the protocol and to avoid a situation where two nodes in different zones (or even in the same zone) are issued certificates at the same moment with different timestamps. Hence, the type of synchronization needed for our protocol is maintaining relative clocks rather than having the clocks synchronized (adjusted) to a reference clock in the network, i.e. nodes run their local clocks independently, but keep information about the difference between their clocks and the system's clock so that at any instant the local time of the node can be converted to the system's time.

This synchronization can be achieved simply via *GPS* (El-Rabbany 2002; Kaplan & Hegarty 2005) or a synchronization scheme can be used to maintain synchronization among different *LCA*s in the network. A simple synchronization scheme is proposed in the subsequent paragraphs.

As a starting point, the *PCA* includes a timestamp within the *NROLE* message sent to *LCA*s during the network setup phase. So each *LCA* will be aware of the difference between its local clock and the *PCA*'s clock. Also, a timestamp is included in the information sent to a newly elected *LCA*.

Moreover, all clocks are subject to clock drift, i.e. oscillator frequency will vary unpredictably due to various physical effects (Sivrikaya & Yener 2004). Hence, periodically (each *Tls* seconds), one of the *LCA*s sends a *CLocks SYNchronization (CLSYN)* packet containing a timestamp (*t*) to other *LCA*s in the network to eliminate the effect of clocks drifts. In order to increase the robustness of the system, *LCA*s alternate this job between them. A nonce is used to avoid replay attack. The *LCA* includes its zone *LCA*s certificate with the message, signs the content of the packet and appends its own certificate. These packets are sent to all *LCA*s using *LCA* flooding technique, using source routing between *LCA*s in the same zone and using one-hop unicast or restricted directional flooding between adjacent *LCA*s.

Suppose that it is LCA_{34} (node *I*) turn to send the *CLSYN* packet. It sends the following packet using source routing to LCA_{32} (node *B*):

$$LCA_{34} \xrightarrow{Src} LCA_{32}\text{: } [CLSYN, (IP_G, IP_D, IP_B), t, N_I, CertLZ_3]\ K_{I-}, Cert_I$$

Regular nodes can include a timestamp in their certificates to know the system's time and check the validity of other nodes' certificates, so there is no need for extra communications between *LCA*s and regular nodes in a particular zone.

4.3.6 Network Maintenance Phase Summary

This section has explained the required communications among the nodes to maintain the network structure in the case of updating nodes' certificates, *LCA*s synchronization, movements of nodes in and out the network as well as corrupted and destroyed nodes. The following table summarizes packets sent to deal with these issues and the pseudocode of this phase is presented in *Section A.5* of *Appendix A*.

Table 4.15: Packets sent during the network maintenance phase of *ARANz*

Case	***Pid***	**Stand for**	**Description**	**From**	**To**
Certificate update	*CREQ*	Certificate REQuest	• Sent periodically requesting to update the certificate of node *n*. • Sent using restricted directional flooding.	Each regular node *n*	Nearest *LCA* in its zone
	CREP	Certificate REPly	• Contains the updated certificate of node *n*. • Sent through the reverse path of the *CREQ*.	Nearest *LCA* to *n*	Node *n*
	ACREQ	Acceptance of Certificate REQuest	• Sent to ask whether to update the certificate for *n* or not. • Sent using source routing.	Nearest *LCA* to *n*	Other *LCA*s in the zone
	ACREP	Acceptance of Certificate REPly	• Sent in the case of accepting the certificate update request. • Sent through the reverse path of the *ACREQ*.	Other *LCA*s in the zone	Nearest *LCA* to *n*
	NCERT	Node CERTificate	• Contains the newly issued certificate to enable zone's *LCA*s to store identical information. • Sent using source routing.	Nearest *LCA* to *n*	Other *LCA*s in the zone
Node failure	*FLCA*	Failed *LCA*	• Sent to initiate a new *LCA* election in the case of sudden *LCA* failure which is discovered if other *LCA*s in the zone *z* do not receive the *ACREQ* packet from the failed *LCA* in a pre-determined time (*Tcu*). • Sent using zone flooding.	One of the other *LCA*s in zone *z* (voluntary *LCA*)	All nodes in zone *z*
	FALCA	Failed Adjacent *LCA*	• Sent to inform the adjacent *LCA* about the failed *LCA*. • Sent using restricted directional flooding.	Voluntary *LCA*	Adjacent *LCA* of the failed one
	FNODE	Failed NODE	• Contains the *IP* address and public key of a failed node n to enable it to join the network from any zone. • Sent using *LCA* flooding, i.e. using source routing between *LCA*s in the same zone and using one-hop unicast or restricted directional flooding between adjacent *LCA*s.	*LCA* that issued the last certificate for n (if n is a regular node) or voluntary *LCA* (if n is a *LCA*)	All *LCA*s in the network

Table 4.15: Packets sent during the network maintenance phase of *ARANz* (continued)

Case	***Pid***	**Stand for**	**Description**	**From**	**To**
Node mobility	*UNPOS*	Update Node POSition	• Contains the new position of a node *n* that has moved a pre-defined distance (*Dmov*) from its last known position. • Sent using restricted directional flooding.	Moving node	Nearest *LCA* to *n*
	DNODE	Departing NODE	• Sent when a node *n* departs to a neighbouring zone to indicate that this node is trusted and contains the node's position. • Sent using one-hop unicast if the adjacent *LCA* is within the transmission range of the departed zone's *LCA*, else restricted directional flooding is used.	Nearest *LCA* to the zone that node *n* is departing to	Adjacent *LCA* in the neighbouring zone
	NNODE	New NODE	• Contains information about the new node. • Sent using source routing.	Adjacent *LCA* in the new zone	Other *LCA*s in its zone
	NZONE	New ZONE	• Contains the number and public key of the new zone as well as *IP* addresses and positions of the zone's *LCA*s. • Sent using source routing.	Adjacent *LCA* in the new zone	Departing node *n*
	ULPOS	Update *LCA* POSition	• Contains the new position of a *LCA* that has moved *Dmov* from its last known position. • Sent using zone flooding.	Moving *LCA*	All nodes in its zone
	UALPOS	Update Adjacent *LCA* POSition	• Contains the new position of a *LCA* that has moved *Dmov* from its last known position. • Sent using one-hop unicast or restricted directional flooding.	Moving *LCA*	Adjacent *LCA*
	NLCAE	New *LCA* Election	• Sent to initiate a new *LCA* election if a *LCA* has decided to depart its zone *z*, or its distance from the middle point of the zone boundary became higher than a pre-defined distance (*Dsid*). • Sent using zone flooding.	Departing *LCA*	All nodes in zone *z*

Table 4.15: Packets sent during the network maintenance phase of *ARANz* (continued)

Case	***Pid***	**Stand for**	**Description**	**From**	**To**
LCA election	*NPROB*	Node PROBability	• Contains the probability of a node in the corresponding zone z to replace the departing (or failed) *LCA*. • Sent through the reverse path of the *NLCAE* (or *FLCA*).	All nodes in zone *z*	Departing (or voluntary) *LCA*
	NLCA	New *LCA*	• Contains the *IP* address and position of the new *LCA*. • Sent using zone flooding.	Departing (or voluntary) *LCA*	All nodes in zone *z*
	NALCA	New Adjacent *LCA*	• Contains the *IP* address and position of the new *LCA*. • Sent using one-hop unicast or restricted directional flooding.	Departing (or voluntary) *LCA*	Adjacent *LCA*
Empty zone	*EZONE*	Empty ZONE	• Sent to inform *LCA*s of the 8-neighbouring zones that this zone will be empty. • Sent between *LCA*s in the same zone using source routing and between adjacent *LCA*s using one-hop unicast or restricted directional flooding.	Last node *n* leaving a particular zone *z1*	*LCA*s of the 8-neighbouring zones of *z1*
	PKPREQ	Zone Private Key Part REQuest	• Sent when a node leaves zone *z2* and enters an empty zone *z1*. • Sent to request the empty zone private key parts that the 4 adjacent *LCA*s have. • Sent using one-hop unicast or restricted directional flooding.	Nearest *LCA* in *z2*	4 adjacent *LCA*s of *z1*
	PKPREP	Zone Private Key Part REPly	• Contains the empty zone private key part they have. • Sent through the reverse path of the *PKPREQ*.	4 adjacent *LCA*s of *z1*	Nearest *LCA* in *z2*
	SNODE	Sole NODE	• Sent upon receiving and combining the private key parts. • Sent to inform *n* that it is the only node in the zone and giving it the needed information. • Sent using one-hop unicast or restricted directional flooding.	Nearest *LCA* in *z2*	node *n*
LCA synchronization	*CLSYN*	CLocks SYNchronization	• Sent periodically (each pre-defined time *Tls*) and contains a timestamp to help *LCA*s keep synchronized clocks. To increase the system robustness, *LCA*s alternate this job. • Sent using *LCA* flooding.	Any *LCA*	All *LCA*s in the network

4.4 Location Service

This section discusses the location service used to enable the source node to obtain the position of a specific destination. Two cases are considered, local communications (source and destination are in the same zone) and external communications (source and destination are in different zones).

Before starting route discovery, the source is supposed to get the destination’s position. The source *S* sends a *Position Discovery Packet (PDP)* to the nearest *LCA* in its zone using restricted directional flooding to ask the *LCA* about the position of the destination *D*. Thus, source *S* in Figure 4.9 sends the following *PDP* packet to LCA_{34} (node *I*):

$$S \xrightarrow{Rdf} LCA_{34}\text{: } [PDP, IP_I, N_S, IP_D]\ K_{S-}, Cert_S$$

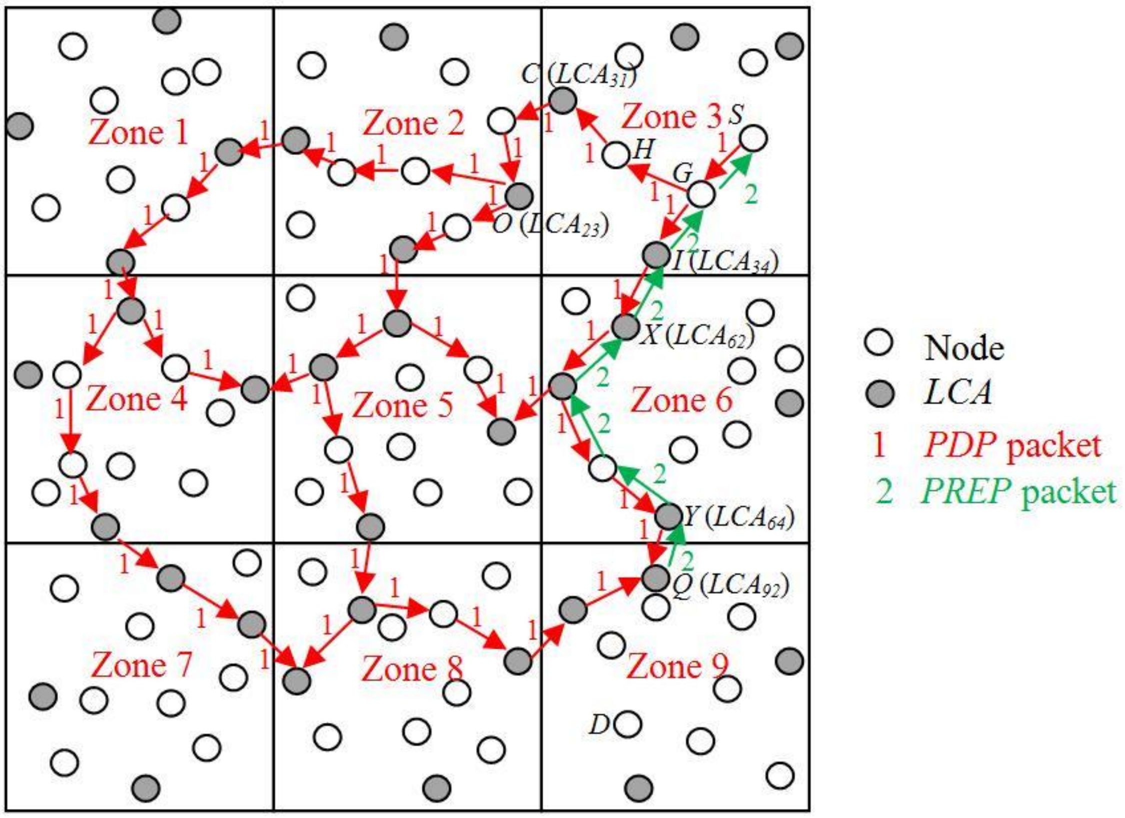


Figure 4.9: Location service phase of *ARANz*

The purpose of the source’s nonce (N_S) is to uniquely identify a *PDP* packet coming from a particular node. The first node receiving this *PDP* adds a new field (*Dist*) indicating the distance from itself to the intended destination (LCA_{34} in our example) to enable other nodes to continue the restricted directional flooding.

Upon receiving the first *PDP* packet, the anticipated *LCA* checks whether the destination is in its zone. If the destination is in the same zone of the source, the destination will be found in the authentication table of the *LCA*. Hence, the *LCA* will unicast a *Position REPly (PREP)* packet to the source. This *PREP* packet contains the destination's position and goes back along the reverse path to the source:

$$LCA_{34} \xrightarrow{Rev} S\text{: } [PREP, IP_S, IP_G, N_S, P_D, CertLZ_3]\ K_{I\text{-}}, Cert_I$$

Consider if the destination is in a different zone than the source, in this case, the destination will not be found in the authentication table of the nearest *LCA*. As a result, the *PDP* packet is forwarded to other *LCA*s in the network. So the nearest *LCA* will send multiple unicast *PDP* packets (using source routing) to other *LCA*s in its zone that have adjacent *LCA*s in neighbouring zones. For example, LCA_{34} will unicast the following *PDP* to LCA_{31} (node *C*):

$$LCA_{34} \xrightarrow{Src} LCA_{31}\text{: } [PDP, (IP_G, IP_H, IP_C), IP_S, N_S, IP_D, CertLZ_3]\ K_{I\text{-}}, Cert_I$$

Each *LCA* in that zone sends this *PDP* packet to its adjacent *LCA* in the neighbouring zone. This *PDP* is sent using unicast if the adjacent *LCA* is reachable in one-hop or using restricted directional flooding if the adjacent *LCA* is not within the transmission range of the first *LCA*. LCA_{34} for example will send the following one-hop unicast *PDP* packet to LCA_{62} (node *X*):

$$LCA_{34} \xrightarrow{Src} LCA_{62}\text{: } [PDP, (IP_X), 3, NZ_3, IP_D, CertLZ_3]\ K_{I\text{-}}, Cert_I$$

On the other hand, LCA_{31} sends the following *PDP* packet to its adjacent *LCA* (LCA_{23} (node *O*)) using restricted directional flooding:

$$LCA_{31} \xrightarrow{Rdf} LCA_{23}\text{: } [PDP, IP_O, P_O, 3, NZ_3, IP_D, CertLZ_3]\ K_{C\text{-}}, Cert_C$$

The purpose of the zone nonce (NZ_3) is to uniquely identify a packet initiated from a particular zone. Additionally, the position of LCA_{23} (P_O) is included in the request since the nodes in Zone 3 are not aware of the position of *LCA*s in Zone 2.

Now each *LCA* in the neighbouring zones checks whether it has received the packet from other *LCA*s in its zone, and if so, the packet is ignored. Else, the corresponding *LCA* will unicast the *PDP* packet (using source routing) to other *LCA*s in its zone that have adjacent *LCA*s in the neighbouring zones.

These steps are repeated until one of the *LCA*s finds the destination in its authentication table. This *LCA*, in turn, will unicast a *PREP* back along the reverse path to source. Suppose that LCA_{92} (node *Q*) is the *LCA* that found the destination's position and that LCA_{64} (node *Y*) can be reached within one-hop from LCA_{92}, then LCA_{92} unicasts the following *PREP* packet.

$$LCA_{92} \xrightarrow{Rev} LCA_{64}\text{: } [PREP, IP_Y, 3, NZ_3, P_D, CertLZ_9]\, K_{Q-}, Cert_Q$$

This packet is forwarded through the reverse path until it reaches the source node.

All position discovery steps are performed using the authentication steps used with *ARAN* protocol. Each node along the *PDP* path and the reverse (*PREP*) path validates the previous node's signature, removes the previous node's certificate and signature, signs the original content of the packet and appends its own certificate. There is only one difference between the behaviour of nodes upon receiving a request or a reply. When a node receives a *PDP* packet it records the previous node's *IP* address and forwards the packet. On the other hand, upon receiving a *PREP* packet it forwards the reply back to the predecessor from which it received the *PDP*.

In case that the source node does not receive a *PREP* packet within a pre-defined time (*Tpd*), it resubmits a *PDP* packet. After sending two *PDP* packets without receiving any reply, the source node starts the next phase (route instantiation) by broadcasting the route discovery packet (as in the original *ARAN* protocol).

Table 4.16 summarizes the packets sent during the location service phase. *Section A.6* in *Appendix A* provides the pseudocode of this phase.

Table 4.16: Packets sent during the location service phase of *ARANz*

Pid	**Stand for**	**Description**	**From**	**To**
PDP	Position Discovery Packet	• Initiated to ask for the position of destination *D*. • Sent using restricted directional flooding or source routing.	Source node *S*	Nearest *LCA* in its zone (or all *LCA*s having adjacent *LCA* in case of external communications)
PREP	Position REPly	• Contains position of *D*. • Sent along the reverse path of the *PDP*.	*LCA* that finds *D* in its authentication table	Source node *S*

4.5 Route Instantiation and Maintenance

This section explains the steps required to accomplish route discovery, setup and maintenance. After getting the destination's position (whether local or external) the source begins route instantiation to the destination by sending a *Route Discovery Packet (RDP)*. Unlike *ARAN*, sending *RDP* packets in *ARANz* is done using restricted directional flooding towards the anticipated destination node:

$$S \xrightarrow{Rdf} D\text{: } [RDP, IP_D, N_S, P_D]\ K_{S-}, Cert_S$$

The purpose of the source's nonce (N_S) is to uniquely identify a packet initiated by a specific node. As in other packets sent using restricted directional flooding, the first node that receives the packet adds a new field (*Dist*) indicating the distance between itself and the destination node. When the destination receives the first *RDP*, it unicasts a *Route REPly (RREP)* packet back along the reverse path to the source. Let the first node that receives the *RREP* sent by *D* be *C*:

$$D \xrightarrow{Rev} S\text{: } [RREP, IP_S, IP_C, N_S]\ K_{D-}, Cert_D$$

All route discovery steps are done using the authentication steps used with *ARAN* protocol. Each node along the *RDP* path and the reverse (*RREP*) path validates the previous node's signature, removes the previous node's certificate and signature, signs the original content of the packet and appends its own certificate. There is only one difference between the behaviour of the nodes upon receiving a request or a reply.

When a node receives a *RDP*, it records the previous node's *IP* address and forwards the packet. Upon receiving a *RREP*, it sends the reply to the predecessor from which it received the original request. If the source node does not receive a *RREP* packet within a pre-defined time (*Trd*), it resends another *RDP* packet. In case that the source node does not receive a *RREP* for the second *RDP* packet, the route instantiation is initiated by broadcasting the *RDP* packet (as it is the case in the original *ARAN* protocol).

ARANz, as *ARAN*, is an on-demand routing protocol, i.e. nodes keep track of whether routes are active or not. When no data is received on an existing route during the route's lifetime, the route is simply deactivated. Data received on an inactive route causes nodes to generate an *ERRor (ERR)* packet. Nodes also use *ERR* packets to report links in active routes that are broken due to node movement. For a route between source *S* and destination *D*, a node *B*, for example, generates the following *ERR* packet:

$$B \xrightarrow{Rev} S\text{: } [ERR, IP_S, IP_D, N_B]K_B\text{-}, Cert_B$$

This packet is signed and forwarded along the path toward the source without modification. The nonce (N_B) ensures that the *ERR* packet is fresh. Table 4.17 summarizes the packets exchanged during the route instantiation and maintenance phase, while *Section A.7* in *Appendix A* provides the corresponding pseudocode.

Table 4.17: Packets sent during route instantiation and maintenance phase of *ARANz*

Pid	**Stand for**	**Description**	**From**	**To**
RDP	Route Discovery Packet	• Sent to initiate route establishment to destination. • Sent using restricted directional flooding towards the destination node.	Source	Destination
RREP	Route REPly	• Initiated when the destination receives the first *RDP*. • Sent along the reverse path of the *RDP*.	Destination	Source
ERR	ERRor packet	• Generated if data is received on an inactive route or to report broken links in active routes. • All *ERR* packets must be signed. • Forwarded along the path toward the source without modification.	Node that notices the problem	Source

4.6 Data Transmission

After finishing the route discovery and setup, the source node begins sending the data to the destination. As in *ARAN* protocol, only the control messages between nodes are subject to signing and verifying, once the route reply reaches the originator, it is guaranteed that the route found is authentic. Thus, data packets exchanged between nodes after a route has been set up are not processed by *ARANz* in any way, i.e. they do not have attached certificates and are not signed. Accordingly, each node simply relays a data packet as is to its successor in the route obtained during the route instantiation process.

$$S \xrightarrow{Rly} D\text{: } [DATA, IP_S, IP_D, \ldots]$$

To ensure data privacy and prevent other trusted nodes from reading the data, the data can be encrypted using the public key of the destination, which the source may obtain during the position or route discovery phase. The pseudocode of the data transmission phase is presented in *Section A.8* in *Appendix A*.

4.7 Misbehaviour Detection System

Malicious nodes may cause some erratic actions, such as the use of invalid certificates, improperly signed packets and misuse of some packets. *ARANz* responds to all erratic behaviours in the same way, dropping all packets that show any erratic behaviour.

Malicious nodes, however, may cause more severe misbehaving actions and attacks, such as altering some fields in control packets, dropping data packets and fabricating error packets. In these cases, our protocol can collaborate with a misbehaviour detection system to help in detecting and isolating malicious nodes, such as the one proposed in this section.

The proposed system is flexible and can be used to protect against a wide set of attacks. The main concept is that each node has a trust table (*TT*) to maintain reputation information about each neighbouring node. In the *TT*, values regarding several events are stored. A node uses this value to evaluate its neighbour as misbehaving (malicious) or well-behaving node. Each node is responsible for collecting events from direct relations and computing its own trust values for its neighbours.

Section 4.7.1 discusses the process of collecting data about different trust metrics. After that, dealing with malicious and compromised nodes are explained in *Sections 4.7.2* and *4.7.3*, respectively.

4.7.1 Data Collection and Trust Metrics Calculation

One of the most important aspects of trust management schemes is the process of data collection. Therefore, it is essential to identify what events can provide a useful feedback to the system and assist in making the proper decision. Many trust metrics can be considered to disclose the cooperation willingness of nodes during route establishment and maintenance as well as data forwarding phases, however, as trade-off between security and implementation cost, a set of these metrics have been selected in this work. The behaviour aspects that have been chosen for monitoring are:

- Control packet modification: A node can collect trust information about neighbouring nodes during interactions regarding the try to modify some fields in *PDP*, *PREP*, *RDP* or *RREP* packets.
- Data packet dropping: To protect against black-hole and grey-hole attacks, a node should be evaluated regarding its willingness and sincerity in forwarding data packets. This can be checked either through overhearing, or based on link layer acknowledgements (Zahariadis et al. 2009).
- Error packet fabrication: To protect against fabricating *ERR* packets, each node keeps information about the number of *ERR* packets issued by each neighbour.

Coming to the quantification of trust, for the first two trust metrics, node *A* calculates trust values regarding node *B* considering modification attacks ($TrstVm_{AB}$) and dropping attacks ($TrstVd_{AB}$) using the following equations:

$$TrstVm_{AB} = \frac{Sm_{AB}}{Sm_{AB} + Fm_{AB}}$$

$$TrstVd_{AB} = \frac{Sd_{AB}}{Sd_{AB} + Fd_{AB}}$$

Where Sm_{AB} and Sd_{AB} stand for the number of successful co-operations, whereas Fm_{AB} and Fd_{AB} stand for the number of failed ones. In other words, for the first metric Sm_{AB} is the number of unmodified control packets and Fm_{AB} stands for the number of modified control packets received by node *A* from node *B*. For the second metric, Sd_{AB} stands for the number of delivered data packets and Fd_{AB} is the number of dropped data packets by node *B* that it already received from node *A*.

For the last trust metric, node *A* calculates a trust value regarding neighbour *B* considering *ERR* packets fabrication attack ($TrstVf_{AB}$) by counting the number of *ERR* packets issued by *B* that passes through node *A* towards the source.

4.7.2 Malicious Nodes

Once $TrstVm_{AB}$ or $TrstVd_{AB}$ become less than a threshold *Thm* or *Thd* respectively, node *A* considers node *B* as a malicious node. Also, if $TrstVf_{AB}$ becomes higher than a threshold *Thf*, node *A* believes that node *B* is a malicious node. In these cases node *A* excludes node *B* from future communications. Moreover, node *A* sends a *Misbehaving NODE (MNODE)* packet to report this misbehaviour to the nearest *LCA* in its zone. This packet is sent using restricted directional flooding. Suppose that the nearest *LCA* to node *A* is node *I*, then node *A* will send the following *MNODE* packet to node *I*:

$$A \xrightarrow{Rdf} I\text{: } [MNODE, IP_I, N_A, IP_B]\ K_{A-}, Cert_A$$

The *MNODE* packet includes a packet type identifier (*MNODE*), the *IP* address of the nearest *LCA* (IP_I), the sending node’s nonce (N_A) and the *IP* address of the misbehaving node (IP_B). The packet is signed by the node’s private key (K_{A-}) and the node’s certificate ($Cert_A$) is appended to the packet to enable other nodes to validate the signature and verify that *A*'s certificate has not been expired.

4.7.3 Compromised Nodes

If the majority of *LCA*s in a particular zone have received a pre-defined number (*Nm*) of *MNODE* packets indicating the misbehaviour of a particular node, then they can collaborate and broadcast a *Compromised NODE (CNODE)* packet. Consequently other nodes exclude this node from the future routes until its certificate expires normally. Suppose that the nearest *LCA* to the compromised node is node *I*, then node *I* will broadcast the following *CNODE* packet:

$$I \xrightarrow{Fln} ALL\text{: } [CNODE, N_I, [IP_B]\ K_{NET-}]\ K_{I-}, Cert_I$$

The *CNODE* packet includes a packet type identifier (*CNODE*), the nonce of the sending node (N_I) and the *IP* address of the compromised node (IP_B). The packet is signed by node's private key (K_{I-}) and node's certificate ($Cert_I$) is appended to the packet to enable other nodes to validate the signature and verify that *I*'s certificate has not expired. To ensure that the node initiated the packet is truly one of the *LCA*s in the network, the *IP* address of the compromised node is signed by K_{NET-}.

This technique is applicable also when the misbehaving node is a *LCA*. For example, if three *LCA*s of a particular zone received the pre-defined number of *MNODE* packets indicating the misbehaviour of the fourth *LCA* in their zone, they will remove this *LCA* from the $LCAsZ_z$ list of this zone, broadcast a *CNODE* packet and initiate a new *LCA* election process. Even before revoking the certificate of the misbehaving *LCA*, the other three *LCA*s are still able to issue certificates for trusted nodes in their zone though the compromised *LCA* may refuse to send *ACREP* packets for the *ACREQ* packets it receives.

In the case of two compromised *LCA*s at the same time in the same zone, neither the two trusted *LCA*s nor the compromised *LCA*s will be able to update certificates for the nodes in the zone. This situation may continue until the certificates of all nodes within the zone expire, in this state, they will not be able to participate in any future network activity. This situation may also end (before having nodes' certificates expired) by leaving one of the compromised *LCA*s to a neighbouring zone or having its battery energy run out. In these cases, a new *LCA* needs to be elected to replace the compromised one. Having a third trusted *LCA*, the three trusted *LCA*s will be able to continue their tasks as usual.

On the other hand, this situation may end by replacing one of the well-behaving *LCA*s with a compromised one (e.g. the well-behaving *LCA* has moved to a neighbouring zone and the newly elected one is a compromised node). This results in having three compromised *LCA*s of a particular zone at the same time. In this case the security of the whole network is compromise and these *LCA*s can collaborate together and issue certificates for untrusted nodes.

4.7.4 Misbehaviour Detection System Summary

This section has discussed the misbehaviour detection system. Table 4.18 summarizes the packets sent during the misbehaviour detection system phase.

Table 4.18: Packets sent during the misbehaviour detection phase of *ARANz*

Pid	**Stand for**	**Description**	**From**	**To**
MNODE	Misbehaving NODE	• Sent to report the misbehaviour of other nodes. • Sent using restricted directional flooding.	Any regular node *n*	Nearest *LCA* in its zone *z*
CNODE	Compromised NODE	• Sent after collaboration of the majority of *LCA*s in zone *z* upon receiving a pre-defined number (*Nm*) of *MNODE* packets for a particular misbehaving node. • Sent using network flooding.	*LCA*s of zone *z*	All nodes in the network

4.8 *ARANz* Performance and Security Analysis

In this section, a summary of the characteristics of the *ARANz* protocol is given as well as an analysis of its robustness in the presence of different attacks.

4.8.1 *ARANz* Performance Analysis

ARANz uses cryptographic certificates to prevent most of the attacks against Ad-Hoc routing protocols (such as modification, impersonation and fabrication of routing packets) and detect erratic behaviour (including the use of invalid certificates and improperly signed packets). *ARANz* divides the network area into zones and distributes trust among multiple *LCAs* in an attempt to achieve robustness, availability, load distribution and solving the single point of failure and attack problems.

Introducing multiple *LCA*s helps in distributing load and trust resulting in increasing security and robustness. High level security is achieved by avoiding a single point of attack problem. In *ARANz,* security of the network is compromised only if three *LCA*s in the same zone are compromised at the same time. Moreover, a high level of robustness is achieved due to avoiding a single point of failure problem. The failure of a single *LCA* in *ARANz* does not affect updating nodes' certificates since other *LCA*s in its zone are able to discover its failure (via periodic certificate update process) and elect another *LCA* to replace it. However, in original *ARAN* the *CA* is vital to the network and its failure prevents all nodes from updating their certificates.

Having multiple *LCA*s, on the other hand, gives rise to a need to synchronize their clocks to ensure protocol correctness. *LCAs* in *ARANz* run their local clocks independently while keeping information about the difference between their clocks and the system's clock. Hence, at any instant, the local time of a *LCA* can be converted to the system's time. Accordingly, there is no need to have the clocks synchronized (adjusted) to a reference clock.

Another point to mention is that because the *LCA*s may move freely in the network, new election operations may be required to maintain the network structure. It is anticipated that the communication carried out among nodes in *ARANz* to maintain the network structure, and update nodes' certificates and positions are outweighed by the overhead required to update the nodes' certificates in *ARAN*. In *ARAN* protocol, all certificate update request packets are broadcast to the entire network because nodes are unaware of the *CA*'s position. On the contrary, in *ARANz*, most packets are sent using restricted directional flooding, source routing, zone flooding or *LCA* flooding to significantly reduce the overhead.

RDP packets in *ARAN*z are sent using restricted directional flooding towards the destination in order to achieve better scalability and performance. Moreover, the use of restricted directional flooding helps in reducing overall overhead as well as saving network bandwidth.

Using restricted directional flooding requires that each node should be equipped with positioning instruments (such as *GPS* receivers) to be able to obtain its own geographical position. This assumption is acceptable since the recent availability of small, inexpensive and low-power positioning instruments justifies adopting position-based routing algorithms in mobile Ad-Hoc networks (Giordano et al. 2003). Moreover, utilizing restricted directional flooding in *ARAN*z results in longer route discovery latency, compared to original *ARAN*, due to time required to inquire about the destination's position.

Table 4.19 summarizes the characteristics of *ARAN*z protocol.

Table 4.19: Characteristics of *ARAN*z protocol

Criterion	***ARANz***
Approach	Position-based (restricted directional flooding)
Secure extension for	*AODV*
Basic security mechanism	Certificates and timestamps
Synchronization	Yes
Central trust	No
Main idea/ contribution	Solving scalability as well as single point of compromise and failure problems existing in *ARAN* protocol.

Table 4.19: Characteristics of *ARANz* protocol (continued)

Criterion	***ARANz***
Proposal	• Divides area into zones and introduces multiple *LCA*s in each zone. • Requires sending a *PDP* if the position of the destination is unknown. • Uses restricted directional flooding to forward *RDP*. • Uses cryptographic certificates to prevent most security attacks that face Ad-Hoc routing protocols. • Control messages are authenticated at each hop from source to destination, as well as on the reverse path from destination to source.
Scalability	High
Advantages	• Robust against most security attacks. • No centralized trust. • High scalability. • Reduced packet overhead.
Disadvantages	• Synchronization among *LCA*s. • Extra hardware (*GPS*). • Extra delay to inquiry about the destination's position.

4.8.2 *ARANz* Security Analysis

ARANz, like *ARAN*, uses cryptographic certificates to prevent most of the security attacks that Ad-Hoc routing protocols face. It introduces authentication, message integrity and non-repudiation as part of a minimal security policy for the Ad-Hoc environment.

Since all control packets in *ARANz* must be signed, a node cannot participate in the routing process without authorization from the *LCA*s of its zone. This access control, therefore, relies on keeping the *LCA*s uncompromised. Even if a *LCA* is compromised, the revocation mechanism discussed in *Section 4.3.5* can be executed to exclude this *LCA* from the used routes and elect a new *LCA*.

One may think that introducing multiple *LCA*s in *ARANz* may cause compromising the network if any of these *LCA*s is compromised. However, as mentioned in *Section 4.7.3*, the security of the whole network is compromised only if three *LCA*s of a particular zone are compromised at the same time without being able to identify them as compromised. In this case, these *LCA*s can collaborate together to issue certificates for untrusted nodes in their zone. Unlike *ARAN*, in which the network is compromised upon compromising a single node (*CA*). Consequently, a higher level of availability is achieved by *ARANz* due to avoiding a single point of attack problem.

The following is an analysis of *ARANz* robustness in the presence of different attacks introduced in earlier sections:

- Passive attacks: Detecting passive attacks is very difficult since the operation of the network itself is not affected. One way of overcoming such problems is to use powerful encryption mechanisms to encrypt the data being transmitted, thereby making it impossible for eavesdroppers to obtain any useful information from the data overheard. *ARANz* uses cryptographic operations to protect control packets from eavesdropping.
- Active attacks: *ARAN*z protocol is robust against most active attacks, as shown in the following discussion:
    - Spoofed routing signaling: All request packets are signed with the source's private key and contain its certificate. Similarly, reply packets include the destination's certificate and signature, ensuring that only the destination can respond to a particular request. This prevents impersonation attacks where either the source or destination is spoofed.
    - Fabricated routing messages: *ARAN*z does not prevent fabrication of routing messages, but it offers a deterrent by ensuring non-repudiation since all routing messages must contain the sender's certificate and signature. Therefore, a node that injects false messages into the network can be excluded from future communications.
    - Alteration of routing messages: *ARANz* specifies that all fields of a control packet remain unchanged between source and destination nodes. Since packet is signed by the initiating node, any alterations in transit would be detected, and the altered

packet would be subsequently discarded. Thus, modification attacks are prevented in *ARANz* and nodes altering routing messages are detected and prohibited from participating in future routes.

- Forwarding attacks: As in *ARAN* protocol, secure forwarding is accomplished by preventing unauthorized nodes from participating in forwarding data packets. This can be achieved using the cryptographic material available to *ARANz,* but would add overhead to the cost of data transmission, or by using shared keys instantiated between neighbours during the route reply process. This ensures end-to-end integrity but does not prevent certificate holders from replay attacks. Nodes dropping data packets can be detected and excluded using the proposed misbehaviour detection scheme.

The following two tables give a summary of the security requirements satisfied by *ARANz* protocol as well as the different attacks that *ARANz* defends against.

Table 4.20: Security requirements satisfied by *ARANz* protocol

Requirement	**Satisfied**
Availability	Medium
Authentication	Yes
Confidentiality	No
Integrity	Yes
Non-repudiation	Yes

Table 4.21: Robustness of *ARANz* against existing attacks

Type	**Attack**	**Robust against**
Passive attacks	Eavesdropping	Prevented via the used encryption techniques.
Active attacks	Impersonation	Yes
	Fabrication	No, but provides non-repudiation, hence, nodes fabricating packets can be excluded using the proposed misbehaviour detection scheme.
	Modification	Yes
Forwarding attacks	Modification	Yes
	Dropping	Yes

4.9 Chapter Summary

In this chapter, a detailed discussion of the phases of our proposed routing model has been presented. This discussion starts with the network setup phase, including certification process, dividing the area into zones and electing the certificate authority servers. After that we look at the location service. Next, route discovery, setup and maintenance are discussed. Then, the data transmission phase is explained. After that, the misbehaviour detection system is presented. Finally, an analysis of *ARAN*z performance and security is provided.

Chapter 5 presents a detailed discussion of our simulation methodology and scenarios as well as evaluating and comparing *ARAN*z to other existing routing protocols.

Chapter 5

Simulations, Results and Performance Analysis

This chapter explains our simulation methodology and scenarios as well as discussing the results of the experiments carried out. Simulation methodology and scenarios are explained in *Section 5.1*. *Section 5.2 g*ives an introduction about the conducted statistical analysis tests. After that the results we obtained are discussed in *Section 5.3*, while *Section 5.4* summarizes the obtained results.

5.1 Simulation Methodology and Scenarios

In this work, we have studied the effect of different scenarios on the network performance and compared the proposed protocol *ARANz* with existing protocols. We shall compare our protocol with the original *ARAN* protocol because our protocol is based on it. We also consider *AODV* protocol for comparison because *AODV* is often considered as a benchmark for evaluating the performance of Ad-Hoc routing protocols (Li et al. 2008; Nanda 2008) and because *ARAN* was proposed based on it. Hence, we compare the performance of our protocol against both *ARAN* and *AODV* routing protocols.

Section 5.1.1 describes simulation setup step. *Sections 5.1.2* and *5.1.3* present constant and variable simulation parameters, respectively. Finally, the performance metrics used to evaluate the performance of the proposed protocol are explained in *Section 5.1.4*.

5.1.1 Simulation Setup

GloMoSim is used as a simulation tool to evaluate the performance of *ARAN*, *ARANz* and *AODV* protocols. *AODV* is already implemented in *GloMoSim*, so two new models called “ARAN” and “ARANz” have been added to *GloMoSim* to simulate the original *ARAN* protocol and our new proposed protocol, respectively. The needed steps to add a new protocol to *GloMoSim* are explained in *Section B.8* in *Appendix B*.

For simulating these protocols we assumed that every host can be uniquely identified and its identity cannot be changed throughout the lifetime of the Ad-Hoc network. The identity is used in the identification procedure. In *GloMoSim*, each node has a “node address” field which is a sequence number, so this filed is used as the node’s identity.

5.1.2 Constant Simulation Parameters

Node transmission range of 250 meters (m) is used as it is a typical value for wireless local area networks in a free area without any obstacles (Farkas et al. 2006). The initial positions of the nodes are chosen randomly. After that, all nodes are granted full mobility, i.e. they are allowed to move any where inside the whole area. Node mobility is simulated according to the random waypoint mobility model, in which each node travels to a randomly selected location at a configured speed and then pauses for a configured pause time, before choosing another random location and repeating the same steps. This mobility model is used since experiments in (Camp et al. 2002) and (Izuan & Zukarnain 2009) show that the random waypoint mobility model has the highest data packet delivery ratio, the lowest end-to-end delay and the lowest average hop count compared to the random walk mobility as well as random direction mobility models. Moreover, this model is the most common model used by researchers for simulating node mobility in Ad-Hoc network (Camp et al. 2002). A fixed pause time of 30 seconds (s) is used for simulating different scenarios (Sanzgiri et al. 2005).

For the three protocols, the simulation time is chosen to be 1000s according to simulator clock. This value is chosen since data packets are generated by the sources only during first 800 seconds of simulation time. Also, we note that most studded metrics remain constant after 1000s and no new results are obtained when increasing the simulation time longer than this period. We use 802.11 *MAC* layer and *Constant Bit Rate (CBR)* traffic generator over *User Datagram Protocol (UDP)* (Sanzgiri et al. 2005). The source and destination pairs are chosen randomly in both local and external communications.

Local communications mean that the two communicating nodes reside in the same zone. On the other hand, if a node has data to be sent to a node in another zone, this is referred to as external communication. Five *CBR* sessions are simulated in each run, including both local and external communications. Each session generates 1000 data packets of 512 bytes each at the rate of 4 packets/s.

In managed-open environments, keys are generated and exchanged a priori (Sanzgiri et al. 2005). Accordingly, for simulating *ARAN* and *ARANz*, we assume that the key distribution procedure is completed.

ARAN and *ARANz* are simulated using a 512-bit key and 16-byte signature (Sanzgiri et al. 2005). Trusted nodes are required to update their certificates every 60s (Lim & Lakshminarayanan 2007) from the *CA* and the nearest *LCA* upon simulating *ARAN* and *ARANz*, respectively.

For either protocol, a routing packet processing delay of 1 millisecond (ms) is assumed. This value is obtained through field testing of the *AODV* protocol implementation (Perkins & Royer 1999). Additionally, a processing delay of 2.2ms is added to account for the cryptographic operations for *ARAN* and *ARANz*. This value is adopted from (Sanzgiri et al. 2005), which they obtained through the implementation testing of measuring processing routing messages of *ARAN* for both a laptop and a handheld computer.

A random delay between 0ms and 10ms is introduced before the retransmission of a broadcast packet in order to minimize collisions. This is required since the 802.11 *MAC* protocol does not perform a ready-to-send/clear-to-send exchange for broadcast packets. Upon having fairly dense networks, the probability of collision of broadcast packets becomes quite high without using such a random delay (Sanzgiri et al. 2005).

In order to have a consistent comparison of results, a basic version of *AODV* is used, which does not include optimizations, such as the expanding ring search and local repair of routes (Sanzgiri et al. 2005).

For simulating *ARANz*, nodes update their positions to the nearest *LCA* if they moved 50m from their last known position. Also, a new *LCA* election is initiated if the distance between a *LCA* position and middle point of the zone boundary becomes higher than 250m. *LCA*s in *ARANz* should synchronize their clocks every 60s. Moreover, to help preserve resources of the nodes selected to play the role of *LCA*s, a *LCA* is able to start a new *LCA* election procedure if it has served as a *LCA* for 120s.

Table 5.1 summarizes the constant simulation parameters that have been used.

Table 5.1: Constant simulation parameters

Parameter	**Value**
Routing protocols	*AODV*, *ARAN* and *ARANz*
MAC layer protocol	IEEE 802.11
Transport layer protocol	*UDP*
Simulation time	1000s
Nodes transmission range	250m
Initial node placement	Random
Node mobility model	Random waypoint
Node pause time	30s
Data traffic generator	*CBR*
Number of *CBR* sessions	5 sessions
Number of data packets per session	1000 packets
Data packet size	512 bytes
Packet rate	4 packets/s
Packet processing delay	1ms
Cryptographic operations delay	2.2ms (for *ARAN* and *ARANz*)
key size	512 bits (for *ARAN* and *ARANz*)
Signature size	16 bytes (for *ARAN* and *ARANz*)
Certificates update	Every 60s (for *ARAN* and *ARANz*)
Position update	Upon moving 50m from last known position (for *ARANz*)
*LCA*s election	After serving as a *LCA* for 120s or if *LCA* distance from the zone boundary become more than 250m (for *ARANz*)
Clock synchronization	Every 60s (for *ARANz*)

5.1.3 Variable Simulation Parameters

We tested the effect of seven important parameters of the network. These parameters are the *node speed, network size, node density, local communication percentage, zone size, node failure percentage* and *malicious node percentage*. The following is a discussion of these parameters:

1. Node speed: Different values for node speed are considered ranging from 0 m/s to 10 m/s. Upon studying other parameters effect, a maximum speed of 5 m/s is used, as it is considered as a moderate speed (Sanzgiri et al. 2005).
2. Network size: Multiple network sizes are considered ranging from 1 kilometer × 1 kilometer (1km×1km) to 3km×3km. Upon studying other parameters, a 2km×2km terrain is used since it is considered as a relatively large size Ad-Hoc network (Sanzgiri et al. 2005).
3. Node density: Several node densities are considered ranging from 40 nodes/km^2 to 100 nodes/km^2. Upon studying other parameters, node density of 60 nodes/km^2 is used. This value is chosen after conducting experiments to study the effect of node density on network performance (*Section 5.3.3*). The results of these experiments demonstrate that the minimum node density that resulted in the minimum path length is 60 nodes/km^2. Moreover, this node density value result in having a moderate density network.
4. Local communication percentage: Numerous local communication percentages are considered ranging from 0% to 100%. Upon studying other parameters effect, local communication percentage of 60% is used. Hence, in each run, three of the five *CBR* sessions are chosen to be local and the other two are external. The motive behind choosing this percentage is that we believe that the chance for a node to communicate with a nearby node is higher than communicating with a node which is far away from it.

5. Zone size: Different zone sizes are considered ranging from 1km×1km to 3km×3km. Upon studying other parameters, the simulated 2km×2km terrain is divided into 4 zones each of 1km×1km size. This zone size is chosen after testing the zone size effect on network performance (*Section 5.3.5*). Results of this testing show that moderate performance is obtained upon dividing the area into 4 or 9 zones.
6. Node failure percentage: Multiple failed node percentages are considered ranging from 0% to 40%. Upon studying the effect of other parameters, all nodes in the network are considered as well-functioning, i.e. a node failure percentage of 0% is used.
7. Malicious node percentage: Many malicious node percentages are considered ranging from 0% to 40%. Upon studying other parameters effect, all nodes in the network are considered as well-behaving, a malicious node percentage of 0% is used.

Table 5.2 summarizes the used variable simulation parameters.

Table 5.2: Variable simulation parameters

Parameter	**Value**
Node maximum mobility speed	0 m/s-10 m/s
Network size	1km×1km-3km×3km
Node density	40 nodes/km^2-100 nodes/km^2
Local communication percentage	0% -100% (for *ARANz*)
Zone size	1km×1km-3km×3km (for *ARANz*)
Node failure percentage	0%-40%
Malicious node percentage	0%-40%

5.1.4 Performance Metrics

To evaluate the performance of the proposed scheme, eight performance metrics are evaluated for each aforementioned parameter. These metrics are *packet delivery fraction*, *average path number of hops*, *packet network load*, *byte network load*, *packet* *routing load*, *byte routing load*, *average route acquisition latency* as well as *average end-to-end delay of data packets*. These metrics are defined as follows:

1. Packet Delivery Fraction (*PDF*): The fraction of the data packets generated by the *CBR* sources that are received by the intended destinations. This metric evaluates the ability of the protocol to discover and maintain routes (Sanzgiri et al. 2005). This metric also determines the completeness and correctness of the routing protocol (Buruhanudeen et al. 2007).
2. Average Path Number of Hops (*APNH*): The average number of hops of the paths discovered by the protocol. It is calculated by averaging the number of hops taken by each data packet to reach the destination. This metric evaluates the protocol's ability to discover shortest routes.
3. Packet Network Load (*PNL*): The overhead packets resulted from constructing and maintaining the network structure as well as updating nodes' positions and certificates. It is calculated in *ARANz* as the summation of all packets sent during the setup and maintenance phases. This is unlike in *ARAN*, where *PNL* is calculated as the summation of all packets sent to update nodes' certificates. The transmission at each hop along the paths is also counted upon calculating this metric. Related to *AODV*, it is a flat non-secure topology-based routing protocol. Hence, in *AODV* there is no need for network structure maintenance or nodes' positions and certificates update. Thus, *PNL* for *AODV* is excluded from the figures because it is considered as zero.
4. Byte Network Load (*BNL*): Similar to the above metric, but considering the resulted overhead bytes.
5. Packet Routing Load (*PRL*): The ratio of routing packets to delivered data packets. Routing packets in *AODV* and *ARAN* are defined as those sent during route instantiation and maintenance phases. In *ARANz*, all packets sent during the

location service phase are also included in calculating this metric. The transmission at each hop along the route is also counted in the calculation of this metric.

6. Byte Routing Load (*BRL*): Similar to the previous metric, however it is defined as the ratio of routing bytes to delivered data bytes. *ARAN* and *ARANz* suffer from larger control bytes due to certificates and signatures stored in packets.
7. Average Route Acquisition Latency (*ARAL*): The average delay incurred to discover a route to a destination. It is defined in *ARAN* and *AODV* as the average delay between sending a route request/discovery packet by a source and the receipt of the first corresponding route reply packet. In *ARANz*, it is defined as the average delay to both discover the position of the destination and to set up a route to it. If a request timed out and has to be retransmitted, the sending time of the first transmission is used for calculating the latency.
8. Average End-to-End Delay of data packets (*AEED*): The average delay between the sending of data packet by the *CBR* source and its receipt at the corresponding *CBR* receiver. This includes all delays caused during position inquiry, route establishment, buffering and processing at intermediate nodes and retransmission delays at the *MAC* layer.

5.2 Statistical Analysis

The significance of the obtained results is tested statistically for all the conducted experiments. The performed statistical testing is done with the help of the t-Test and the chi-square Test. The t-Test is used to verify the significance of the difference in a specific metric for two protocols. Whereas the chi-square Test is used to study the significance of the increase or decrease of performance metrics for each protocol separately. *Section 5.2.1* and *Section 5.2.2* introduce these tests.

5.2.1 The t-Test

The t-Test assesses whether the means of two groups are statistically different from each other (Malla, D. 2005). The formula for the t-Test is a ratio, the top part of the ratio is the difference between the means of the two groups ($\bar{X}_1$ and $\bar{X}_2$) and the bottom part is a measure of the variability of the groups (Trochim, W. 2001). The formula for the t-Test is defined as:

$$t = \frac{\text{Difference between the means of the two groups}}{\text{Variability of the two groups}}$$

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{Var(X_1)}{n_1} + \frac{Var(X_2)}{n_2}}}$$

The top part of the formula is simply the difference between the means. The bottom part is computed by taking the variance for each group (*Var*(X_1) and *Var*(X_2)) and dividing it by the number of samples in that group (n_1 and n_2). These two values are added and then their square root is taken. Note that the variance is the square of the standard deviation (Trochim W. 2001).

The value of the t-Test is positive if the first mean is larger than the second and negative if it is smaller. To test whether the t-Test value is small enough to say that the difference between the groups is not likely to have been a chance finding, a risk level, called the alpha level, need to be set. In most research, the alpha level is set at 0.05. This means that five times out of a hundred you would find a statistically significant difference between the means. Given the alpha level and the t-Test value, the t-Test is used to test the null hypothesis, which states that there is no significant difference between the means for the two groups. If the calculated t-Test value is greater than the alpha level, the null hypothesis of independence is accepted, i.e. the difference between the means for the two groups is statistically insignificant. Whereas, if the t-Test value is less than the alpha level, the hypothesis is rejected, and it can be concluded that the difference between the means for the two groups is statistically significant and that some factor other than chance causes that difference (Trochim, W. 2001; Kirkman 1996).

5.2.2 The chi-square Test

The chi-square Test (χ^2) is a statistical test that is commonly used to compare observed data with data we would expect to obtain according to a specific hypothesis (Fisher & Yates1974). The chi-square Test value is calculated by adding up the square of the differences between each observed value and the group mean ($\bar{X}$), divided by the group mean (McClean P. 2000; Fisher & Yates1974; Kirkman 1996). The chi-square Test formula is defined as:

$$\chi^2 = \sum \frac{\text{The square of the difference between observed value and group mean}}{\text{Group mean}}$$

$$\chi^2 = \sum \frac{(\text{Observed value} - \text{Group mean})^2}{\text{Group mean}}$$

$$\chi^2 = \sum \frac{(\text{Observed value} - \bar{X})^2}{\bar{X}}$$

The chi-square Test is used to check whether the differences between the observed values and the mean of the group are a result of chance, or due to other factors. In other words, the chi-square Test is used to test the null hypothesis, which states that there is no significant difference between observed values and the mean (Fisher & Yates1974). The probability value, p-value, determines how much deviation can occur to conclude that something other than chance is causing the observed values to differ from the mean. By statistical convention, 0.05 is used as the p-value (McClean P. 2000). If the calculated chi-square Test value is greater than the p-value, the null hypothesis of independence is accepted, i.e. the deviation is small enough that chance alone accounts for it. Whereas, if the calculated chi-square Test value is less than the p-value, the hypothesis is rejected, and we can conclude that the deviation is statistically significant and that some factor other than chance is operating for the deviation (Fisher & Yates1974).

5.3 Results and Performance Analysis

This section discusses the results of our experiments considering different scenarios. *Section 5.3.1* studies the effect of node mobility speed on different performance metrics. Then, *Section 5.3.2* evaluates the performance of the proposed protocol for different network sizes. Next, *Section 5.3.3* examines the effect of node density. The effect of local communication percentage is presented in *Section 5.3.4*. Afterward, *Section 5.3.5* examines the effect of zone size. The effect of node failure percentage is studied in *Section 5.3.6*. Finally, *Section 5.3.7* investigates the efficiency of our secure routing protocol in detecting malicious nodes.

For all figures in this chapter that represent performance parameters effect, each data point is an average of ten simulation runs with identical configuration, but different randomly generated numbers.

5.3.1 Node Mobility Speed Effect

To study the effect of the node mobility speed, a 2km×2km network is considered. This network contains 240 nodes (i.e. node density of 60 nodes/km^2) and it is divided into 4 equal-sized square-shape zones. Simulations are run with 0m/s, 3m/s, 6m/s and 10m/s speeds with a pause time fixed at 30s. Five *CBR* sessions are simulated in each run three of them are local and two are external.

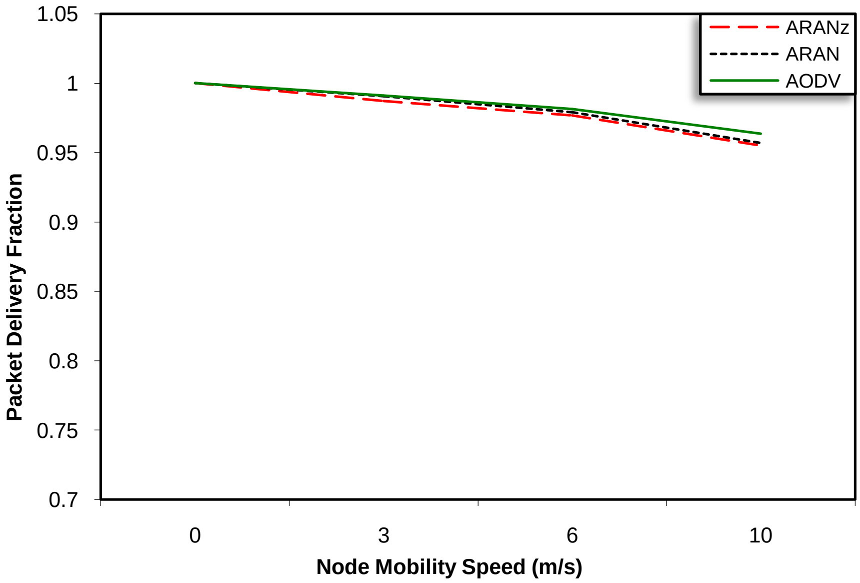


Figure 5.1: *PDF* vs. node mobility speed

Table 5.3: *PDF* vs. node mobility speed (t-Test)

	ARANz	***ARAN***
Mean	9.80E-01	9.82E-01
Variance	3.62E-04	3.45E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-1.45E-01	
P(T<=t) one-tail	4.45E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	8.89E-01	
t Critical two-tail	2.45E+00	

	ARANz	***AODV***
Mean	9.80E-01	9.84E-01
Variance	3.62E-04	2.44E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-3.49E-01	
P(T<=t) one-tail	3.69E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	7.39E-01	
t Critical two-tail	2.45E+00	

Table 5.4: *PDF* vs. node mobility speed (chi-square Test)

ARANz	***ARAN***	***AODV***
0.990516726	0.991187853	0.994737892

As shown in Figure 5.1, *PDF* for *ARANz* and *ARAN* are identical to that for *AODV* in the low node mobility, but they are slightly less when the mobility increases. This difference in *PDF* is due to higher packet processing and authentication delay at each node in the case of using *ARAN* and *ARANz* protocols. In other words, longer time means higher probability for losing the link connection due to nodes movement, which results in dropping some packets. However, the results of the t-Test (presented in Table 5.3) show that the differences in *PDF* for the three protocols are statistically insignificant (p one-tail > 0.05). It is also obvious from Figure 5.1 that *PDF* for the three protocols decreases slightly with increasing the node mobility. Higher node mobility means higher probability for losing the link connection and dropping some data packets. However, the results of the chi-square Test (presented in Table 5.4) show that chi-square Test values are greater than 0.05, i.e. the decrease in *PDF* for the three protocols is statistically insignificant. Moreover, *PDF* obtained using either protocol is above 95% in all scenarios. This indicates that the three protocols are highly effective in discovering and maintaining routes for delivery of data packets even with relatively high node mobility.

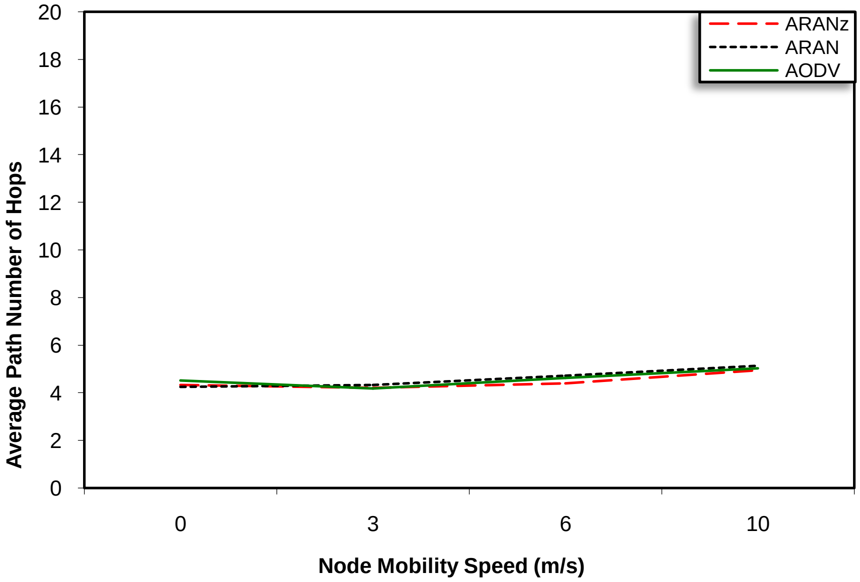


Figure 5.2: *APNH* vs. node mobility speed

Table 5.5: *APNH* vs. node mobility speed (t-Test)

	ARANz	***ARAN***
Mean	4.46E+00	4.60E+00
Variance	1.06E-01	1.71E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-5.34E-01	
P(T<=t) one-tail	3.06E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	6.13E-01	
t Critical two-tail	2.45E+00	

	ARANz	***AODV***
Mean	4.46E+00	4.59E+00
Variance	1.06E-01	1.21E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-5.15E-01	
P(T<=t) one-tail	3.12E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	6.25E-01	
t Critical two-tail	2.45E+00	

Table 5.6: *APNH* vs. node mobility speed (chi-square Test)

ARANz	*ARAN*	*AODV*
0.995052524	0.990465197	0.994247407

Even though *ARAN* and *ARANz* do not explicitly seek shortest paths, the first *RDP* packet to reach the destination usually travels along the shortest path (as shown in Figure 5.2). Moreover, the results of the t-Test presented in Table 5.5, show that the differences in *APNH* for the three protocols are statistically insignificant. Hence, it is obvious that *ARAN* and *ARANz* are as efficient as *AODV* in discovering shortest paths. *APNH* for the three protocols increases slightly with increasing the node mobility. Nodes mobility may result in separating the source and destination nodes from each other and so using longer paths. However, Table 5.6 shows that the increase in *APNH* for the three protocols is statistically insignificant.

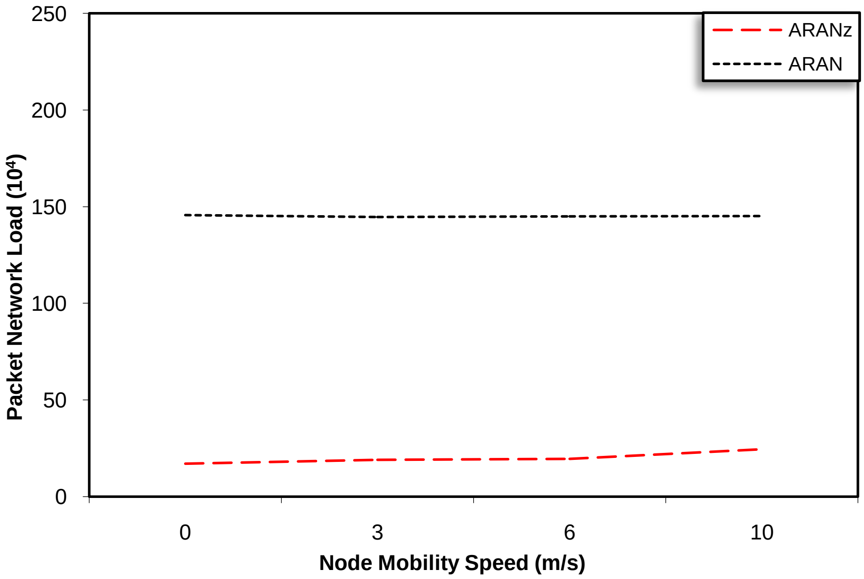


Figure 5.3: *PNL* vs. node mobility speed

Table 5.7: *PNL* vs. node mobility speed (t-Test)

	ARANz	***ARAN***
Mean	2.01E+01	1.45E+02
Variance	1.02E+01	2.11E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-7.77E+01	
P(T<=t) one-tail	2.35E-06	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	4.70E-06	
t Critical two-tail	3.18E+00	

Table 5.8: *PNL* vs. node mobility speed (chi-square Test)

ARANz	***ARAN***
0.677828191	0.999923656

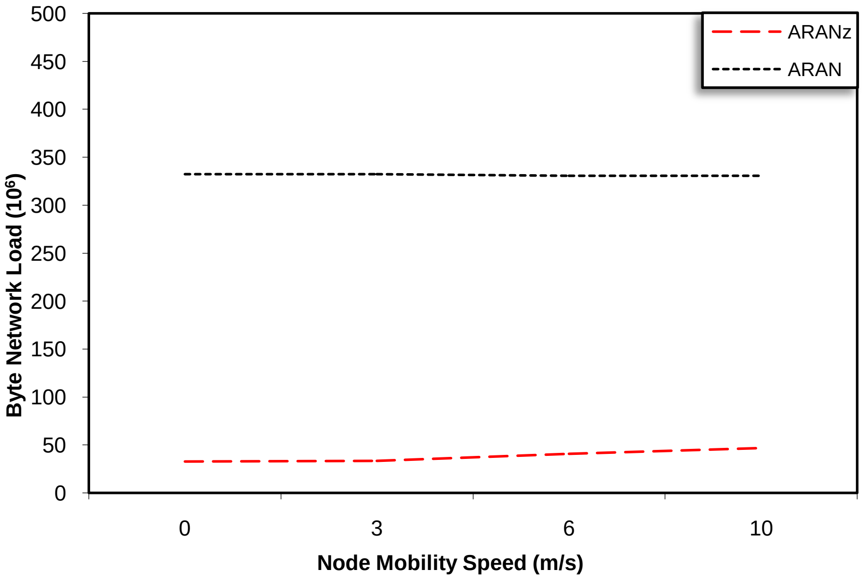


Figure 5.4: *BNL* vs. node mobility speed

Table 5.9: *BNL* vs. node mobility speed (t-Test)

	ARANz	***ARAN***
Mean	3.84E+01	3.32E+02
Variance	4.48E+01	8.01E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-8.68E+01	
P(T<=t) one-tail	1.68E-06	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.36E-06	

Table 5.10: *BNL* vs. node mobility speed (chi-square Test)

ARANz	***ARAN***
0.321546188	0.999836184

Figure 5.3 and Figure 5.4 show that *PNL* and *BNL* for *ARANz* are significantly lower than these for *ARAN*. The results of the t-Test (presented in Table 5.7 and Table 5.9) assist that the differences in *PNL* and *BNL* for the two protocols are statistically significant (p one-tail < 0.05). The main reason behind this gap is that nodes in *ARAN* are unaware of *CA*'s position, hence all certificate update request packets sent from nodes to *CA* are broadcast to the entire network. In *ARANz*, however, certificate update request packets as well as position update packets are sent from a node towards the nearest *LCA* using restricted directional flooding. After that these packets are forwarded from the nearest *LCA* to other *LCA*s in its zone using source routing. Even packets initiated upon updating *LCA*'s position or electing a new *LCA* are sent only to nodes in the intended zone and adjacent *LCA* in the immediate neighbouring zone. Moreover, packets sent in case of node movement to a neighbouring zone are sent using source routing or restricted directional flooding. Additionally, packets initiated to inform nodes failure and to achieve *LCA*s synchronization are sent only to *LCA*s in the network using *LCA* flooding. Finally, the only two packets that are broadcast to the entire network are *NETSET* packet during the network setup and *CNODE* packet to indicate a compromised node.

In *ARAN*, certificate update request packets are broadcast to the entire network regardless of node mobility speed. As such, *PNL* and *BNL* for *ARAN* are almost not affected by mobility speed. *PNL* and *BNL* for *ARANz* increase slightly as the node mobility increases. Frequent node mobility results in increasing the number of packets sent for updating nodes' positions as well as electing new *LCA*s. However, the results of the chi-square Test (presented in Table 5.8 and Table 5.10) show that the increase in *PNL* and *BNL* for *ARANz* is statistically insignificant.

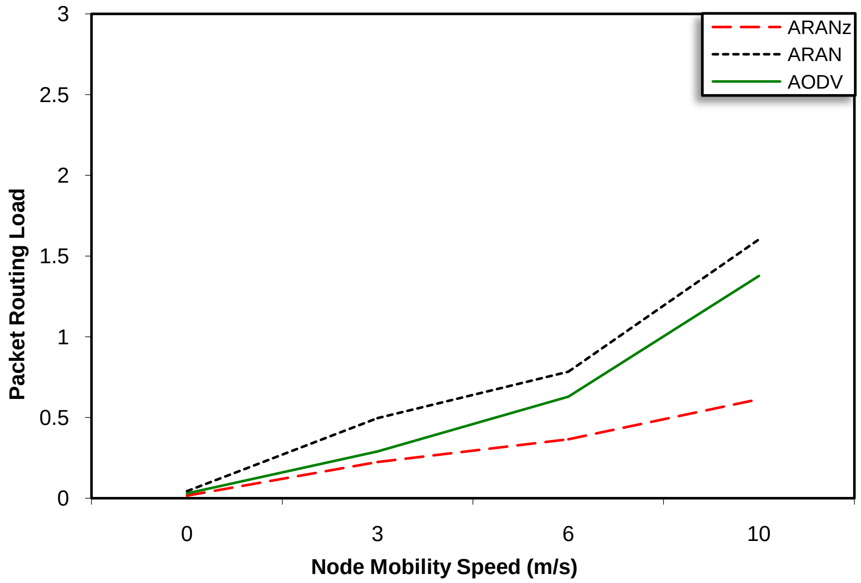


Figure 5.5: *PRL* vs. node mobility speed

Table 5.11: *PRL* vs. node mobility speed (t-Test)

	ARANz	***ARAN***
Mean	3.02E-01	7.09E-01
Variance	3.42E-02	2.26E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.60E+01	
t Stat	-2.88E+00	
P(T<=t) one-tail	5.40E-03	
t Critical one-tail	1.75E+00	
P(T<=t) two-tail	1.08E-02	
t Critical two-tail	2.12E+00	

	ARANz	***AODV***
Mean	3.02E-01	5.53E-01
Variance	3.42E-02	1.85E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.60E+01	
t Stat	-1.93E+00	
P(T<=t) one-tail	3.55E-02	
t Critical one-tail	1.75E+00	
P(T<=t) two-tail	7.10E-02	
t Critical two-tail	2.12E+00	

Table 5.12: *PRL* vs. node mobility speed (chi-square Test)

ARANz	***ARAN***	***AODV***
0.891129142	0.622431902	0.622852746

Figure 5.5 shows that *ARANz* has the minimum *PRL* and Table 5.11 confirms that the differences between *PRL* for the three protocols are statistically significant. *ARANz* does not broadcast the *RDP* packet to the whole area. In *ARANz*, the *RDP* is sent using restricted directional flooding towards the destination, this is the reason behind reducing the overall *PRL*. Even *PDP* packets are sent using restricted directional flooding or source routing. Hence, *PDP* packets should not significantly affect *PRL* especially if the source and the destination nodes are in the same zone.

It is obvious also from Figure 5.5 that *ARAN* has higher *PRL* than *AODV*, although both protocols propagate *RDP* packets to the entire area. This is due to two reasons. The first, *ARAN* has higher packet processing and authentication delay at each node, which in turn, increases the chance of a link break due to nodes movement. This break causes the source to reinitiate a new *RDP* packet, which increases the overall *PRL* for *ARAN*. Secondly, in case that an intermediate node in *AODV* has a valid route towards the destination, it can respond with a *RREP* packet to the source. Hence, there is no need to rebroadcast the *RREQ* to its neighbours, which in turn reduces the overall *AODV*'s *PRL*.

To study the effect of node mobility speed on *PRL*, let us refer again to Figure 5.5. It is clear that *PRL* for the three protocols increases with increasing node mobility. This is because increasing mobility increases the chance for losing the link connection and reinitiating *RDP* packets which increases the overall *PRL* for the three protocols. However, the results of the chi-square Test (presented in Table 5.12) show that the increase in *PRL* for the three protocols is statistically insignificant.

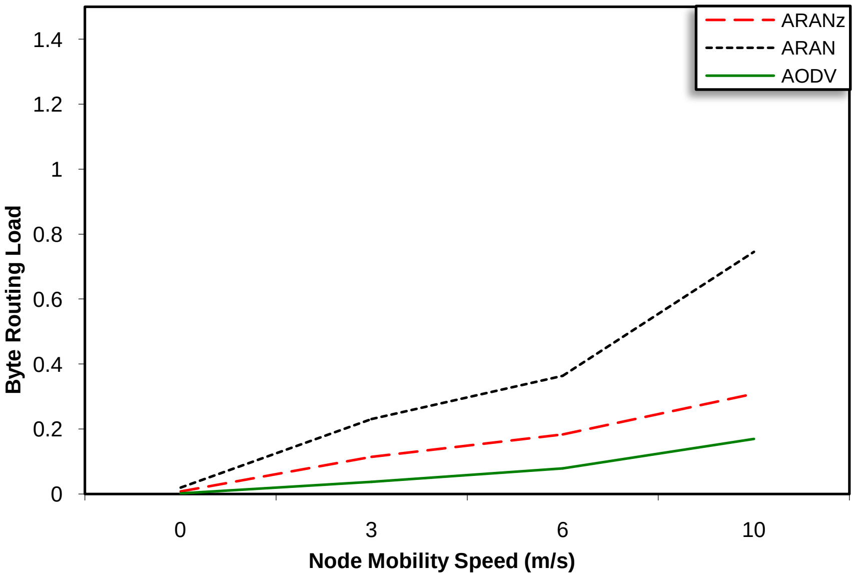


Figure 5.6: *BRL* vs. node mobility speed

Table 5.13: *BRL* vs. node mobility speed (t-Test)

	ARANz	***ARAN***
Mean	1.52E-01	3.30E-01
Variance	8.63E-03	4.89E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.60E+01	
t Stat	-2.67E+00	
P(T<=t) one-tail	8.31E-03	
t Critical one-tail	1.75E+00	
P(T<=t) two-tail	1.66E-02	
t Critical two-tail	2.12E+00	

	ARANz	***AODV***
Mean	1.52E-01	6.87E-02
Variance	8.63E-03	2.82E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.90E+01	
t Stat	2.82E+00	
P(T<=t) one-tail	5.52E-03	
t Critical one-tail	1.73E+00	
P(T<=t) two-tail	1.10E-02	
t Critical two-tail	2.09E+00	

Table 5.14: *BRL* vs. node mobility speed (chi-square Test)

ARANz	***ARAN***	***AODV***
0.957685587	0.844191629	0.974689698

Although *ARANz* has smaller *PRL* than *AODV*, Figure 5.6 shows that it has higher *BRL* due to the larger *ARANz* control packets that contain security data which results in larger control bytes. The figure also shows that as with *PRL*, increasing node mobility will increase *BRL* for the three protocols due to loosing links and reinitiating *RDP* packets. Results of the t-Test (refer to Table 5.13) show that the differences between *BRL* for the three protocols are statistically significant. On the other hand, the results of the chi-square Test (presented in Table 5.14) show that the increase in *BRL* for each protocol is statistically insignificant. This indicates that *BRL* for the three protocols is roughly not affected by increasing the node mobility speed.

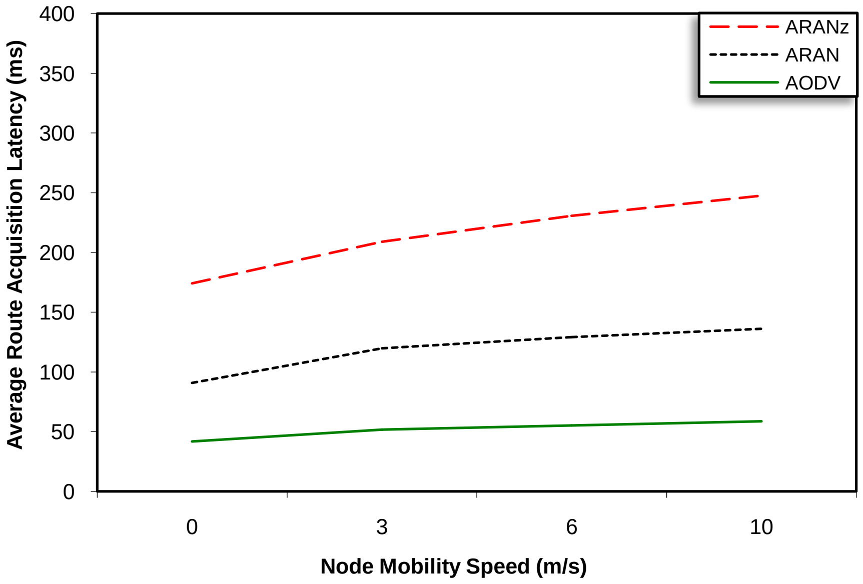


Figure 5.7: *ARAL* vs. node mobility speed

Table 5.15: *ARAL* vs. node mobility speed (t-Test)

	ARANz	***ARAN***
Mean	2.15E+02	1.19E+02
Variance	1.00E+03	3.94E+02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	5.16E+00	
P(T<=t) one-tail	1.79E-03	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	3.57E-03	
t Critical two-tail	2.57E+00	

	ARANz	***AODV***
Mean	2.15E+02	5.18E+01
Variance	1.00E+03	5.41E+01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	1.01E+01	
P(T<=t) one-tail	1.04E-03	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.08E-03	
t Critical two-tail	3.18E+00	

Table 5.16: *ARAL* vs. node mobility speed (chi-square Test)

ARANz	***ARAN***	***AODV***
0.002996019	0.019117188	0.0471507908

Referring to Figure 5.7 it is clear that *ARAL* for *ARAN* and *ARANz* protocols is higher than that for *AODV*. While processing *ARAN* and *ARANz* routing control packets, each node has to verify the digital signature of the previous node and replace this signature with its own digital signature, in addition to the normal packet processing as done by *AODV*. Signature generation and verification cause additional delay at each hop, and so *ARAL* increases. Also, the figure shows that *ARAL* for *ARANz* is higher than that for

ARAN due to time required to get the destination node's position. Results of the t-Test (presented in Table 5.15) show that the differences between *ARAL* for the three protocols are statistically significant. Figure 5.7 and Table 5.16 show that *ARAL* for the three protocols increases as the node mobility increases. Mobility may cause the nodes to move far away from each other which produce longer paths, and so higher *ARAL*.

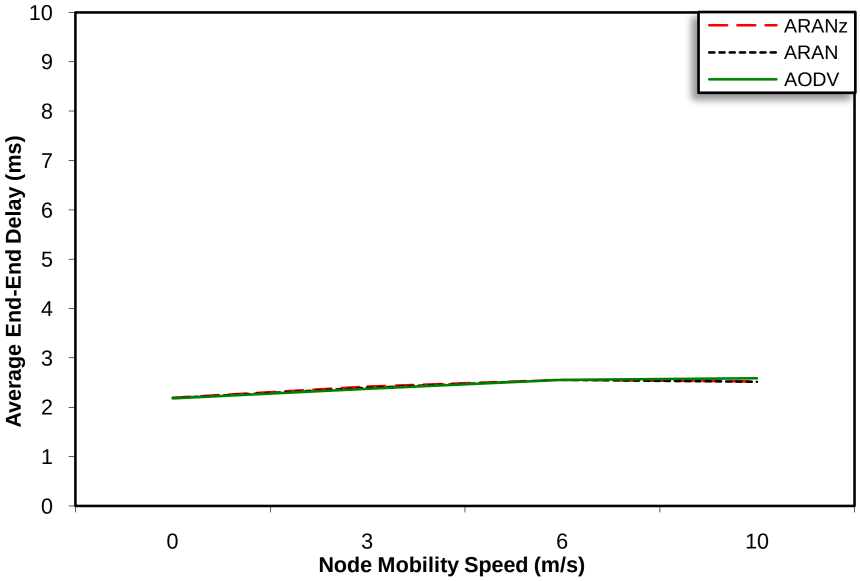


Figure 5.8: *AEED* vs. node mobility speed

Table 5.17: *AEED* vs. node mobility speed (t-Test)

	ARANz	***ARAN***
Mean	2.42E+00	2.41E+00
Variance	2.69E-02	2.76E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	9.52E-02	
P(T<=t) one-tail	4.64E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	9.27E-01	
t Critical two-tail	2.45E+00	

	ARANz	***AODV***
Mean	2.42E+00	2.42E+00
Variance	2.69E-02	3.60E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-9.80E-03	
P(T<=t) one-tail	4.96E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	9.92E-01	
t Critical two-tail	2.45E+00	

Table 5.18: *AEED* vs. node mobility speed (chi-square Test)

ARANz	***ARAN***	***AODV***
0.99839369	0.998324118	0.997523097

The *AEED* is almost identical in all three protocols (Figure 5.8). The processing of data packets at each hop is identical when using either protocol since none of them encrypts the data. So the three protocols have nearly the same average latency for data packets. Moreover, although *ARAN* and *ARANz* has higher *ARAL*, the number of position enquiries and route discoveries performed is a small fraction of the number of data packets delivered, as Figure 5.5 shows. Hence, the effect of *ARAL* on *AEED* of data packets is not significant. Table 5.17 confirms that the differences in *AEED* between the three protocols are statistically insignificant. Moreover, the results of the chi-square Test (presented in Table 5.18) assure that the *AEED* for the three protocols is roughly not affected by increasing node mobility speed.

5.3.2 Network Size Effect

To study the effect of network size, three networks of 1km×1km, 2km×2km and 3km×3km network sizes are tested. These networks are divided into multiple zones each of 1km×1km. Simulations are run with 60 nodes/km^2. These nodes move at a maximum speed of 5m/s and a pause time of 30s. Five *CBR* sessions are simulated in each run, three of them are chosen to be local and the other two are external.

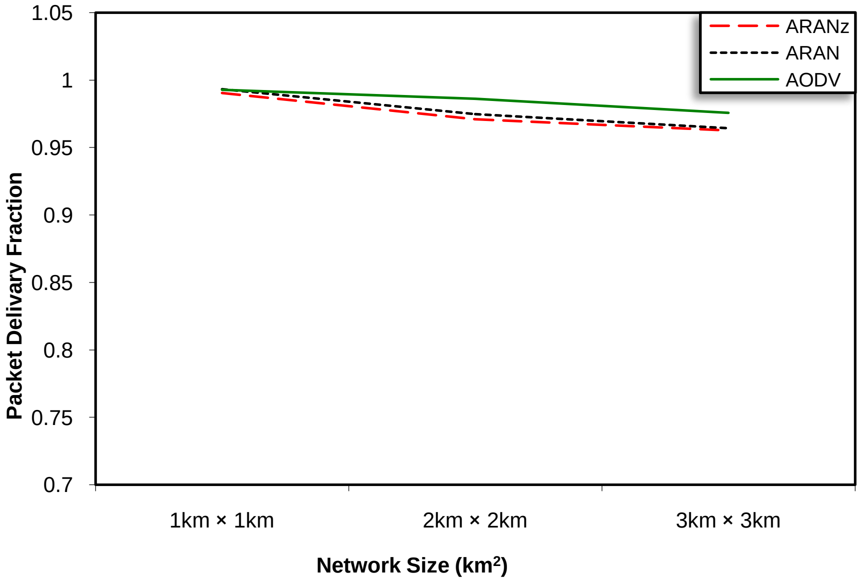


Figure 5.9: *PDF* vs. network size

Table 5.19: *PDF* vs. network size (t-Test)

	ARANz	***ARAN***
Mean	9.75E-01	9.77E-01
Variance	2.03E-04	2.15E-04
Observations	3.00E+00	3.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	-2.33E-01	
P(T<=t) one-tail	4.14E-01	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	8.27E-01	
t Critical two-tail	2.78E+00	

	ARANz	***AODV***
Mean	9.75E-01	9.85E-01
Variance	2.03E-04	7.39E-05
Observations	3.00E+00	3.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.07E+00	
P(T<=t) one-tail	1.82E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.64E-01	
t Critical two-tail	3.18E+00	

Table 5.20: *PDF* vs. network size (chi-square Test)

ARANz	***ARAN***	***AODV***
0.979419523	0.978260815	0.992519721

Referring to Figure 5.9 it is obvious that *PDF* for *ARAN* and *ARANz* is slightly less than *AODV*, especially in large area network. This is due to higher packet processing and authentication delay at each node in the case of using *ARAN* and *ARANz* protocols. This delay increases the probability for losing the link connection due to nodes movement, which results in dropping some packets. Also, *PDF* for the three protocols decreases as the network size increases due to the longer paths that the *RDP* packet passes through which increases the probability of link break and dropping some data packets. However, the results of the t-Test (presented in Table 5.19) and the chi-square Test (presented in Table 5.20) show that the differences in *PDF* between the three protocols and for each protocol separately are statistically insignificant. Moreover, *PDF* obtained using either protocol is above 95% for all simulated network sizes. This indicates that the three protocols are highly effective in discovering and maintaining routes for delivery of data packets even with relatively large network sizes.

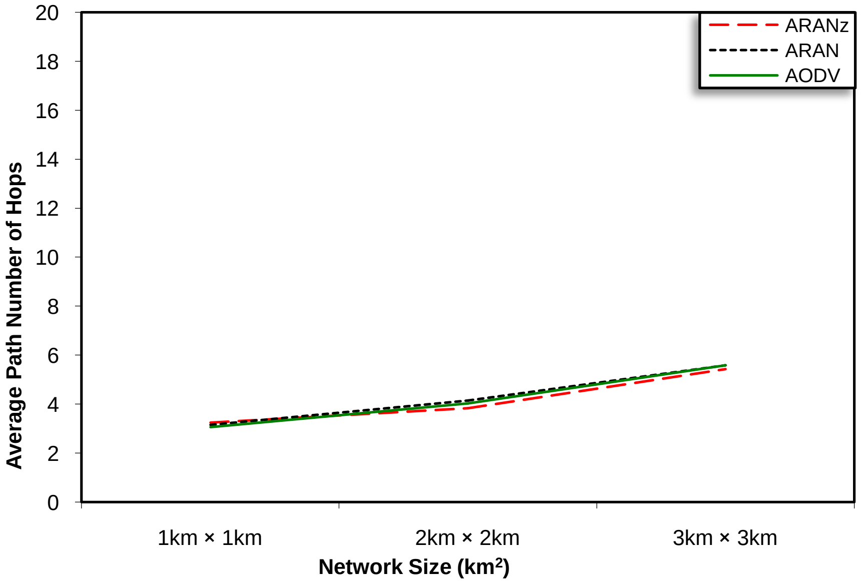


Figure 5.10: *APNH* vs. network size

Table 5.21: *APNH* vs. network size (t-Test)

	ARANz	*ARAN*
Mean	4.17E+00	4.30E+00
Variance	1.27E+00	1.50E+00
Observations	3.00E+00	3.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	-1.34E-01	
P(T<=t) one-tail	4.50E-01	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	9.00E-01	

	ARANz	*AODV*
Mean	4.17E+00	4.23E+00
Variance	1.27E+00	1.62E+00
Observations	3.00E+00	3.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	-6.12E-02	
P(T<=t) one-tail	4.77E-01	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	9.54E-01	

Table 5.22: *APNH* vs. network size (chi-square Test)

ARANz	*ARAN*	*AODV*
0.736778391	0.704781544	0.682009842

Figure 5.10 shows that *APNH* for the three protocols is almost identical for a specified network size. This indicates that *ARAN* and *ARANz* are as efficient as *AODV* in discovering the shortest paths and Table 5.21 assures that. It is clear also that the *APNH* increases with increasing the network size, due to the longer paths the packet passes through if the source and destination are apart from each other, which means larger number of hops. However, Table 5.22 shows that the increase in *APNH* for the three protocols is statistically insignificant.

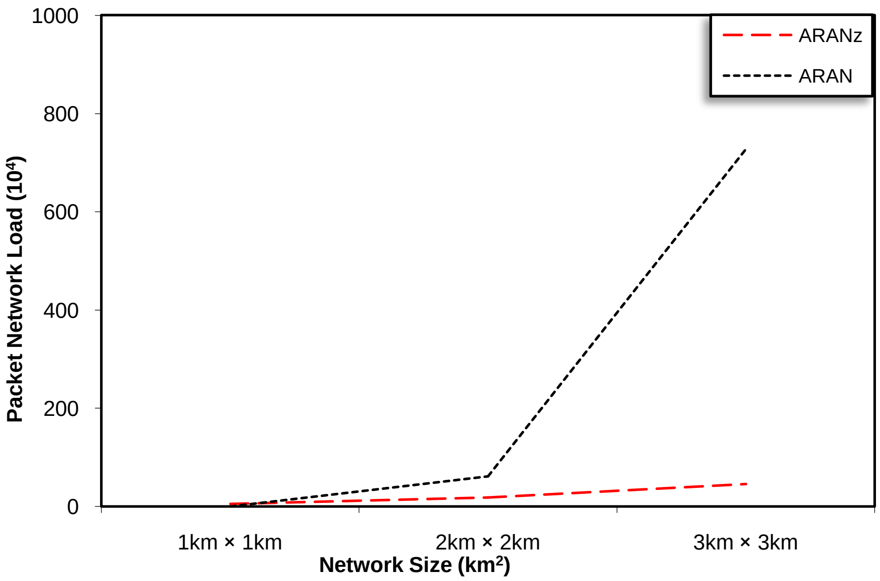


Figure 5.11: *PNL* vs. network size

Table 5.23: *PNL* vs. network size (t-Test)

	ARANz	***ARAN***
Mean	2.30E+01	2.63E+02
Variance	4.29E+02	1.63E+05
Observations	3.00E+00	3.00E+00
Hypothesized Mean Difference	0.00E+00	
df	2.00E+00	
t Stat	-1.03E+00	
P(T<=t) one-tail	2.06E-02	
t Critical one-tail	2.92E+00	
P(T<=t) two-tail	4.11E-01	
t Critical two-tail	4.30E+00	

Table 5.24: *PNL* vs. network size (chi-square Test)

ARANz	***ARAN***
8.103E-09	2E-269

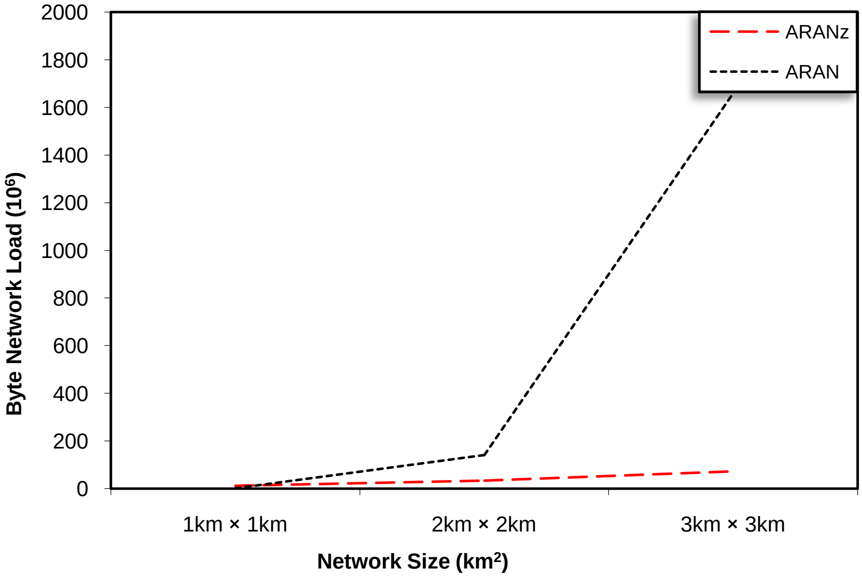


Figure 5.12: *BNL* vs. network size

Table 5.25: *BNL* vs. network size (t-Test)

	ARANz	***ARAN***
Mean	3.88E+01	6.00E+02
Variance	9.56E+02	8.47E+05
Observations	3.00E+00	3.00E+00
Hypothesized Mean Difference	0.00E+00	
df	2.00E+00	
t Stat	-1.06E+00	
P(T<=t) one-tail	2.01E-02	
t Critical one-tail	2.92E+00	
P(T<=t) two-tail	4.02E-01	
t Critical two-tail	4.30E+00	

Table 5.26: *BNL* vs. network size (chi-square Test)

ARANz	***ARAN***
2.01E-11	5.42521E-62

As shown in Figure 5.11 and Figure 5.12, the *PNL* and *BNL* for *ARAN* and *ARANz* are almost identical when a network of 1km×1km size is used. *ARANz* deals with this network size as one zone. Hence, some packets, such as those sent upon updating *LCA*s' positions or electing a new *LCA* are broadcast to the entire network. In large area networks, however, *PNL* and *BNL* for *ARAN* become much higher than these for *ARANz* reaching to more than ten times in 3km×3km area network. This large gap results from

ARAN broadcasting certificate update request packets to the entire network. Moreover, *PNL* and *BNL* for both protocols increase with increasing the network size. Larger network size results in increasing the number of packets sent for updating nodes' positions and certificates due to increasing the number of nodes existing in the network. Results of the t-Test and the chi-square Test (presented in Table 5.23 through Table 5.26) show that the differences in *PNL* and *BNL* for the two protocols are statistically significant.

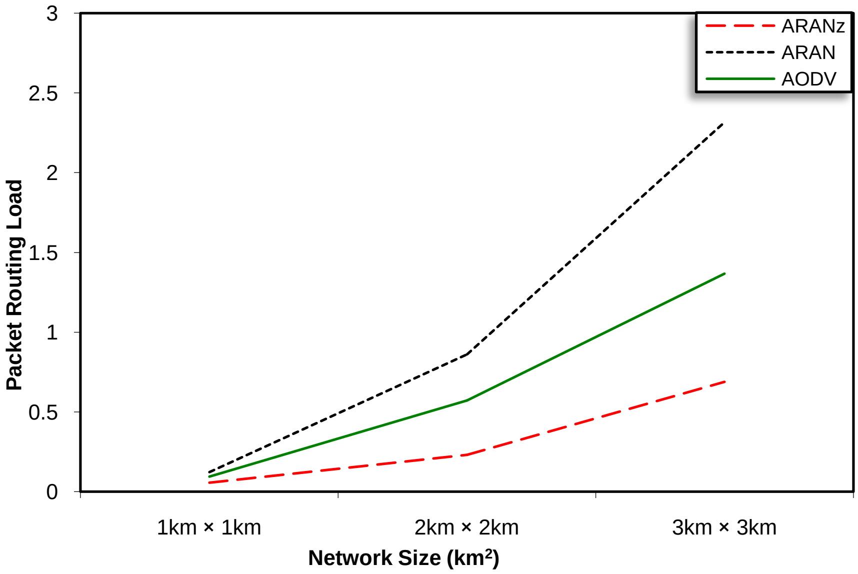


Figure 5.13: *PRL* vs. network size

Table 5.27: *PRL* vs. network size (t-Test)

	ARANz	*ARAN*
Mean	3.09E-01	1.06E+00
Variance	4.93E-02	5.80E-01
Observations	3.00E+00	9.00E+00
Hypothesized Mean Difference	0.00E+00	
df	9.00E+00	
t Stat	-2.84E+00	
P(T<=t) one-tail	9.76E-03	
t Critical one-tail	1.83E+00	
P(T<=t) two-tail	1.95E-02	

	ARANz	*AODV*
Mean	3.09E-01	6.59E-01
Variance	4.93E-02	1.92E-01
Observations	3.00E+00	9.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-2.13E+00	
P(T<=t) one-tail	2.70E-02	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	5.41E-02	

Table 5.28: *PRL* vs. network size (chi-square Test)

ARANz	*ARAN*	*AODV*
0.720338674	0.321636671	0.543654369

Figure 5.13 demonstrates that the *PRL* for the three protocols increases as the network size increases due to the higher probability of links breakage that requires reinitiating a *RDP* packet. However, the results of the chi-square Test (presented in Table 5.28) show that the increase in *PRL* for the three protocols is statistically insignificant.

ARANz still has the minimum *PRL* as a result of using restricted directional flooding in sending *RDP* packets. Additionally, it is clear that *ARAN* has higher *PRL* than *AODV* since *ARAN* has higher packet processing and authentication delay, which increases the chance of having a link break and reinitiating a new *RDP*. Table 5.27 confirms that the differences between *PRL* for the three protocols are statistically significant.

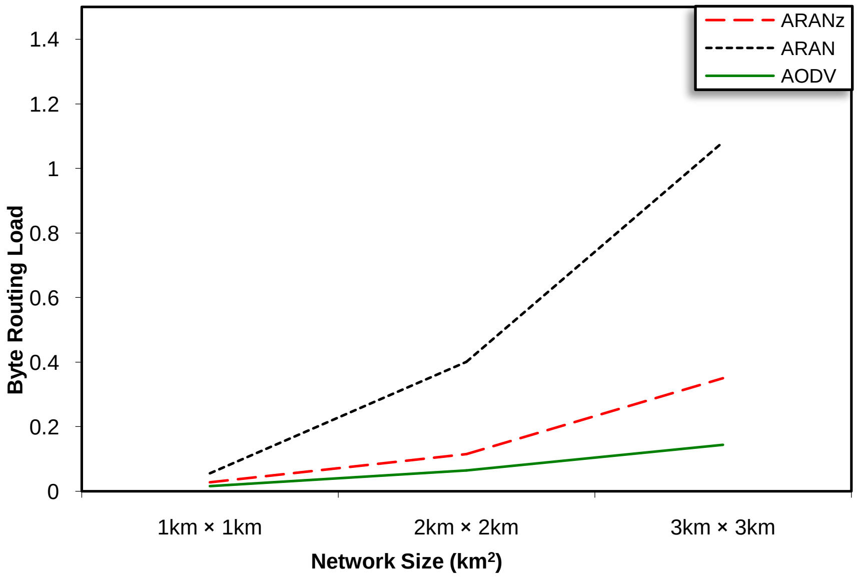


Figure 5.14: *BRL* vs. network size

Table 5.29: *BRL* vs. network size (t-Test)

	ARANz	*ARAN*
Mean	1.56E-01	4.94E-01
Variance	1.28E-02	1.27E-01
Observations	3.00E+00	9.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.00E+01	
t Stat	-2.71E+00	
P(T<=t) one-tail	1.10E-02	
t Critical one-tail	1.81E+00	
P(T<=t) two-tail	2.19E-02	

	ARANz	*AODV*
Mean	1.56E-01	7.33E-02
Variance	1.28E-02	1.95E-03
Observations	3.00E+00	9.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.00E+01	
t Stat	2.05E+00	
P(T<=t) one-tail	3.37E-02	
t Critical one-tail	1.81E+00	
P(T<=t) two-tail	6.73E-02	

Table 5.30: *BRL* vs. network size (chi-square Test)

ARANz	*ARAN*	*AODV*
0.844601757	0.587712101	0.945819155

Figure 5.14 shows that even though *ARANz* has smaller *PRL* than *AODV*, it has higher *BRL* due to the larger *ARANz*'s control packets that contain security data. As in the *PRL*, the results of the t-Test and the chi-square Test (presented in Table 5.29 and Table 5.30) show that the differences between *BRL* for the three protocols are statistically significant but the increase in *BRL* for each protocol is statistically insignificant.

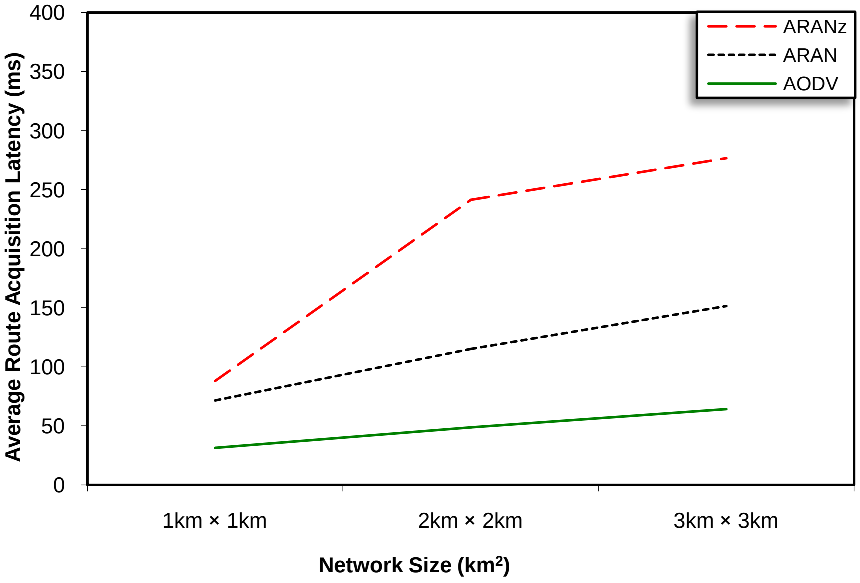


Figure 5.15: *ARAL* vs. network size

Table 5.31: *ARAL* vs. network size (t-Test)

	ARANz	*ARAN*
Mean	2.02E+02	1.13E+02
Variance	1.00E+04	1.60E+03
Observations	3.00E+00	3.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	1.43E+00	
P(T<=t) one-tail	1.24E-02	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.47E-01	
t Critical two-tail	3.18E+00	

	ARANz	*AODV*
Mean	2.02E+02	4.81E+01
Variance	1.00E+04	2.67E+02
Observations	3.00E+00	3.00E+00
Hypothesized Mean Difference	0.00E+00	
df	2.00E+00	
t Stat	2.63E+00	
P(T<=t) one-tail	5.98E-03	
t Critical one-tail	2.92E+00	
P(T<=t) two-tail	1.20E-01	
t Critical two-tail	4.30E+00	

Table 5.32: *ARAL* vs. network size (chi-square Test)

ARANz	***ARAN***	***AODV***
2.538E-22	7E-07	0.003833869

Figure 5.15 shows that *ARAL* for the three protocols increases as the network size increases due to increasing the number of nodes (hops) that the control packets pass through. Results of the chi-square Test (presented in Table 5.32) show that the increase in *ARAL* for the three protocols is statistically significant. As the figure shows, *ARANz* has the highest *ARAL* due to the time required for position enquiry processes. Moreover, *ARAL* for *ARAN* is higher than that for *AODV* due to higher *ARAN* packet processing and authentication delay. Results of the t-Test (presented in Table 5.31) show that the differences between *ARAL* for the three protocols are statistically significant. Finally, the difference in *ARAL* between *ARANz* and the other two protocols is small when a network size of 1km×1km is simulated because *ARANz* deals with this network as one zone hence destination's position is absolutely obtained from the nearest *LCA* to the source without the need to communicate other *LCA*s in the network.

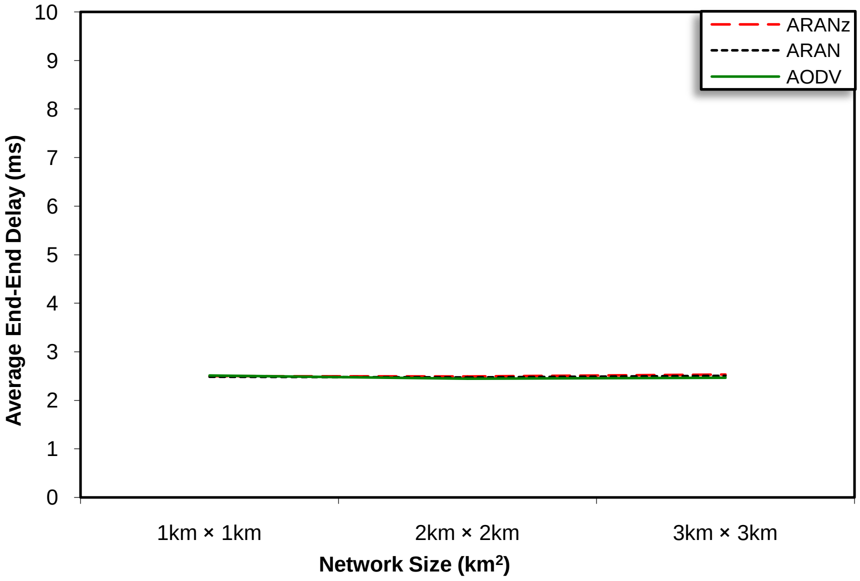


Figure 5.16: *AEED* vs. network size

Table 5.33: *AEED* vs. network size (t-Test)

	ARANz	***ARAN***
Mean	2.50E+00	2.49E+00
Variance	3.76E-04	2.89E-04
Observations	3.00E+00	3.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	1.26E+00	
P(T<=t) one-tail	1.39E-01	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	2.78E-01	
t Critical two-tail	2.78E+00	

	ARANz	***AODV***
Mean	2.50E+00	2.47E+00
Variance	3.76E-04	1.26E-03
Observations	3.00E+00	3.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	1.35E+00	
P(T<=t) one-tail	1.35E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.69E-01	
t Critical two-tail	3.18E+00	

Table 5.34: *AEED* vs. network size (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999850073	0.999883692	0.999492006

The three protocols produce almost identical *AEED* values (as shown in Figure 5.16). Although *ARAN* and *ARANz* have higher *ARAL*, the number of route discoveries and position enquiries performed is a small fraction of the number of data packets delivered. Hence, the effect of *ARAL* on *AEED* of the data packets is not significant (as confirmed in Table 5.33). Moreover, the results of the chi-square Test (presented in Table 5.34) assure that the *AEED* for the three protocols is not affected by increasing the simulated network size.

5.3.3 Node Density Effect

To test the effect of node density, a 2km×2km network that is divided into 4 zones is considered. Nodes inside this network move at a maximum speed of 5m/s. Five *CBR* sessions are simulated in each run, three of them are local and two are external. Simulations are run with 40 nodes/km^2, 60 nodes/km^2, 80 nodes/km^2 and 100 nodes/km^2.

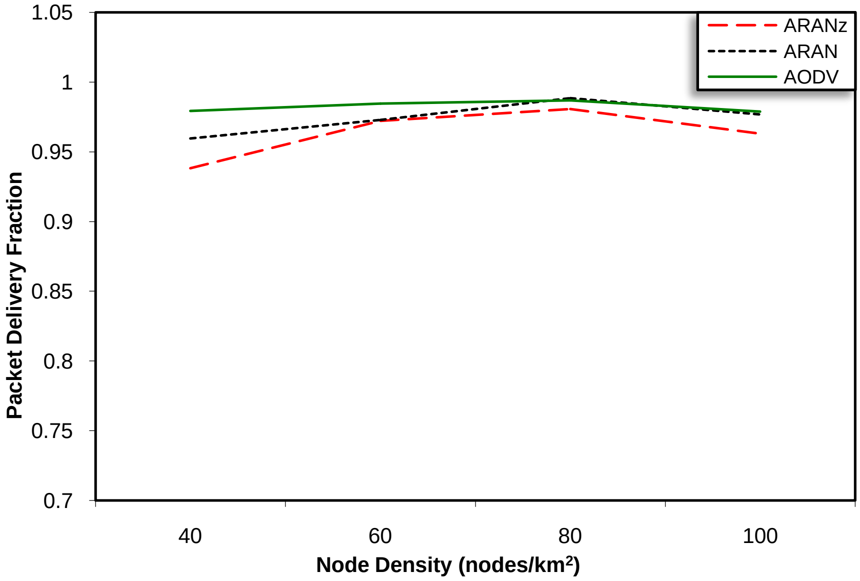


Figure 5.17: *PDF* vs. node density

Table 5.35: *PDF* vs. node density (t-Test)

	ARANz	*ARAN*
Mean	9.63E-01	9.74E-01
Variance	3.36E-04	1.42E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	-9.92E-01	
P(T<=t) one-tail	1.83E-01	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	3.67E-01	
t Critical two-tail	2.57E+00	

	ARANz	*AODV*
Mean	9.63E-01	9.82E-01
Variance	3.36E-04	1.58E-05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-2.01E+00	
P(T<=t) one-tail	6.88E-02	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.38E-01	
t Critical two-tail	3.18E+00	

Table 5.36: *PDF* vs. node density (chi-square Test)

ARANz	*ARAN*	*AODV*
0.991280477	0.99760336	0.999911054

As Figure 5.17 shows, higher *PDF* for all protocols is obtained for node density values between 60 nodes/km^2 and 80 nodes/km^2. As density decreases below 60 nodes/km^2, the probability of finding a path between the source and destination decreases. On the other hand, as density increases above 80 nodes/km^2, the number of nodes participating in rebroadcasting the control packets increases. In other words, an intermediate node receives multiple copies of the same *RDP* packet from its neighbours. Processing these control packets may cause delay in processing data packets as well as causing some packet drops. However, the results of the t-Test (presented in Table 5.35) and the chi-square Test (presented in Table 5.36) show that the differences in *PDF* between the three protocols and for each protocol separately are statistically insignificant. Also, Figure 5.17 shows that the *PDF* for all protocols is above 93% for all simulated node density values. This suggests that the three protocols are highly effective in discovering and maintaining routes for delivery of data packets regardless of node density.

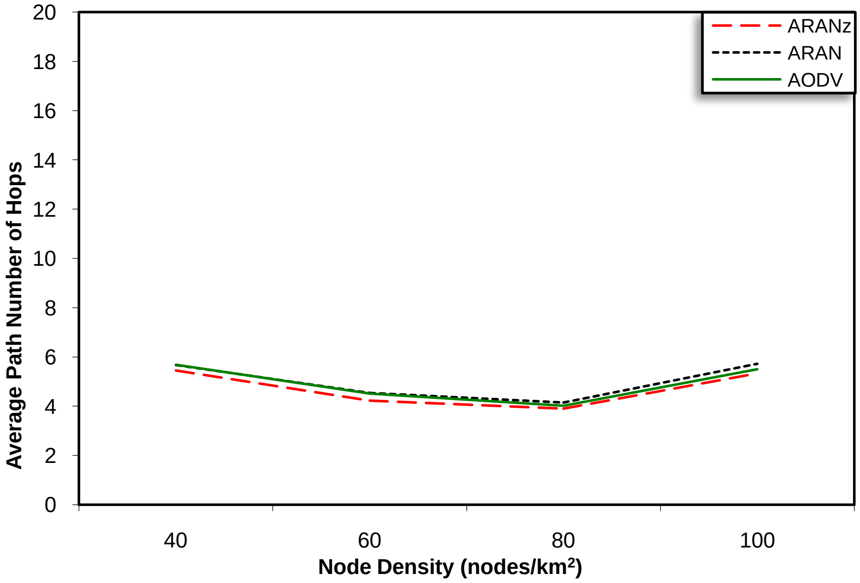


Figure 5.18: *APNH* vs. node density

Table 5.37: *APNH* vs. node density (t-Test)

	ARANz	*ARAN*
Mean	4.83E+00	5.02E+00
Variance	5.93E-01	6.32E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-3.38E-01	
P(T<=t) one-tail	3.73E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	7.47E-01	

	ARANz	*AODV*
Mean	4.83E+00	4.92E+00
Variance	5.93E-01	6.34E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-1.67E-01	
P(T<=t) one-tail	4.36E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	8.73E-01	

Table 5.38: *APNH* vs. node density (chi-square Test)

ARANz	*ARAN*	*AODV*
0.94675694	0.944773539	0.943017741

It is clear from Figure 5.18 that the *APNH* decreases with increasing the node density, until reaching its minimum values at node densities ranging from 60 nodes/km^2 to 80 nodes/km^2. This suggests that increasing the node density increases the chance to find faster/shorter path until reaching 80 nodes/km^2. As density increases above 80 nodes/km^2 *APNH* starts to increase. This indicates that increasing the density more than 80 nodes/km^2 will only make the nodes closer to each other while not serving in finding shorter paths. In fact, increasing the number of control packets received from the neighbours may result in dropping some control packets that may have already passed through the shortest path.

Results of the t-Test (presented in Table 5.37) and the chi-square Test (presented in Table 5.38) show that the differences in *APNH* among the three protocols and for each protocol are statistically insignificant. This is an indication that the three protocols are efficient in discovering the shortest paths regardless of node density.

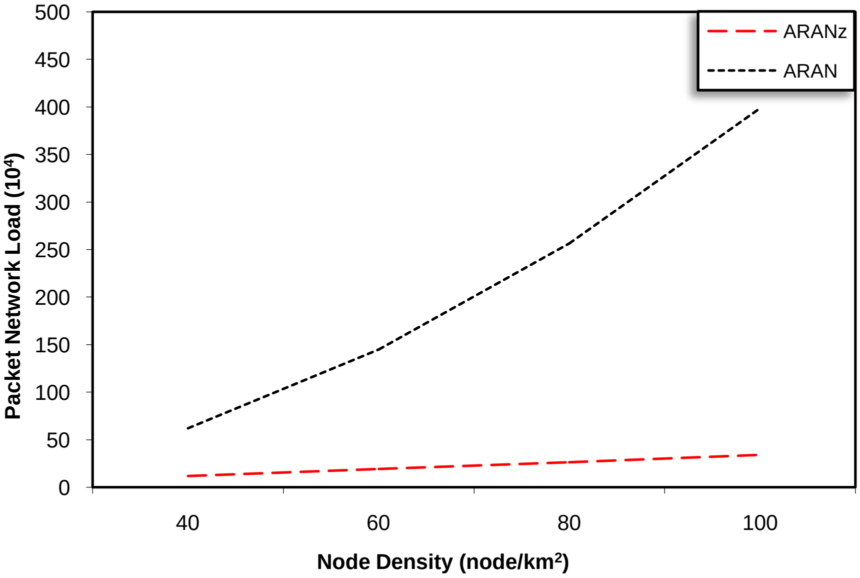


Figure 5.19: *PNL* vs. node density

Table 5.39: *PNL* vs. node density (t-Test)

	ARANz	*ARAN*
Mean	2.29E+01	2.16E+02
Variance	8.98E+01	2.13E+04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-2.64E+00	
P(T<=t) one-tail	3.89E-02	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	7.79E-02	
t Critical two-tail	3.18E+00	

Table 5.40: *PNL* vs. node density (chi-square Test)

ARANz	*ARAN*
0.00815608	7.22449E-64

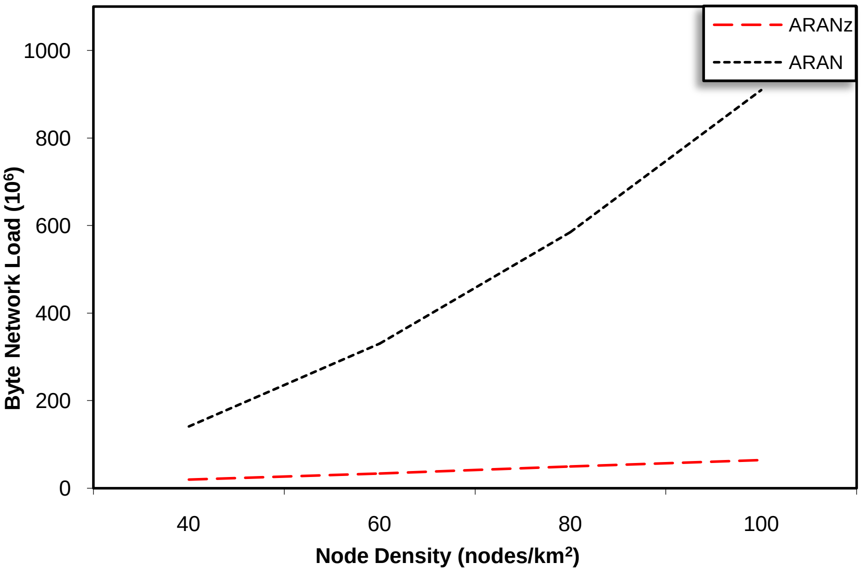


Figure 5.20: *BNL* vs. node density

Table 5.41: *BNL* vs. node density (t-Test)

	ARANz	*ARAN*
Mean	4.20E+01	4.91E+02
Variance	3.84E+02	1.11E+05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-2.70E+00	
P(T<=t) one-tail	3.69E-02	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	7.39E-02	

Table 5.42: *BNL* vs. node density (chi-square Test)

ARANz	***ARAN***
4.909E-06	5.731E-146

Figure 5.19 and Figure 5.20 show that, as in other scenarios, the *PNL* and *BNL* for *ARAN* are significantly higher than *ARANz*. Results of the t-Test (presented in Table 5.39 and Table 5.41) assure that the differences in these metrics between the two protocols are statistically significant. Moreover, these figures show that *PNL* and *BNL* for both *ARAN* and *ARANz* increase as the node density increases due to increasing the number of nodes updating their certificates and positions. Results of the chi-square Test (presented in Table 5.40 and Table 5.42) show that the increase in *PNL* and *BNL* for both protocols is statistically significant. However, these tables show that the increase in these metrics is more significant upon simulating *ARAN* protocol. This large difference results from *ARAN* broadcasting certificate update request packets to the entire network. On the other hand, *ARANz* sends packets related to updating nodes' positions and certificates only to the nearest *LCA*.

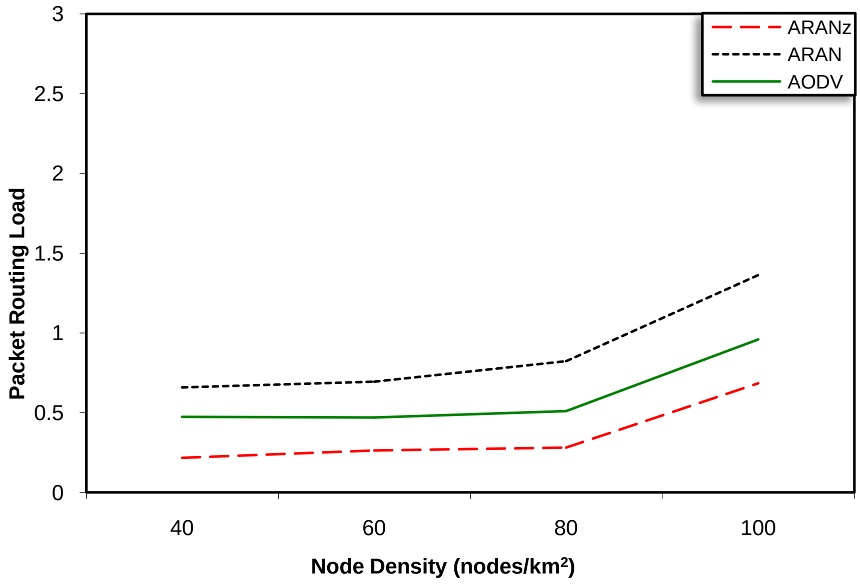


Figure 5.21: *PRL* vs. node density

Table 5.43: *PRL* vs. node density (t-Test)

	ARANz	*ARAN*
Mean	3.41E-01	8.55E-01
Variance	2.24E-02	5.46E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	2.00E+01	
t Stat	-6.68E+00	
P(T<=t) one-tail	8.40E-07	
t Critical one-tail	1.72E+00	
P(T<=t) two-tail	1.68E-06	
t Critical two-tail	2.09E+00	

	ARANz	*AODV*
Mean	3.41E-01	5.76E-01
Variance	2.24E-02	2.75E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	2.40E+01	
t Stat	-3.81E+00	
P(T<=t) one-tail	4.27E-04	
t Critical one-tail	1.71E+00	
P(T<=t) two-tail	8.54E-04	
t Critical two-tail	2.06E+00	

Table 5.44: *PRL* vs. node density (chi-square Test)

ARANz	*ARAN*	*AODV*
0.941877946	0.948010878	0.963224417

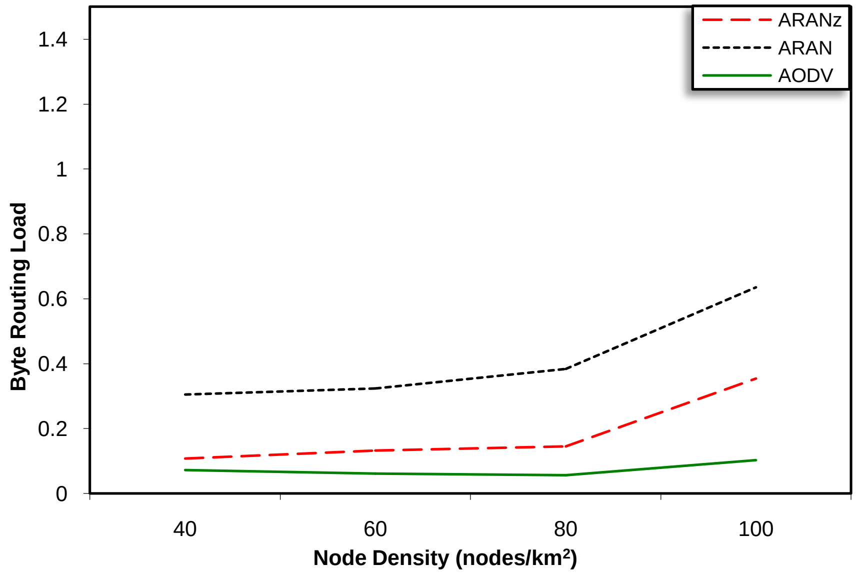


Figure 5.22: *BRL* vs. node density

Table 5.45: *BRL* vs. node density (t-Test)

	ARANz	*ARAN*
Mean	1.74E-01	3.98E-01
Variance	6.21E-03	1.20E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	2.20E+01	
t Stat	-5.99E+00	
P(T<=t) one-tail	2.49E-06	
t Critical one-tail	1.72E+00	
P(T<=t) two-tail	4.98E-06	

	ARANz	*AODV*
Mean	1.74E-01	6.98E-02
Variance	6.21E-03	1.96E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.30E+01	
t Stat	4.71E+00	
P(T<=t) one-tail	2.06E-04	
t Critical one-tail	1.77E+00	
P(T<=t) two-tail	4.11E-04	

Table 5.46: *BRL* vs. node density (chi-square Test)

ARANz	*ARAN*	*AODV*
0.975859253	0.982263221	0.9993587

It is conspicuous from Figure 5.21 and Figure 5.22 that the *PRL* and *BRL* for the three protocols increase as the node density increases, due to the larger number of nodes receiving and broadcasting *RDP* packets. However, the results of the chi-square Test, presented in Table 5.44 and Table 5.46, show that the increase in *PRL* and *BRL* for the three protocols is statistically insignificant.

ARANz still has the minimum *PRL* as a result of using restricted directional flooding in sending *RDP* packets. Even though *ARANz* has smaller *PRL* than *AODV*, Figure 5.22 shows that its *BRL* is higher due to the larger *ARANz* control packet that contains security data. Table 5.43 and Table 5.45 confirm that the differences in *PRL* and *BRL* between the three protocols are statistically significant.

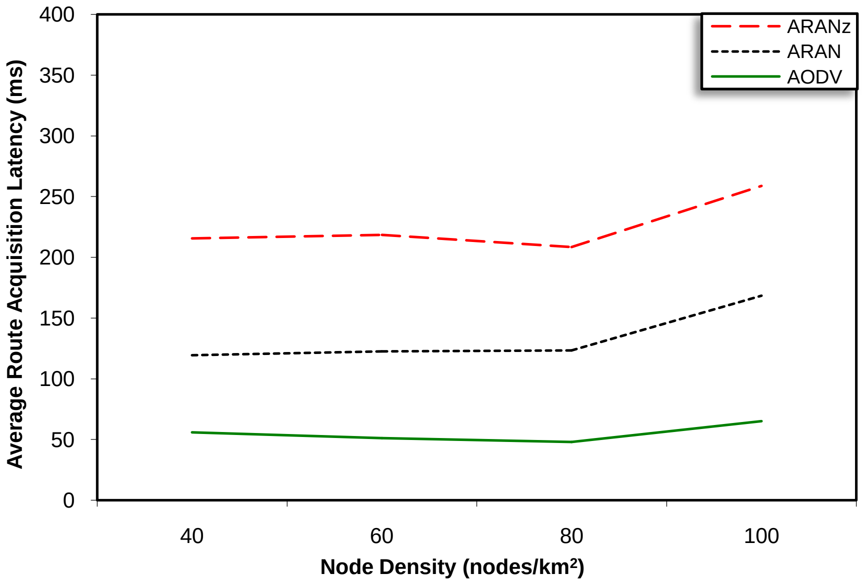


Figure 5.23: *ARAL* vs. node density

Table 5.47: *ARAL* vs. node density (t-Test)

	ARANz	***ARAN***
Mean	2.25E+02	1.33E+02
Variance	5.19E+02	5.46E+02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	5.64E+00	
P(T<=t) one-tail	6.67E-04	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	1.33E-03	
t Critical two-tail	2.45E+00	

	ARANz	***AODV***
Mean	2.25E+02	5.49E+01
Variance	5.19E+02	5.70E+01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	1.42E+01	
P(T<=t) one-tail	7.15E-05	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	1.43E-04	
t Critical two-tail	2.78E+00	

Table 5.48: *ARAL* vs. node density (chi-square Test)

ARANz	***ARAN***	***AODV***
0.074756351	0.064732218	0.174897357

Figure 5.23 and Table 5.48 show that the *ARAL* for the three protocols increases with increasing node density. This increase is a result of the increased number of nodes participating in broadcasting *RDP* packets, which causes congestion as well as delay in processing control packets.

As in other scenarios, *ARAL* for *ARAN* and *ARANz* protocols is higher than *AODV* due to digital signature generation and verification. Also, *ARAL* for *ARANz* is higher than that for *ARAN* due to time required to get the destination node’s position. Results of the t-Test (presented in Table 5.47) show that the differences between *ARAL* for the three protocols are statistically significant.

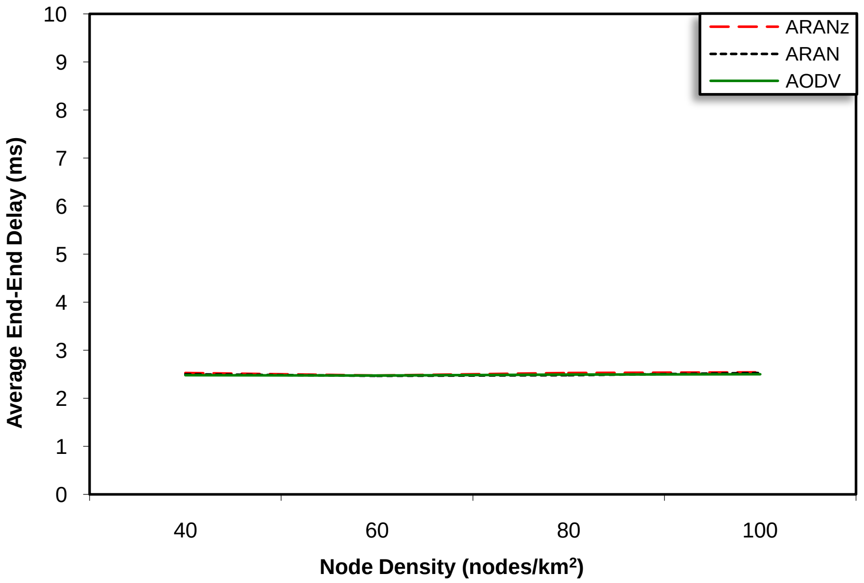


Figure 5.24: *AEED* vs. node density

Table 5.49: *AEED* vs. node density (t-Test)

	ARANz	*ARAN*
Mean	2.51E+00	2.49E+00
Variance	9.28E-04	7.60E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	1.05E+00	
P(T<=t) one-tail	1.68E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	3.36E-01	
t Critical two-tail	2.45E+00	

	ARANz	*AODV*
Mean	2.51E+00	2.49E+00
Variance	9.28E-04	1.34E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	1.75E+00	
P(T<=t) one-tail	7.71E-02	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	1.54E-01	
t Critical two-tail	2.78E+00	

Table 5.50: *AEED* vs. node density (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999990214	0.999992642	0.999999455

Figure 5.24 demonstrates that *AEED* curves for the three protocols are almost identical to each other. Although *ARAN* and *ARANz* have higher *ARAL*, the number of route discoveries and position enquiries performed is a small fraction of the number of data packets delivered. Hence, the effect of *ARAL* on *AEED* of the data packets is insignificant. Table 5.49 confirms that the differences in *AEED* between the three protocols are statistically insignificant. Moreover, the results of the chi-square Test,

presented in Table 5.50, assure that the *AEED* for the three protocols is not affected by increasing node density.

5.3.4 Local Communication Effect

To evaluate our protocol considering local communication percentage, a 2km×2km network which is divided into 4 zones is considered. A total of 240 nodes are randomly placed in this network. These nodes are allowed to move at 5m/s speed. Five *CBR* sessions are simulated in each run. Simulations are run with 0%, 40%, 60% and 100% local communication. These percentages are adjusted by specifying the local and external *CBR* sessions. For example, to simulate 40% local communication, two of the *CBR* sessions are chosen as local and the other three are external.

As shown in Figure 5.25, *PDF* obtained using either protocol slightly increases as the percentage of local communication increases and nearly reaches 100% when all communications are local. Larger percentage of local communications means shorter paths, i.e. lower probability of having link breakage and data packet drops. However, Table 5.51 and Table 5.52 show that the differences between the *PDF* of the three protocols and the increase in the *PDF* for the three protocols are statistically insignificant. Moreover, it is clear from the figure that *PDF* obtained for either protocol is above 96% in all scenarios. This suggests that the three protocols are highly effective in discovering and maintaining routes for delivery of data packets.

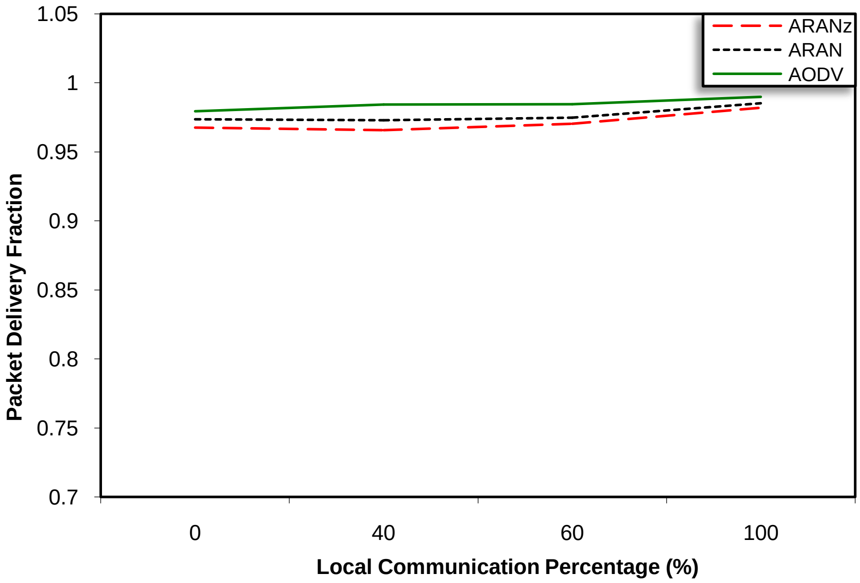


Figure 5.25: *PDF* vs. local communication percentage

Table 5.51: *PDF* vs. local communication percentage (t-Test)

	ARANz	***ARAN***
Mean	9.71E-01	9.77E-01
Variance	5.31E-05	3.31E-05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-1.13E+00	
P(T<=t) one-tail	1.51E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	3.02E-01	
t Critical two-tail	2.45E+00	

	ARANz	***AODV***
Mean	9.71E-01	9.84E-01
Variance	5.31E-05	1.77E-05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	-3.12E+00	
P(T<=t) one-tail	1.12E-01	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	2.63E-02	
t Critical two-tail	2.57E+00	

Table 5.52: *PDF* vs. local communication percentage (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999443571	0.999728641	0.999894517

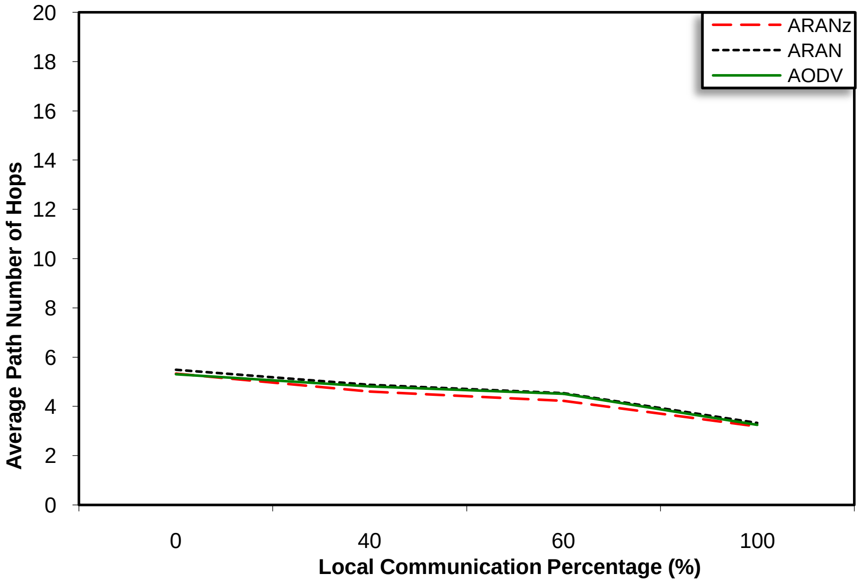


Figure 5.26: *APNH* vs. local communication percentage

Table 5.53: *APNH* vs. local communication percentage (t-Test)

	ARANz	*ARAN*
Mean	4.33E+00	4.56E+00
Variance	7.97E-01	8.28E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-3.49E-01	
P(T<=t) one-tail	3.69E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	7.39E-01	
t Critical two-tail	2.45E+00	

	ARANz	*AODV*
Mean	4.33E+00	4.46E+00
Variance	7.97E-01	7.71E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-2.10E-01	
P(T<=t) one-tail	4.20E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	8.41E-01	
t Critical two-tail	2.45E+00	

Table 5.54: *APNH* vs. local communication percentage (chi-square Test)

ARANz	*ARAN*	*AODV*
0.907373092	0.90884223	0.914826538

Figure 5.26 and the results of the t-Test presented in Table 5.53 show that *ARAN* and *ARANz* are as efficient as *AODV* in discovering the shortest paths regardless of the simulated local communication percentage. The same figure and Table 5.54 indicate that *APNH* slightly decreases for all protocols with increasing local communication because the source and destination nodes are closer to each other.

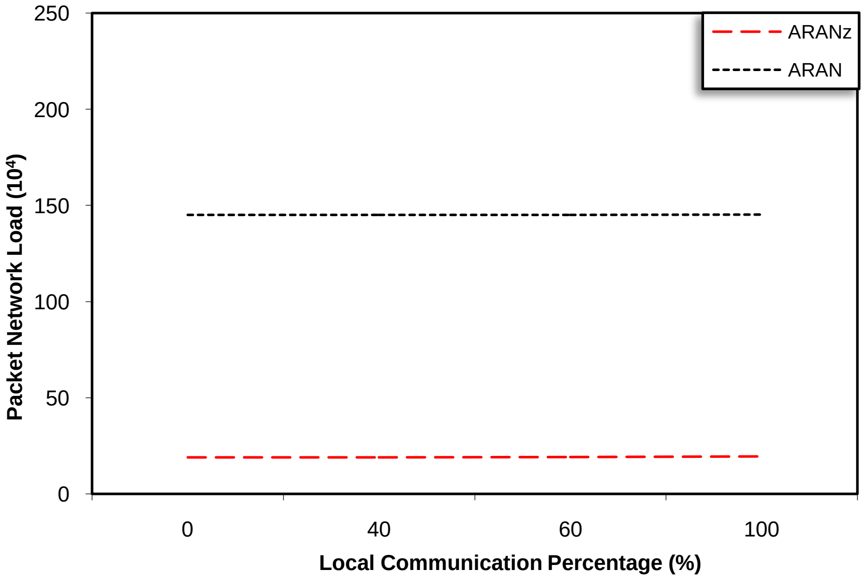


Figure 5.27: *PNL* vs. local communication percentage

Table 5.55: *PNL* vs. local communication percentage (t-Test)

	ARANz	***ARAN***
Mean	1.91E+01	1.45E+02
Variance	7.36E-02	4.84E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-8.99E+02	
P(T<=t) one-tail	1.52E-09	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.03E-09	
t Critical two-tail	3.18E+00	

Table 5.56: *PNL* vs. local communication percentage (chi-square Test)

ARANz	***ARAN***
0.99967069	0.999999734

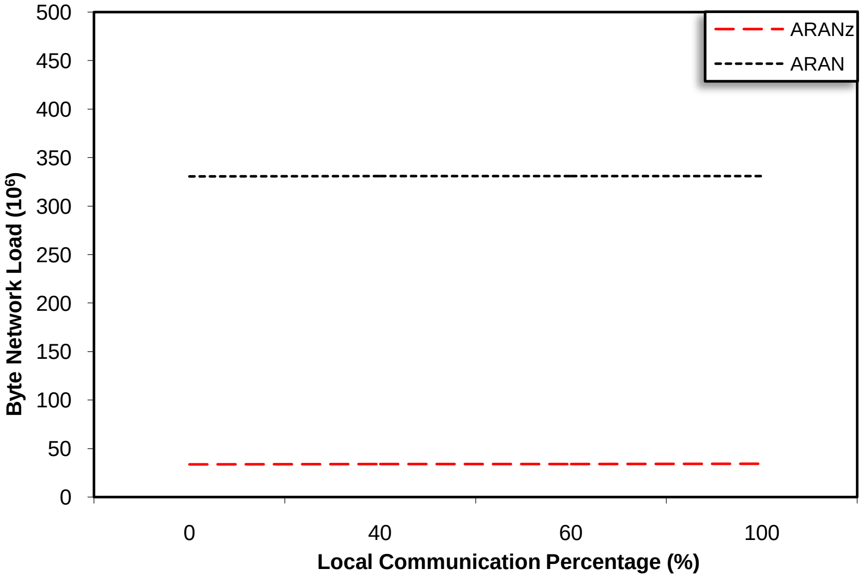


Figure 5.28: *BNL* vs. local communication percentage

Table 5.57: *BNL* vs. local communication percentage (t-Test)

	ARANz	***ARAN***
Mean	3.39E+01	3.31E+02
Variance	8.94E-02	2.52E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	-1.75E+03	
P(T<=t) one-tail	5.70E-16	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	1.14E-15	
t Critical two-tail	2.57E+00	

Table 5.58: *BNL* vs. local communication percentage (chi-square Test)

ARANz	***ARAN***
0.999813226	0.999999083

Figure 5.27, Figure 5.28, Table 5.56 and Table 5.58 show that the *PNL* and *BNL* for both protocols are not affected by local communication percentage because the packets sent for updating nodes certificates and maintaining network structure are sent regardless of the number and type of communication sessions among nodes. Figure 5.27 and Figure 5.28 show that *PNL* and *BNL* for *ARANz* are still much less than these for

ARAN and the results of the t-Test (presented in Table 5.55 and Table 5.57) assist that the differences in *PNL* and *BNL* for the two protocols are statistically significant.

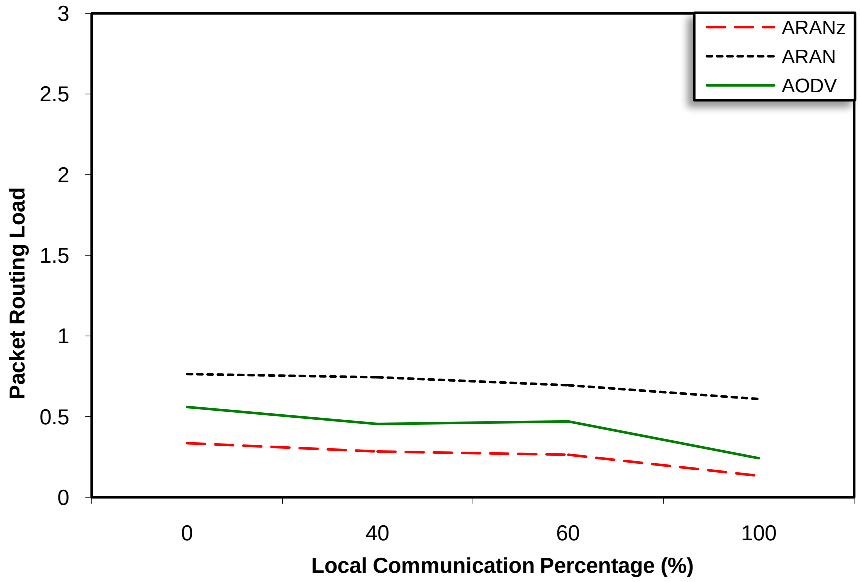


Figure 5.29: *PRL* vs. local communication percentage

Table 5.59: *PRL* vs. local communication percentage (t-Test)

	ARANz	*ARAN*
Mean	2.57E-01	7.06E-01
Variance	3.69E-03	2.64E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	2.30E+01	
t Stat	-2.03E+01	
P(T<=t) one-tail	1.73E-16	
t Critical one-tail	1.71E+00	
P(T<=t) two-tail	3.46E-16	
t Critical two-tail	2.07E+00	

	ARANz	*AODV*
Mean	2.57E-01	4.38E-01
Variance	3.69E-03	8.21E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	2.10E+01	
t Stat	-5.96E+00	
P(T<=t) one-tail	3.26E-06	
t Critical one-tail	1.72E+00	
P(T<=t) two-tail	6.51E-06	
t Critical two-tail	2.08E+00	

Table 5.60: *PRL* vs. local communication percentage (chi-square Test)

ARANz	*ARAN*	*AODV*
0.993201607	0.999233634	0.988520979

Figure 5.29 and Table 5.60 show that the *PRL* curves for the three protocols slightly decrease as the local communication increases due to the shorter paths. Shorter paths decrease the probability of link break, which in turn, reduces the need for reinitiating a new *RDP* packet. The figure shows that *ARANz*'*PRL* is significantly lower than the

other two protocols, since *ARANz* does not broadcast the *RDP* packet to the whole network, instead, it is sent using restricted directional flooding. It is also clear from Figure 5.29 that *ARAN* has higher *PRL* than *AODV*. As discussed earlier, this difference results as a consequence of higher packets processing and authentication delay in *ARAN* along with the possibility of sending *RREP* packets by the intermediate nodes in *AODV*. Table 5.59 confirms that the *PRL* of the three protocols are significantly different.

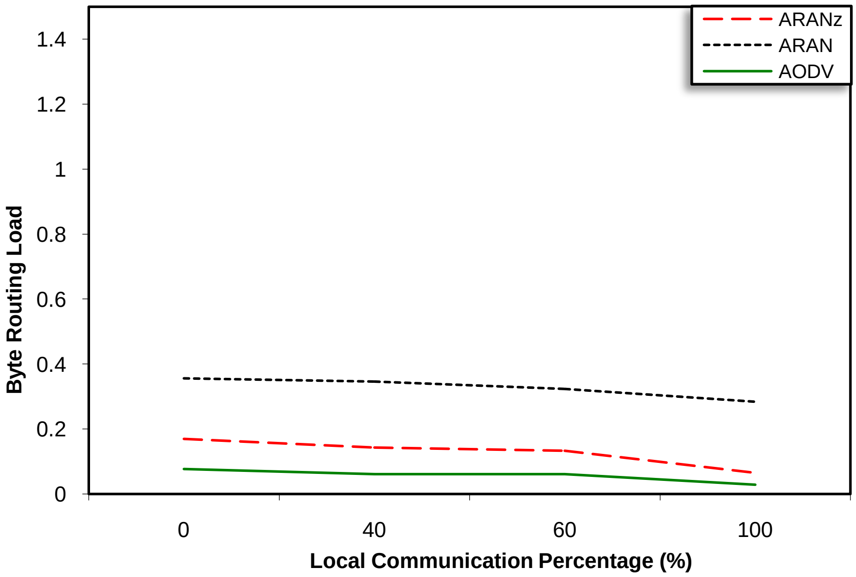


Figure 5.30: *BRL* vs. local communication percentage

Table 5.61: *BRL* vs. local communication percentage (t-Test)

	ARANz	*ARAN*
Mean	1.30E-01	3.29E-01
Variance	9.68E-04	5.67E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	2.20E+01	
t Stat	-1.83E+01	
P(T<=t) one-tail	4.18E-15	
t Critical one-tail	1.72E+00	
P(T<=t) two-tail	8.37E-15	
t Critical two-tail	2.07E+00	

	ARANz	*AODV*
Mean	1.30E-01	5.77E-02
Variance	9.68E-04	1.91E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.70E+01	
t Stat	7.62E+00	
P(T<=t) one-tail	3.51E-07	
t Critical one-tail	1.74E+00	
P(T<=t) two-tail	7.02E-07	
t Critical two-tail	2.11E+00	

Table 5.62: *BRL* vs. local communication percentage (chi-square Test)

ARANz	*ARAN*	*AODV*
0.997407611	0.999759873	0.999159221

Even though *ARANz* has smaller *PRL* than *AODV*, it has higher *BRL* (as shown in Figure 5.30 and Table 5.61). Higher *ARANz*'s *BRL* results from its larger control packets that contain security data. It is also clear from the figure and Table 5.62 that as the local communication increases, *BRL* slightly decreases for all protocols due to shorter paths and reduced probability of having link breakage.

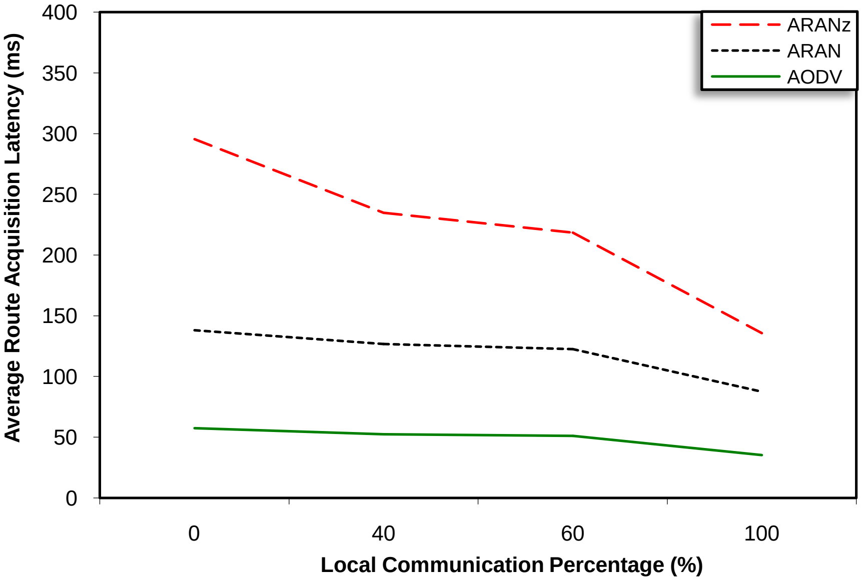


Figure 5.31: *ARAL* vs. local communication percentage

Table 5.63: *ARAL* vs. local communication percentage (t-Test)

	ARANz	*ARAN*
Mean	2.21E+02	1.19E+02
Variance	4.35E+03	4.78E+02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	2.95E+00	
P(T<=t) one-tail	2.10E-02	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	4.20E-02	
t Critical two-tail	2.78E+00	

	ARANz	*AODV*
Mean	2.21E+02	4.90E+01
Variance	4.35E+03	9.33E+01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	5.16E+00	
P(T<=t) one-tail	7.05E-03	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.41E-02	
t Critical two-tail	3.18E+00	

Table 5.64: *ARAL* vs. local communication percentage (chi-square Test)

ARANz	*ARAN*	*AODV*
9.65497E-13	0.007056394	0.126532763

As expected, Figure 5.31 shows that *AODV* is superior in its *ARAL* as it has the shortest processing delay at each node. Moreover, *ARANz* has the highest *ARAL* because *ARANz* needs to carry out a position discovery step. Results of the t-Test (presented in Table 5.63) show that the differences between *ARAL* for the three protocols are statistically significant. However, *ARANz*'s *ARAL* improves rapidly as more and more packets become internal ones because the nearest *LCA*, upon receiving a *PDP* packet, will find the destination in its authentication table, so there is no need to communicate with *LCA*s in other zones. In fact, all protocols have better *ARAL* as more packets are delivered locally due to shorter paths although the results of the chi-square Test (presented in Table 5.64) assure that *ARAL* curves of *AODV* and *ARAN* decrease at a slower rate compared to *ARANz*. The reason behind this difference is that the *RDP* packets in *AODV* and *ARAN* are flooded to the whole area even if the communications are local. This flooding affects *ARAL* for other external communications by increasing the processing delay of other *RDP* packets.

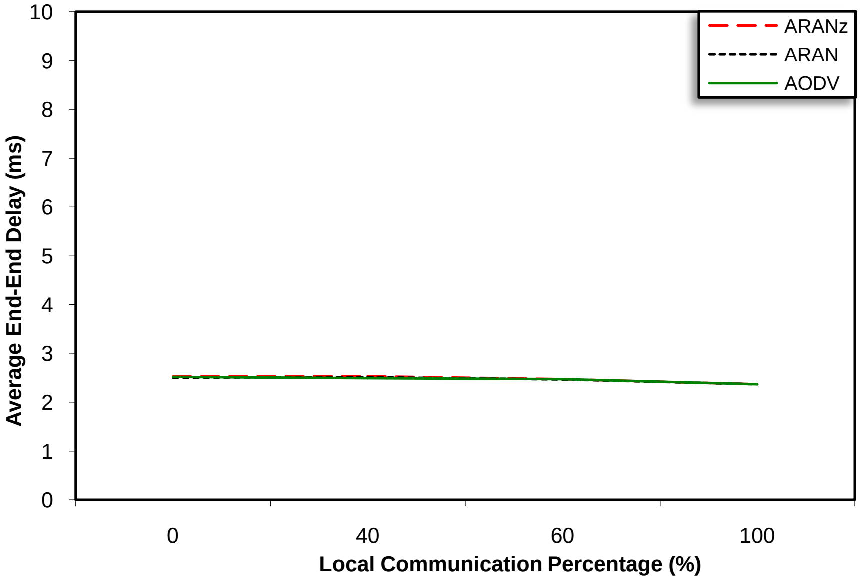


Figure 5.32: *AEED* vs. local communication percentage

Table 5.65: *AEED* vs. local communication percentage (t-Test)

	ARANz	***ARAN***
Mean	2.45E+00	2.45E+00
Variance	1.57E-02	7.23E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	-6.20E-02	
P(T<=t) one-tail	4.76E-01	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	9.53E-01	
t Critical two-tail	2.57E+00	

	ARANz	***AODV***
Mean	2.45E+00	2.44E+00
Variance	1.57E-02	1.47E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	1.30E-01	
P(T<=t) one-tail	4.50E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	9.01E-01	
t Critical two-tail	2.45E+00	

Table 5.66: *AEED* vs. local communication percentage (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999292517	0.999779433	0.99935506

Figure 5.32 shows that *AEED* slightly decreases with increasing local communication due to the shorter paths whether for data or control packets. Though, the results of the t-Test (presented in Table 5.65) and the chi-square Test (presented in Table 5.66) show that the differences in *AEED* between the three protocols and for each protocol separately are statistically insignificant.

5.3.5 Zone Size Effect

To examine the effect of zone size, two networks of 3km×3km and 2km×2km are considered and divided into multiple zones as discussed in the following two sections.

5.3.5.1 Zone Size Effect Considering 3km×3km Network

In this scenario, a network of 3km×3km is considered. This network contains 540 nodes, i.e. the node density is 60 nodes/km^2. The nodes move at a maximum speed of 5m/s. Five *CBR* sessions are simulated in each run, three of them are local and two are external. The network is divided into 1 zone of 3km×3km, 4 zones each of 1.5km×1.5km, 9 zones each of 1km×1km and finally 16 zones each of 750m×750m.

By looking at Figure 5.33 through Figure 5.40 and considering results of the related chi-square Test, it is clear that *ARAN* and *AODV* are not affected by changing zone size since only *ARANz* deals with the network as zones. Accordingly, only *ARANz* protocol is considered in the following discussion.

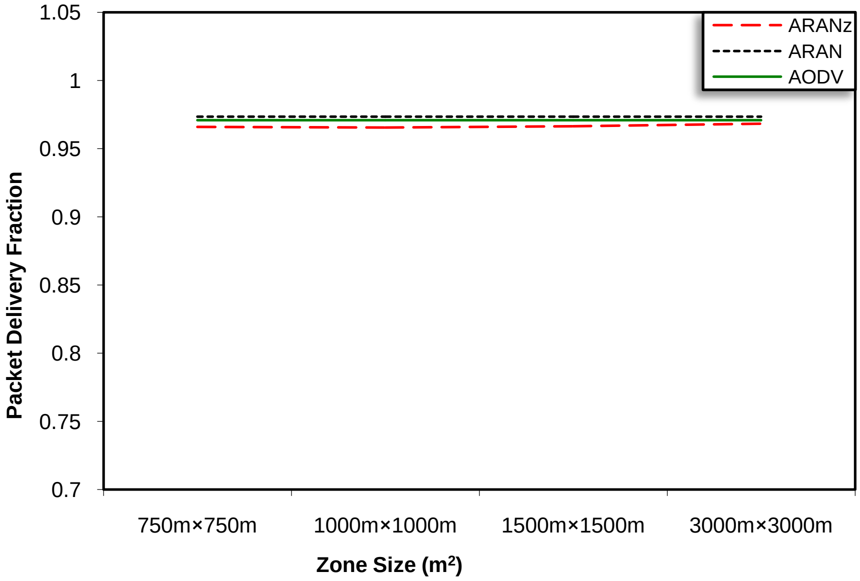


Figure 5.33: *PDF* vs. zone size (considering a 3km×3km network)

Table 5.67: *PDF* vs. zone size (considering a 3km×3km network) (t-Test)

	ARANz	***ARAN***
Mean	9.67E-01	9.73E-01
Variance	1.52E-06	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.11E+01	
P(T<=t) one-tail	3.88E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.58E-03	
t Critical two-tail	3.18E+00	

	ARANz	***AODV***
Mean	9.67E-01	9.71E-01
Variance	1.52E-06	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-6.85E+00	
P(T<=t) one-tail	4.18E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	6.36E-03	
t Critical two-tail	3.18E+00	

Table 5.68: *PDF* vs. zone size (considering a 3km×3km network) (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999997285	1	1

As for *ARAN* and *AODV*, Figure 5.33 show that *ARANz*'s *PDF* is always above 96%. This is an indication that *ARANz*, just like *ARAN* and *AODV*, is highly effective in discovering and maintaining routes regardless of zone size. This result is assisted by the results presented in Table 5.67 and 5.68.

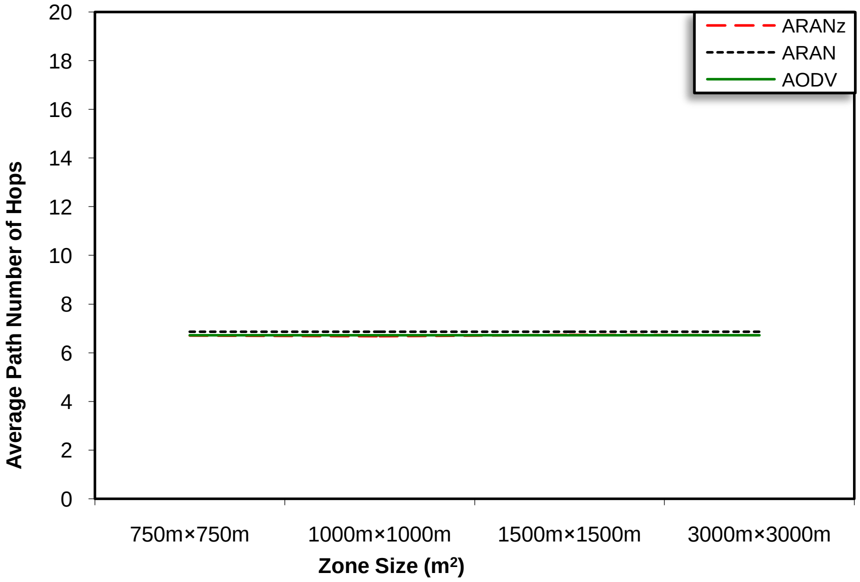


Figure 5.34: *APNH* vs. zone size (considering a 3km×3km network)

Table 5.69: *APNH* vs. zone size (considering a 3km×3km network) (t-Test)

	ARANz	*ARAN*
Mean	6.71E+00	6.86E+00
Variance	4.63E-04	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.36E+01	
P(T<=t) one-tail	9.34E-02	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	8.68E-04	
t Critical two-tail	3.18E+00	

	ARANz	*AODV*
Mean	6.71E+00	6.73E+00
Variance	4.63E-04	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.28E+00	
P(T<=t) one-tail	1.46E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.91E-01	
t Critical two-tail	3.18E+00	

Table 5.70: *APNH* vs. zone size (considering a 3km×3km network) (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999999209	1	1

Figure 5.34 shows that *APNH* for *ARANz* is identical to that for the other two protocols, suggesting that *ARANz* is as efficient as the other two protocols in discovering the shortest paths regardless zone size. Table 5.69 and Table 5.70 confirm that the differences in *APNH* are statistically insignificant.

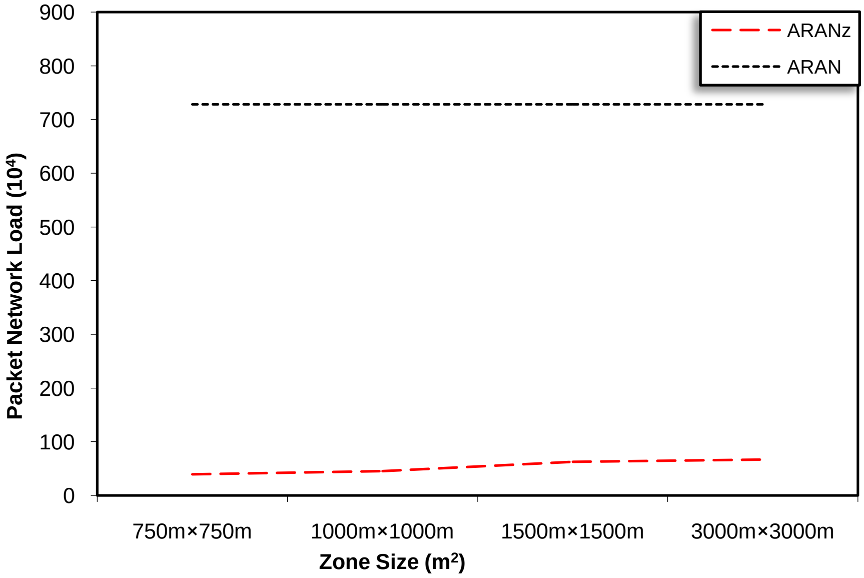


Figure 5.35: *PNL* vs. zone size (considering a 3km×3km network)

Table 5.71: *PNL* vs. zone size (considering a 3km×3km network) (t-Test)

	ARANz	***ARAN***
Mean	5.34E+01	7.29E+02
Variance	1.74E+02	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.02E+02	
P(T<=t) one-tail	1.03E-06	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.05E-06	
t Critical two-tail	3.18E+00	

Table 5.72: *PNL* vs. zone size (considering a 3km×3km network) (chi-square Test)

ARANz	***ARAN***
0.020649798	1

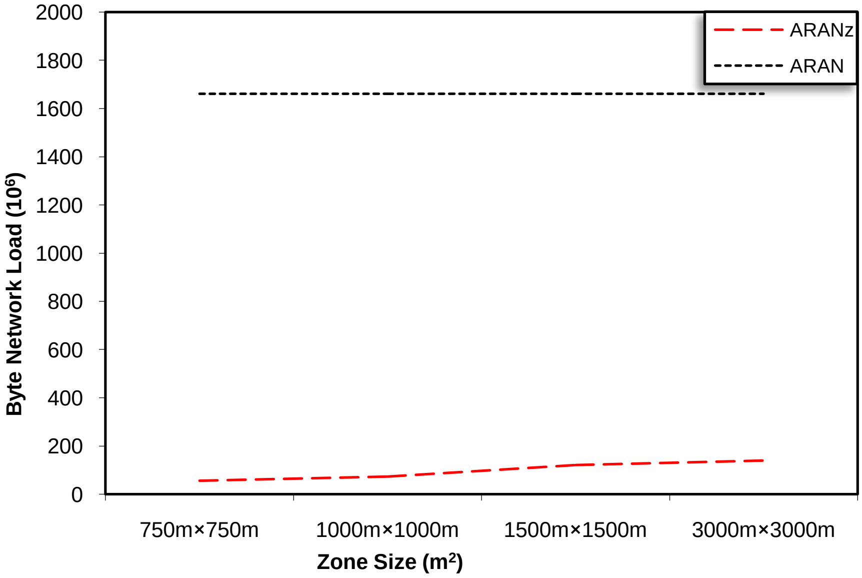


Figure 5.36: *BNL* vs. zone size (considering a 3km×3km network)

Table 5.73: *BNL* vs. zone size (considering a 3km×3km network) (t-Test)

	ARANz	***ARAN***
Mean	9.75E+01	1.66E+03
Variance	1.55E+03	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-7.95E+01	
P(T<=t) one-tail	2.19E-06	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	4.39E-06	
t Critical two-tail	3.18E+00	

Table 5.74: *BNL* vs. zone size (considering a 3km×3km network) (chi-square Test)

ARANz	***ARAN***
2.49796E-10	1

Referring to Figure 5.35 and Figure 5.36, it is clear that *PNL* and *BNL* for *ARANz* increase as the zone size increases. This is because packets sent for updating nodes certificates and maintaining the network structure are sent using restricted directional flooding towards the nearest *LCA* to the node, i.e. as the distance between the node and the nearest *LCA* increases, the number of nodes participating in forwarding these packets also increases. Results of the chi-square Test (presented in Table 5.72 and Table 5.74) show that the increase in *PNL* and *BNL* for *ARANz* is statistically significant.

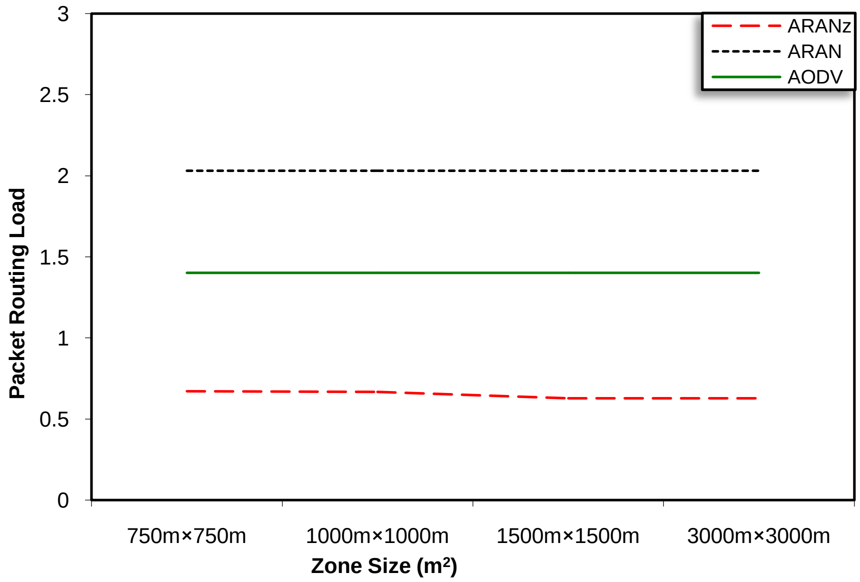


Figure 5.37: *PRL* vs. zone size (considering a 3km×3km network)

Table 5.75: *PRL* vs. zone size (considering a 3km×3km network) (t-Test)

	ARANz	*ARAN*
Mean	6.48E-01	2.03E+00
Variance	3.85E-04	2.14E-31
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-2.54E+02	
P(T<=t) one-tail	4.64E-24	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	9.28E-24	
t Critical two-tail	2.18E+00	

	ARANz	*AODV*
Mean	6.48E-01	1.40E+00
Variance	3.85E-04	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-1.38E+02	
P(T<=t) one-tail	6.81E-21	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	1.36E-20	
t Critical two-tail	2.18E+00	

Table 5.76: *PRL* vs. zone size (considering a 3km×3km network) (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999961878	1	1

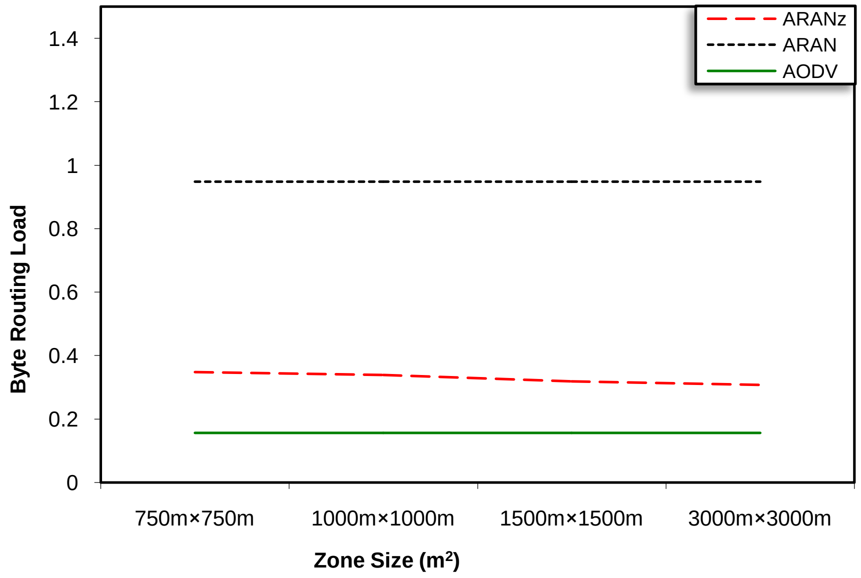


Figure 5.38: *BRL* vs. zone size (considering a 3km×3km network)

Table 5.77: *BRL* vs. zone size (considering a 3km×3km network) (t-Test)

	ARANz	*ARAN*
Mean	3.29E-01	9.48E-01
Variance	2.04E-04	1.34E-32
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-1.56E+02	
P(T<=t) one-tail	1.56E-21	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	3.12E-21	
t Critical two-tail	2.18E+00	

	ARANz	*AODV*
Mean	3.29E-01	1.56E-01
Variance	2.04E-04	8.35E-34
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	4.36E+01	
P(T<=t) one-tail	6.96E-15	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	1.39E-14	
t Critical two-tail	2.18E+00	

Table 5.78: *BRL* vs. zone size (considering a 3km×3km network) (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999954576	1	1

Figure 5.37, Figure 5.38, Table 5.76 and Table 5.78 show that the *PRL* and *BRL* for *ARANz* slightly decrease with increasing the zone size. This is because dividing the area into multiple zones reduces the probability of finding the destination in the authentication table of the nearest *LCA*, therefore, the *PRL* increases due to communicating *LCA*s in other zones. However, in the case of dealing with the network

as one zone, the nearest *LCA* upon receiving *PDP* packet finds the destination in its authentication table, so there is no need to communicate other *LCA*s.

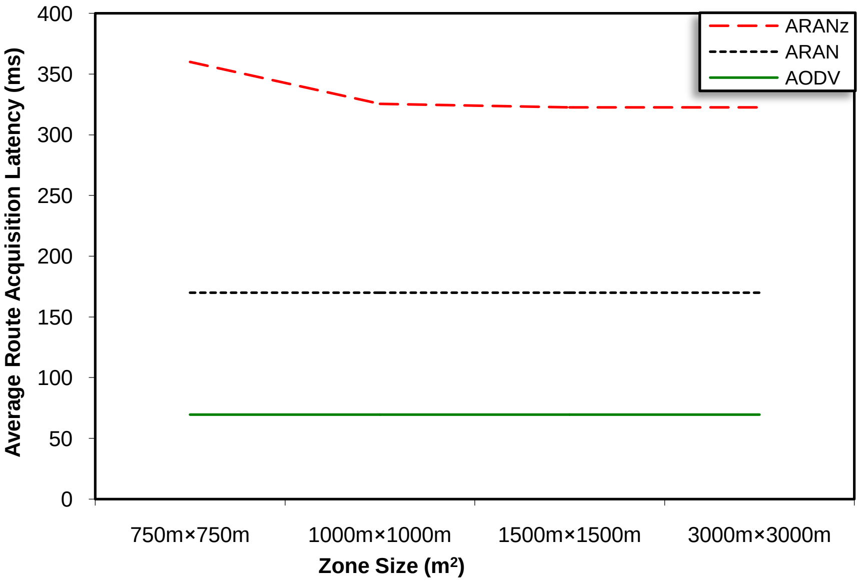


Figure 5.39: *ARAL* vs. zone size (considering a 3km×3km network)

Table 5.79: *ARAL* vs. zone size (considering a 3km×3km network) (t-Test)

	ARANz	***ARAN***
Mean	3.33E+02	1.70E+02
Variance	3.31E+02	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	1.79E+01	
P(T<=t) one-tail	1.91E-04	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.82E-04	
t Critical two-tail	3.18E+00	

	ARANz	***AODV***
Mean	3.33E+02	6.95E+01
Variance	3.31E+02	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	2.89E+01	
P(T<=t) one-tail	4.55E-05	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	9.09E-05	
t Critical two-tail	3.18E+00	

Table 5.80: *ARAL* vs. zone size (considering a 3km×3km network) (chi-square Test)

ARANz	***ARAN***	***AODV***
0.0493234078	1	1

Figure 5.39 and Table 5.80 show that *ARAL* for *ARANz* significantly decreases as the zone size increases. The highest *ARAL* is obtained in the case of 750m×750m zone size due to time required for communicating *LCA*s in other zones to inquiry about the destination's position.

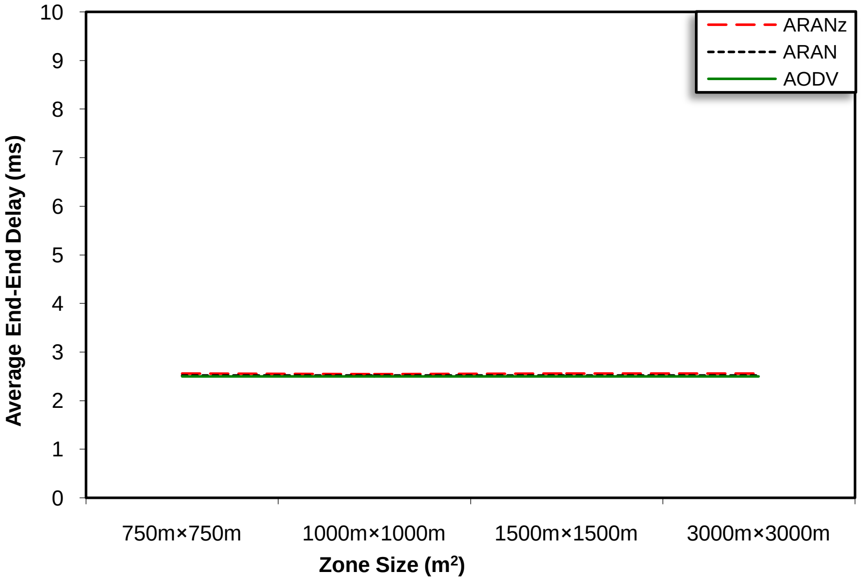


Figure 5.40: *AEED* vs. zone size (considering a 3km×3km network)

Table 5.81: *AEED* vs. zone size (considering a 3km×3km network) (t-Test)

	ARANz	*ARAN*
Mean	2.55E+00	2.52E+00
Variance	4.31E-05	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	9.81E+00	
P(T<=t) one-tail	1.13E-03	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.25E-03	
t Critical two-tail	3.18E+00	

	ARANz	*AODV*
Mean	2.55E+00	2.49E+00
Variance	4.31E-05	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	1.76E+01	
P(T<=t) one-tail	2.02E-04	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	4.03E-04	
t Critical two-tail	3.18E+00	

Table 5.82: *AEED* vs. zone size (considering a 3km×3km network) (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999999904	1	1

Figure 5.40 shows that *ARANz*’s *AEED* is almost not affected by changing zone size. As mentioned previously, the number of route and position discoveries is a small fraction of the number of data packets delivered. Hence the effect of *ARAL* on *AEED* is unnoticeable. Results of the chi-square Test (presented in Table 5.82) assure that the differences in *ARANz*’s *AEED* are statistically insignificant.

It is conspicuous from the analysis that a better performance (significantly reduced *PNL* and *BNL)* is obtained for *ARANz* upon using a small zone size. On the other hand, *PRL* and *BRL* slightly decrease and *ARAL* significantly decreases as the zone size increases. Hence a moderate performance in terms of the five metrics is obtained upon dividing the area into four 1.5km×1.5km or nine 1km×1km zones.

5.3.5.2 Zone Size Effect Considering 2km×2km Network

To ensure the results obtained in the first scenario, another simulation scenario is carried out. In other words, the aim of this scenario is to ensure whether a moderate performance is obtained upon dividing the area into four or nine zones, or upon using 1.5km×1.5km or 1km×1km zone size.

In this scenario, a network size of 2km×2km, a node density of 60 nodes/km^2 and a maximum mobility speed of 5m/s are considered. Three local and two external *CBR* sessions are simulated. The network is divided into 1 zone of 2km×2km, 4 zones each of 1km×1km, 9 zones each of 666.666m×666.666m and finally 16 zones each of 500m×500m.

Looking at Figure 5.41 through Figure 5.48 and results of the related chi-square Test, it is clear that the *PNL* and *BNL* for *ARANz* significantly decrease with decreasing the zone size (increasing the number of zones). On the other hand, the *PRL* and *BRL* slightly decrease and the *ARAL* significantly decreases with increasing the zone size (decreasing the number of zones). Thus a moderate performance regarding the five metrics is obtained upon dividing the area into four 1km×1km or nine 666.666m×666.666m zones.

From the results of the two scenarios we can conclude that regardless of the network size, a moderate performance regarding the five metrics is obtained upon dividing the area into 4 or 9 zones.

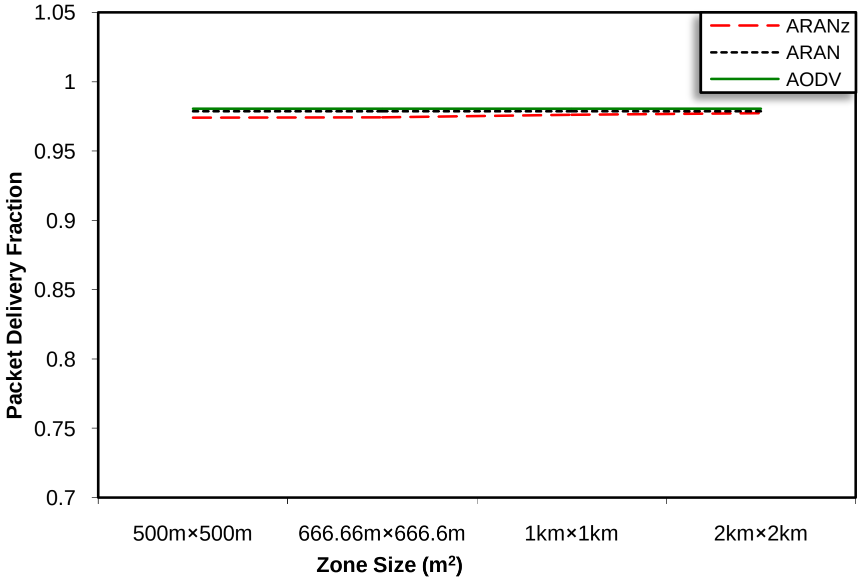


Figure 5.41: *PDF* vs. zone size (considering a 2km×2km network)

Table 5.83: *PDF* vs. zone size (considering a 2km×2km network) (t-Test)

	ARANz	***ARAN***
Mean	9.75E-01	9.79E-01
Variance	2.26E-06	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-4.26E+00	
P(T<=t) one-tail	5.18E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.37E-02	
t Critical two-tail	3.18E+00	

	ARANz	***AODV***
Mean	9.75E-01	9.80E-01
Variance	2.26E-06	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-6.75E+00	
P(T<=t) one-tail	3.32E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	6.63E-03	
t Critical two-tail	3.18E+00	

Table 5.84: *PDF* vs. zone size (considering a 2km×2km network) (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999995127	1	1

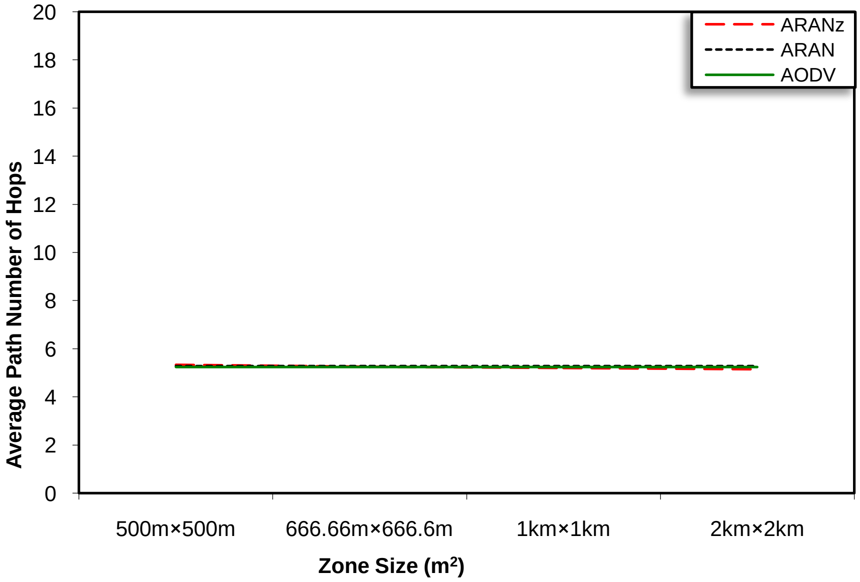


Figure 5.42: *APNH* vs. zone size (considering a 2km×2km network)

Table 5.85: *APNH* vs. zone size (considering a 2km×2km network) (t-Test)

	ARANz	***ARAN***
Mean	5.24E+00	5.28E+00
Variance	6.56E-03	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.05E+00	
P(T<=t) one-tail	1.86E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.73E-01	
t Critical two-tail	3.18E+00	

	ARANz	***AODV***
Mean	5.24E+00	5.24E+00
Variance	6.56E-03	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-3.64E-02	
P(T<=t) one-tail	4.87E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	9.73E-01	
t Critical two-tail	3.18E+00	

Table 5.86: *APNH* vs. zone size (considering a 2km×2km network) (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999938826	1	1

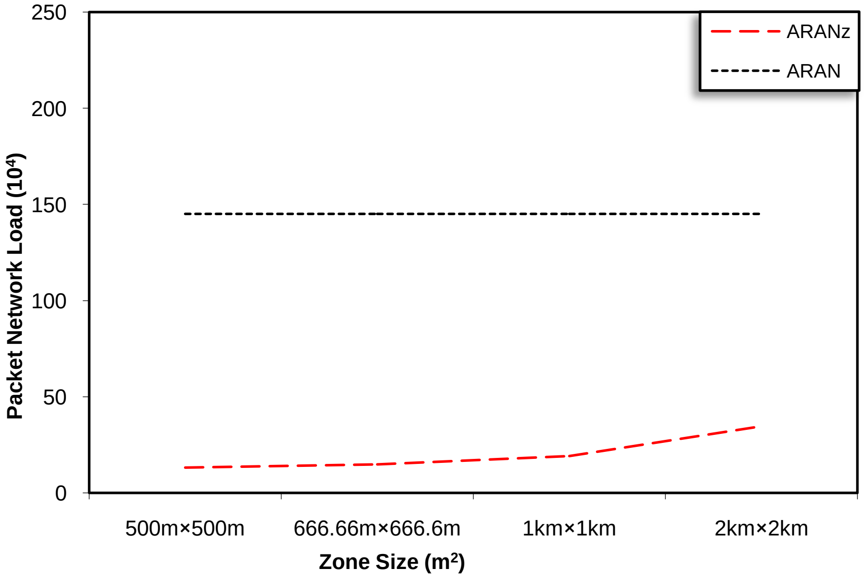


Figure 5.43: *PNL* vs. zone size (considering a 2km×2km network)

Table 5.87: *PNL* vs. zone size (considering a 2km×2km network) (t-Test)

	ARANz	***ARAN***
Mean	2.04E+01	1.45E+02
Variance	9.58E+01	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-2.55E+01	
P(T<=t) one-tail	6.64E-05	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.33E-04	
t Critical two-tail	3.18E+00	

Table 5.88: *PNL* vs. zone size (considering a 2km×2km network) (chi-square Test)

ARANz	***ARAN***
0.002802303	1

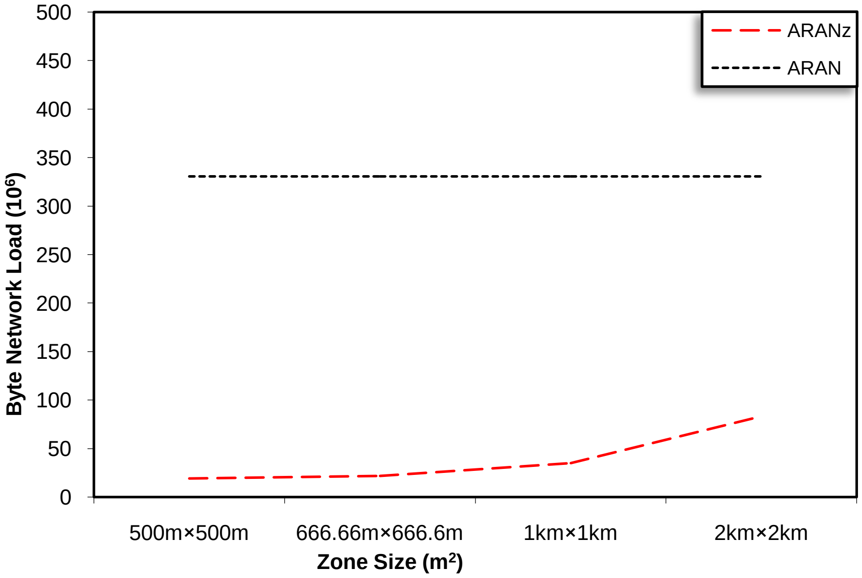


Figure 5.44: *BNL* vs. zone size (considering a 2km×2km network)

Table 5.89: *BNL* vs. zone size (considering a 2km×2km network) (t-Test)

	ARANz	***ARAN***
Mean	3.96E+01	3.31E+02
Variance	8.85E+02	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.96E+01	
P(T<=t) one-tail	1.46E-04	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.92E-04	
t Critical two-tail	3.18E+00	

Table 5.90: *BNL* vs. zone size (considering a 2km×2km network) (chi-square Test)

ARANz	***ARAN***
1.83669E-14	1

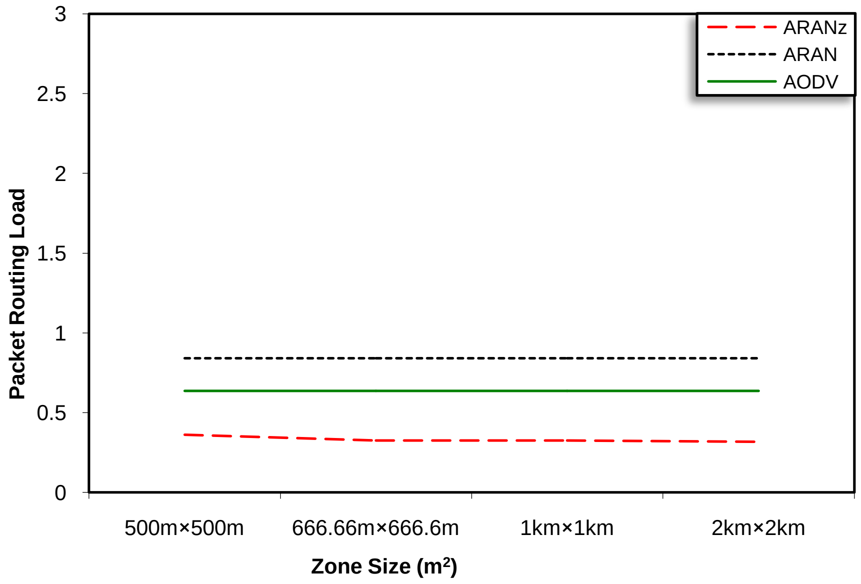


Figure 5.45: *PRL* vs. zone size (considering a 2km×2km network)

Table 5.91: *PRL* vs. zone size (considering a 2km×2km network) (t-Test)

	ARANz	***ARAN***
Mean	3.30E-01	8.41E-01
Variance	1.81E-04	1.34E-32
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-1.37E+02	
P(T<=t) one-tail	7.65E-21	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	1.53E-20	
t Critical two-tail	2.18E+00	

	ARANz	***AODV***
Mean	3.30E-01	6.36E-01
Variance	1.81E-04	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-8.20E+01	
P(T<=t) one-tail	3.61E-18	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	7.21E-18	
t Critical two-tail	2.18E+00	

Table 5.92: *PRL* vs. zone size (considering a 2km×2km network) (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999946242	1	1

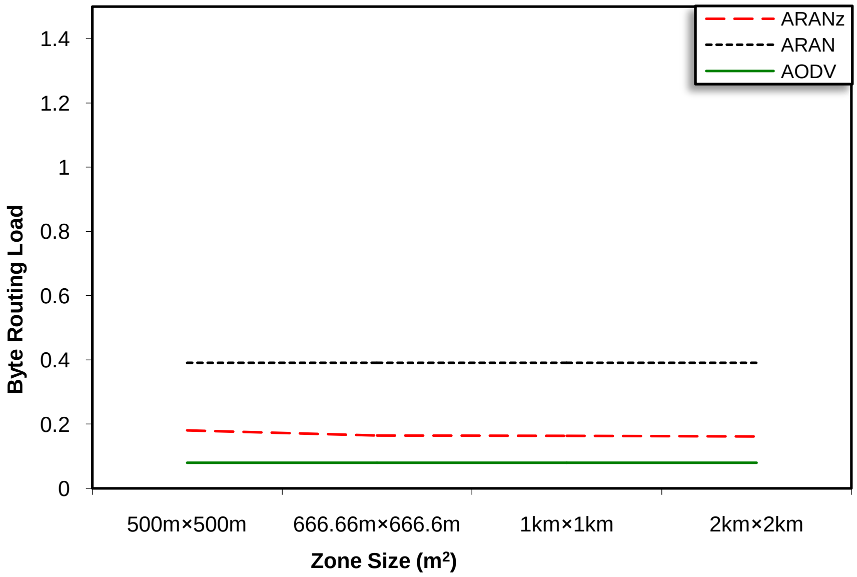


Figure 5.46: *BRL* vs. zone size (considering a 2km×2km network)

Table 5.93: *BRL* vs. zone size (considering a 2km×2km network) (t-Test)

	ARANz	***ARAN***
Mean	1.67E-01	3.91E-01
Variance	3.79E-05	1.34E-32
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-1.32E+02	
P(T<=t) one-tail	1.25E-20	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	2.49E-20	
t Critical two-tail	2.18E+00	

	ARANz	***AODV***
Mean	1.67E-01	7.99E-02
Variance	3.79E-05	2.09E-34
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	5.08E+01	
P(T<=t) one-tail	1.10E-15	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	2.20E-15	
t Critical two-tail	2.18E+00	

Table 5.94: *BRL* vs. zone size (considering a 2km×2km network) (chi-square Test)

ARANz	***ARAN***	***AODV***
0.99998608	1	1

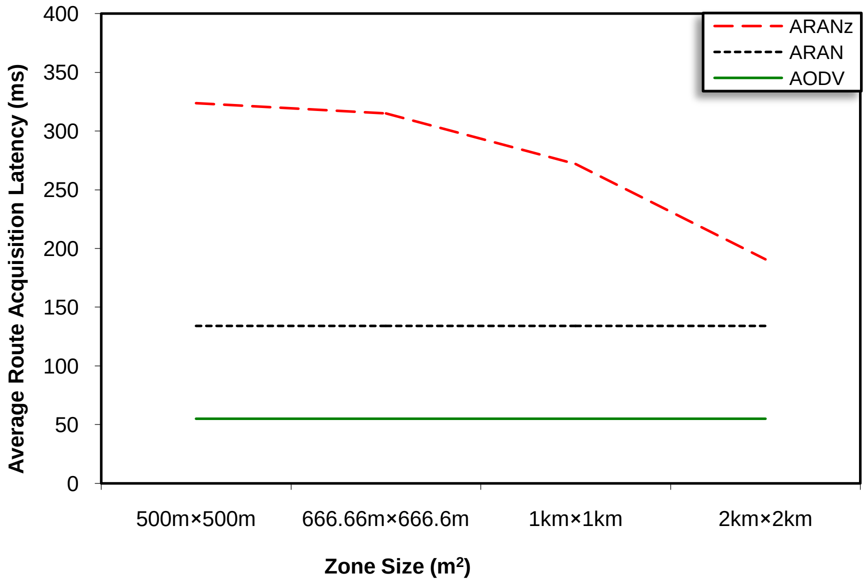


Figure 5.47: *ARAL* vs. zone size (considering a 2km×2km network)

Table 5.95: *ARAL* vs. zone size (considering a 2km×2km network) (t-Test)

	ARANz	***ARAN***
Mean	2.75E+02	1.34E+02
Variance	3.69E+03	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	4.66E+00	
P(T<=t) one-tail	9.35E-03	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.87E-02	
t Critical two-tail	3.18E+00	

	ARANz	***AODV***
Mean	2.75E+02	5.50E+01
Variance	3.69E+03	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	7.25E+00	
P(T<=t) one-tail	2.70E-03	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	5.40E-03	
t Critical two-tail	3.18E+00	

Table 5.96: *ARAL* vs. zone size (considering a 2km×2km network) (chi-square Test)

ARANz	***ARAN***	***AODV***
9.57289E-09	1	1

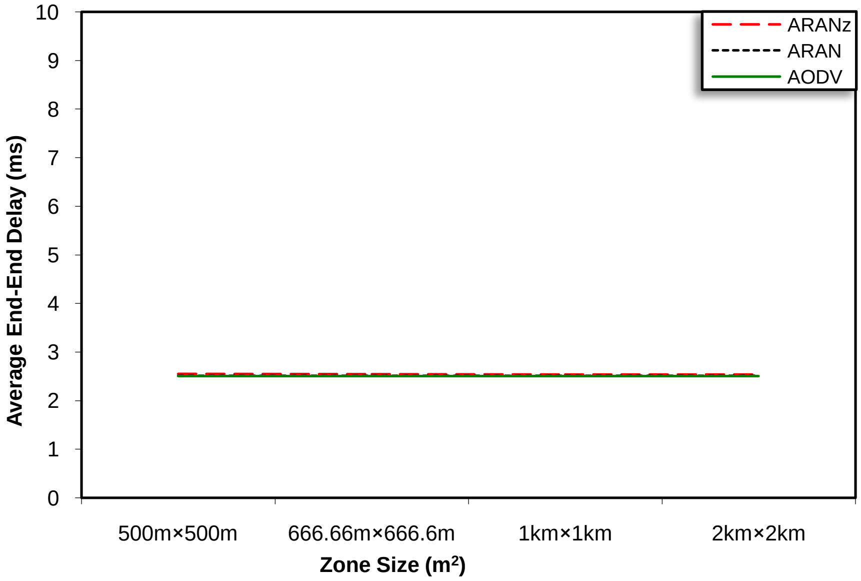


Figure 5.48: *AEED* vs. zone size (considering a 2km×2km network)

Table 5.97: *AEED* vs. zone size (considering a 2km×2km network) (t-Test)

	ARANz	*ARAN*
Mean	2.54E+00	2.51E+00
Variance	4.20E-05	0.01E-05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	8.64E+00	
P(T<=t) one-tail	1.63E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.27E-03	
t Critical two-tail	3.18E+00	

	ARANz	*AODV*
Mean	2.54E+00	2.50E+00
Variance	4.20E-05	0.01E-05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	1.19E+01	
P(T<=t) one-tail	6.42E-02	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.28E-03	
t Critical two-tail	3.18E+00	

Table 5.98: *AEED* vs. zone size (considering a 2km×2km network) (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999999907	1	1

5.3.6 Node Failure Percentage Effect

In the previously studied scenarios, all participating nodes are assumed as well-functioning. In this section, we try to inspect our protocol efficiency and compare it with *AODV* and *ARAN* protocols, in case of having some malfunctioning (failed) nodes.

To examine the effect of node failure percentage a 2km×2km network that is divided into 4 zones is considered. The nodes inside this network move at a maximum speed of

5m/s. Five *CBR* sessions are simulated in each run, three of them are local and two are external. Simulations are run with 0%, 10%, 20% and 40% node failure percentages.

To simulate the node failure, a node periodically draws a random number between 0 and 1. If the drawn number is less than the failure probability, then the node deletes all information about the zone it is residing in and becomes unable to participate in the network activities. Node failure continues until a randomly chosen period between 10s and 60s (Owen & Adda 2009). By the end of this period, the failed node is placed at a random place in the simulation area. After that, the recovered node starts communicating with *LCA*s in the new zone so that it is issued a fresh certificate and rejoins the network.

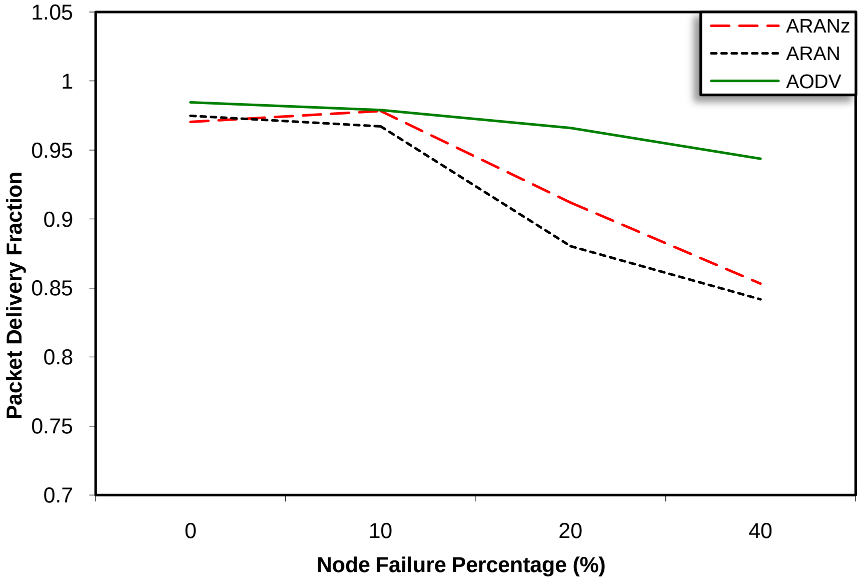


Figure 5.49: *PDF* vs. node failure

Table 5.99: *PDF* vs. node failure (t-Test)

	ARANz	***ARAN***
Mean	9.28E-01	9.16E-01
Variance	3.39E-03	4.28E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	2.83E-01	
P(T<=t) one-tail	4.93E-02	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	7.87E-01	
t Critical two-tail	2.45E+00	

	ARANz	***AODV***
Mean	9.28E-01	9.68E-01
Variance	3.39E-03	3.29E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	-1.31E+00	
P(T<=t) one-tail	1.30E-02	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	2.60E-01	
t Critical two-tail	2.78E+00	

Table 5.100: *PDF* vs. node failure (chi-square Test)

ARANz	***ARAN***	***AODV***
0.777798019	0.704997183	0.9916082

Figure 5.49 and Table 5.100 show that the *PDF* for the three simulated protocols decreases as the node failure percentage increases. A higher node failure percentage leads to a higher probability of having link break resulting in dropping some data packets and reinitiating *RDP* packets. The probability of link breakage is significantly higher for *ARANz* and *ARAN* due to higher packet processing and authentication delay at each node. The situation becomes worse in *ARAN* protocol if the failed node is the *CA* itself. In this case, all other nodes will be unable to update their certificates and take part in sending data packets, resulting in dropping some packets. In *ARANz*, however, only nodes inside a particular zone will not be able to update their certificates upon the failure of the four *LCA*s in that zone at the same time. Results of the t-Test (presented in Table 5.99) show that the differences between *PDF* for the three protocols are statistically significant.

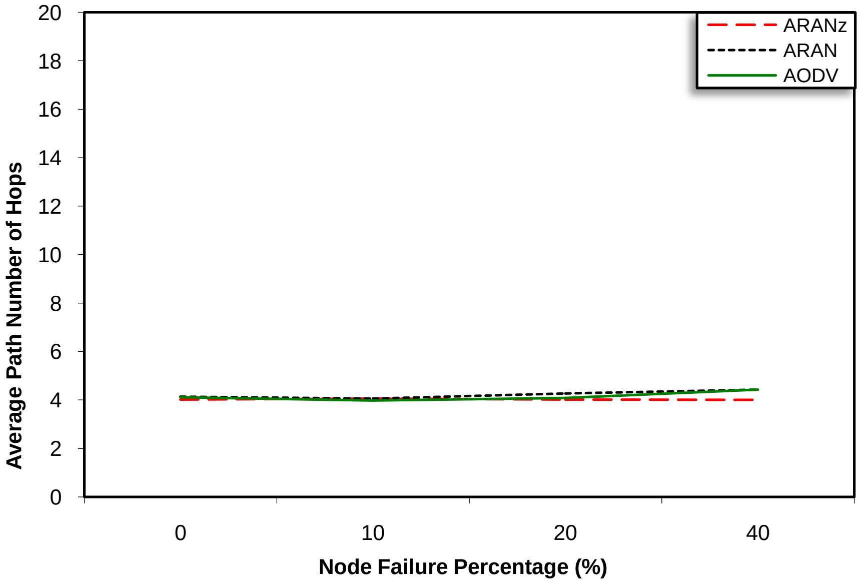


Figure 5.50: *APNH* vs. node failure

Table 5.101: *APNH* vs. node failure (t-Test)

	ARANz	***ARAN***
Mean	4.02E+00	4.22E+00
Variance	4.40E-04	2.62E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-2.46E+00	
P(T<=t) one-tail	1.24E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	9.09E-02	
t Critical two-tail	3.18E+00	

	ARANz	***AODV***
Mean	4.02E+00	4.15E+00
Variance	4.40E-04	3.63E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.35E+00	
P(T<=t) one-tail	1.35E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.71E-01	
t Critical two-tail	3.18E+00	

Table 5.102: *APNH* vs. node failure (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999998418	0.999328559	0.998879594

It is apparent from Figure 5.50 and Table 5.102 that the *APNH* increases slightly with increasing node failure percentage. Higher node failure percentage means higher probability of link breakage and the select of alternate non-optimal paths, increasing *APNH*.

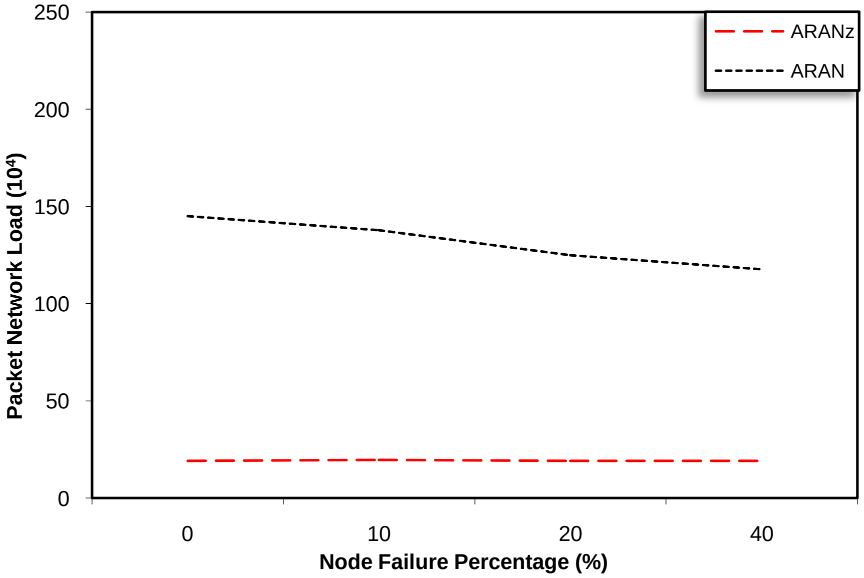


Figure 5.51: *PNL* vs. node failure

Table 5.103: *PNL* vs. node failure (t-Test)

	ARANz	***ARAN***
Mean	1.93E+01	1.31E+02
Variance	5.80E-02	1.51E+02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.82E+01	
P(T<=t) one-tail	1.80E-04	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.60E-04	
t Critical two-tail	3.18E+00	

Table 5.104: *PNL* vs. node failure (chi-square Test)

ARANz	***ARAN***
0.999772954	0.327194515

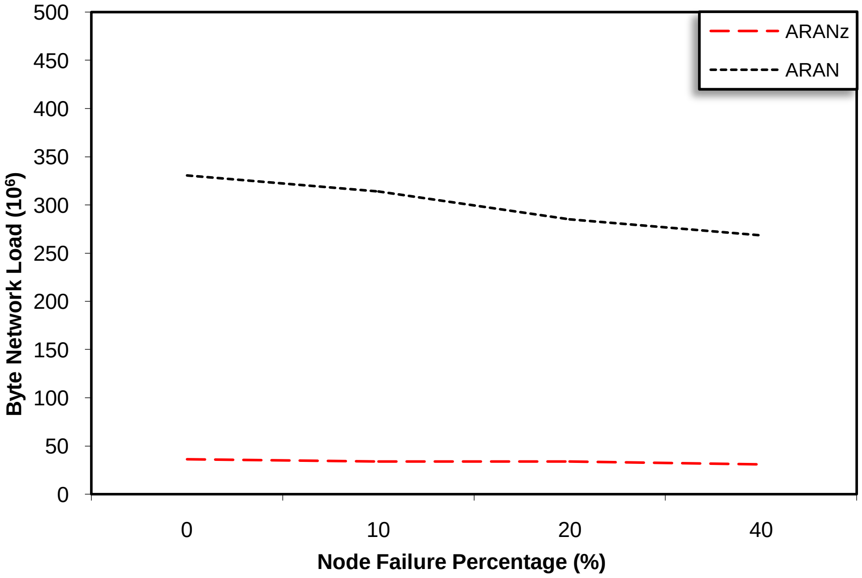


Figure 5.52: *BNL* vs. node failure

Table 5.105: *BNL* vs. node failure (t-Test)

	ARANz	***ARAN***
Mean	3.43E+01	3.00E+02
Variance	1.85E+00	7.86E+02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.89E+01	
P(T<=t) one-tail	1.61E-04	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.23E-04	
t Critical two-tail	3.18E+00	

Table 5.106: *BNL* vs. node failure (chi-square Test)

ARANz	***ARAN***
0.983492702	0.050821981

Figure 5.51 and Figure 5.52 show that the *PNL* and *BNL* for both protocols (*ARAN* and *ARANz*) decrease as the node failure percentage increases. This decrease in *PNL* and *BNL* is due to the decrease in the number of nodes updating their certificates and positions as a result of their failure. However, the results of the chi-square Test (presented in Table 5.104 and Table 5.106) show that the decrease in *PNL* and *BNL* is statistically insignificant.

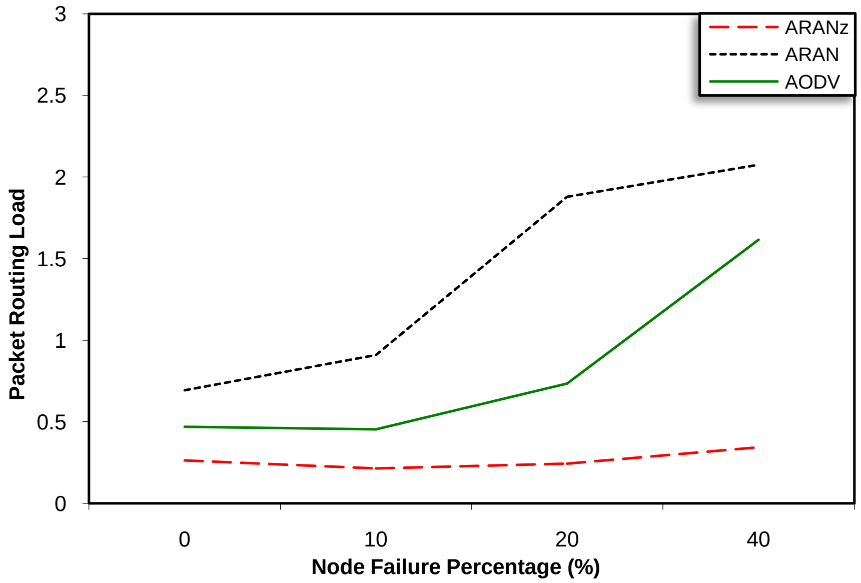


Figure 5.53: *PRL* vs. node failure

Table 5.107: *PRL* vs. node failure (t-Test)

	ARANz	*ARAN*
Mean	2.57E-01	1.39E+00
Variance	1.57E-03	3.01E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-7.42E+00	
P(T<=t) one-tail	4.04E-06	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	8.08E-06	
t Critical two-tail	2.18E+00	

	ARANz	*AODV*
Mean	2.57E-01	7.66E-01
Variance	1.57E-03	1.57E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-4.61E+00	
P(T<=t) one-tail	3.00E-04	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	6.01E-04	
t Critical two-tail	2.18E+00	

Table 5.108: *PRL* vs. node failure (chi-square Test)

ARANz	*ARAN*	*AODV*
0.99826424	0.795571117	0.877939462

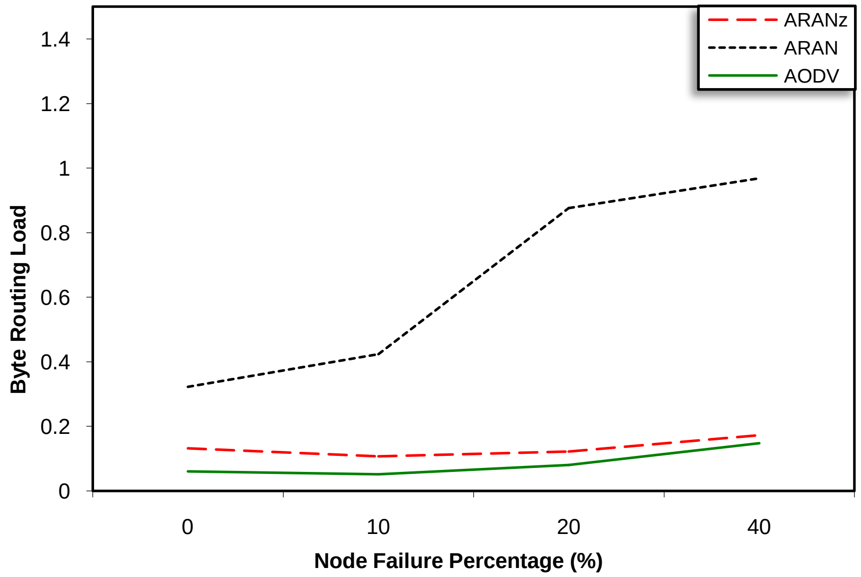


Figure 5.54: *BRL* vs. node failure

Table 5.109: *BRL* vs. node failure (t-Test)

	ARANz	*ARAN*		*ARANz*	*AODV*
Mean	1.30E-01	6.48E-01	Mean	1.30E-01	8.10E-02
Variance	3.86E-04	6.59E-02	Variance	3.86E-04	1.01E-03
Observations	4.00E+00	4.00E+00	Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00		Hypothesized Mean Difference	0.00E+00	
df	1.20E+01		df	2.00E+01	
t Stat	-7.27E+00		t Stat	4.68E+00	
P(T<=t) one-tail	4.95E-06		P(T<=t) one-tail	7.24E-05	
t Critical one-tail	1.78E+00		t Critical one-tail	1.72E+00	
P(T<=t) two-tail	9.89E-06		P(T<=t) two-tail	1.45E-04	
t Critical two-tail	2.18E+00		t Critical two-tail	2.09E+00	

Table 5.110: *BRL* vs. node failure (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999406171	0.823410659	0.995570626

Figure 5.53 and Figure 5.54 show that *PRL* and *BRL* increase for the three protocols as the node failure percentage increases. The reason behind this increase is the need to reinitiate *RDP* packets subsequent to link breaks resulting from nodes failure. *ARANz* still has the minimum *PRL* in all experiments due to sending *RDP* packets using restricted directional flooding towards the destination. On the other hand, *ARAN* protocol has the maximum (worst) *PRL*. *ARAN* has a high probability of link breakage

due to the high packet processing and authentication delay at each node. Increased number of failed nodes results in resending *RDP* packets several times in an attempt to secure a route between the communicating nodes, resulting in higher *PRL* and *BRL*. Moreover, a worse case may appear in *ARAN* protocol if the *CA* itself malfunctions. In this case, other nodes will not be able to update their certificates nor participate in constructing a route between the source and destination nodes.

Table 5.107 and Table 5.109 confirm that the differences in *PRL* and *BRL* between the three protocols are statistically significant. Results of the chi-square Test (presented in Table 5.108 and Table 5.110) show that the increase in *PNL* and *BNL* for *ARAN* is more significant than *AODV* and *ARANz*. This observation indicates that *AODV* and *ARANz* are more stable against node failure percentage.

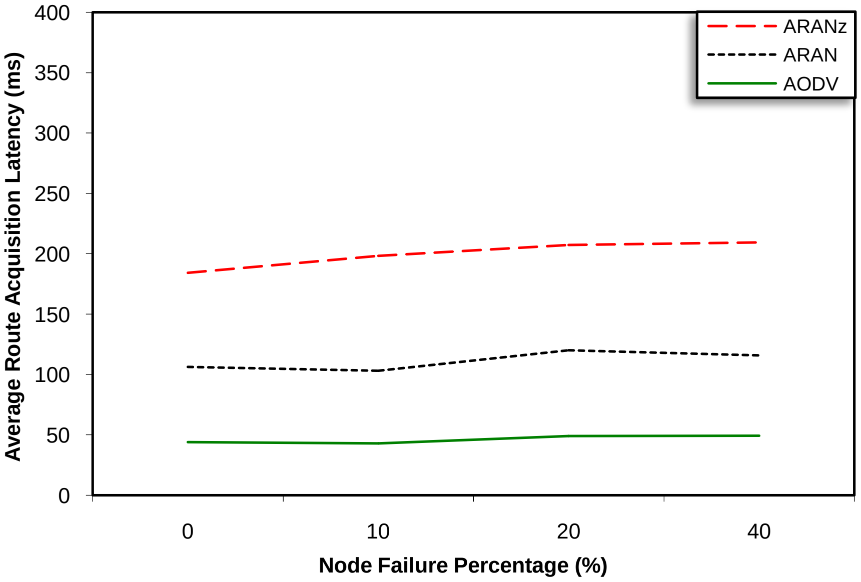


Figure 5.55: *ARAL* vs. node failure

Table 5.111: *ARAL* vs. node failure (t-Test)

	ARANz	*ARAN*
Mean	2.00E+02	1.11E+02
Variance	1.30E+02	6.34E+01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	1.27E+01	
P(T<=t) one-tail	2.66E-05	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	5.32E-05	

	ARANz	*AODV*
Mean	2.00E+02	4.63E+01
Variance	1.30E+02	1.11E+01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	2.59E+01	
P(T<=t) one-tail	6.62E-06	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	1.32E-05	

Table 5.112: *ARAL* vs. node failure (chi-square Test)

ARANz	*ARAN*	*AODV*
0.583685424	0.63522018	0.869214671

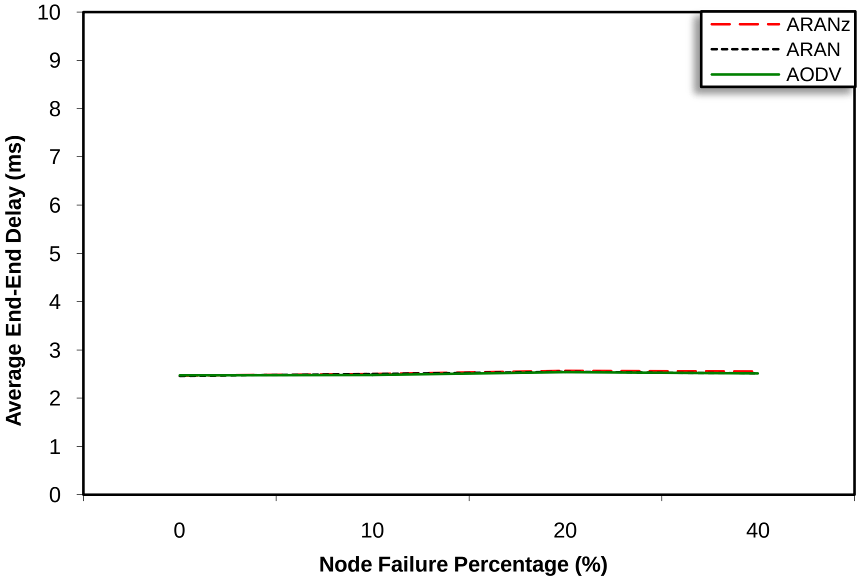


Figure 5.56: *AEED* vs. node failure

Table 5.113: *AEED* vs. node failure (t-Test)

	ARANz	*ARAN*
Mean	2.52E+00	2.51E+00
Variance	2.11E-03	1.18E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	6.23E-01	
P(T<=t) one-tail	2.78E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	5.56E-01	

	ARANz	*AODV*
Mean	2.52E+00	2.50E+00
Variance	2.11E-03	9.84E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	8.13E-01	
P(T<=t) one-tail	2.27E-01	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	4.53E-01	

Table 5.114: *AEED* vs. node failure (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999966511	0.999985854	0.999989231

By looking at Figure 5.55 and Table.112, it is clear that *ARAL* for the three protocols slightly increases as the node failure percentage increases. Higher node failure percentage means higher link break probability and the select of alternate non-optimal paths leading to higher delay in processing control packets. On the other hand, *AEED* is almost identical for the three protocols (refer to Figure 5.56, Table.113 and Table.114). The effect of *ARAL* on *AEED* of data packets is not significant since the number of the performed route discoveries and position enquiries is a small fraction of the sent data packets.

5.3.7 Malicious Node Percentage Effect

The experiments described in the previous sections compare the performance of the three protocols where all nodes in the network are considered as well-behaving ones. This section investigates the efficiency of our secure routing protocol in detecting malicious nodes. We have run many scenarios for different attacks and with a varying number of attacking nodes.

The effect of malicious node behaviour is studied on a 2km×2km network that contains 240 nodes and divided into 4 zones. These nodes move at a maximum speed of 5m/s. Five *CBR* sessions are simulated in each run, three of them are local and two are external. Simulations are run with 0%, 10%, 20% and 40% malicious nodes. The malicious nodes are selected randomly.

To study the effect of the malicious node percentage, five scenarios have been simulated. Malicious nodes simulate the following types of attacks against data and/or control packets:

1. Modification attack: Malicious nodes performing modification attack selectively reset the hop count field to 0 in the route discovery and setup packets passing through them. By assigning the hop count field to 0, a malicious node makes other nodes believe that it is only one hop away from the source or destination.
2. Black hole attack: Malicious nodes dump all data packets that they are supposed to forward.
3. Grey hole attack: Malicious nodes selectively drop data packets at random intervals.
4. Fabrication attack: Malicious nodes performing this attack periodically fabricate error packets with a specific probability.
5. Multi-attack: Malicious nodes carry out multiple attacks with a specific probability.

For these scenarios some or all the following metrics have been added, as necessary, to the set of the studied performance metrics:

1. Malicious Route Percentage (*MRP*): The fraction of the used routes that have malicious nodes within them. It is calculated as the number of routes passing through malicious nodes over the total number of routes.
2. Packet Loss Percentage (*PLP*): The fraction of data packets that are abandoned by malicious nodes without any notification.
3. Fabricated Error Packets (*FEP*): The number of error packets that are fabricated by malicious nodes.
4. Compromised Node Percentage (*CNP*): The percentage of nodes that have been considered as compromised as a result of recognizing their misbehaviour.
5. Packet Malicious Load (*PML*): The overhead packets resulted from sending misbehaviour detection packets such as *MNODE* and *CNODE* packets. The transmission at each hop along the paths is also counted in the calculation of this metric.

6. Byte Malicious Load (*BML*): Similar to the above metric, while considering the resulted overhead bytes.

The last three metrics are specified for *ARANz* protocol since neither *ARAN* nor *AODV* has a misbehaviour detection system. Some initial experiments have been carried out to choose the best values for modification threshold (*Thm*), dropping threshold (*Thd*), fabrication threshold (*Thf*) and the number of *MNODE* packets that should be received by *LCA*s to consider a specific node as compromised (*Nm*). The details of these experiments are discussed in *Appendix C* as they are somehow long. Different values for *Nm* are considered ranging from 1 to 3, also *Thm* and *Thd* are assigned values ranging from 0.3 to 0.7. Finally values of *Thf* range from 3 to 7. Results of these experiments show that a larger number of malicious nodes are discovered and identified as compromised nodes upon setting *Nm*, *Thm*, *Thd* and *Thf* to 1, 0.5, 0.5 and 3, respectively. Accordingly these are the values that are assigned for these parameters upon simulating different scenarios.

5.3.7.1 Malicious Node Percentage Effect Considering Modification Attack

The malicious behaviour simulated in this scenario is an example of the modification attack. Whenever a malicious node receives a route discovery or a route reply, it draws a random number between 0 and 1. If the drawn number is less than 0.5, then it illegally resets the hop count field to 0, pretending to be only one hop away from the source or destination. Otherwise, the control packet is forwarded without modification.

It is clear from Figure 5.57 through Figure 5.64 and the related chi-square Tests that the first eight metrics for the three protocols are not affected by malicious node percentage except *APNH* and *ARAL* for *AODV*. This fact indicates that the three protocols are able to deliver data while having acceptable routing load regardless the malicious nodes percentage. In case of *ARAN* and *ARANz*, data delivery is almost guaranteed without affecting either the time required to obtain the routes or the number of hops in the selected paths. In *AODV*, however, *APNH* and *ARAL* slightly increase with increasing malicious node percentage since malicious nodes can exploit *AODV* so that non-shortest paths are selected, while such exploitation is not possible with *ARAN* and *ARANz*.

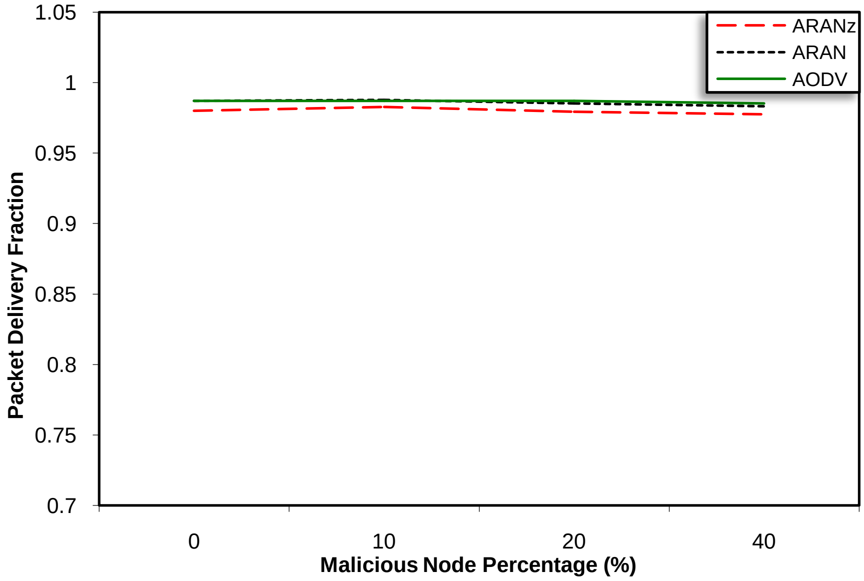


Figure 5.57: *PDF* vs. malicious node percentage considering modification attack

Table 5.115: *PDF* vs. malicious node percentage considering modification attack (t-Test)

	ARANz	***ARAN***
Mean	9.80E-01	9.86E-01
Variance	4.61E-06	4.17E-06
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-3.96E+00	
P(T<=t) one-tail	3.71E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	7.42E-03	
t Critical two-tail	2.45E+00	

	ARANz	***AODV***
Mean	9.80E-01	9.87E-01
Variance	4.61E-06	1.00E-06
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	-5.70E+00	
P(T<=t) one-tail	2.34E-01	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	4.68E-03	
t Critical two-tail	2.78E+00	

Table 5.116: *PDF* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999985925	0.999987993	0.99999859

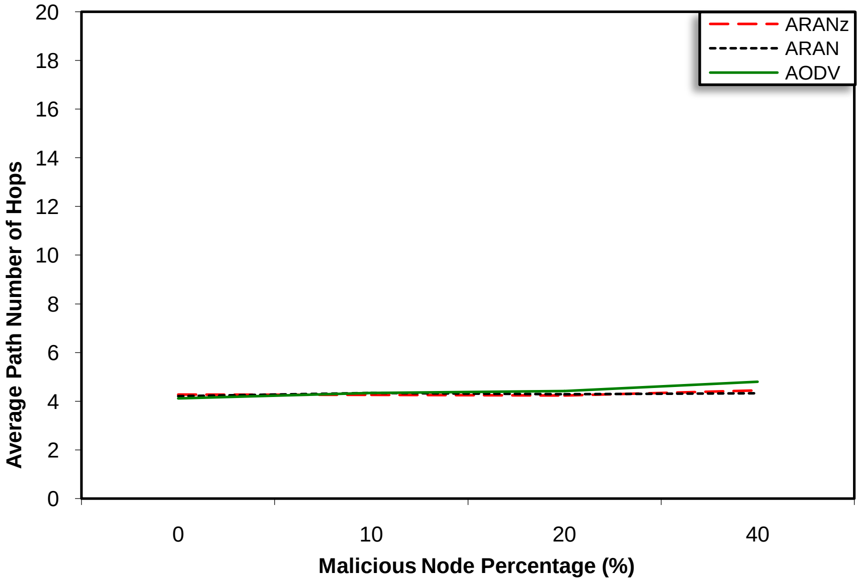


Figure 5.58: *APNH* vs. malicious node percentage considering modification attack

Table 5.117: *APNH* vs. malicious node percentage considering modification attack (t-Test)

	ARANz	*ARAN*
Mean	4.31E+00	4.29E+00
Variance	9.46E-03	3.69E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	3.09E-01	
P(T<=t) one-tail	3.85E-01	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	7.70E-01	
t Critical two-tail	2.57E+00	

	ARANz	*AODV*
Mean	4.31E+00	4.42E+00
Variance	9.46E-03	8.08E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	-7.57E-01	
P(T<=t) one-tail	2.45E-01	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	4.91E-01	
t Critical two-tail	2.78E+00	

Table 5.118: *APNH* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999858017	0.99996512	0.986639904

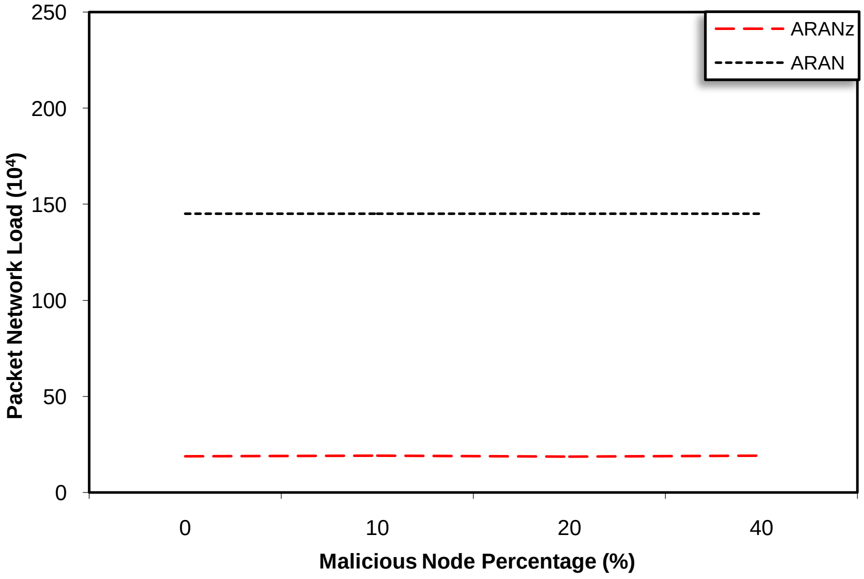


Figure 5.59: *PNL* vs. malicious node percentage considering modification attack

Table 5.119: *PNL* vs. malicious node percentage considering modification attack (t-Test)

	ARANz	***ARAN***
Mean	1.89E+01	3.47E+01
Variance	7.39E-02	8.59E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	0.00E+00
df	3.00E+00	3.00E+00
t Stat	-9.25E+02	-6.38E+02
P(T<=t) one-tail	1.40E-09	4.25E-09
t Critical one-tail	2.35E+00	2.35E+00
P(T<=t) two-tail	2.79E-09	8.49E-09
t Critical two-tail	3.18E+00	3.18E+00

Table 5.120: *PNL* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz	***ARAN***
0.999664584	0.999999989

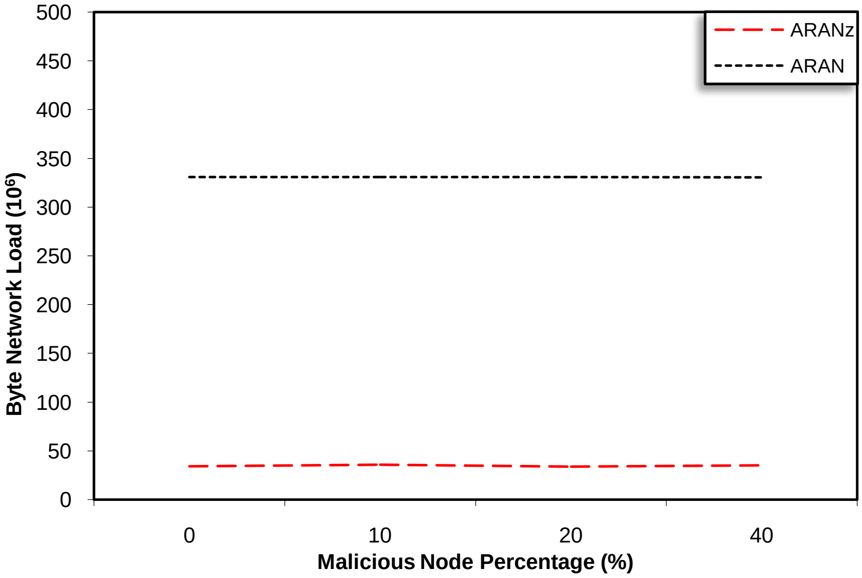


Figure 5.60: *BNL* vs. malicious node percentage considering modification attack

Table 5.121: *BNL* vs. malicious node percentage considering modification attack (t-Test)

	ARANz	***ARAN***
Mean	3.47E+01	3.31E+02
Variance	8.59E-01	2.95E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-6.38E+02	
P(T<=t) one-tail	4.25E-09	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	8.49E-09	
t Critical two-tail	3.18E+00	

Table 5.122: *BNL* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz	***ARAN***
0.994730164	0.999999963

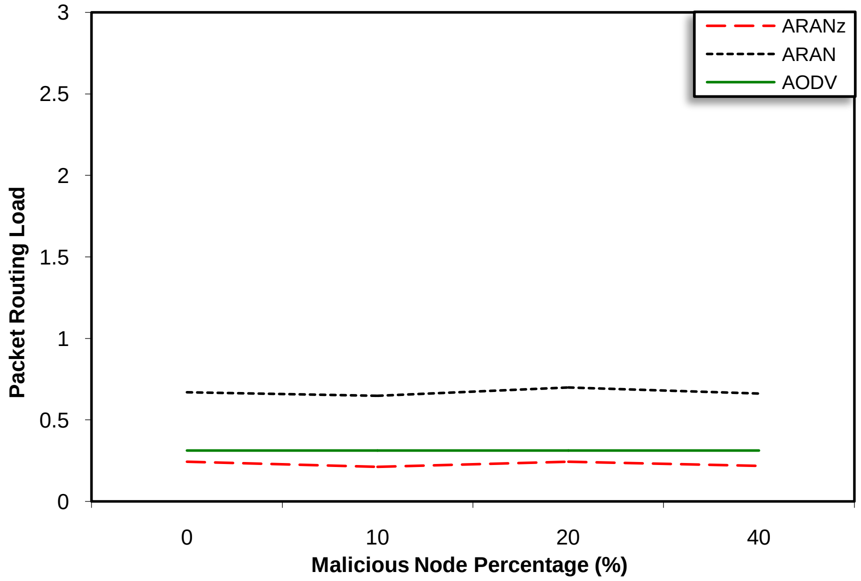


Figure 5.61: *PRL* vs. malicious node percentage considering modification attack

Table 5.123: *PRL* vs. malicious node percentage considering modification attack (t-Test)

	ARANz	*ARAN*
Mean	2.29E-01	6.70E-01
Variance	1.06E-04	2.19E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	2.10E+01	
t Stat	-8.81E+01	
P(T<=t) one-tail	9.21E-29	
t Critical one-tail	1.72E+00	
P(T<=t) two-tail	1.84E-28	
t Critical two-tail	2.08E+00	

	ARANz	*AODV*
Mean	2.29E-01	3.13E-01
Variance	1.06E-04	1.34E-32
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-2.93E+01	
P(T<=t) one-tail	7.86E-13	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	1.57E-12	
t Critical two-tail	2.18E+00	

Table 5.124: *PRL* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999934425	0.9999749	1

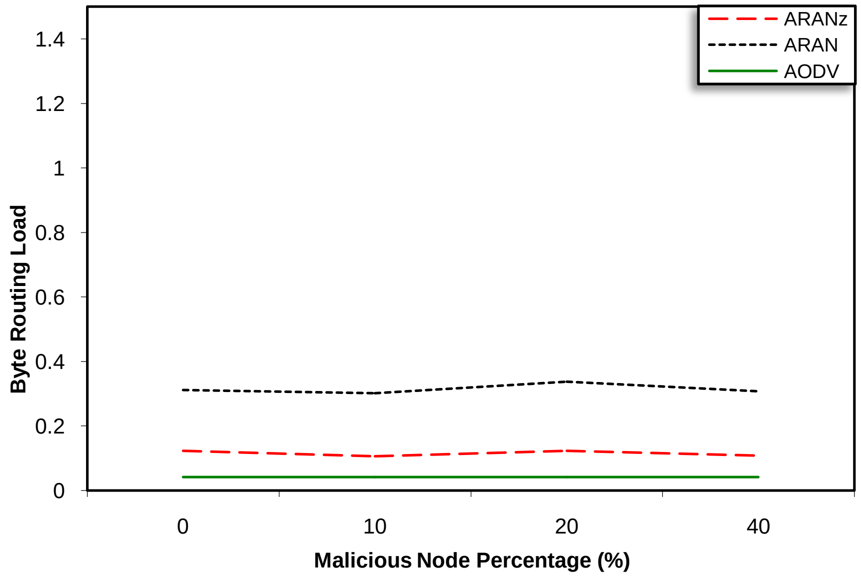


Figure 5.62: *BRL* vs. malicious node percentage considering modification attack

Table 5.125: *BRL* vs. malicious node percentage considering modification attack (t-Test)

	ARANz	*ARAN*
Mean	1.15E-01	3.16E-01
Variance	2.93E-05	1.18E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.80E+01	
t Stat	-5.95E+01	
P(T<=t) one-tail	1.99E-22	
t Critical one-tail	1.73E+00	
P(T<=t) two-tail	3.98E-22	
t Critical two-tail	2.10E+00	

	ARANz	*AODV*
Mean	1.15E-01	4.20E-02
Variance	2.93E-05	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	4.89E+01	
P(T<=t) one-tail	1.77E-15	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	3.55E-15	
t Critical two-tail	2.18E+00	

Table 5.126: *BRL* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999972614	0.999970249	1

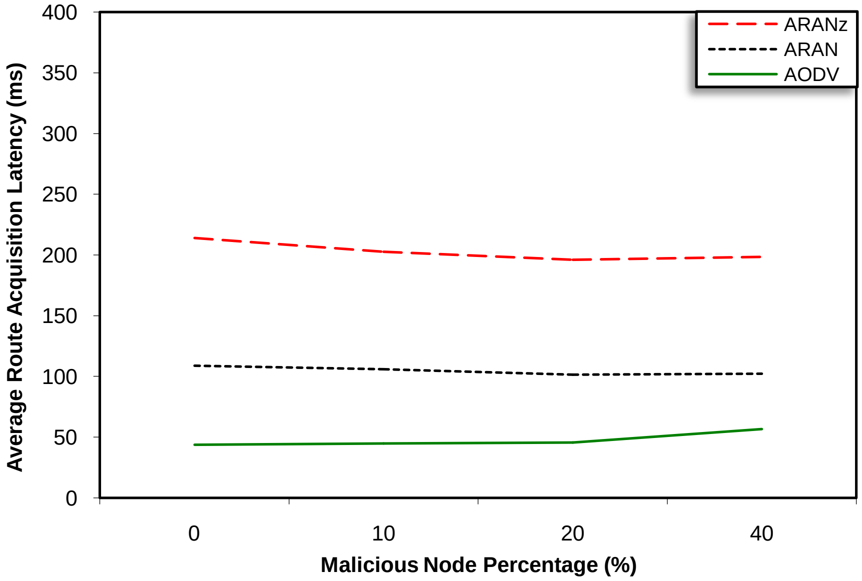


Figure 5.63: *ARAL* vs. malicious node percentage considering modification attack

Table 5.127: *ARAL* vs. malicious node percentage considering modification attack (t-Test)

	ARANz	***ARAN***
Mean	2.01E+02	1.03E+02
Variance	8.46E+01	2.55E+01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	1.87E+01	
P(T<=t) one-tail	4.02E-06	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	8.03E-06	
t Critical two-tail	2.57E+00	

	ARANz	***AODV***
Mean	2.01E+02	4.51E+01
Variance	8.46E+01	1.67E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	3.37E+01	
P(T<=t) one-tail	2.88E-05	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	5.76E-05	
t Critical two-tail	3.18E+00	

Table 5.128: *ARAL* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.863478409	0.990501927	0.738756951

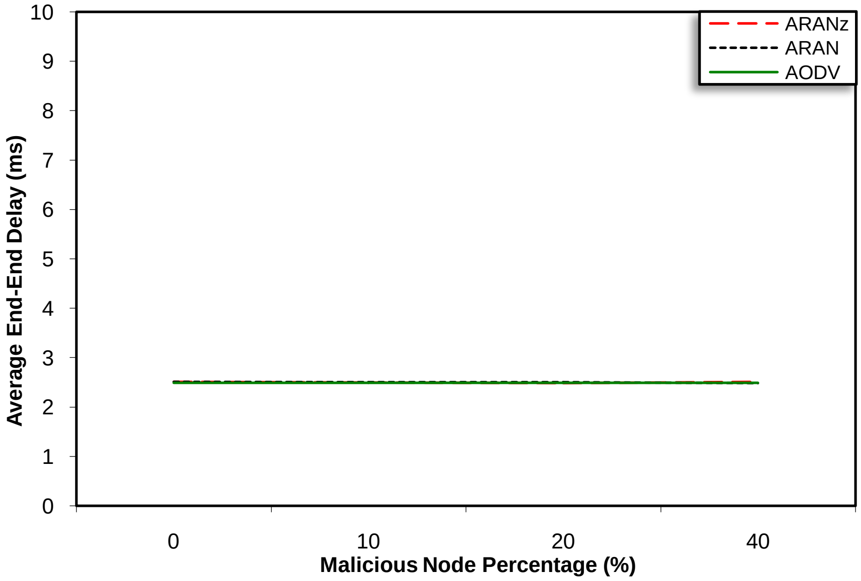


Figure 5.64: *AEED* vs. malicious node percentage considering modification attack

Table 5.129: *AEED* vs. malicious node percentage considering modification attack (t-Test)

	ARANz	***ARAN***
Mean	2.50E+00	2.50E+00
Variance	1.23E-04	1.20E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	1.33E-01	
P(T<=t) one-tail	4.49E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	8.99E-01	
t Critical two-tail	2.45E+00	

	ARANz	***AODV***
Mean	2.50E+00	2.49E+00
Variance	1.23E-04	0.00E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	1.84E+00	
P(T<=t) one-tail	8.13E-02	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.63E-01	
t Critical two-tail	3.18E+00	

Table 5.130: *AEED* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999999523	0.999999538	0.999999518

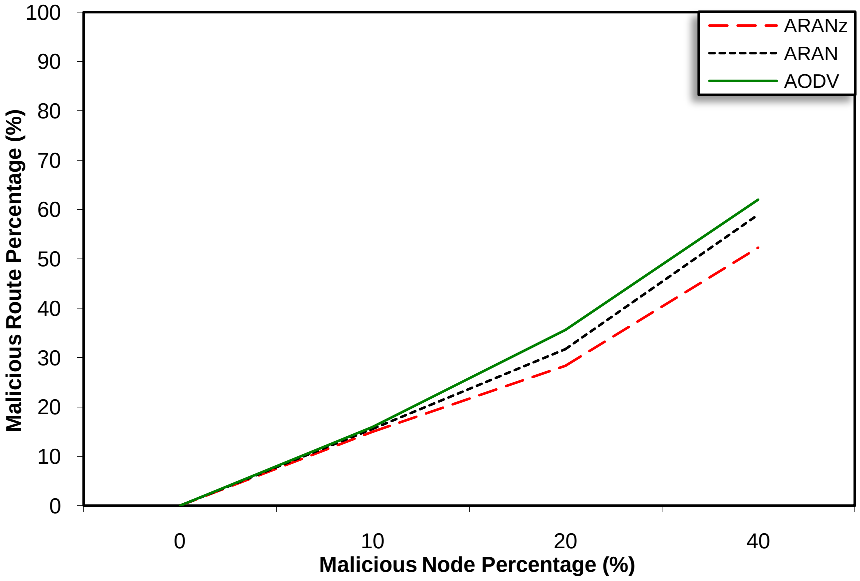


Figure 5.65: *MRP* vs. malicious node percentage considering modification attack

Table 5.131: *MRP* vs. malicious node percentage considering modification attack (t-Test)

	ARANz	*ARAN*
Mean	2.39E+01	2.65E+01
Variance	4.91E+02	6.34E+02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-1.59E-01	
P(T<=t) one-tail	4.39E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	8.79E-01	
t Critical two-tail	2.45E+00	

	ARANz	*AODV*
Mean	2.39E+01	2.84E+01
Variance	4.91E+02	7.14E+02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-2.59E-01	
P(T<=t) one-tail	4.02E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	8.04E-01	
t Critical two-tail	2.45E+00	

Table 5.132: *MRP* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz	*ARAN*	*AODV*
2.57866E-13	1.88579E-15	2.86445E-16

Figure 5.65 and Table 5.132 show that *MRP* significantly increases for the three protocols when the malicious node percentage is increased. However, the figure and Table 5.131 show that upon using *AODV*, a larger fraction of routes have malicious nodes within them. When the malicious node resets the hop count field to 0, it forces the

selected routes to pass through itself because *AODV* selects the shorter paths. *ARAN* and *ARANz*, on the other hand, cannot be exploited in this fashion. The selected route could pass through a malicious node but not forced to do this.

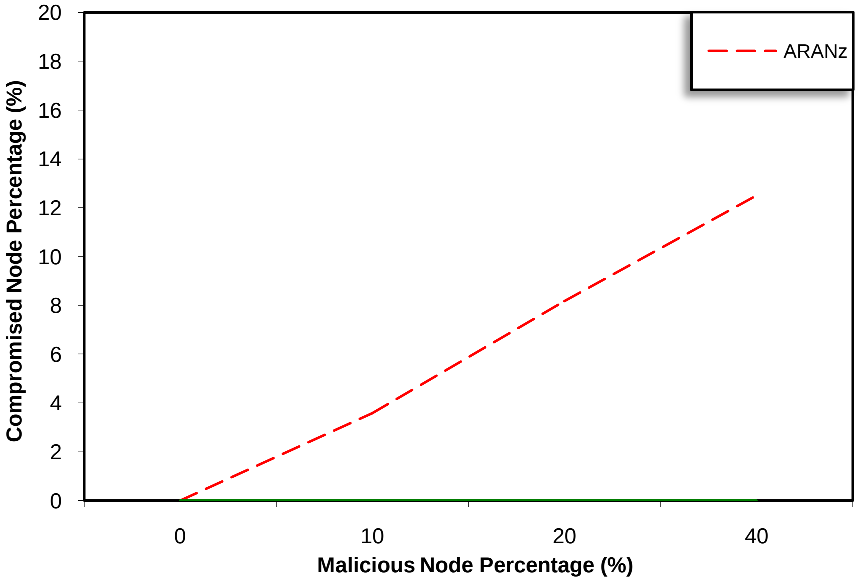


Figure 5.66: *CNP* vs. malicious node percentage considering modification attack

Table 5.133: *CNP* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz
0.00214929

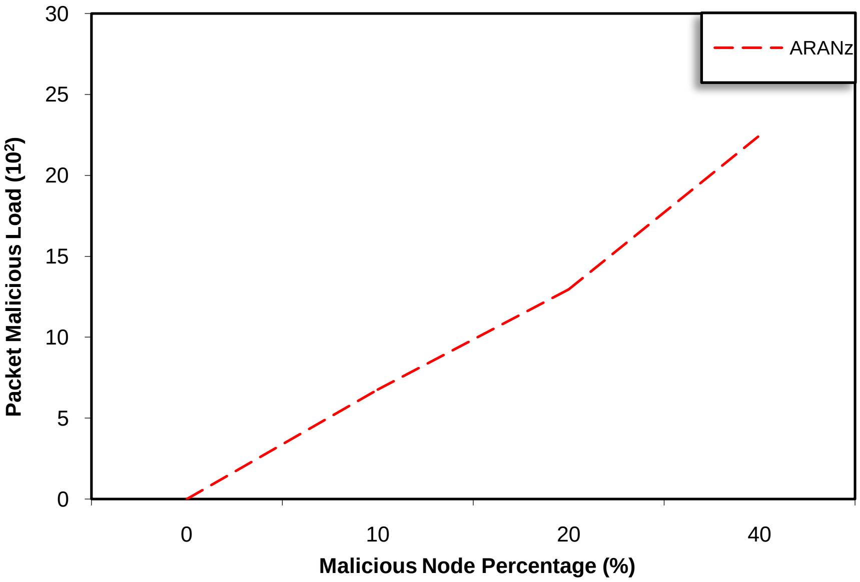


Figure 5.67: *PML* vs. malicious node percentage considering modification attack

Table 5.134: *PML* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz
9.864E-06

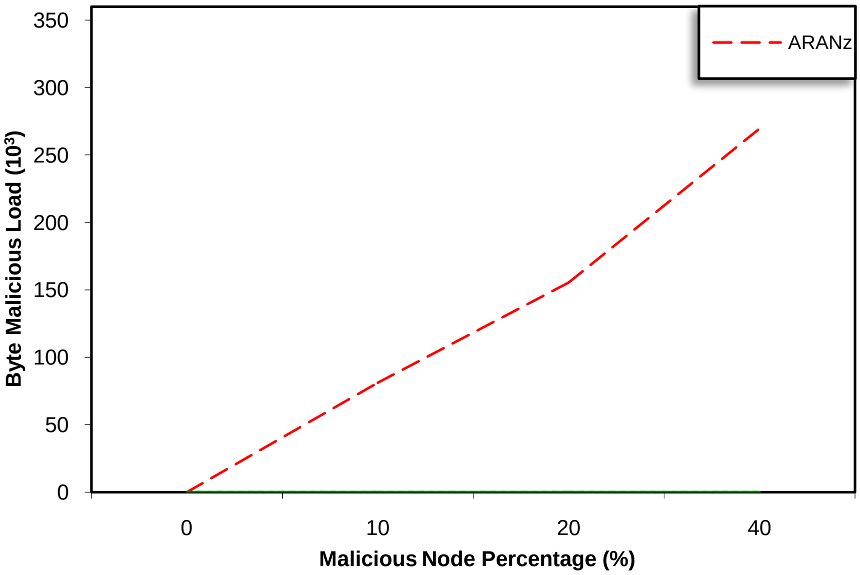


Figure 5.68: *BML* vs. malicious node percentage considering modification attack

Table 5.135: *BML* vs. malicious node percentage considering modification attack (chi-square Test)

ARANz
3.8184E-67

Referring to Figure 5.66 through Figure 5.68 and the related chi-square Tests (presented in Table 5.133 through Table 5.135), it is apparent that *CNP*, *PML* and *BML* for *ARANz* increase as the malicious node percentage increases. This suggests that *ARANz* is efficient in identifying and isolating modification attacks.

5.3.7.2 Malicious Node Percentage Effect Considering Black Hole Attack

The black hole attack is considered in this scenario. In this attack, malicious nodes dump all data packets that they receive.

From Figure 5.70 through Figure 5.72 and Figure 5.75 through Figure 5.76, it is obvious that the *APNH*, *PNL*, *BNL*, *ARAL* and *AEED* for the three protocols are roughly not affected by the simulated percentage of malicious nodes. The almost constant *APNH*, *ARAL* and *AEED* indicate that the three protocols are able to discover the shortest paths without affecting the time required to obtain the routes even with increasing the malicious node percentage. *PNL* and *BNL* for *ARAN* and *ARANz* protocols have almost fixed values for the reason that packets initiated to update nodes certificates and maintaining network structure are sent regardless of the number of nodes dropping data packets.

It is noticeable from Figure 5.69 that *PDF* decreases for the three protocols upon increasing the malicious node percentage. The decrease in *PDF* is justifiable as the malicious nodes in this scenario perform the black hole attack, they drop the data packets they receive. However, the figure and Table 5.137 assure that the decrease in *PDF* is slower and insignificant in *ARANz*, suggesting that *ARANz* is efficient in identifying and isolating the black hole attackers.

By looking at Figure 5.73, Figure 5.74 and the related chi-square Tests, we can observe that *PRL* and *BRL* for *AODV* and *ARAN* are approximately not affected by malicious node percentage. For *ARANz*, these two metrics slightly increase as the malicious node percentage increases since detecting malicious nodes in *ARANz* causes reinitiating *RDP* packets in a try to find another secure route, i.e. slightly increasing the routing overhead.

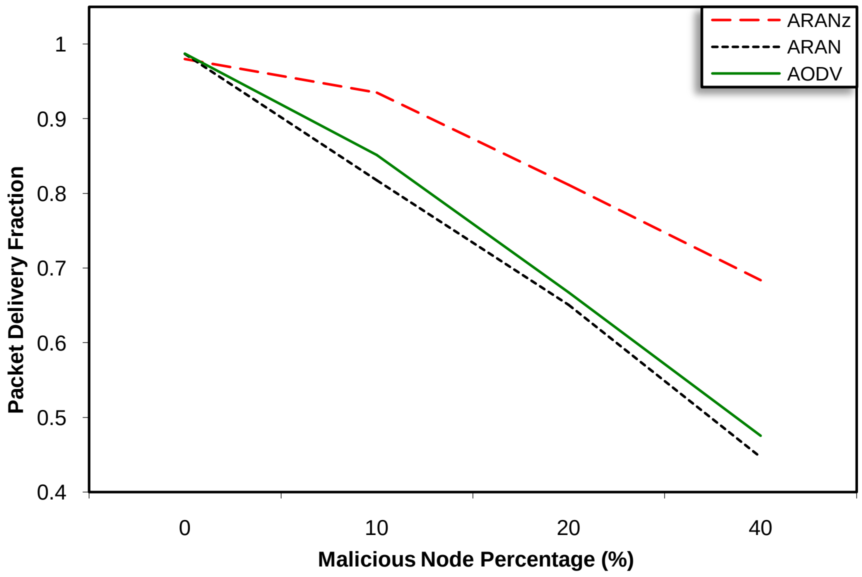


Figure 5.69: *PDF* vs. malicious node percentage considering black hole attack

Table 5.136: *PDF* vs. malicious node percentage considering black hole attack (t-Test)

	ARANz	*ARAN*
Mean	8.53E-01	7.25E-01
Variance	1.77E-02	5.34E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	9.53E-01	
P(T<=t) one-tail	1.92E-02	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	3.84E-01	
t Critical two-tail	2.57E+00	

	ARANz	*AODV*
Mean	8.53E-01	7.45E-01
Variance	1.77E-02	4.95E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	8.27E-01	
P(T<=t) one-tail	2.23E-02	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	4.46E-01	
t Critical two-tail	2.57E+00	

Table 5.137: *PDF* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.100628435	6.28226E-05	0.000175365

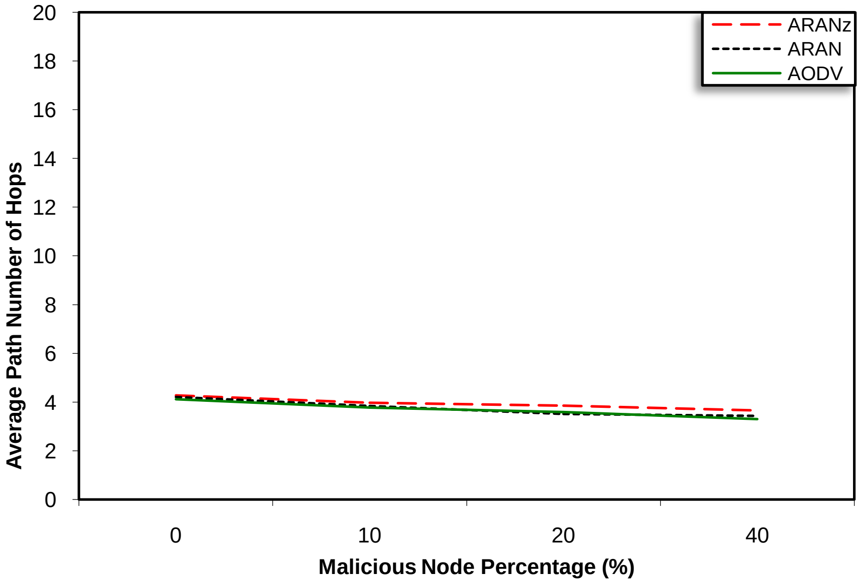


Figure 5.70: *APNH* vs. malicious node percentage considering black hole attack

Table 5.138: *APNH* vs. malicious node percentage considering black hole attack (t-Test)

	ARANz	***ARAN***
Mean	3.94E+00	3.75E+00
Variance	6.90E-02	1.26E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	8.65E-01	
P(T<=t) one-tail	2.10E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	4.20E-01	
t Critical two-tail	2.45E+00	

	ARANz	***AODV***
Mean	3.94E+00	3.70E+00
Variance	6.90E-02	1.17E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	1.12E+00	
P(T<=t) one-tail	1.54E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	3.07E-01	
t Critical two-tail	2.45E+00	

Table 5.139: *APNH* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.996844547	0.991706018	0.992406814

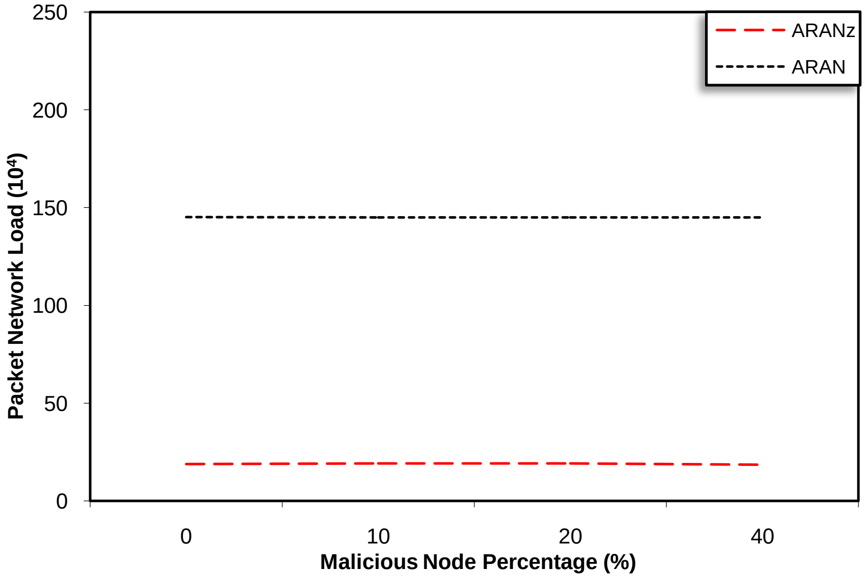


Figure 5.71: *PNL* vs. malicious node percentage considering black hole attack

Table 5.140: *PNL* vs. malicious node percentage considering black hole attack (t-Test)

	ARANz	***ARAN***
Mean	1.89E+01	1.45E+02
Variance	7.83E-02	2.23E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-8.89E+02	
P(T<=t) one-tail	1.57E-09	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.14E-09	
t Critical two-tail	3.18E+00	

Table 5.141: *PNL* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz	***ARAN***
0.99963338	0.999999917

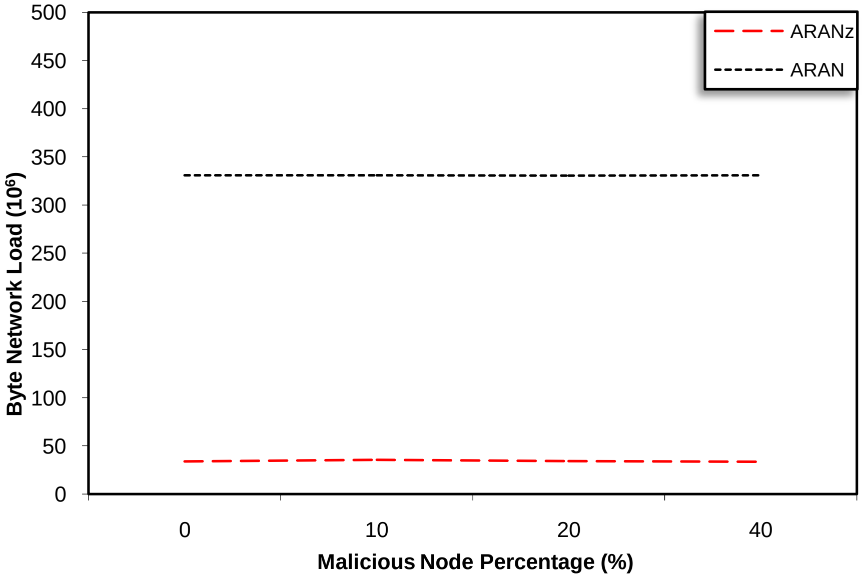


Figure 5.72: *BNL* vs. malicious node percentage considering black hole attack

Table 5.142: *BNL* vs. malicious node percentage considering black hole attack (t-Test)

	ARANz	***ARAN***
Mean	3.44E+01	3.31E+02
Variance	6.54E-01	1.16E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-7.27E+02	
P(T<=t) one-tail	2.87E-09	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	5.75E-09	
t Critical two-tail	3.18E+00	

Table 5.143: *BNL* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz	***ARAN***
0.996438447	0.999999713

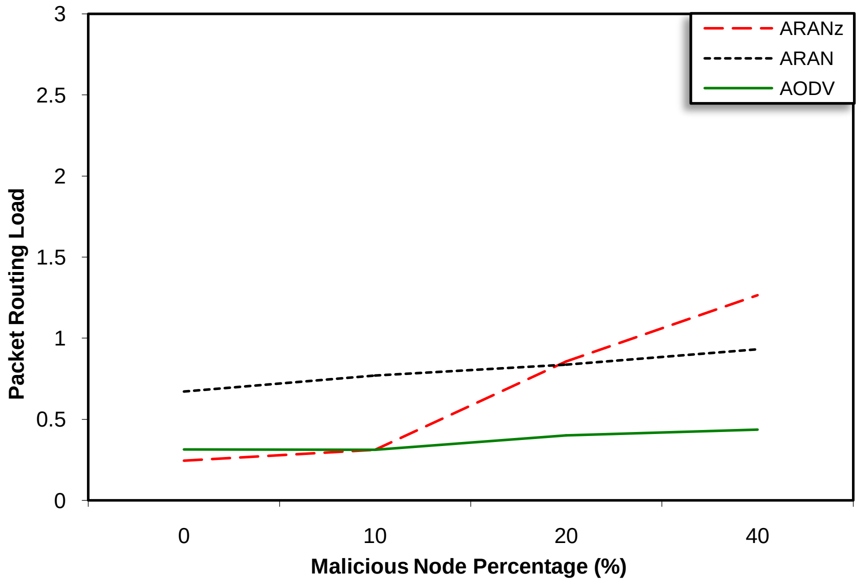


Figure 5.73: *PRL* vs. malicious node percentage considering black hole attack

Table 5.144: *PRL* vs. malicious node percentage considering black hole attack (t-Test)

	ARANz	***ARAN***
Mean	6.50E-01	8.02E-01
Variance	1.41E-01	6.66E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.30E+01	
t Stat	-1.43E+00	
P(T<=t) one-tail	8.79E-02	
t Critical one-tail	1.77E+00	
P(T<=t) two-tail	1.76E-01	
t Critical two-tail	2.16E+00	

	ARANz	***AODV***
Mean	6.50E-01	3.63E-01
Variance	1.41E-01	2.50E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	2.73E+00	
P(T<=t) one-tail	9.21E-03	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	1.84E-02	
t Critical two-tail	2.18E+00	

Table 5.145: *PRL* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.791302557	0.997457337	0.99846509

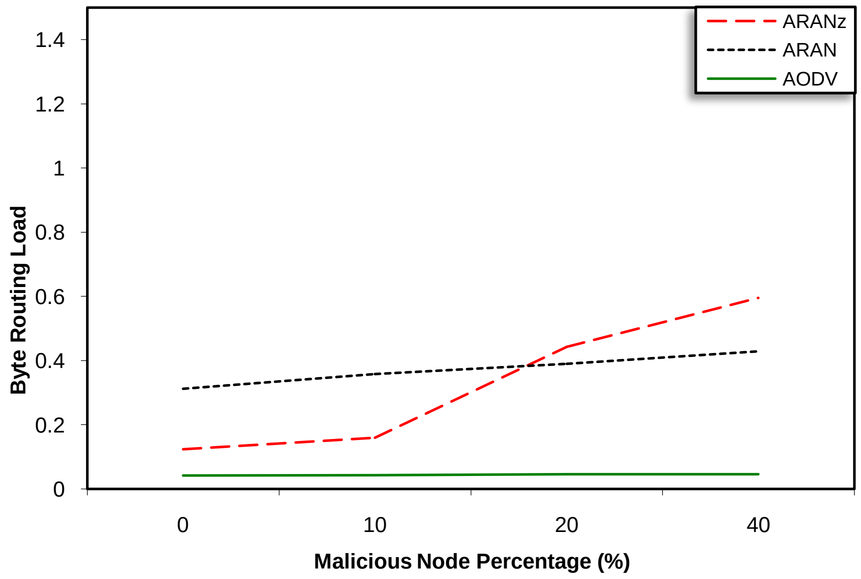


Figure 5.74: *BRL* vs. malicious node percentage considering black hole attack

Table 5.146: *BRL* vs. malicious node percentage considering black hole attack (t-Test)

	ARANz	***ARAN***
Mean	3.24E-01	3.73E-01
Variance	3.20E-02	1.35E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.30E+01	
t Stat	-9.71E-01	
P(T<=t) one-tail	1.75E-01	
t Critical one-tail	1.77E+00	
P(T<=t) two-tail	3.49E-01	
t Critical two-tail	2.16E+00	

	ARANz	***AODV***
Mean	3.24E-01	4.39E-02
Variance	3.20E-02	2.47E-06
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	5.64E+00	
P(T<=t) one-tail	5.47E-05	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	1.09E-04	
t Critical two-tail	2.18E+00	

Table 5.147: *BRL* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.925704688	0.999264521	0.999998863

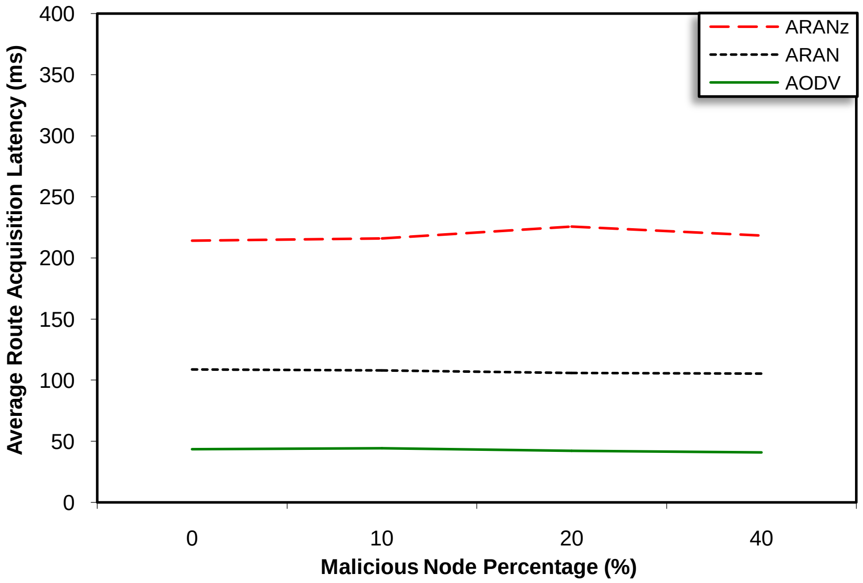


Figure 5.75: *ARAL* vs. malicious node percentage considering black hole attack

Table 5.148: *ARAL* vs. malicious node percentage considering black hole attack (t-Test)

	ARANz	*ARAN*
Mean	2.18E+02	1.07E+02
Variance	2.57E+01	2.84E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	4.17E+01	
P(T<=t) one-tail	9.86E-07	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	1.97E-06	
t Critical two-tail	2.78E+00	

	ARANz	*AODV*
Mean	2.18E+02	4.27E+01
Variance	2.57E+01	2.23E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	6.65E+01	
P(T<=t) one-tail	1.53E-07	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	3.06E-07	
t Critical two-tail	2.78E+00	

Table 5.149: *ARAL* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.949757983	0.994177043	0.984288903

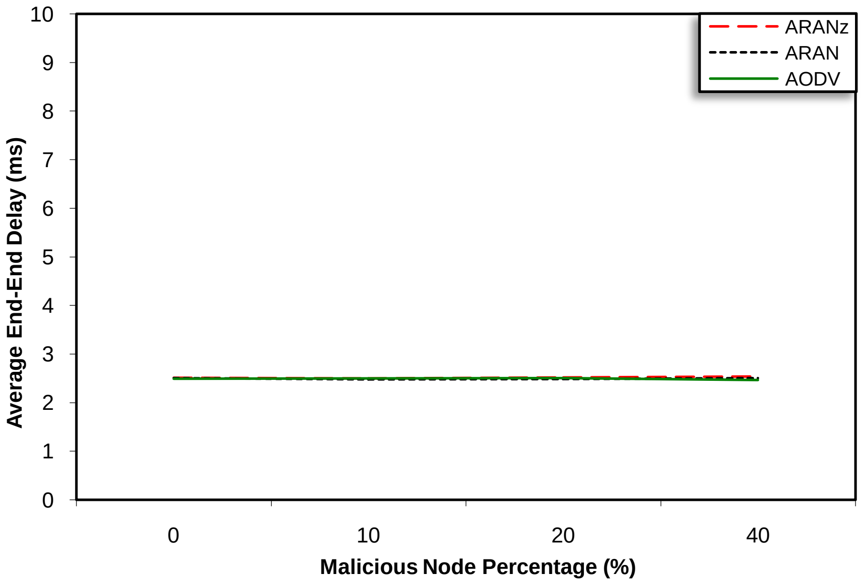


Figure 5.76: *AEED* vs. malicious node percentage considering black hole attack

Table 5.150: *AEED* vs. malicious node percentage considering black hole attack (t-Test)

	ARANz	*ARAN*
Mean	2.51E+00	2.49E+00
Variance	2.61E-04	1.72E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	2.00E+00	
P(T<=t) one-tail	4.61E-02	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	9.22E-02	
t Critical two-tail	2.45E+00	

	ARANz	*AODV*
Mean	2.51E+00	2.49E+00
Variance	2.61E-04	3.48E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	2.25E+00	
P(T<=t) one-tail	3.27E-02	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	6.53E-02	
t Critical two-tail	2.45E+00	

Table 5.151: *AEED* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999998537	0.999999209	0.999997717

Figure 5.77 and Table 5.153 show that *PLP* increases for the three protocols as the malicious node percentage increases. However, upon using *ARANz* the increase in *PLP* is much slower. This suggests that *ARANz* is efficient in detecting and isolating black hole attackers and justifies the increase in *CNP*, *PML* and *BML* for *ARANz* with

increasing the malicious node percentage (refer to Figure 5.78 through Figure 5.80 and Table 5.154 through Table 5.156).

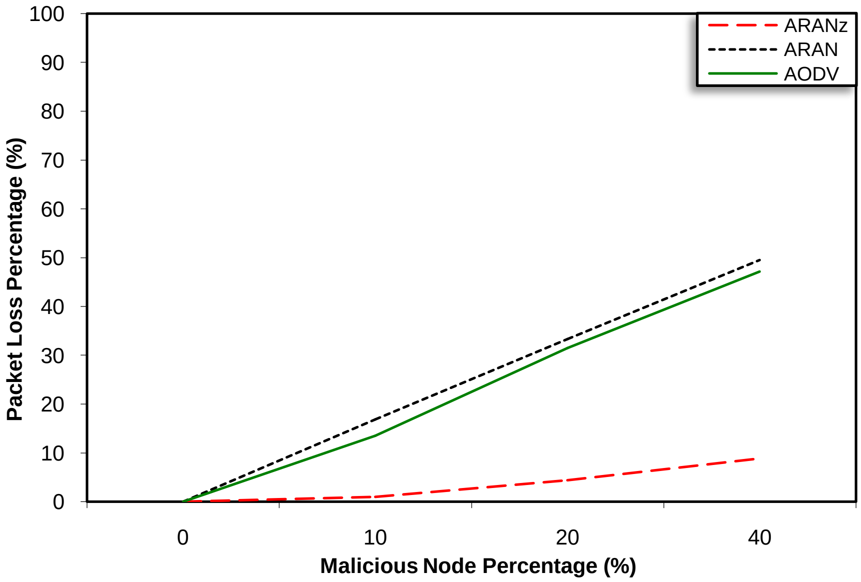


Figure 5.77: *PLP* vs. malicious node percentage considering black hole attack

Table 5.152: *PLP* vs. malicious node percentage considering black hole attack (t-Test)

	ARANz	*ARAN*
Mean	3.54E+00	2.49E+01
Variance	1.59E+01	4.54E+02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.97E+00	
P(T<=t) one-tail	7.15E-02	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.43E-01	
t Critical two-tail	3.18E+00	

	ARANz	*AODV*
Mean	3.54E+00	2.30E+01
Variance	1.59E+01	4.24E+02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.86E+00	
P(T<=t) one-tail	8.01E-02	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.60E-01	
t Critical two-tail	3.18E+00	

Table 5.153: *PLP* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.003642852	8.30098E-12	6.02649E-12

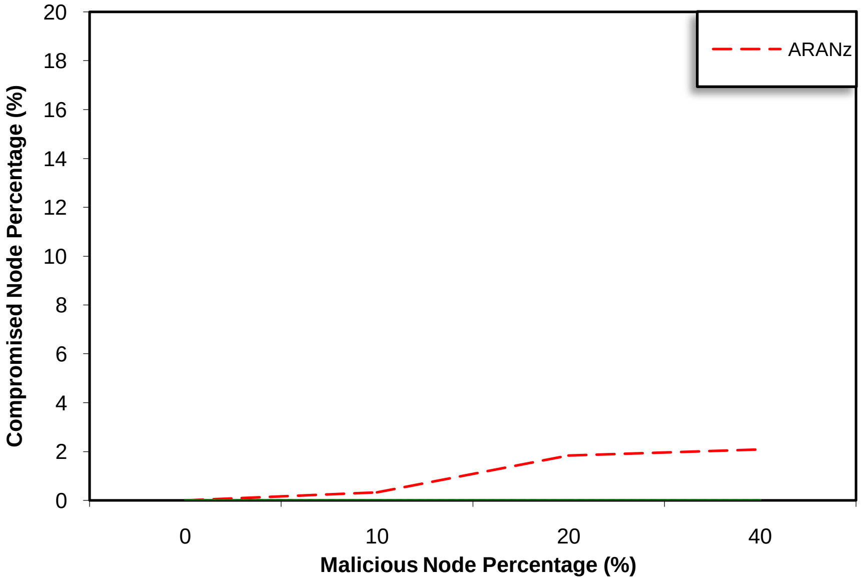


Figure 5.78: *CNP* vs. malicious node percentage considering black hole attack

Table 5.154: *CNP* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz
0.376018258

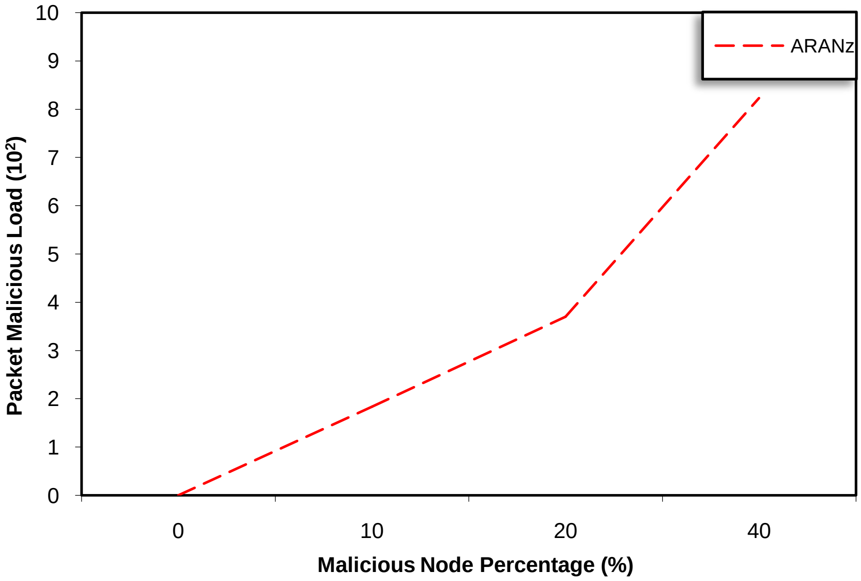


Figure 5.79: *PML* vs. malicious node percentage considering black hole attack

Table 5.155: *PML* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz
0.0125204

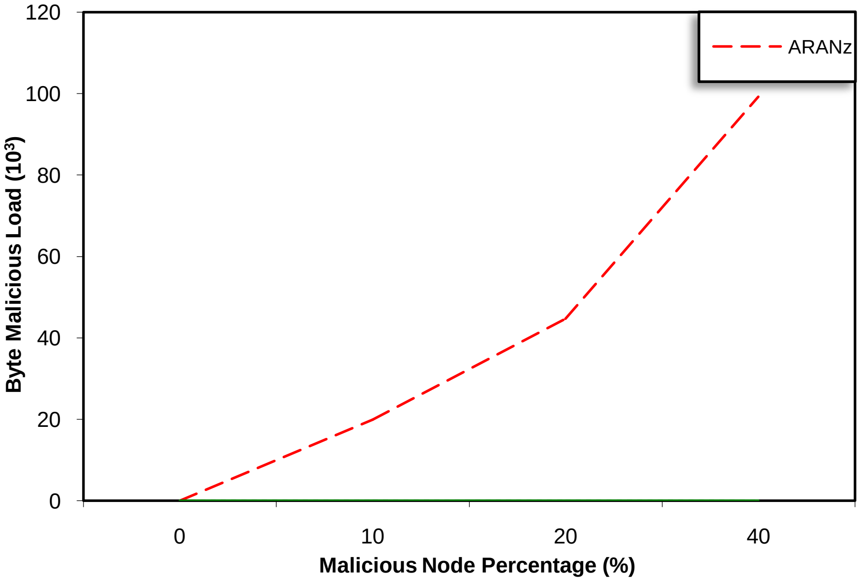


Figure 5.80: *BML* vs. malicious node percentage considering black hole attack

Table 5.156: *BML* vs. malicious node percentage considering black hole attack (chi-square Test)

ARANz
4.72465E-29

5.3.7.3 Malicious Node Percentage Effect Considering Grey Hole Attack

In this scenario, the grey hole attack is considered. In grey hole attack, malicious nodes selectively drop data packets at random intervals. To simulate this attack, whenever a malicious node receives a data packet, it draws a random number between 0 and 1. If the number is less than 0.5, then the node drops the data packet. Otherwise, the data packet is forwarded to the successor node.

As in the previous scenario, Figure 5.82 through Figure 5.84 and Figure 5.87 through Figure 5.88 show that *APNH*, *PNL*, *BNL*, *ARAL* and *AEED* for the three protocols are roughly not affected by the percentage of malicious nodes existing in the network. The related chi-square Tests assure this result.

Figure 5.81 shows that *PDF* decreases for the three protocols as the number of malicious nodes dropping data packets is increased. However, Table 5.158 shows that the decrease in *PDF* is slower in *ARANz* implying that *ARANz* is efficient in detecting and isolating grey hole attackers.

Figure 5.85, Figure 5.86 and related chi-square Tests show that *PRL* and *BRL* for *AODV* and *ARAN* are not affected by malicious node percentage. On the other hand, these two metrics for *ARANz* slightly increase with increasing the malicious node percentage. This increase in *PRL* and *BRL* is due to reinitiating *RDP* packets as a result of detecting malicious nodes.

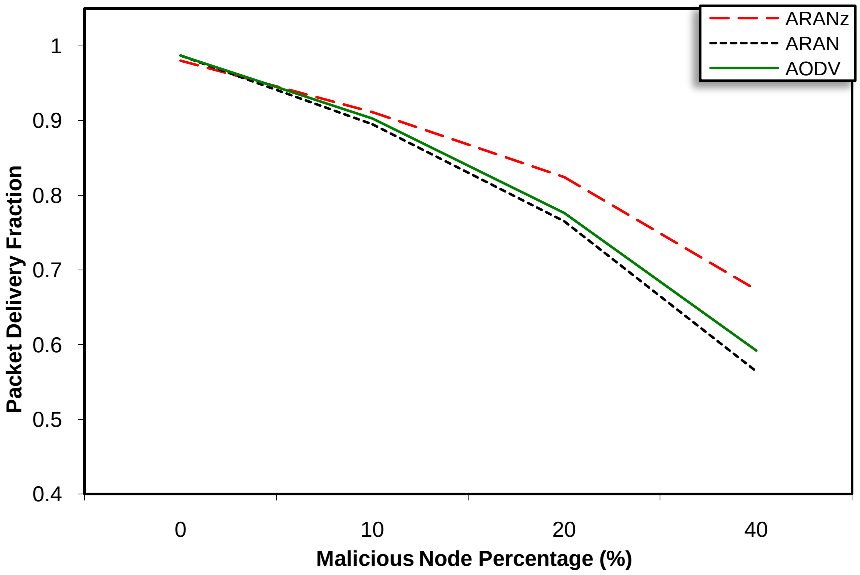


Figure 5.81: *PDF* vs. malicious node percentage considering grey hole attack

Table 5.157: *PDF* vs. malicious node percentage considering grey hole attack (t-Test)

	ARANz	*ARAN*
Mean	8.47E-01	8.03E-01
Variance	1.75E-02	3.36E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	3.94E-01	
P(T<=t) one-tail	3.55E-02	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	7.10E-01	
t Critical two-tail	2.57E+00	

	ARANz	*AODV*
Mean	8.47E-01	8.14E-01
Variance	1.75E-02	2.95E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	3.02E-01	
P(T<=t) one-tail	3.86E-02	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	7.73E-01	
t Critical two-tail	2.45E+00	

Table 5.158: *PDF* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.102625481	0.005657606	0.012408533

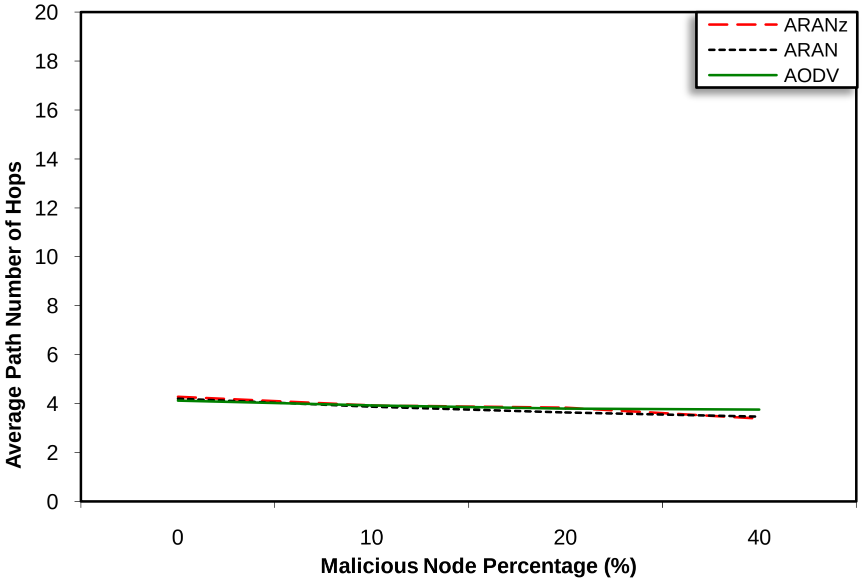


Figure 5.82: *APNH* vs. malicious node percentage considering grey hole attack

Table 5.159: *APNH* vs. malicious node percentage considering grey hole attack (t-Test)

	ARANz	*ARAN*
Mean	3.86E+00	3.80E+00
Variance	1.31E-01	1.02E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	2.52E-01	
P(T<=t) one-tail	4.05E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	8.09E-01	
t Critical two-tail	2.45E+00	

	ARANz	*AODV*
Mean	3.86E+00	3.90E+00
Variance	1.31E-01	2.62E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	-2.34E-01	
P(T<=t) one-tail	4.13E-01	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	8.26E-01	
t Critical two-tail	2.78E+00	

Table 5.160: *APNH* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.991610316	0.994045487	0.999244052

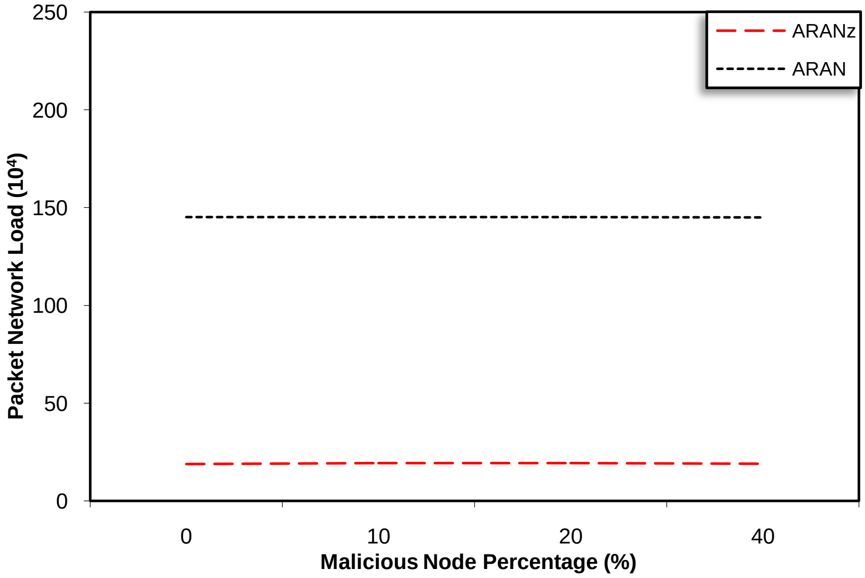


Figure 5.83: *PNL* vs. malicious node percentage considering grey hole attack

Table 5.161: *PNL* vs. malicious node percentage considering grey hole attack (t-Test)

	ARANz	***ARAN***
Mean	1.91E+01	1.45E+02
Variance	5.33E-02	7.18E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.08E+03	
P(T<=t) one-tail	8.65E-10	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.73E-09	
t Critical two-tail	3.18E+00	

Table 5.162: *PNL* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz	***ARAN***
0.99979736	0.999999985

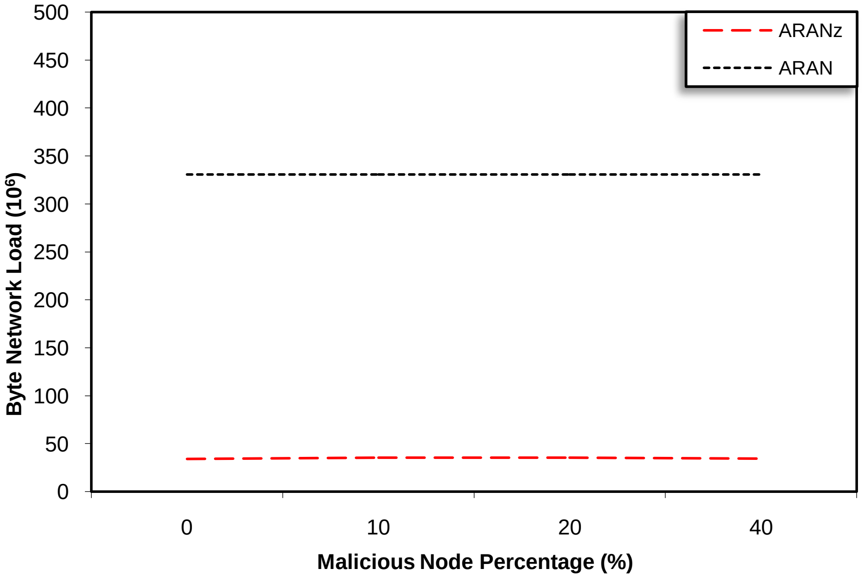


Figure 5.84: *BNL* vs. malicious node percentage considering grey hole attack

Table 5.163: *BNL* vs. malicious node percentage considering grey hole attack (t-Test)

	ARANz	***ARAN***
Mean	3.47E+01	3.31E+02
Variance	4.20E-01	3.73E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-9.10E+02	
P(T<=t) one-tail	1.46E-09	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.93E-09	
t Critical two-tail	3.18E+00	

Table 5.164: *BNL* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz	***ARAN***
0.998178852	0.999999948

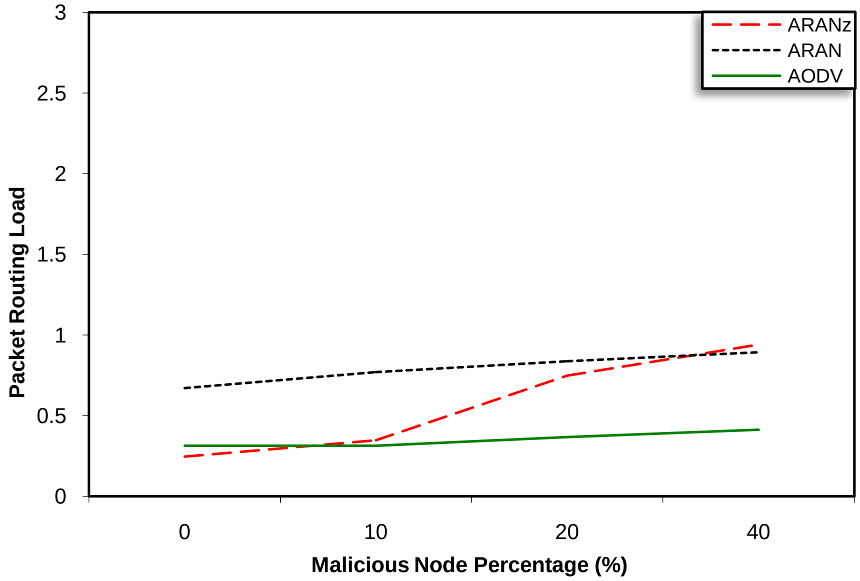


Figure 5.85: *PRL* vs. malicious node percentage considering grey hole attack

Table 5.165: *PRL* vs. malicious node percentage considering grey hole attack (t-Test)

	ARANz	*ARAN*
Mean	5.64E-01	7.94E-01
Variance	6.67E-02	5.09E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.40E+01	
t Stat	-3.10E+00	
P(T<=t) one-tail	3.92E-03	
t Critical one-tail	1.76E+00	
P(T<=t) two-tail	7.84E-03	
t Critical two-tail	2.14E+00	

	ARANz	*AODV*
Mean	5.64E-01	3.48E-01
Variance	6.67E-02	1.38E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	2.98E+00	
P(T<=t) one-tail	5.73E-03	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	1.15E-02	
t Critical two-tail	2.18E+00	

Table 5.166: *PRL* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.903676816	0.998318049	0.999278945

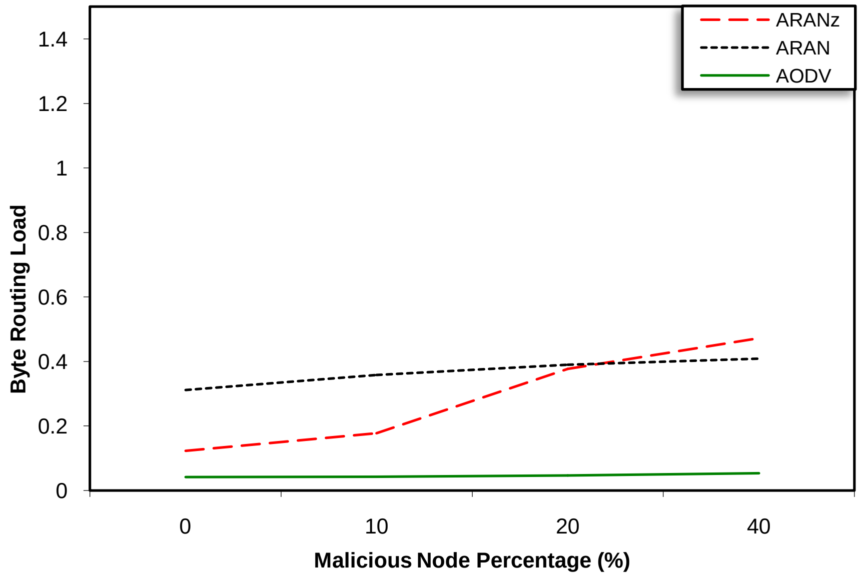


Figure 5.86: *BRL* vs. malicious node percentage considering grey hole attack

Table 5.167: *BRL* vs. malicious node percentage considering grey hole attack (t-Test)

	ARANz	***ARAN***
Mean	2.85E-01	3.69E-01
Variance	1.68E-02	1.01E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.30E+01	
t Stat	-2.26E+00	
P(T<=t) one-tail	2.09E-02	
t Critical one-tail	1.77E+00	
P(T<=t) two-tail	4.17E-02	
t Critical two-tail	2.16E+00	

	ARANz	***AODV***
Mean	2.85E-01	4.61E-02
Variance	1.68E-02	1.79E-05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	6.66E+00	
P(T<=t) one-tail	1.16E-05	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	2.33E-05	
t Critical two-tail	2.18E+00	

Table 5.168: *BRL* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.963262739	0.999530053	0.999975854

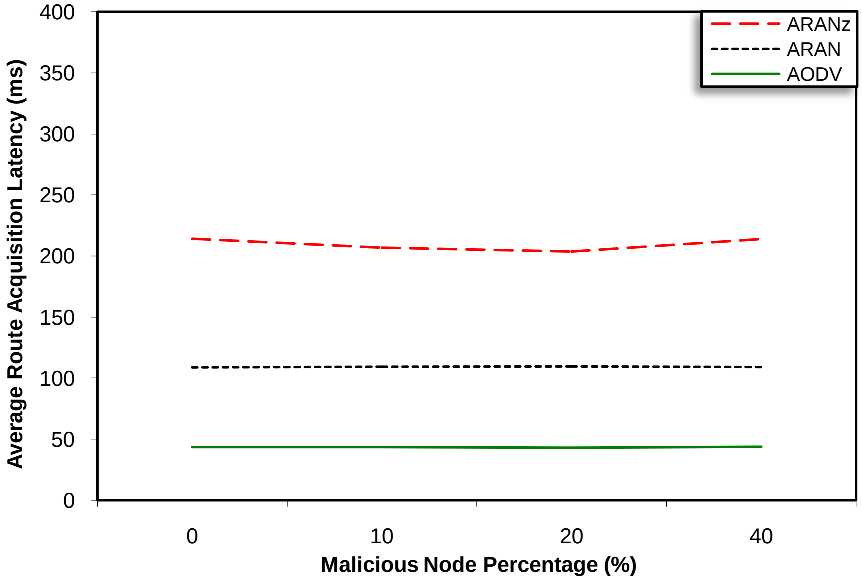


Figure 5.87: *ARAL* vs. malicious node percentage considering grey hole attack

Table 5.169: *ARAL* vs. malicious node percentage considering grey hole attack (t-Test)

	ARANz	***ARAN***
Mean	2.10E+02	1.09E+02
Variance	2.68E+01	1.62E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	3.87E+01	
P(T<=t) one-tail	1.90E-05	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.79E-05	
t Critical two-tail	3.18E+00	

	ARANz	***AODV***
Mean	2.10E+02	4.35E+01
Variance	2.68E+01	1.39E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	6.40E+01	
P(T<=t) one-tail	4.19E-06	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	8.39E-06	
t Critical two-tail	3.18E+00	

Table 5.170: *ARAL* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.943673344	0.999921266	0.999750515

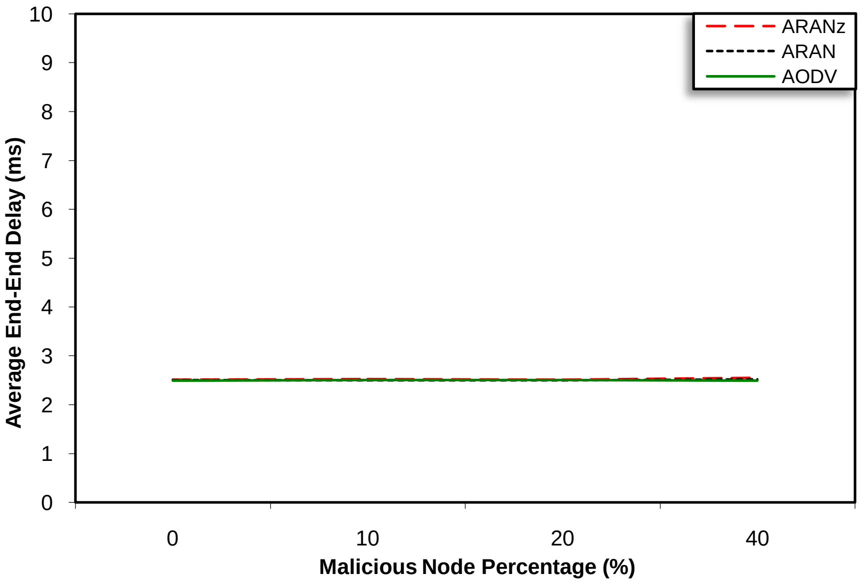


Figure 5.88: *AEED* vs. malicious node percentage considering grey hole attack

Table 5.171: *AEED* vs. malicious node percentage considering grey hole attack (t-Test)

	ARANz	***ARAN***
Mean	2.52E+00	2.50E+00
Variance	3.90E-04	9.75E-05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	1.72E+00	
P(T<=t) one-tail	8.00E-02	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	1.60E-01	
t Critical two-tail	2.78E+00	

	ARANz	***AODV***
Mean	2.52E+00	2.50E+00
Variance	3.90E-04	6.37E-05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	2.61E+00	
P(T<=t) one-tail	2.98E-02	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	5.97E-02	
t Critical two-tail	2.78E+00	

Table 5.172: *AEED* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999997348	0.999999665	0.999999822

Looking at Figure 5.89 and Table 5.174, it is clear that upon increasing the malicious node percentage *PLP* increases for the three protocols. However upon using *ARANz*, the increase in *PLP* is much slower, which is an evidence that *ARANz* is efficient in identifying grey hole attackers and justifies the increase in *CNP*, *PML* and *BML* for

ARANz with increasing malicious node percentage (refer to Figure 5.90 through Figure 5.92 and Table 5.175 through Table 5.177).

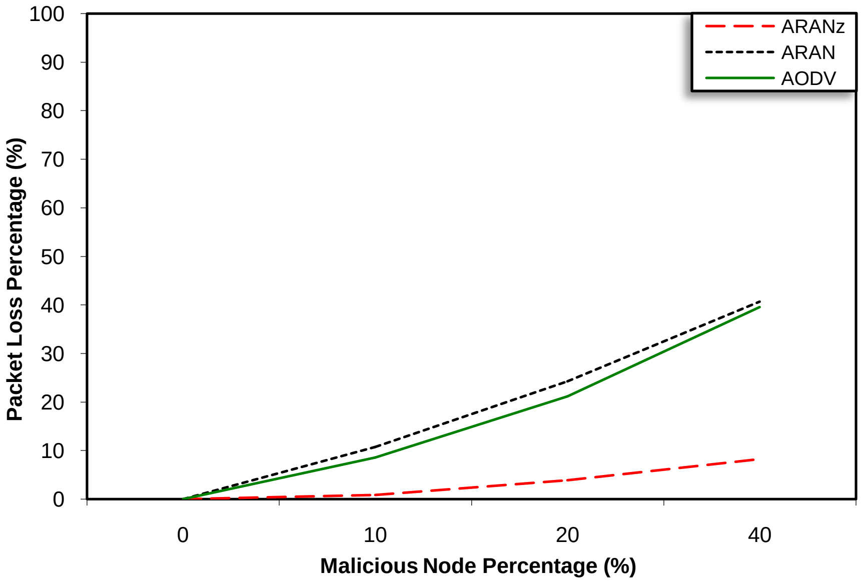


Figure 5.89: *PLP* vs. malicious node percentage considering grey hole attack

Table 5.173: *PLP* vs. malicious node percentage considering grey hole attack (t-Test)

	ARANz	***ARAN***
Mean	3.25E+00	1.89E+01
Variance	1.39E+01	3.09E+02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.74E+00	
P(T<=t) one-tail	8.99E-02	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.80E-01	
t Critical two-tail	3.18E+00	

	ARANz	***AODV***
Mean	3.25E+00	1.73E+01
Variance	1.39E+01	2.95E+02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.60E+00	
P(T<=t) one-tail	1.04E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.08E-01	
t Critical two-tail	3.18E+00	

Table 5.174: *PLP* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.005046258	1.26263E-10	4.5091E-11

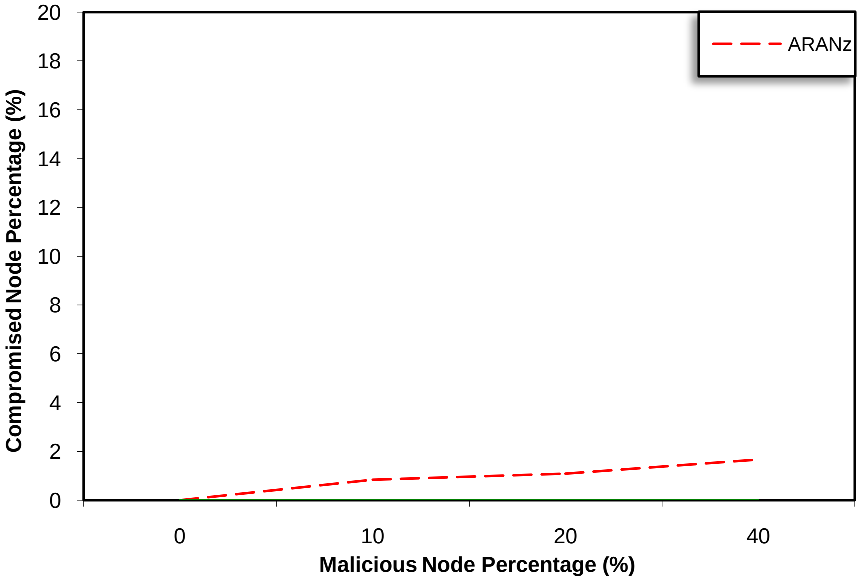


Figure 5.90: *CNP* vs. malicious node percentage considering grey hole attack

Table 5.175: *CNP* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz
0.658777509

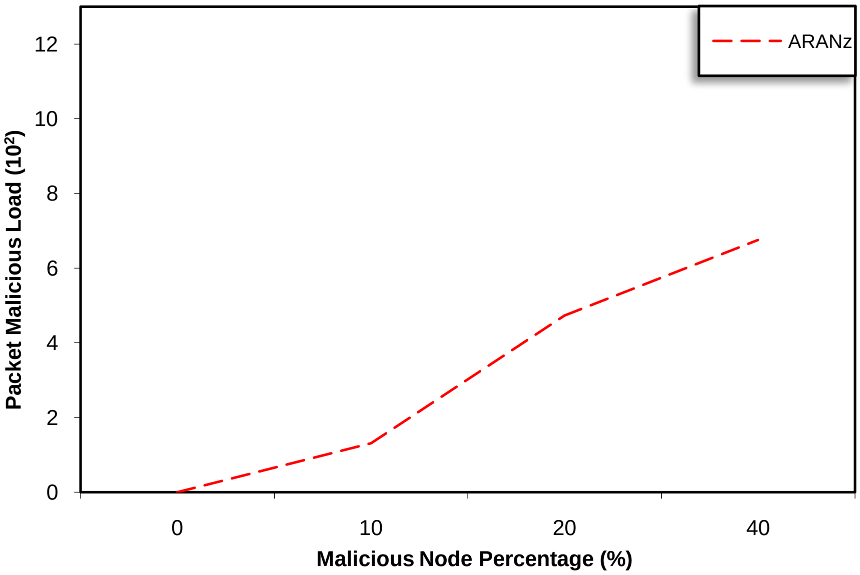


Figure 5.91: *PML* vs. malicious node percentage considering grey hole attack

Table 5.176: *PML* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz
0.0292448

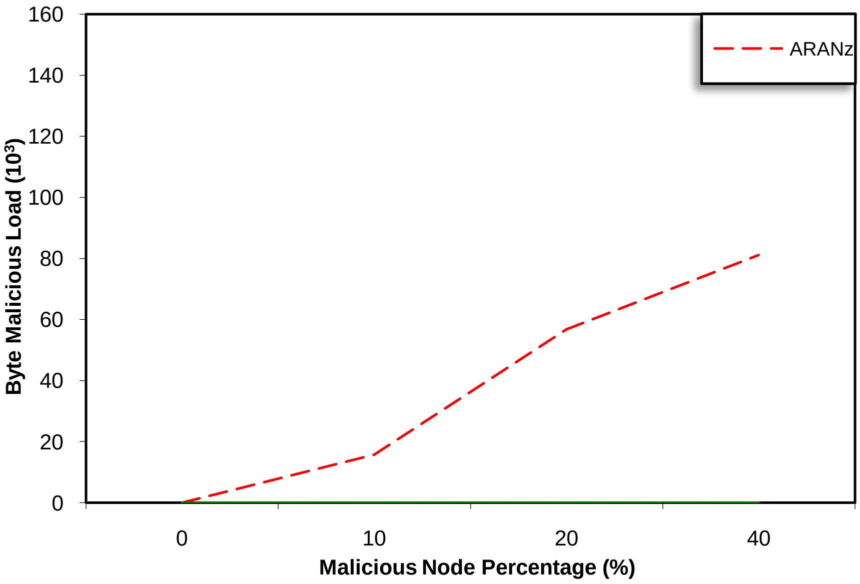


Figure 5.92: *BML* vs. malicious node percentage considering grey hole attack

Table 5.177: *BML* vs. malicious node percentage considering grey hole attack (chi-square Test)

ARANz
2.89574E-23

In comparison with the previous scenario (black hole attack effect), results of the conducted chi-square Tests show that the increase in *PRL*, *BRL*, *CNP*, *PML* and *BML* for *ARANz* is slower in this scenario. This means that discovering grey hole attackers is more difficult and requires a longer time compared to discovering black hole attackers because grey hole attackers drop only some of the data packets they receive, so it takes time to detect them.

Another point to mention here is that even though discovering grey hole attackers is slower than discovering black hole attackers, black hole attackers drop all packets they receive. Consequently, the increase in *PLP* and the decrease in *PDF* are almost the same in both cases.

5.3.7.4 Malicious Node Percentage Effect Considering Fabrication Attack

This scenario is conducted to examine the effect of the fabrication attack. In this attack, malicious nodes periodically fabricate *ERR* packets with a specific probability. To simulate this attack, malicious nodes existing in the path between the source and destination nodes periodically draws a random number between 0 and 1. If the drawn number is less than 0.5, then they send an *ERR* packets along the path toward the source to report false broken links.

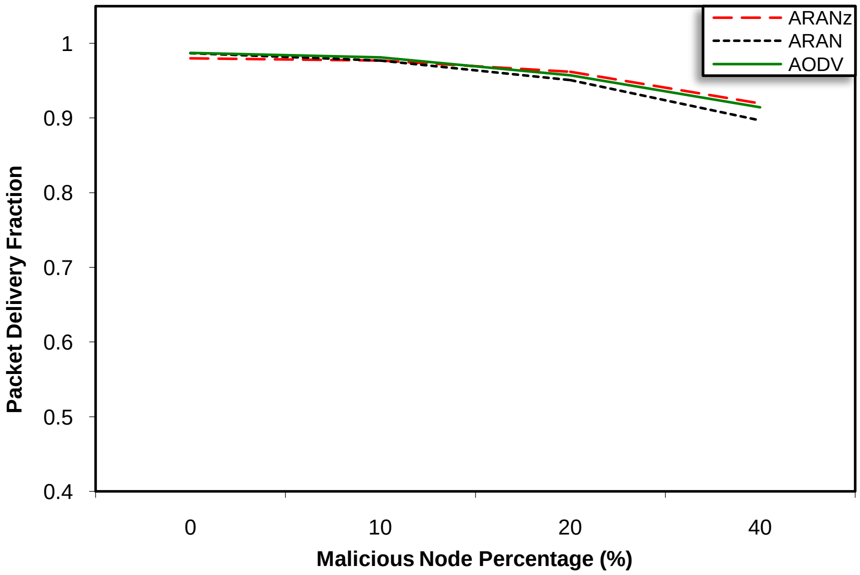


Figure 5.93: *PDF* vs. malicious node percentage considering fabrication attack

Table 5.178: *PDF* vs. malicious node percentage considering fabrication attack (t-Test)

	ARANz	***ARAN***
Mean	9.60E-01	9.53E-01
Variance	7.70E-04	1.63E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	2.73E-01	
P(T<=t) one-tail	3.98E-01	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	7.96E-01	
t Critical two-tail	2.57E+00	

	ARANz	***AODV***
Mean	9.60E-01	9.60E-01
Variance	7.70E-04	1.08E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-2.00E-02	
P(T<=t) one-tail	4.92E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	9.85E-01	
t Critical two-tail	2.45E+00	

Table 5.179: *PDF* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.970781346	0.915902388	0.952582268

Figure 5.93 shows that *PDF* decreases slightly for the three protocols as the malicious node percentage increases due to dropping some data packets as a result of receiving the fabricated *ERR* packets. However, the *PDF* for the three protocols is still above 90% even with the existence of large percentage of fabrication attackers.

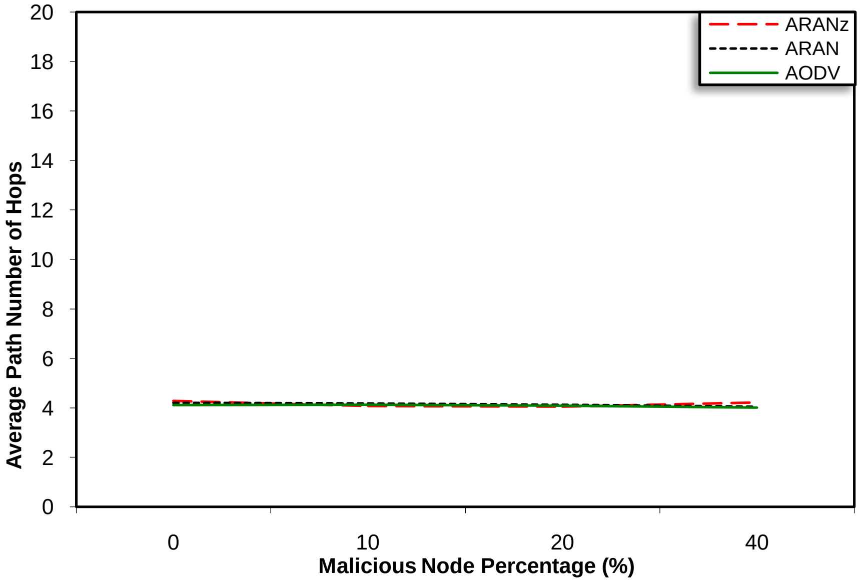


Figure 5.94: *APNH* vs. malicious node percentage considering fabrication attack

Table 5.180: *APNH* vs. malicious node percentage considering fabrication attack (t-Test)

	ARANz	*ARAN*
Mean	4.15E+00	4.14E+00
Variance	1.20E-02	4.84E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	2.02E-01	
P(T<=t) one-tail	4.24E-01	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	8.48E-01	
t Critical two-tail	2.57E+00	

	ARANz	*AODV*
Mean	4.15E+00	4.09E+00
Variance	1.20E-02	3.13E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	1.12E+00	
P(T<=t) one-tail	1.63E-01	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	3.27E-01	
t Critical two-tail	2.78E+00	

Table 5.181: *APNH* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999785232	0.999944785	0.999970768

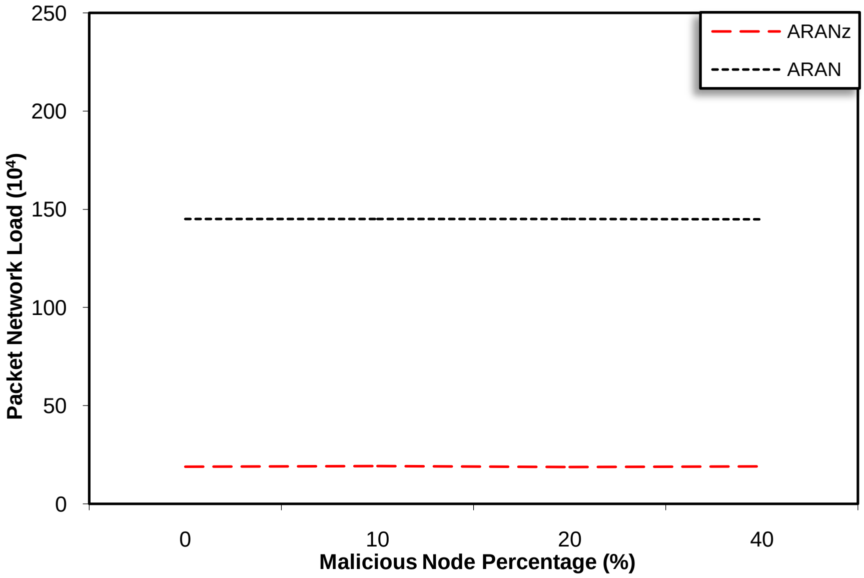


Figure 5.95: *PNL* vs. malicious node percentage considering fabrication attack

Table 5.182: *PNL* vs. malicious node percentage considering fabrication attack (t-Test)

	ARANz	*ARAN*
Mean	1.90E+01	1.45E+02
Variance	3.86E-02	1.19E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	-1.12E+03	
P(T<=t) one-tail	5.34E-15	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	1.07E-14	
t Critical two-tail	2.57E+00	

Table 5.183: *PNL* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz	*ARAN*
0.999873717	0.999998973

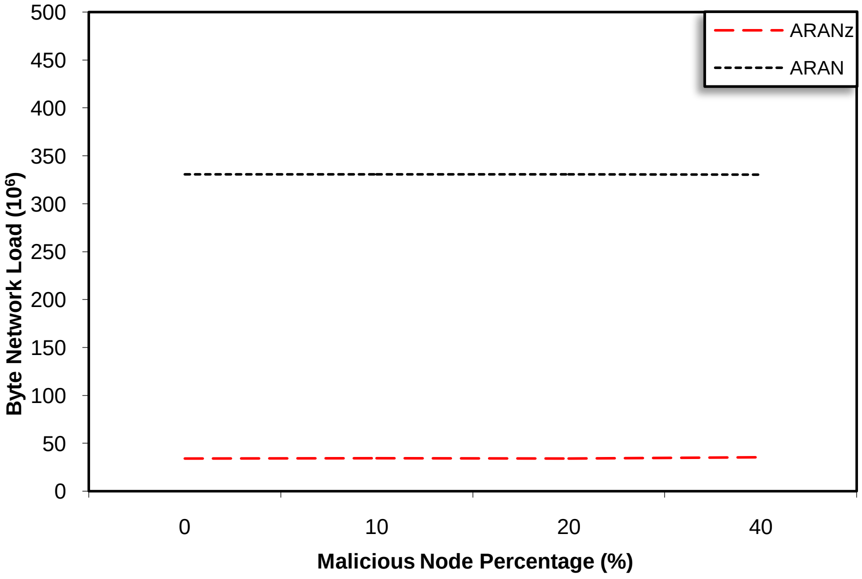


Figure 5.96: *BNL* vs. malicious node percentage considering fabrication attack

Table 5.184: *BNL* vs. malicious node percentage considering fabrication attack (t-Test)

	ARANz	***ARAN***
Mean	3.45E+01	3.31E+02
Variance	4.69E-01	6.18E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	-8.13E+02	
P(T<=t) one-tail	6.87E-12	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	1.37E-11	
t Critical two-tail	2.78E+00	

Table 5.185: *BNL* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz	***ARAN***
0.997831873	0.999996466

As in the preceding three scenarios, Figure 5.94 through Figure 5.96 and the related chi-square Tests show that *APNH*, *PNL* and *BNL* for the three protocols are general speaking not affected by malicious node percentage. This suggests that the three protocols are still able to discover the shortest paths even with the existence of some malicious nodes.

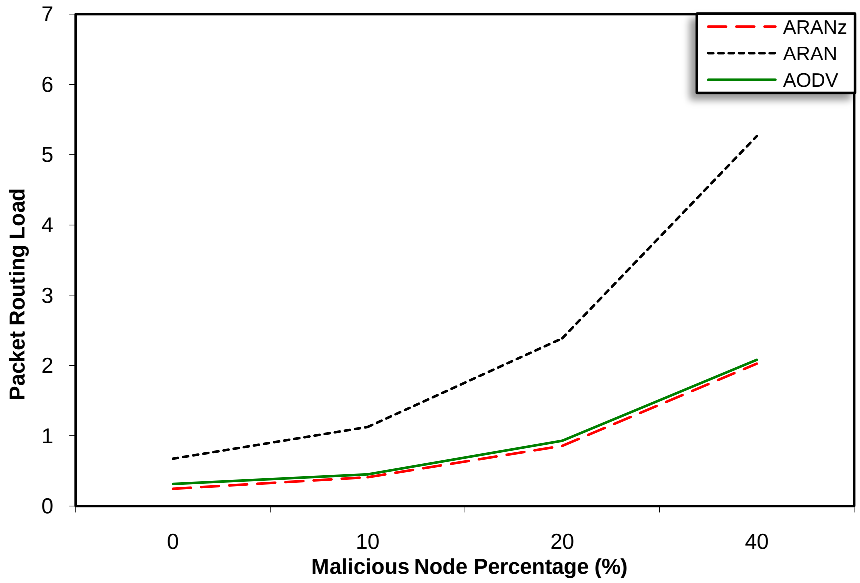


Figure 5.97: *PRL* vs. malicious node percentage considering fabrication attack

Table 5.186: *PRL* vs. malicious node percentage considering fabrication attack (t-Test)

	ARANz	***ARAN***
Mean	8.25E-01	2.22E+00
Variance	3.45E-01	2.31E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.60E+01	
t Stat	-3.08E+00	
P(T<=t) one-tail	3.56E-03	
t Critical one-tail	1.75E+00	
P(T<=t) two-tail	7.12E-03	
t Critical two-tail	2.12E+00	

	ARANz	***AODV***
Mean	8.25E-01	8.84E-01
Variance	3.45E-01	3.46E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	2.40E+01	
t Stat	-2.54E-01	
P(T<=t) one-tail	4.01E-01	
t Critical one-tail	1.71E+00	
P(T<=t) two-tail	8.01E-01	
t Critical two-tail	2.06E+00	

Table 5.187: *PRL* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.53314849	0.143118785	0.562855753

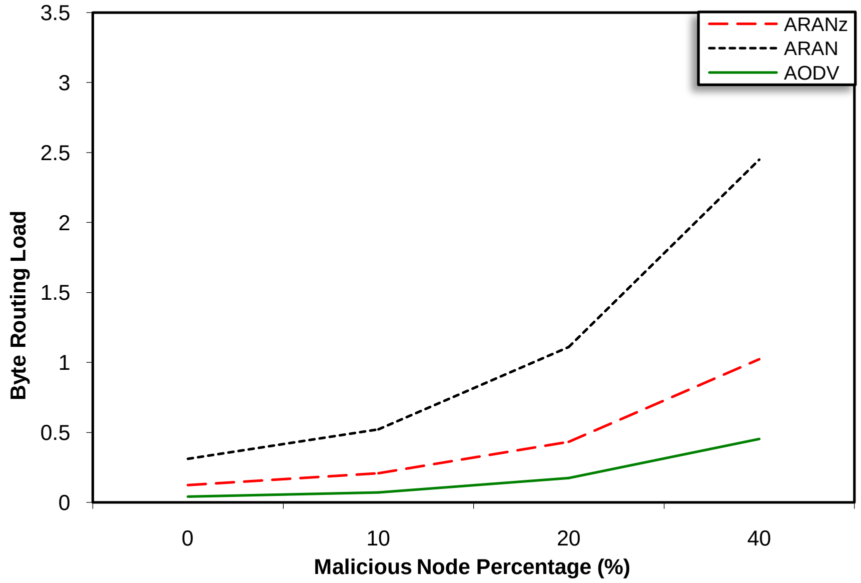


Figure 5.98: *BRL* vs. malicious node percentage considering fabrication attack

Table 5.188: *BRL* vs. malicious node percentage considering fabrication attack (t-Test)

	ARANz	*ARAN*
Mean	4.18E-01	1.03E+00
Variance	8.80E-02	5.01E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.60E+01	
t Stat	-2.89E+00	
P(T<=t) one-tail	5.32E-03	
t Critical one-tail	1.75E+00	
P(T<=t) two-tail	1.06E-02	
t Critical two-tail	2.12E+00	

	ARANz	*AODV*
Mean	4.18E-01	1.70E-01
Variance	8.80E-02	1.88E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.70E+01	
t Stat	2.73E+00	
P(T<=t) one-tail	7.14E-03	
t Critical one-tail	1.74E+00	
P(T<=t) two-tail	1.43E-02	
t Critical two-tail	2.11E+00	

Table 5.189: *BRL* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.775734529	0.470700994	0.902745149

By looking at Figure 5.97 and Figure 5.98, it is noticeable that *PRL* and *BRL* for either protocol increase as the malicious node percentage increases. This increase in *PRL* and *BRL* is due to reinitiating *RDP* packets by the source node as a result of receiving the fabricated *ERR* packets.

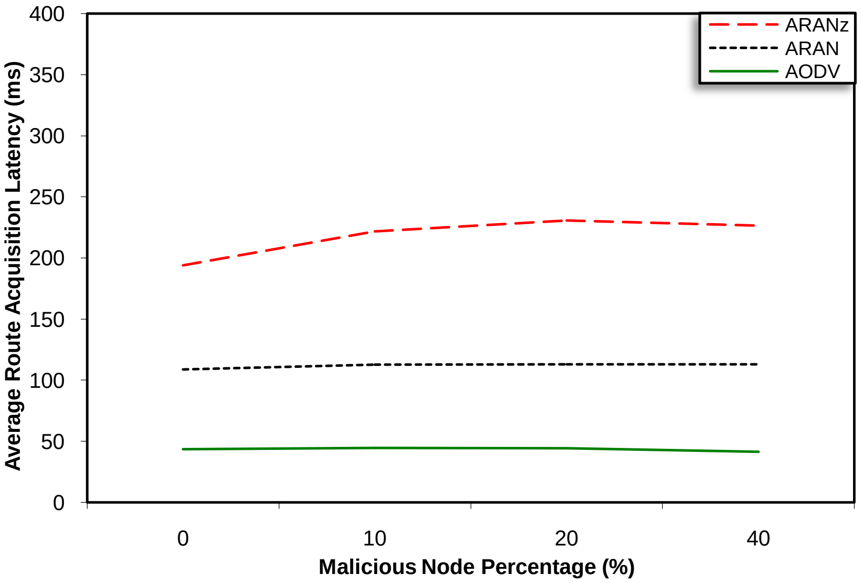


Figure 5.99: *ARAL* vs. malicious node percentage considering fabrication attack

Table 5.190: *ARAL* vs. malicious node percentage considering fabrication attack (t-Test)

	ARANz	*ARAN*
Mean	2.18E+02	1.12E+02
Variance	2.73E+02	4.24E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	1.28E+01	
P(T<=t) one-tail	5.18E-04	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.04E-03	
t Critical two-tail	3.18E+00	

	ARANz	*AODV*
Mean	2.18E+02	4.34E+01
Variance	2.73E+02	1.95E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	2.11E+01	
P(T<=t) one-tail	1.17E-04	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.33E-04	
t Critical two-tail	3.18E+00	

Table 5.191: *ARAL* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.289390604	0.990141568	0.987412326

Figure 5.99 and Table 5.191 show that *ARAL* for *AODV* and *ARAN* protocols is not affected by malicious node percentage. However, this metric for *ARANz* increases slightly with increasing the malicious node percentage. In *ARANz*, discovered malicious

nodes are not included in future route selections which may result in choosing non-optimal paths that do not contain malicious nodes.

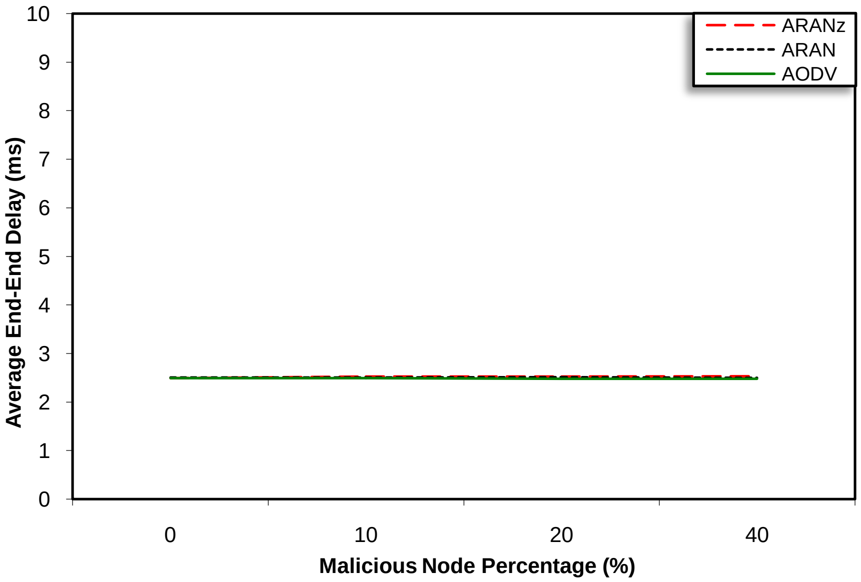


Figure 5.100: *AEED* vs. malicious node percentage considering fabrication attack

Table 5.192: *AEED* vs. malicious node percentage considering fabrication attack (t-Test)

	ARANz	***ARAN***
Mean	2.52E+00	2.50E+00
Variance	3.35E-04	3.14E-05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	1.39E+00	
P(T<=t) one-tail	1.18E-01	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	2.36E-01	
t Critical two-tail	2.78E+00	

	ARANz	***AODV***
Mean	2.52E+00	2.48E+00
Variance	3.35E-04	7.35E-05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	3.21E+00	
P(T<=t) one-tail	1.63E-02	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	3.25E-02	
t Critical two-tail	2.78E+00	

Table 5.193: *AEED* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999997875	0.999999939	0.999999777

Referring to Figure 5.100, it is clear that *AEED* is almost identical for the three protocols. Although *ARANz* has higher *ARAL*, the number of route discoveries and position enquiries performed is a small fraction of the number of data packets delivered. Hence, the effect of *ARAL* on *AEED* of data packets is not significant. Table 5.193 confirms that the differences in *AEED* for the three protocols are statistically insignificant.

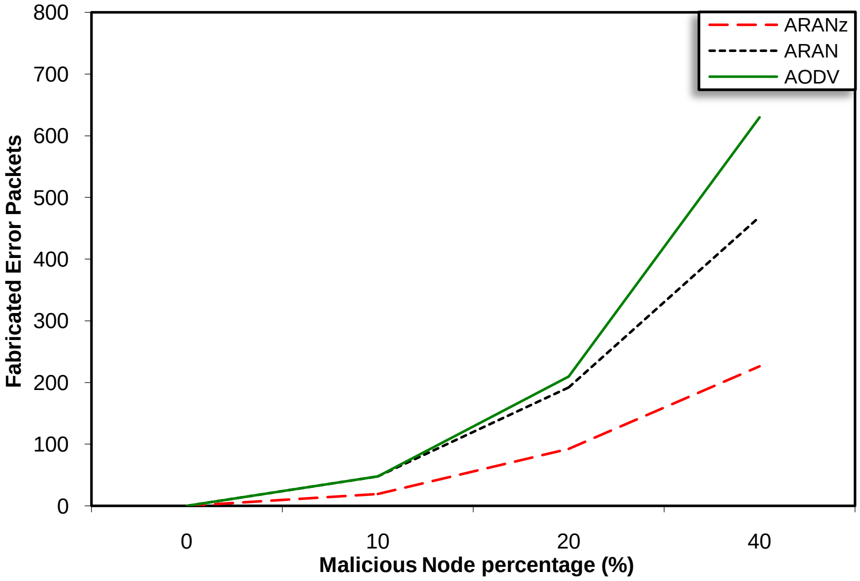


Figure 5.101: *FEP* vs. malicious node percentage considering fabrication attack

Table 5.194: *FEP* vs. malicious node percentage considering fabrication attack (t-Test)

	ARANz	***ARAN***
Mean	8.42E+01	1.77E+02
Variance	1.05E+04	4.44E+04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	-7.92E-01	
P(T<=t) one-tail	2.36E-01	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	4.73E-01	
t Critical two-tail	2.78E+00	

	ARANz	***AODV***
Mean	8.42E+01	2.22E+02
Variance	1.05E+04	8.20E+04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	-9.04E-01	
P(T<=t) one-tail	2.09E-01	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	4.17E-01	
t Critical two-tail	2.78E+00	

Table 5.195: *FEP* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz	***ARAN***	***AODV***
9.13673E-81	6.8468E-163	2.6724E-240

Figure 5.101 and results of the chi-square Test (presented in Table 5.195) show that *FEP* increases for the three protocols upon increasing the malicious node percentage. However, Table 5.195 shows that the increase in *FEP* is much slower upon simulating *ARANz*, which is an indication that *ARANz* is effective in detecting and isolating nodes performing fabrication attack.

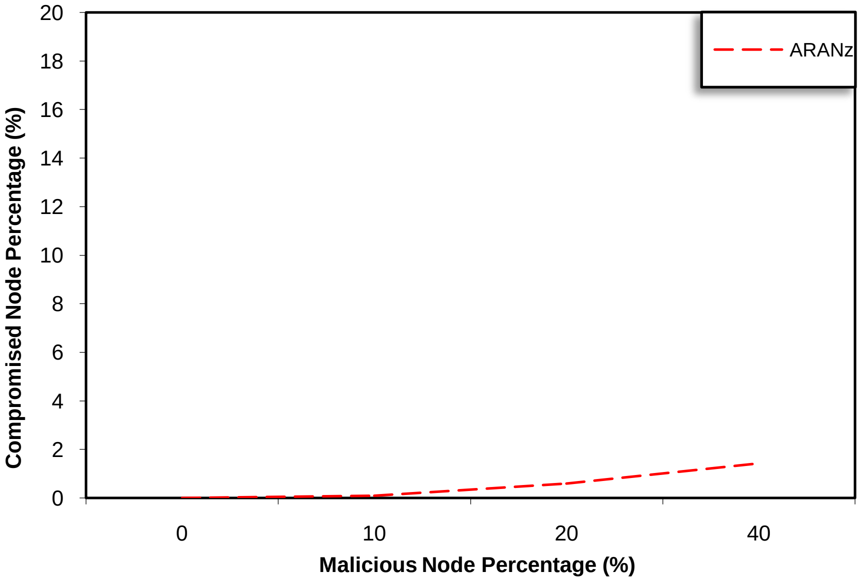


Figure 5.102: *CNP* vs. malicious node percentage considering fabrication attack

Table 5.196: *CNP* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz
0.4868447

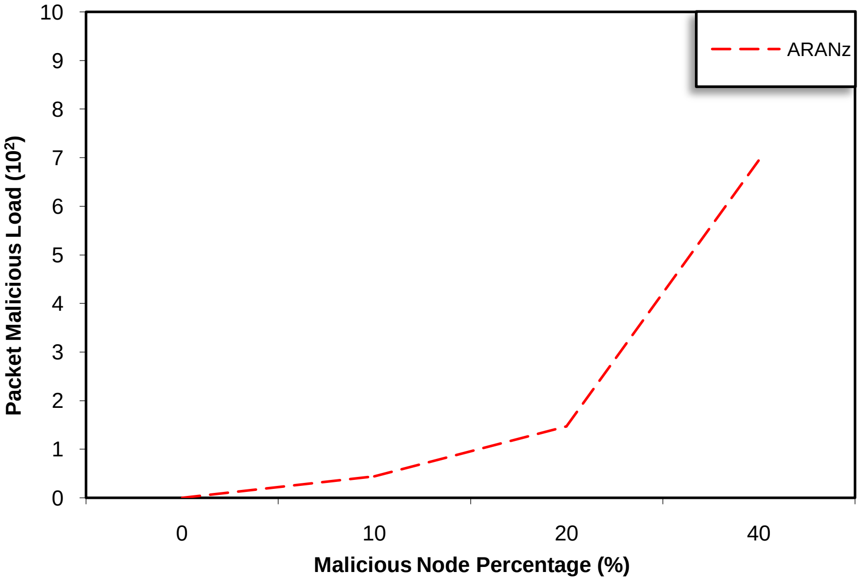


Figure 5.103: *PML* vs. malicious node percentage considering fabrication attack

Table 5.197: *PML* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz
0.0029

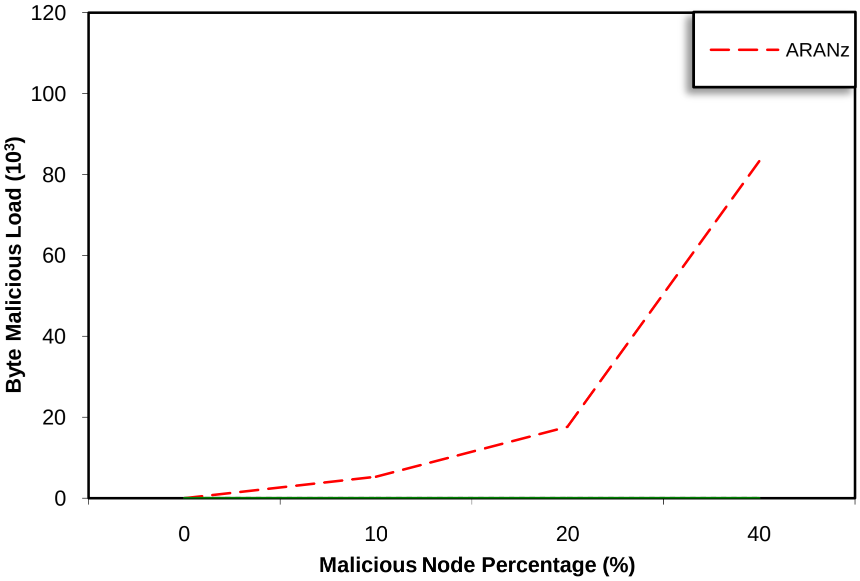


Figure 5.104: *BML* vs. malicious node percentage considering fabrication attack

Table 5.198: *BML* vs. malicious node percentage considering fabrication attack (chi-square Test)

ARANz
3.36156E-36

Figure 5.102 through Figure 5.104 show that *CNP*, *PML* and *BML* increase as the malicious node percentage increases. In other words, as malicious node percentage increases *ARANz* demonstrates its effectiveness in distinguishing more and more malicious nodes.

5.3.7.5 Malicious Node Percentage Effect Considering Multi-Attack

In this scenario, the effect of multi-attack is studied. In this attack, malicious nodes perform multiple attacks with a specific probability. To simulate multi-attack, malicious nodes perform modification, grey hole and fabrication attacks. The same details used to simulate each attack separately in the previous scenarios are used to simulate multi-attack. In other words, malicious nodes performing multi-attack illegally reset the hop count field to 0 in a received route discovery or route reply, if a drawn number is less than 0.5. They also drop a received data packet if a drawn number is less than 0.5 and periodically fabricate *ERR* packet if a drawn number is less than 0.5.

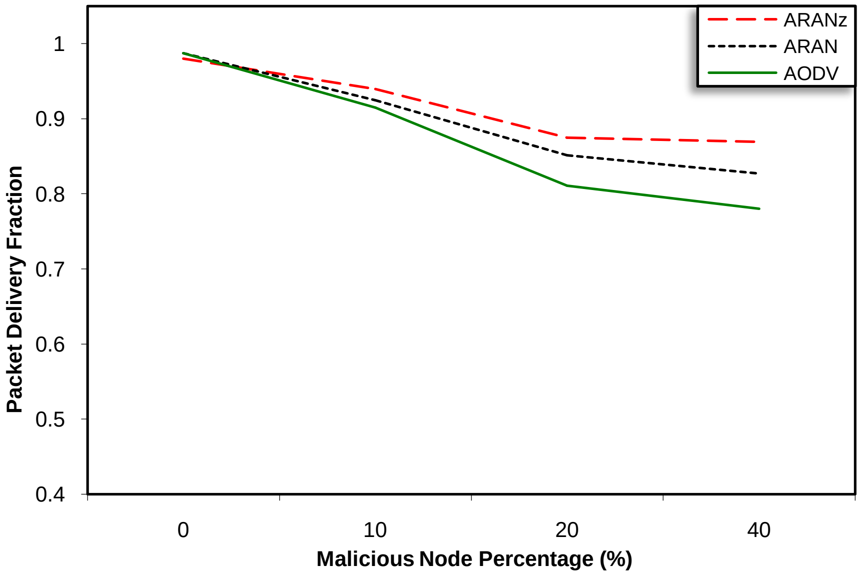


Figure 5.105: *PDF* vs. malicious node percentage considering multi-attack

Table 5.199: *PDF* vs. malicious node percentage considering multi-attack (t-Test)

	ARANz	*ARAN*
Mean	9.16E-01	8.97E-01
Variance	2.84E-03	5.29E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	4.09E-01	
P(T<=t) one-tail	1.48E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	6.97E-01	
t Critical two-tail	2.45E+00	

	ARANz	*AODV*
Mean	9.16E-01	8.73E-01
Variance	2.84E-03	9.08E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	7.81E-01	
P(T<=t) one-tail	4.35E-02	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	4.70E-01	
t Critical two-tail	2.57E+00	

Table 5.200: *PDF* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.817934264	0.622179015	0.373346007

Referring to Figure 5.105 it is clear that increasing malicious node percentage results in decreasing *PDF* for all protocols. This is mainly due to data packets dropped upon performing grey hole attack. The slower decrease in *ARANz*'s *PDF* is an indication that *ARANz* is effective in identifying and isolating multi-attack malicious nodes even if the simulated percentage is large.

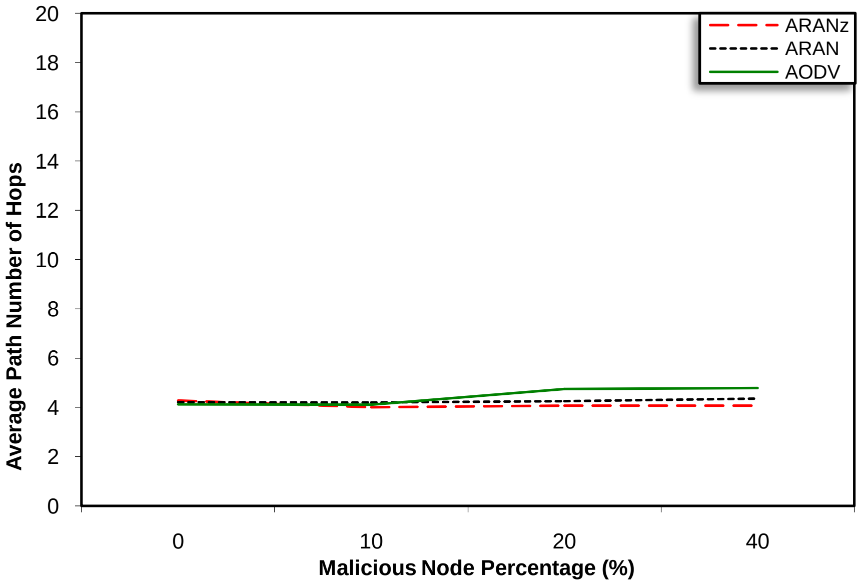


Figure 5.106: *APNH* vs. malicious node percentage considering multi-attack

Table 5.201: *APNH* vs. malicious node percentage considering multi-attack (t-Test)

	ARANz	*ARAN*
Mean	4.11E+00	4.25E+00
Variance	1.38E-02	5.73E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	-2.04E+00	
P(T<=t) one-tail	4.82E-02	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	9.64E-02	
t Critical two-tail	2.57E+00	

	ARANz	*AODV*
Mean	4.11E+00	4.44E+00
Variance	1.38E-02	1.40E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	-1.70E+00	
P(T<=t) one-tail	8.23E-02	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	1.65E-01	
t Critical two-tail	2.78E+00	

Table 5.202: *APNH* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999730564	0.999931702	0.989465681

Figure 5.106 and Table 5.202 show that *APNH* for *AODV* slightly increases as the malicious node percentage increases. Malicious nodes can exploit *AODV*, via modification attack, so that non-shortest paths are selected. *ARAN* and *ARANz* are not exploitable in this way. The selected route could pass through a malicious node but not forced to do this.

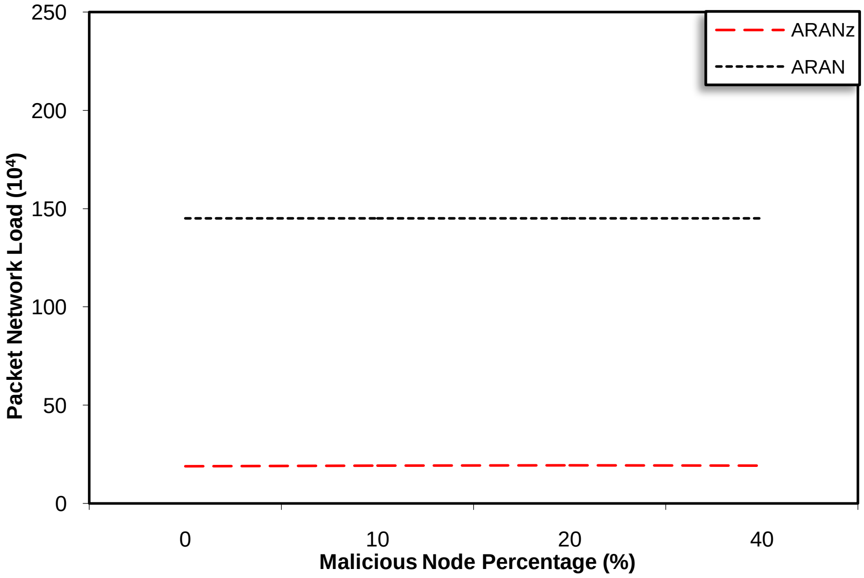


Figure 5.107: *PNL* vs. malicious node percentage considering multi-attack

Table 5.203: *PNL* vs. malicious node percentage considering multi-attack (t-Test)

	ARANz	***ARAN***
Mean	1.92E+01	1.45E+02
Variance	4.89E-02	5.47E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.13E+03	
P(T<=t) one-tail	7.59E-10	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.52E-09	
t Critical two-tail	3.18E+00	

Table 5.204: *PNL* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz	***ARAN***
0.999822242	0.99999999

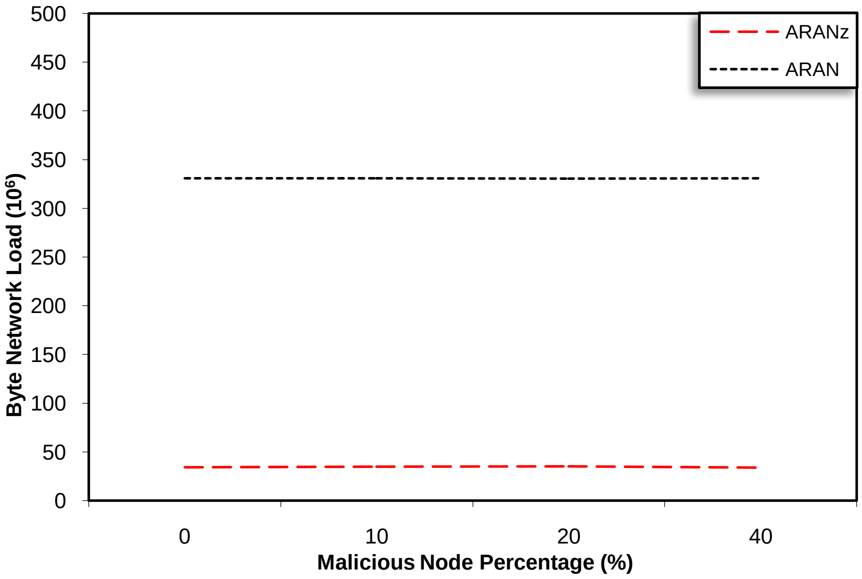


Figure 5.108: *BNL* vs. malicious node percentage considering multi-attack

Table 5.205: *BNL* vs. malicious node percentage considering multi-attack (t-Test)

	ARANz	***ARAN***
Mean	3.45E+01	3.31E+02
Variance	4.58E-01	2.85E-03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-8.73E+02	
P(T<=t) one-tail	1.65E-09	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.31E-09	
t Critical two-tail	3.18E+00	

Table 5.206: *BNL* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz	***ARAN***
0.997913592	0.999999965

It is conspicuous from Figure 5.107, Figure 5.108, Table 5.204 and Table 5.206 that malicious node percentage definitely does not affect *PNL* and *BNL* for *ARAN* and *ARANz* protocols. The reason behind the stable *PNL* and *BNL* is that updating nodes' certificates and positions is carried out regardless the number of existing malicious nodes.

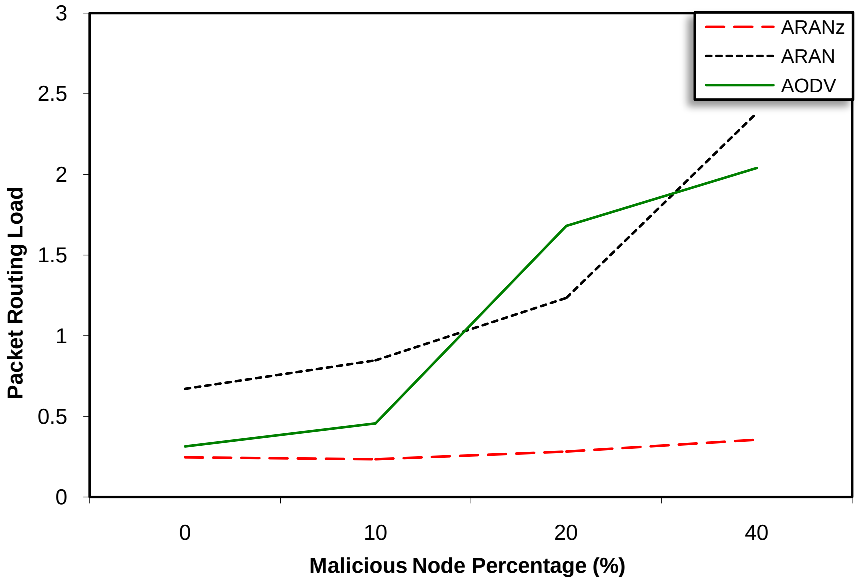


Figure 5.109: *PRL* vs. malicious node percentage considering multi-attack

Table 5.207: *PRL* vs. malicious node percentage considering multi-attack (t-Test)

	ARANz	*ARAN*
Mean	2.73E-01	1.23E+00
Variance	1.67E-03	3.12E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-6.14E+00	
P(T<=t) one-tail	2.53E-05	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	5.05E-05	
t Critical two-tail	2.18E+00	

	ARANz	*AODV*
Mean	2.73E-01	1.11E+00
Variance	1.67E-03	4.77E-01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-4.36E+00	
P(T<=t) one-tail	4.67E-04	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	9.35E-04	
t Critical two-tail	2.18E+00	

Table 5.208: *PRL* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.998491445	0.709625878	0.571111512

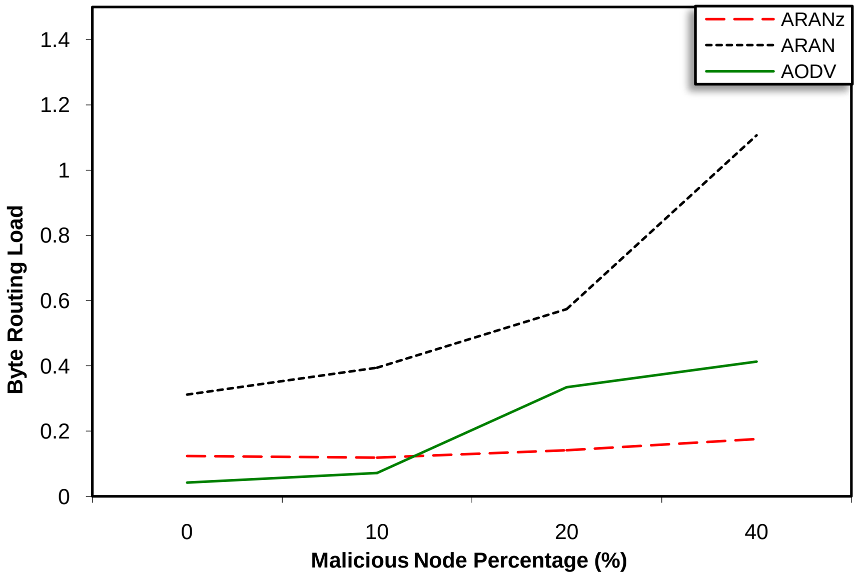


Figure 5.110: *BRL* vs. malicious node percentage considering multi-attack

Table 5.209: *BRL* vs. malicious node percentage considering multi-attack (t-Test)

	ARANz	***ARAN***
Mean	1.38E-01	5.71E-01
Variance	3.77E-04	6.74E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-6.00E+00	
P(T<=t) one-tail	3.11E-05	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	6.21E-05	
t Critical two-tail	2.18E+00	

	ARANz	***AODV***
Mean	1.38E-01	2.13E-01
Variance	3.77E-04	2.21E-02
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	1.20E+01	
t Stat	-1.81E+00	
P(T<=t) one-tail	4.79E-02	
t Critical one-tail	1.78E+00	
P(T<=t) two-tail	9.58E-02	
t Critical two-tail	2.18E+00	

Table 5.210: *BRL* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.999548096	0.886924747	0.9226007

Figure 5.109 and Figure 5.110 show that *PRL* and *BRL* for the three protocols increase with increasing malicious node percentage. This increase in *PRL* and *BRL* is mainly due to reinitiating *RDP* packets by the source upon receiving the fabricated *ERR* packets. Also, it is apparent that *ARANz* has the minimum *PRL* and chi-square Test assures that *ARANz* has the slowest increase in *PRL*, which reflects *ARANz* effectiveness in detecting and isolating the fabrication attackers. Furthermore, Table 5.208 shows that *AODV* is highly affected by the fabrication attack because the selected routes in *AODV* are forced to pass through malicious nodes via modification attack. After that, these malicious nodes start to fabricate *ERR* packets resulting in higher *PRL* and *BRL*. In *ARAN* and *ARANz*, however, routes are not forced to go through malicious nodes due to their robustness against the modification attacks.

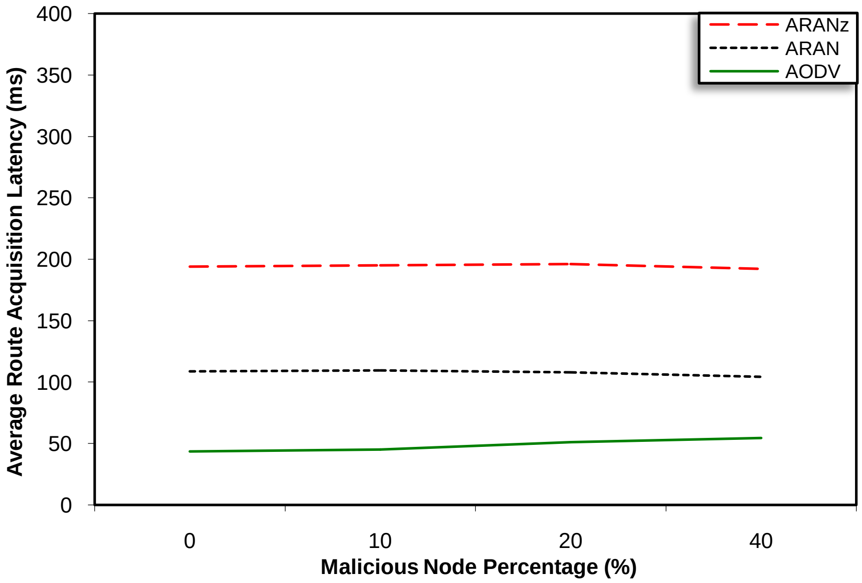


Figure 5.111: *ARAL* vs. malicious node percentage considering multi-attack

Table 5.211: *ARAL* vs. malicious node percentage considering multi-attack (t-Test)

	ARANz	*ARAN*
Mean	1.94E+02	1.08E+02
Variance	2.58E+00	5.64E+00
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	5.00E+00	
t Stat	6.05E+01	
P(T<=t) one-tail	1.17E-08	
t Critical one-tail	2.02E+00	
P(T<=t) two-tail	2.34E-08	
t Critical two-tail	2.57E+00	

	ARANz	*AODV*
Mean	1.94E+02	4.77E+01
Variance	2.58E+00	1.46E+01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	4.00E+00	
t Stat	7.07E+01	
P(T<=t) one-tail	1.20E-07	
t Critical one-tail	2.13E+00	
P(T<=t) two-tail	2.40E-07	
t Critical two-tail	2.78E+00	

Table 5.212: *ARAL* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.997910806	0.984201061	0.820286878

Figure 5.111 and Table 5.212 show that *ARAL* for *AODV* slightly increases upon increasing malicious node percentage due to selecting non-shortest paths (since it is susceptible to modification attack). *ARAL* for *ARAN* and *ARANz* protocols is not affected by increasing malicious node percentage since both protocols are robust against modification attacks.

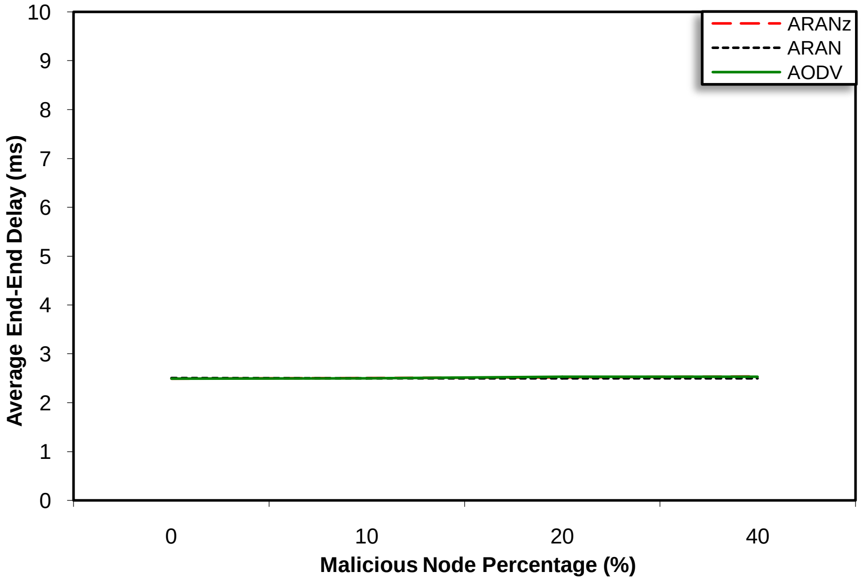


Figure 5.112: *AEED* vs. malicious node percentage considering multi-attack

Table 5.213: *AEED* vs. malicious node percentage considering multi-attack (t-Test)

	ARANz	*ARAN*
Mean	2.51E+00	2.50E+00
Variance	3.88E-04	1.18E-05
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	1.18E+00	
P(T<=t) one-tail	1.62E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.23E-01	
t Critical two-tail	3.18E+00	

	ARANz	*AODV*
Mean	2.51E+00	2.51E+00
Variance	3.88E-04	5.50E-04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-1.38E-01	
P(T<=t) one-tail	4.47E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	8.95E-01	
t Critical two-tail	2.45E+00	

Table 5.214: *AEED* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz	*ARAN*	*AODV*
0.999997346	0.999999986	0.999995533

It is obvious from Figure 5.112 that *AEED* curves for the three protocols are almost identical since the number of route discoveries and position enquiries performed is limited compared to the number of data packets delivered. Hence, the effect of *ARAL* on *AEED* of data packets is not significant.

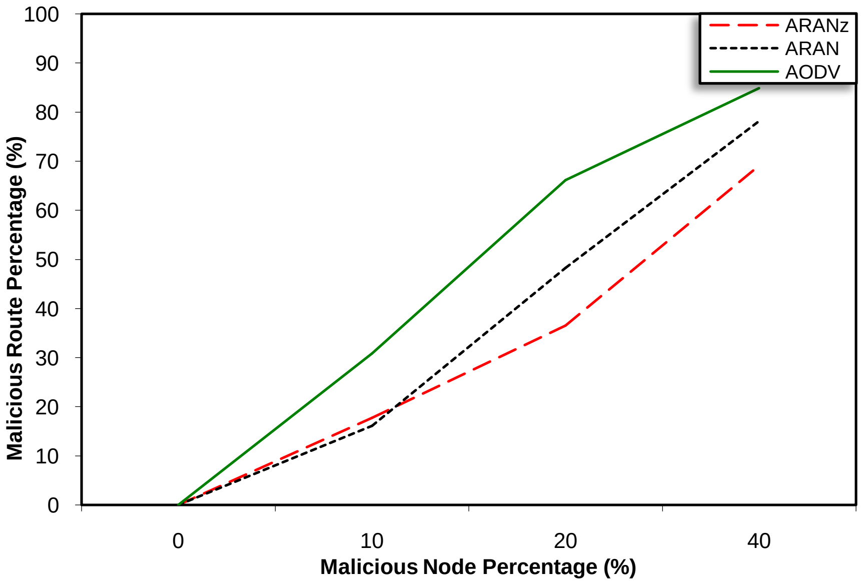


Figure 5.113: *MRP* vs. malicious node percentage considering multi-attack

Table 5.215: *MRP* vs. malicious node percentage considering multi-attack (t-Test)

	ARANz	*ARAN*
Mean	3.08E+01	3.56E+01
Variance	8.72E+02	1.20E+03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-2.10E-01	
P(T<=t) one-tail	4.20E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	8.40E-01	
t Critical two-tail	2.45E+00	

	ARANz	*AODV*
Mean	3.08E+01	4.55E+01
Variance	8.72E+02	1.42E+03
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	6.00E+00	
t Stat	-6.11E-01	
P(T<=t) one-tail	2.82E-01	
t Critical one-tail	1.94E+00	
P(T<=t) two-tail	5.64E-01	
t Critical two-tail	2.45E+00	

Table 5.216: *MRP* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz	*ARAN*	*AODV*
2.8818E-18	7.68648E-22	3.49172E-20

As shown in Figure 5.113 and Table 5.216, the *MRP* increases for the three protocols as the malicious node percentage increases. As the figure also shows, more routes with malicious nodes within them are used upon simulating *AODV*. When the malicious node resets the hop count field to 0, it forces *AODV* to select the route passes through itself because *AODV* selects the shortest path. *ARAN* and *ARANz*, on the other hand, cannot be exploited in this way.

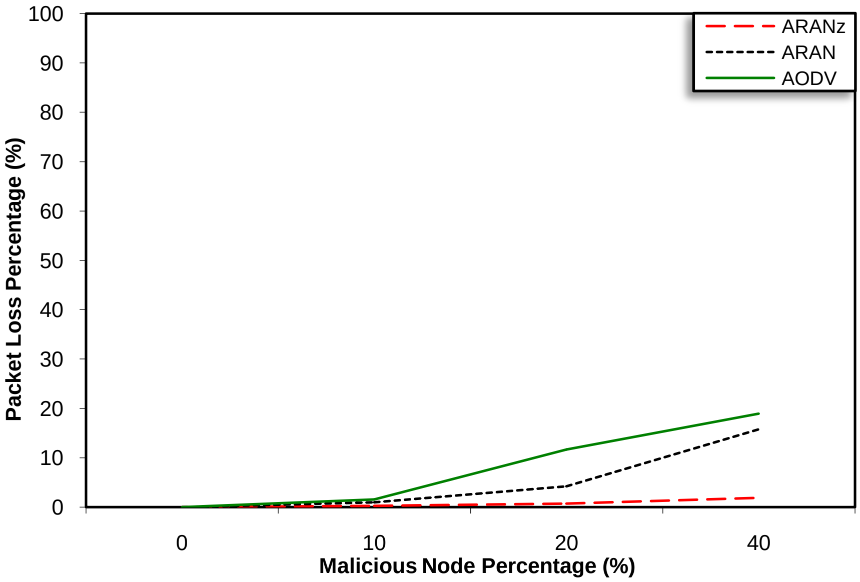


Figure 5.114: *PLP* vs. malicious node percentage considering multi-attack

Table 5.217: *PLP* vs. malicious node percentage considering multi-attack (t-Test)

	ARANz	***ARAN***
Mean	7.09E-01	5.23E+00
Variance	6.74E-01	5.25E+01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.24E+00	
P(T<=t) one-tail	1.52E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	3.03E-01	
t Critical two-tail	3.18E+00	

	ARANz	***AODV***
Mean	7.09E-01	8.04E+00
Variance	6.74E-01	7.96E+01
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.64E+00	
P(T<=t) one-tail	1.00E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.00E-01	
t Critical two-tail	3.18E+00	

Table 5.218: *PLP* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz	***ARAN***	***AODV***
0.415034932	1.299E-06	1.60587E-06

It is clear from Figure 5.114 that the *PLP* for the three protocols increases with increasing malicious node percentage due to dropping data packets via the grey hole attack. However, Table 5.218 shows that upon using *ARANz*, the increase in *PLP* is significantly slower indicating that *ARANz* is efficient in isolating grey hole attackers.

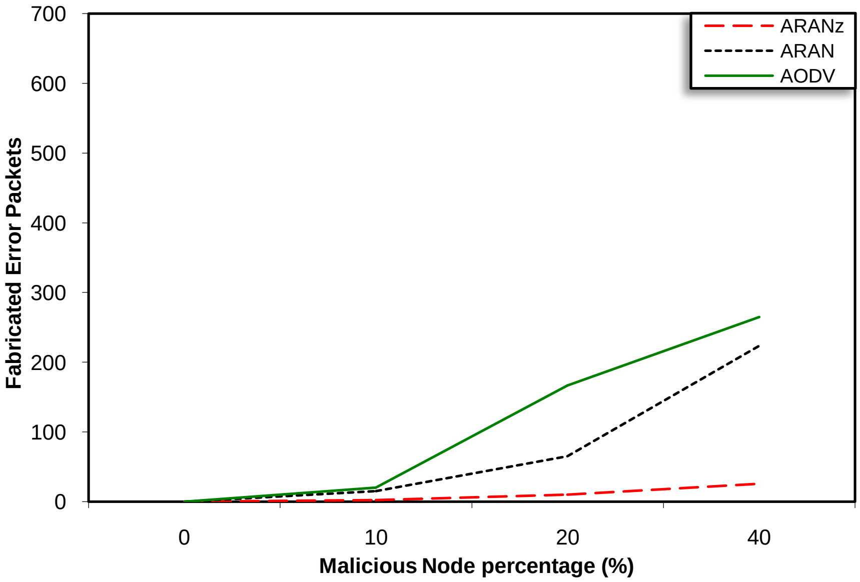


Figure 5.115: *FEP* vs. malicious node percentage considering multi-attack

Table 5.219: *FEP* vs. malicious node percentage considering multi-attack (t-Test)

	ARANz	*ARAN*
Mean	9.55E+00	7.60E+01
Variance	1.33E+02	1.05E+04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.29E+00	
P(T<=t) one-tail	1.44E-01	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	2.88E-01	
t Critical two-tail	3.18E+00	

	ARANz	*AODV*
Mean	9.55E+00	1.13E+02
Variance	1.33E+02	1.57E+04
Observations	4.00E+00	4.00E+00
Hypothesized Mean Difference	0.00E+00	
df	3.00E+00	
t Stat	-1.64E+00	
P(T<=t) one-tail	9.97E-02	
t Critical one-tail	2.35E+00	
P(T<=t) two-tail	1.99E-01	
t Critical two-tail	3.18E+00	

Table 5.220: *FEP* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz	*ARAN*	*AODV*
4.16908E-09	2.74011E-89	2.71368E-90

The *FEP* for the three protocols increases upon increasing the malicious node percentage (as shown in Figure 5.115). However, the results of the chi-square Test (presented in Table 5.220) assure that the increase in *FEP* is much slower upon using *ARANz*, which illustrates that *ARANz* is effective in identifying and extracting nodes performing fabrication attack. Also, the increase in *FEP* is faster in *AODV* protocol since it is forced to use routes containing malicious nodes (via modification attack). Afterward, these nodes start sending fabricated *ERR* packets, resulting in higher *FEP*.

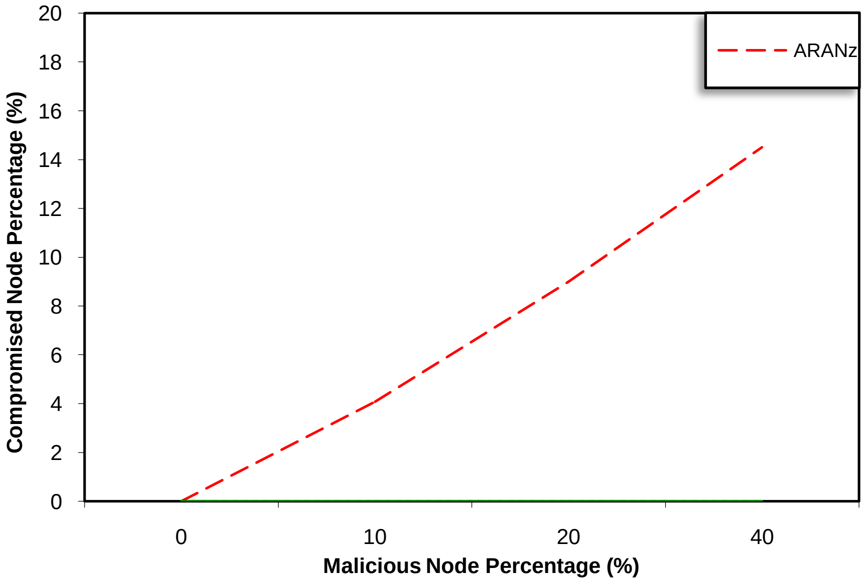


Figure 5.116: *CNP* vs. malicious node percentage considering multi-attack

Table 5.221: *CNP* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz
0.000683621

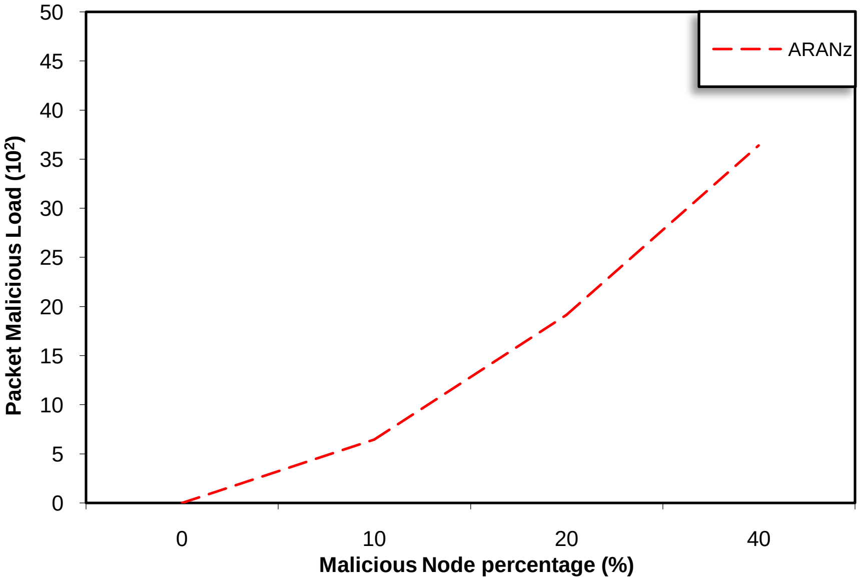


Figure 5.117: *PML* vs. malicious node percentage considering multi-attack

Table 5.222: *PML* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz
8.887E-11

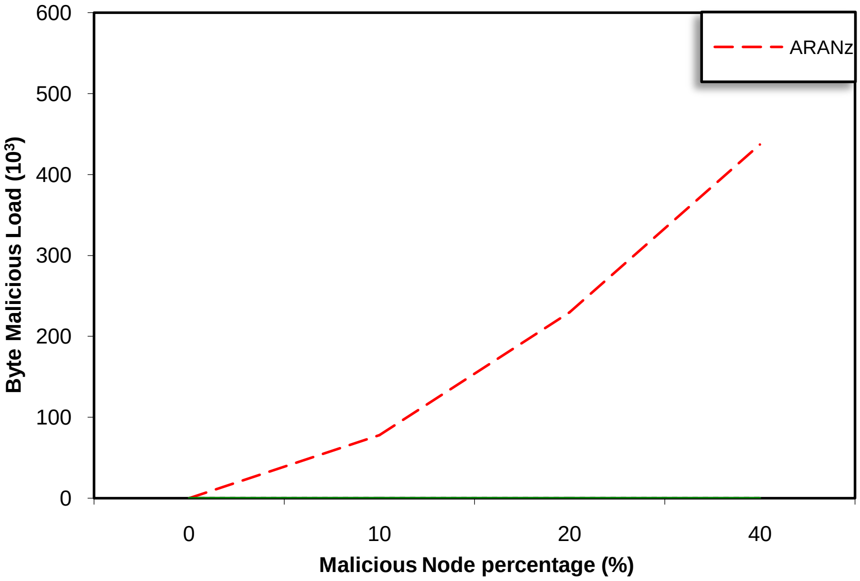


Figure 5.118: *BML* vs. malicious node percentage considering multi-attack

Table 5.223: *BML* vs. malicious node percentage considering multi-attack (chi-square Test)

ARANz
3.7009E-129

Figure 5.116 through Figure 5.118 show that as malicious node percentage increases *ARANz* demonstrates its effectiveness in detecting more and more malicious nodes, i.e. *CNP*, *PML* and *BML* significantly increase as the number of malicious nodes performing multi-attack increases.

5.4 Results Summary

From the results the following points are concluded:

- *PDF* for the three protocols is above 95% in most scenarios. This indicates that the three protocols are highly effective in discovering and maintaining routes for delivery of data packets even with relatively high node mobility and large area networks. Upon studying the effect of malicious node percentage, however, results show that the decrease in *PDF* is much slower in *ARANz* in most cases, implying that *ARANz* is efficient in detecting and isolating malicious nodes even with relatively large percentage of them.
- *PNL* and *BNL* for *ARANz* are significantly less than *ARAN*. The main reason behind this gap is that nodes in *ARAN* are unaware of the position of the *CA* server, hence, all certificate update request packets sent from nodes to *CA* are broadcast to the entire network. In *ARANz*, however, most packets are sent using restricted directional flooding, source routing, zone flooding or *LCA* flooding.
- *ARANz* has the minimum *PRL* in all experiments and the conducted statistical analysis tests confirm that the differences between *PRL* for the three protocols are statistically significant. In contrast to *AODV* and *ARAN*, *ARANz* does not broadcast the *RDP* packets to the whole area, instead, these packets are sent using restricted directional flooding towards the destination. Even *PDP* packets are sent using restricted directional flooding or source routing. Hence, *PDP* packets do not significantly affect *PRL*, especially if the source and destination are in the same zone.

- *ARAN* has higher *PRL* compared to *AODV* as a consequence of higher packet processing and authentication delay in *ARAN* protocol. In other words, higher delay increases the chance of having link break and reinitiating *RDP* packets, i.e. higher *PRL*.
- Even though *ARANz* has smaller *PRL* compared to *AODV*, it has higher *BRL* due to the larger control packets that contain security data.
- *APNH* is almost identical for the three protocols for a specified network parameters setting. In other words, even though *ARAN* and *ARANz* do not explicitly seek the shortest paths, the first *RDP* packet to reach the destination usually travels along the shortest path. Hence, it is obvious that *ARAN* and *ARANz* are as efficient as *AODV* in discovering shortest paths.
- *AODV* is superior in its *ARAL* as it has the shortest processing delay at each node. On the other hand, while processing routing control packets in *ARAN* and *ARANz*, each node has to verify the digital signature of the previous node and replace this signature with its own digital signature, in addition to the normal packet processing done by *AODV*. This signature generation and verification results in additional delay at each hop, and so *ARAL* increases. Moreover, *ARANz* has the highest *ARAL* since it needs to carry out a destination's position discovery step. However *ARANz*'s *ARAL* improves rapidly as more and more packets become internal ones. Upon increasing local communications, *ARANz*'s *ARAL* significantly decreases since the position of the destination is found in the authentication table of the nearest *LCA* to the source, so there is no need to communicate with *LCA*s in other zones.
- Differences in *AEED* between the three protocols are almost negligible since the number of route discoveries and position enquiries performed is limited compared to the number of data packets delivered. Hence, the effect of *ARAL* on *AEED* of data packets is not significant.

- High *PDF* and low *APNH* for all protocols are obtained for node density values between 60 nodes/km^2 and 80 nodes/km^2. However, *PDF* for all protocols is above 93% for all simulated node density values. Moreover, results of the conducted statistical analysis tests show that the differences in *APNH* between the three protocols and for each protocol separately are statistically insignificant. This suggests that the three protocols are highly effective in discovering and maintaining the shortest routes regardless of node density.
- Better performance in terms of *PNL* and *BNL* is obtained upon decreasing the zone size (increasing the number of zones). Decreasing the zone size results in decreasing the distance between the node and the nearest *LCA*, and accordingly, decreasing the number of nodes participating in forwarding the packets needed for updating nodes certificates and maintaining the network structure, i.e. significantly decreasing *PNL* and *BNL*. On the other hand, better performance in terms of *PRL*, *BRL* and *ARAL* is obtained with increasing the zone size (decreasing the number of zones). Increasing the zone size results in increasing the probability that the nearest *LCA*, upon receiving *PDP*, finds the destination in its authentication table. So, there is no need to communicate other *LCA*s, i.e. *PRL* and *BRL* slightly decrease and *ARAL* significantly decreases as the zone size increases. Accordingly, a moderate performance in terms of the five metrics is obtained upon dividing the area into four or nine zones.
- A higher node failure percentage results in a significant decrease in *PDF* and a slight increase in *PRL* and *BRL* for the three tested protocols because a higher probability of link break results in dropping some data packets, reinitiating *RDP* packets as well as selecting non-optimal paths. *ARANz* and *ARAN* protocols robustness against node failure is less than that for *AODV* due to having some nodes, such as *LCA*s in *ARANz* and centralized *CA* in *ARAN*, whose failure may affect other nodes in the network. The situation is worse in *ARAN* protocol since the failure of the *CA* will cause all

other nodes to be unable to either update their certificates or be included in the network operations. In *ARANz* however, only nodes inside a particular zone will not be capable of updating their certificates upon the failure of the four *LCA*s in that zone. Results of the conducted statistical analysis tests indicate that the increase in *PNL* and *BNL* for *ARAN* is more significant than *AODV* and *ARANz*, assuring that *AODV* and *ARANz* are more stable against node failure percentage.

- Increasing malicious node percentage results in decreasing *PDF* and/or increasing *PRL*, *BRL*, *MRP*, *PLP* and *FEP* for the three protocols. In most cases, however, the decrease or increase in these metrics is much slower upon using *ARANz*. This suggests that *ARANz* is efficient in identifying and isolating the malicious nodes.
- As malicious node percentage increases, *ARANz* effectiveness in distinguishing and isolating malicious nodes is increasingly demonstrated by achieving higher *CNP*. *ARANz* is efficient in identifying and isolating malicious nodes performing modification attack against control packets, black hole and grey hole attacks against data packets, *ERR* packets fabrication attack as well as multi-attack against control and data packets. Discovering malicious nodes and excluding them from future routes may result in reinitiating *RDP* packets and choosing non-optimal paths that do not contain malicious nodes within them, hence, causing higher *PML*, *BML*, *PRL*, *BRL* and *ARAL*.

As a summary, the simulation results illustrate the efficiency of the three protocols in discovering and maintaining not only routes, but also the shortest paths. The results suggest that *ARANz* has achieved the scalability issue by maintaining the minimum packet routing load even with large networks and high node mobility. *ARANz*’s reduced packet routing load is a normal result of using restricted directional flooding to send *RDP* packets. The cost of *ARAN* and *ARANz* security is higher routing load and latency in the route discovery process due to cryptographic computation that must occur.

Moreover, *ARANz* reduced packet routing load comes in the price of higher latency in the route discovery due to the time required to obtain destination's position.

5.5 Chapter Summary

In this chapter, a detailed discussion of our simulation methodology and scenarios has been presented. After that, the proposed protocol is evaluated and compared to other existing routing protocols considering the following performance parameters: node mobility speed, network size, node density, local communication percentage, zone size, node failure percentage and malicious nodes percentage. A wide range of performance metrics are used including: packet delivery fraction, average path number of hops, packet network load, byte network load, packet routing load, byte routing load, average route acquisition latency, average end-to-end delay of data packets, malicious route percentage, packet loss percentage, fabricated error packets, compromised node percentage, packet malicious load and byte malicious load.

The following chapter discusses the evaluated protocols and analyzes the obtained simulated results.

Chapter 6

Discussion

This chapter presents a discussion of the studied protocols along with an analysis of the results obtained via the simulated performance evaluation. *Section 6.1* discusses the evaluated protocols while *Section 6.2* summarizes the obtained results.

6.1 Discussion of the Evaluated Routing Protocols

AODV is a non-secure reactive routing protocol, hence it has less processing overhead compared to *ARAN* and *ARAN*z because nodes in *AODV* do not apply cryptographic operations, such as validating the previous node’s signature, signing the routing packets and appending certificates. *AODV* uses broadcasting to discover routes on-demand in the route discovery phase. This strategy increases *AODV*’s robustness against node failure on one hand, while on the other hand, broadcasting increases the packet overhead. Due to its packet overhead, *AODV* is considered as an unscalable protocol.

Like *AODV*, *ARAN* is a reactive routing protocol that uses broadcasting in the route discovery process. *ARAN* uses cryptographic certificates to prevent most of the attacks against Ad-Hoc routing protocols, such as impersonation, modification and fabrication of routing packets, as well as to detect erratic behaviours, such as the use of invalid certificates, improperly signed packets and misuse of some packets. However, the encryption/decryption processes along with route request broadcast increase the route acquisition latency as well as packet and processing overhead compared to *AODV*. *ARAN* also suffers from the centralized trust and load, i.e. a single point of attack and failure. The single *Certificate Authority (CA)* server can also be the operation bottleneck. Thus, using blind broadcasts to discover routes, applying encryption/decryption processes along with using one *CA* server contribute to scalability problem in *ARAN* protocol.

With *ARANz*, a scalable and secure solution can be achieved. Adopting the authentication methods used in *ARAN*, *ARANz* is a secure routing protocol. Additionally, by dealing with the network as zones and the use of restricted directional flooding, the new model aims to exhibit better scalability and performance. As opposed to *ARAN*, *ARANz* distributes load and trust by dividing the area into zones and introducing multiple certificate authorities (i.e. *Local CA*s (*LCA*s)) in each zone. Distributing load and trust helps in achieving the following:

- A high level of security by avoiding the single point of attack problem. In *ARANz*, the security of the network is compromised only if three *LCA*s in the same zone are compromised at the same time, which increases the availability of the system.
- A high level of robustness due to avoiding a single point of failure problem. A failure of a single *LCA* in *ARANz* does not affect node certificate update because other *LCA*s in its zone are able to detect the failure (via periodic certificate update process) and elect another *LCA* to replace it. This is unlike in *ARAN* where the *CA* is a vital part of the network and its failure prevents all nodes from updating their certificates.

The deployment of multiple *LCA*s in *ARANz* gives rise to the need to keep them synchronized. Moreover, there is a need to carry out new election operations to maintain the network structure since *LCA*s are able to move freely in and out of their zones. However, the communications that take place among nodes in *ARANz* to maintain the network structure and to update nodes' certificates and positions is minor compared to the overhead required for updating nodes' certificates in *ARAN*. In *ARAN* protocol, all certificate update request packets are broadcast to the entire network, since nodes are not aware of the *CA*'s position. In *ARANz*, however, most packets are sent using restricted directional flooding, source routing, zone flooding or *LCA* flooding.

Utilizing restricted directional flooding to send *RDP* packets in *ARANz* reduces packet routing load on one hand, and increases route discovery latency on the other hand, due to time required to inquiry about the destination's position. Moreover, using restricted directional flooding requires that nodes should be equipped with positioning instruments (such as *GPS*) or be able to use any other non-*GPS* positioning techniques to obtain their geographical position. This assumption is acceptable due to the recent availability of small, inexpensive and low-power positioning instruments.

Table 6.1 summarizes the properties of the evaluated routing protocols.

Table 6.1: Summary of the evaluated routing protocols

Protocol **Criterion**	***AODV***	***ARAN***	***ARANz***
Approach	Topology-based (reactive)	Topology-based (reactive)	Position-based (restricted directional flooding)
Secure extension for	-	*AODV*	*AODV*
Basic security mechanism	-	Certificates and timestamps	Certificates and timestamps
Synchronization	No	No	Yes
Central trust	No	Certificate Authority	No
Main idea/ contribution	Initiate a route discovery process only when the route is needed.	Protect routing packets against attacks from malicious nodes in managed-open environments.	Solve scalability as well as single point of compromise and failure problems existing in *ARAN* protocol.
Proposal	Uses next hop information stored in the nodes of the route with the least number-of-hop field.	• Provides authentication of route discovery, setup and maintenance. • Uses cryptographic certificates to prevent most security attacks faced by Ad-Hoc routing protocols. • Routing messages are authenticated at each hop from source to destination, as well as on the reverse path from the destination to the source.	• Divides area into zones and introduces multiple *LCA*s in each zone. • Requires sending a *PDP* if the position of the destination is unknown to the source. • Uses cryptographic certificates to prevent most Ad-Hoc security attacks. • Control messages are authenticated at each hop from source to destination, as well as on the reverse path from destination to source.

Table 6.1: Summary of the evaluated routing protocols (continued)

Protocol Criterion	*AODV*	*ARAN*	*ARANz*
Scalability	Low	Low	High
Advantages	No single point of failure, and so high robustness against nodes failure.	Robust against most security attacks.	• Robust against most security attacks. • No single point of compromise and failure, i.e. higher availability and robustness. • High scalability. • Reduced packet overhead.
Disadvantages	• Relies on blind broadcasts to discover routes, resulting in increased control overhead and decreased scalability. • May have security vulnerabilities.	• Single point of compromise and failure, and so low availability and robustness. • Scalability problem with the number of nodes inherited from *AODV* as well as increased packet overhead and route discovery delay compared to original *AODV* due to the encryption/decryption processes.	• Synchronization among *LCA*s. • Extra hardware (*GPS*). • Extra delay to inquiry about the destination’s position.

Now let us consider the security of *ARAN* and *ARANz* protocols. Both protocols introduce authentication, message integrity and non-repudiation as part of a minimal security policy for the Ad-Hoc environment. The centralized *CA* in *ARAN* protocol results in lower availability since the compromise of this *CA* affects the security of the entire network. Unlike *ARANz,* which distributes trust among multiple *LCA*s resulting in a higher level of availability due to avoiding single point of attack problem.

Basically, like *ARAN* protocol, *ARANz* uses cryptographic certificates to prevent most of the security attacks that Ad-Hoc routing protocols face, such as impersonation of other nodes and modification of routing packets. *ARAN* and *ARANz* encryption techniques lead to preventing passive attacks. Both protocols do not prevent fabrication of routing messages, but they offer a deterrent by ensuring non-repudiation since all routing messages must contain the sender's certificate and signature. In *ARANz*, a node that injects false messages into the network can be detected and excluded using the proposed misbehaviour detection scheme.

Secure forwarding of data packets is accomplished in *ARAN* and *ARANz* by preventing unauthorized nodes from participating in forwarding data. As discussed in *Section 2.5.6* and *Section 4.8.2*, there are many opportunities to achieve end-to-end integrity of data packets in both protocols. Some of these suggestions use the available cryptographic material or shared keys instantiated between neighbours during the route reply process. Additionally, nodes dropping data packets in *ARANz* are detected and excluded from future operations using the proposed misbehaviour detection scheme.

The following two tables summarize security requirements satisfied by *ARAN* and *ARANz* along with different attacks they defend against.

Table 6.2: Security requirements satisfied by *ARAN* and *ARANz* protocols

Protocol **Requirement**	***ARAN***	***ARANz***
Availability	Low	Medium
Authentication	Yes	Yes
Confidentiality	No	No
Integrity	Yes	Yes
Non-repudiation	Yes	Yes

Table 6.3: Robustness of *ARAN* and *ARANz* against existing attacks

Type	**Attack**	***ARAN***	***ARANz***
Passive attacks	Eavesdropping	Prevented using encryption techniques.	Prevented using encryption techniques.
Active attacks	Impersonation	Robust	Robust
	Fabrication	Not robust, but provides non-repudiation.	Not robust, but provides non-repudiation, hence, nodes fabricating packets can be excluded using the proposed misbehaviour detection scheme.
	Modification	Robust	Robust
Forwarding attacks	Modification	Robust	Robust
	Dropping	Not robust	Robust

Our analysis is backed by extensive simulations. The following section presents the list of key points extracted from the simulated performance evaluation.

6.2 Discussion of the Simulated Performance Evaluation

The performance evaluation of *ARANz* as well as a comparative analysis against *AODV* and *ARAN* protocols are presented in the previous chapter. Comparisons are conducted using *GloMoSim* simulator considering a wide range of performance metrics, parameters and scenarios.

The obtained results show that *PDF* for the three protocols is above 95% in most scenarios indicating that the three protocols are highly effective in discovering and maintaining routes for delivery of data packets even with relatively high node mobility and large area networks. Moreover, *APNH* is almost identical for the three protocols for a specified network parameters setting. Hence, it is obvious that *ARAN* and *ARANz* are as efficient as *AODV* in discovering the shortest paths.

Results also suggest that the three protocols are highly effective in discovering and maintaining the shortest routes regardless of node density. Additionally, upon studying the effect of zone size, a moderate performance in terms of *PNL*, *BNL PRL, BRL* and *ARAL* is obtained in *ARANz* upon dividing the area into four or nine zones.

Results also assure that *ARANz* achieves the minimum *PRL*, *PNL* and *BNL* in all experiments and the conducted statistical analysis tests confirm that the differences in these metrics for the three protocols are statistically significant. The main reason behind this gap is utilizing restricted directional flooding to send most *ARANz*'s packets.

AODV is superior in its *ARAL* due to the shortest processing delay at each node. On the other hand, signature generation and verification in *ARAN* and *ARANz* result in additional delay at each hop, and so *ARAL* increases. Moreover, *ARANz* incurs higher *ARAL* because of the destination's position discovery step. However, upon increasing local communications, *ARANz*’s *ARAL* significantly decreases since the position of the destination is found in the authentication table of the nearest *LCA* to the source, so there is no need to communicate with *LCA*s in other zones. Moreover, the differences in *AEED* between the three protocols are almost negligible since the number of route discoveries and position enquiries performed is limited compared to the number of data packets delivered. Hence, the effect of *ARAL* on *AEED* of data packets is not significant.

A higher node failure percentage significantly decreases the *PDF* and slightly increases the *PRL* and *BRL* for the three tested protocols because a higher probability of link break results in dropping some data packets, reinitiating *RDP* packets and selecting non-optimal paths. *ARANz* and *ARAN* protocols robustness against node failure is less than *AODV* due to having some nodes, such as *LCA*s in *ARANz* and centralized *CA* in *ARAN*, whose failure may affect other nodes in the network. The situation is worse in *ARAN* protocol since the failure of the *CA* will prevent all other nodes from updating their certificates and participating in the network operations.

The simulation results illustrate that increasing malicious node percentage results in decreasing *PDF* and/or increasing *PRL*, *BRL*, *MRP*, *PLP* and *FEP* for the three protocols. However, in most cases, the decrease or increase in these metrics is considerably slower upon using *ARANz* suggesting that *ARANz* is efficient in identifying malicious nodes.

As malicious node percentage increases, *ARANz*’s effectiveness in distinguishing and isolating malicious nodes is increasingly demonstrated by achieving higher *CNP*. The obtained results assure that *ARANz* is efficient in identifying and isolating malicious nodes performing modification attack against control packets, black hole and grey hole attacks against data packets, *ERR* packets fabrication attack as well as multi-attack against control and data packets.

6.3 Chapter Summary

In this chapter, a comparison among the evaluated protocols has been presented. After that, an analysis of the results obtained via simulated performance evaluation is provided.

The summary of research findings, conclusion and future work are given in *Chapter 7*.

Chapter 7

Conclusion and Future Work

This chapter presents the concluding remarks together with some of the potential future research areas. *Section 7.1* summarizes the thesis and presents the drawn conclusions. *Section 7.2* discusses our main contributions. In *Section 7.3*, we highlight the key features of *ARANz*. Last but not least, *Section 7.4* provides a few key directions to improve this work.

7.1 Thesis Summary

Routing protocol in Ad-Hoc networks is a fundamental part of network infrastructure that supports the delivery of packets. It has to face the challenge of link instability, frequently changing topology, absence of a fixed infrastructure and low transmission power. Additionally, it is a central aspect to secure routing protocols against attacks, such as eavesdropping, spoofing, misdirection and generating deceptive routing messages. Moreover, wireless networks are generally more susceptible to physical security risks than wired networks.

Without the existence of online trusted certificate authority, it is difficult to be aware of the honesty of different nodes participating in the network. Moreover, the certificate authority should be distributed among multiple servers since having one centralized server is not practical in an Ad-Hoc network as compromising or destroying this server may result in having the whole system broken down.

The need for scalable and energy-efficient routing protocols along with the availability of small, inexpensive and low-power positioning instruments result in making position-based routing protocols a promising choice for mobile Ad-Hoc networks.

For the aforementioned reasons, routing in Ad-Hoc networks is a difficult task to accomplish efficiently, robustly and securely. Hence, this work has concentrated on developing a scalable distributed position-based routing protocol for Ad-Hoc networks while keeping security issues in mind.

In *Chapter 1*, we introduce Ad-Hoc networks considering their applications, characteristics and challenges as well as highlighting our research significance, objectives, scope and expected outcomes. *Chapter 2* addresses the existing routing protocols for Ad-Hoc networks along with their different techniques and categories. In this chapter, we also provide an analysis of the threats against these protocols and the requirements that need to be addressed to secure them.

The literature review presented in *Chapter 2* shows that position-based Ad-Hoc routing protocols have better routing performance than traditional topology-based Ad-Hoc routing protocols, such as *Dynamic Source Routing (DSR)* (Johnson & Maltz 1996) and *Ad-Hoc On-demand Distance Vector (AODV)* (Perkins & Royer 1999), in terms of end-to-end throughput and network scalability. However, most of these protocols use greedy forwarding which suffers from congestion, nodes' energy consumption and uncertainty of finding the optimal route. It is also clear that restricted directional flooding position-based routing protocols have better performance than greedy ones in regards to finding the shortest path. However, both of them are vulnerable to some attacks, as they focus on improving performance while ignoring security issues. Even secure ones (such as *Secure Position-Aided Ad-Hoc Routing (SPAAR)* (Carter & Yasinsac 2002), *Anonymous On-Demand Position-based Routing in Mobile Ad-Hoc Networks (AODPR)* (Mizanur Rahman et al. 2006) and *Secure Geographic Forwarding (SGF)* (Song et al. 2007)) have many weaknesses. These disadvantages include the single point of failure and attack, increased packet and processing overhead and scalability problems.

Considering the conclusions drawn from the conducted literature review, a new model of hierarchal and distributed routing protocol called *ARANz* is proposed in this work. *ARANz* design details and analysis are discussed in *Chapter 4*. *ARANz* is proposed to be implemented in managed-open environments, where it is possible to use already established infrastructure. *ARANz* aims to improve performance of the routing protocol and distribute routing load by dividing the area into zones. It seeks to achieve robustness and a high level of security, solve the single point of failure problem and avoid a single point of attack problem by distributing trust among multiple certificate authority servers.

ARANz aspires to exhibit better scalability and performance by taking advantage of the restricted directional flooding position-based routing protocols. So, in conjunction with the chosen routing strategy, a distributed location service has been proposed. Additionally, a misbehaviour detection system has been suggested to help in identifying malicious nodes in order to exclude them from future communications.

Due to a large number of nodes and a large geographical area of Ad-Hoc networks, a simulation tool is used to evaluate the new protocol and explore the impact of different parameters on its performance. The performance of *ARANz* is tested and compared to *AODV* and *ARAN* protocols using the *GloMoSim* simulator. Our research methodology, discussion of different existing simulators and justifications of choosing *GloMoSim* simulator have been presented in *Chapter 3*.

Simulation results are presented in *Chapter 5* and discussed in *Chapter 6*. The results demonstrate that the packet delivery fraction obtained for the three protocols is above 95% in most scenarios. This percentage illustrates that the three protocols are highly effective in discovering and maintaining routes for delivery of data packets, even with relatively high node mobility and large area networks. Moreover, the average path number of hops for *ARANz* is nearly the same as that for *AODV*, meaning that data packets travel along the shortest path.

Experimental results also indicate that *ARANz* has overcome the scalability issue by maintaining the minimum packet routing load even with large networks and high nodes mobility. *ARANz* has the lowest packet routing load in all experiments and significantly decreases with increasing the local communication percentage. *ARANz*'s reduced packet routing load is a natural result of sending route discovery packets using restricted directional flooding towards the destination.

Moreover, packet and byte network load for *ARANz* are much less than those for *ARAN* since most packets in *ARANz* are sent using restricted directional flooding, source routing, zone flooding or *LCA* flooding. Additionally, as malicious node percentage increases, *ARANz* shows its effectiveness in distinguishing and isolating malicious nodes performing modification attack against control packets, black hole and grey hole attacks against data packets, error packets fabrication attack as well as multi-attack against control and data packets.

ARANz's increased packet delivery fraction, decreased packet routing load, efficiency in discovering and maintaining routes and effectiveness in isolating malicious nodes come at the cost of three things. The first, using larger routing packets and higher route discovery latency due to the cryptographic computations, such as signature generation and verification at each hop, which is also found in *ARAN* protocol.

The second, *ARANz* has the highest average route acquisition latency since it needs to carry out a destination position discovery step. However, this latency improves rapidly as more and more packets are sent between nodes inside the same zone. Moreover, the difference in average end-end delay of data packets between the three protocols are almost negligible since the number of route discoveries and position enquiries performed is limited compared to the number of data packets delivered.

Finally, discovering malicious nodes and excluding them from future routes may result in reinitiating route discovery packets and choosing non-optimal paths that do not have malicious nodes, hence causing a slight increase in packet routing load, byte routing load and average route acquisition latency.

7.2 Research Contributions

In this thesis we have made five major contributions, they are summarized as follows:

- Proposing a new model of scalable, distributed and secure position-based routing protocol, *ARANz*, to be implemented in managed-open environments.
- Proposing a distributed location service to be incorporated with *ARANz*.
- Suggesting a misbehaviour detection system to help in identifying malicious nodes in order to exclude them from future communications.

7.3 Key Features of *ARANz*

In this thesis a new model of routing protocol, *ARANz*, has been proposed for managed-open environments. *ARANz* incorporates the following ideas:

- Adopting the authentication methods used in *ARAN* protocol,
- Dividing the network area into multiple zones,
- Introducing multiple *LCA*s in each zone,
- Utilizing restricted directional flooding.

Hence *ARANz* has the following properties:

- Secure: adopting the authentication methods used in *ARAN*, *ARANz* is a secure routing protocol. *ARANz* security is increased by distributing trust among multiple certificate authority servers which helps in avoiding the single point of attack problem and increasing the protocol availability. In *ARANz,* the security of the network is compromised only if three *LCA*s in the same zone are compromised at the same time. Moreover, utilizing the proposed misbehaviour detection system helps in achieving a higher level of security by identifying malicious nodes and excluding them from future communications.
- Distributed: *ARANz* distributes load and trust by dividing the area into zones and introducing multiple *Local Certificate Authorities (LCAs)* in each zone. Distributing load and trust helps in achieving a high level of security by avoiding single point of

attack problem as well as a high level of robustness due to avoiding single point of failure problem.

- Position-based: *ARANz* utilizes the idea of restricted directional flooding position-based routing protocols. Using restricted directional flooding helps in exhibiting better scalability and performance.
- Scalable: the new model reveals better scalability and performance by dealing with the network as zones as well as using restricted directional flooding. On one hand, dividing an area into multiple zones and distributing load among multiple certificate authority servers result in achieving a high level of scalability and improving performance. In other words, instead of assigning the responsibility of managing nodes' certificates and positions to one centralized node, resulting in exhausting its resources, *LCA*s of a particular zone are only responsible for the nodes in their zone. On the other hand, utilizing the idea of restricted directional flooding results in achieving better scalability and performance since route discovery packets are not broadcast to the entire network, instead, only nodes toward the destination participate in forwarding these packets.

7.4 Future Work

In this section, we suggest some improvements specific to our protocol as well as discussing some open issues facing Ad-Hoc networks in general.

The research presented in this thesis serves as a starting point for future research. First of all, more research is needed in order to comprehensively evaluate *ARANz* protocol performance. For example, *ARANz* performance can be studied under different mobility models and different traffic generation applications.

Second, increased refinement of the routing protocols for Ad-Hoc networks is always possible. The following are some points that can be considered to improve and extend our protocol:

- Other zone shapes and different number of *LCA*s in each zone may be studied.
- On the subject of misbehaviour identification and mitigation, the misbehaviour detection system can be improved to detect other types of attacks.
- As with other position-based routing protocols, there is a possibility of finding other techniques to allow nodes to be aware of their positions without using *GPS*.
- Another area for research in *ARANz* is studying the ability of improving it to be suitable for implementation in 3-Dimensional environments.
- Finally, *ARANz* can be implemented and tested in real life, however, this will require a large number of nodes and broad geographical areas to test its scalability.

Third, in our approach, we focus on security issues in Ad-Hoc routing protocols. However, there are still open issues and interesting challenges facing Ad-Hoc networks which worth investigating in future work. These issues include:

- The design of multicast routing protocols,
- The development of a multipath routing approach,
- The design of power-efficient protocols,
- The development of *MAC* layer protocols,
- Provision of end-to-end *Quality-of-Service (QoS),*
- Cross-layer design for wireless networks.

Bibliography

Abolhasan, M., Wysocki, T. & Dutkiewicz, E. 2004, 'A review of routing protocols for mobile ad hoc networks', *Ad Hoc Networks*, Vol. 2, No. 1, pp. 1-22, Elsevier.

Abrougui, K., Boukerche, A. & Pazzi, R. 2011, 'Design and Evaluation of Context-Aware and Location-Based Service Discovery Protocols for Vehicular Networks', *IEEE Transactions On Intelligent Transportation Systems*, Vol. 12, No. 3, pp. 717-735.

Al-Rabayah, M. & Malaney, R. 2011, 'Scalable Hybrid Location-based Routing in Vehicular Ad Hoc Networks', *proceedings of the 74th IEEE Vehicular Technology Conference*, 5-8 September, San Francisco, California, United States, pp. 1-5.

Arpacioglu, O., Small, T. & Haas, Z. 2003, 'Notes on Scalability of Wireless Ad hoc Networks', Internet Draft, Internet Engineering Task Force, Ad hoc Network Scalability Working Group.
Available at: http://draft-irtf-ans-scalability-definition-01.txt.

Bagrodia, R. & Liao, W. 1994, 'Maisie: A language for the design of efficient discrete-event simulations', *IEEE Transactions in Software Engineering*, Vol. 20, No. 4, pp. 225-238.

Bagrodia, R., Meyer, R., Takai, M., Chen, Y., Zeng, X., Martin, J. & Song., H. 1998, 'Parsec: A parallel simulation environment for complex systems', *IEEE Computer*, Vol. 31, No. 10, pp. 77-85.

Bahl, P. & Padmanabhan, V. 2000, 'RADAR: An In-Building RF-based User Location and Tracking System', *proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, 26-30 March, Tel Aviv, Israel, Vol. 2, pp. 775-784.

Bajaj, L., Takai, M., Ahuja, R., Tang, K., Bagrodia, R. & Gerla, M. 1999, 'Glomosim: A scalable network simulation environment', Technical Report 990027, Computer Science Department, University of California, Los Angeles.
Available at: nrlweb.cs.ucla.edu/publication/download/178/glomosim__a_scalable_network_simulation__2895.pdf

Barba, C., Aguirre, K. & Igartua, M. 2010, 'Performance Evaluation of a Hybrid Sensor and Vehicular Network to Improve Road Safety', *proceedings of the 7th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN 2010)*, 17-18 October, Bodrum, Turkey, pp. 71-78.

Barr, R. 2004, 'An efficient, unifying approach to simulation using virtual machines', Ph.D. thesis, Faculty of the Graduate School, Cornell University.
Available at: jist.ece.cornell.edu/docs/040517-thesis.pdf

Beijar, N. 1998, 'Zone Routing Protocol (ZRP)', Networking Laboratory, Helsinki University of Technology, Finland.
Available at: http://www.netlab.hut.fi/opetus/s38030/k02/Papers/08-Nicklas.pdf

Bettstertter, C. & Konig, S. 2002, 'On the Message and Time Complexity of a Distributed Mobility-Adaptive Clustering Algorithm in Wireless Ad Hoc Networks', *proceedings of the 4th European Wireless conference*, 25-28 February, Florence, Italy, pp. 128-134.

Blazevic, L., Buttyan, L., Capkum, S., Giordano, S., Hubaux, J. & Le Boudec, J. 2001, 'Self-organization in mobile ad-hoc networks: the approach of terminodes', *IEEE Communication Magazine*, Vol. 39, No. 6, pp. 166-174.

Bur, K. & Ersoy, C. 2006, 'Ad Hoc Quality of Service Multicast Routing with Objection Queries for Admission Control', *European Transactions on Telecommunications,* Vol. 17, No. 3, pp. 561-576.

Buruhanudeen, S., Othman, M., Othman, M. & Mohd Ali, B. 2007, 'Mobility Models, Broadcasting Methods and Factors Contributing Towards the Efficiency of the MANET Routing Protocols: Overview', *proceedings of the 2007 IEEE International Conference on Telecommunications and Malaysia International Conference on Communications*, 14-17 May, Penang, Malaysia, pp.226-230.

Camp, T., Boleng, J. & Davies, V. 2002, 'A survey of mobility models for ad hoc network research', *Wireless Communications and Mobile Computing,* Vol. 2, No. 5, pp. 483-502.

Campbell, R., Al-Muhtadi, J., Naldurg, P., Sampemane, G. & Mickunas, M. 2002, 'Towards Security and Privacy for Pervasive Computing', *proceedings of the International Symposium on Software Security (ISSS 2002),* 8-10 November, Tokyo, Japan, pp. 1-15.

Canetti, R., Song, D., Perrig, A. & Tygar, D. 2001, 'Efficient and secure source authentication for multicast', *proceedings of the ISOC Symposium on Network and Distributed System Security (NDSS 2001*), 8-9 February, San Diego, California, United States, pp. 35-46.

Cao, Y. & Xie , S. 2005, 'A Position Based Beaconless Routing Algorithm for Mobile Ad hoc Networks', *proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS 2005),* 27-30 May, Hong Kong, China, Vol. 1, pp. 303-307, IEEE.

Capkun, S. & Hubaux, J. 2005, 'Secure positioning of wireless devices with application to sensor networks', *proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005),* 13-17 March, Miami, Florida, United States, Vol. 3, pp. 1917-1928.

Carter, S. & Yasinsac, A. 2002, 'Secure Position Aided Ad Hoc Routing', *proceedings of the IASTED International Conference on Communications and Computer Networks (CCN 2002),* 4-6 November, Cambridge, United Kingdom, pp. 329-334.

Cavin, D., Sasson, Y. & Schiper, A. 2002, 'On the accuracy of MANET simulators', *proceedings of the second ACM international workshop on Principles of mobile computing,* 30-31 October, Toulouse, France, pp. 38-43.

Cheng, M. & Li, D. 2008, *Advances in Wireless Ad Hoc and Sensor Networks,* ISBN 978-0-387-68565-6, Signals and Communication Technology, Springer, United States.

Chiang, C., Wu, H., Liu, W. & Gerla, M. 1997, ‘Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel’, *proceedings of the IEEE Singapore International Conference on Networks (SICON 1997)*, 16-17 April, Singapore, pp. 197-211.

Chimphlee, W., Abdullah, A., Sap, M., Chimphlee, S. & Srinoy, S. 2007, ‘A Rough-Fuzzy Hybrid Algorithm for Computer Intrusion Detection’, *International Arab Journal of Information Technology*, Vol. 4, No. 3, pp. 247-254.

Christiansen, H., Kuijpers, G., Yomo, H. & Fathi, H. 2003, ‘Simulator requirements, comparative evaluation of tools’, Technical University of Denmark, CNTK.
Available at: http://nsl.csie.nctu.edu.tw/NCTUnsReferences/Simulation%20tool%20evaluation_v21.doc

Corson, S. & Macker, J. 1999, ‘Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations’, Request for Comments 2501, Internet Engineering Task Force, Network Working Group.
Available at: http://www.ietf.org/rfc/rfc2501.txt

Corson, M., Macker, J. & Batsell, S. 1996, ‘Architectural Considerations for Mobile Mesh Networking’, *proceedings of the Military Communications Conference (MILCOM 1996)*, 21-24 October, McLean, Virginia, United States, Vol. 1, pp. 225-229, IEEE.

Desbrandes, F., Bertolotti, S. & Dunand, L. 1993, ‘Opnet 2.4: An environment for communication network modeling and simulation’, *proceedings of the European Simulation Symposium. Society for Computer Simulation (ESS 1993)*, 25-28 October, Delft, Netherlands, pp. 64-74.

Dutta, R. & Dowling, T. 2011, ‘Provably secure hybrid key agreement protocols in cluster-based wireless ad hoc networks’, *Ad Hoc Networks*, Vol. 9, No. 1, pp. 767-787.

El-Rabbany, A. 2002, *Introduction to GPS: the Global Positioning System*, ISBN 1580531830, 1st edition, (Chapter 1, pp.1-11), Artech House Publishers.

Eriksson, J. 2006, ‘Protocols for Routing Scalability and Security in Open Networks’, Ph.D. thesis, Computer Science Department, University of California, Riverside, California, United States.
Available at: http://nms.csail.mit.edu/~jakob/pubs/eriksson_thesis.pdf

Estrin, D., Handley, M., Heidemann, J., McCanne, S., Xu, Y. & Yu, H. 1999, ‘Network visualization with the VINT Network Animator Nam’, Technical Report 99-703b, USC Computer Science Department.
Available at: http://www.isi.edu/~johnh/PAPERS/Estrin99d.pdf

Farkas, K., Budke, D., Plattner, B., Wellnitz, O. & Wolf, L. 2006, ‘QoS Extensions to Mobile Ad Hoc Routing Supporting Real-Time Applications’, *proceedings of the 4th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-06)*, 8-11 March, Dubai, United Arab Emirates, pp.54-61.

Fernandes, N. & Duarte, O. 2011, ‘A lightweight group-key management protocol for secure ad-hoc-network routing’, *Computer Networks*, Vol. 55, No. 3, pp. 759-778, Elsevier.

Fisher, R. & Yates, F. 1974, *Statistical Tables for Biological Agricultural and Medical Research*, ISBN 0582445256, 6th edition, Longman Group Ltd. London.

Fonseca, E. & Festag, A. 2006, ‘A Survey of Existing Approaches for Secure Ad Hoc Routing and Their Applicability to VANETS’, NEC Technical Report NLE-PR-2006-19, NEC Network Laboratories.
Available at: http://www.network-on-wheels.de

Frey, H., Gorgen, D., Lehnert, J. & Sturm, P. 2003, ‘A java-based uniform workbench for simulating and executing distributed mobile applications’, *FIDJI 2003 International Workshop on Scientific Engineering of Distributed Java Applications*, 27-28 November, Kirchberg, Luxembourg, Vol. 2952 , pp. 116-127.

Garcia-Luna-Aceves, J. & Spohn, M. 1999, ‘Source-Tree Routing in Wireless Networks’, *proceedings of the IEEE International Conference on Network Protocols (ICNP1999),* 31 October - 3 November, Toronto, Canada, pp. 273-282.

Gera, P., Garg, K. & Misra, M. 2011, ‘Eliminating Misbehaving nodes by Opinion Based Trust Evaluation Model in MANETs’, *proceedings of the International Conference on Communication, Computing & Security (ICCCS 2011),* 12-14 February, Odisha, India, pp. 50-55.

Gerla, M. 2005, ‘Scalable Ad hoc Networking Routing: From Battlefield to Vehicle Grids and Pervasive Computing’, *proceedings of the 4th Annual Mediterranean Workshop on Ad Hoc Networks*, 21-24 June, Ile de Porquerolles, France, pp. 355-365.

Ghinita, G., Azarmi, M. & Bertino, E. 2010, ‘Privacy-Aware Location-Aided Routing in Mobile Ad Hoc Networks’, *proceedings of the 11th International Conference on Mobile Data Management (MDM 2010),* 23-26 May, Kansas City, Missouri, United States, pp. 65-74.

Giordano, S., Stojmenovic, I. & Blazevic, L. 2003, ‘Position Based Routing Algorithms for Ad Hoc Networks: A Taxonomy’. In Cheng, X., Huang, X., Du, D.Z., *Ad hoc wireless Networking*, (pp. 103-136), Kluwer.
Available at: http://www.site.uottawa.ca/~ivan/routing-survey.pdf

Giruka, V. & Singhal, M. 2005, ‘Angular Routing Protocol for Mobile Ad-hoc Networks’, *proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW 2005),* 6-10 June, Columbus, Ohio, United States, pp. 551-557.

Goyal, P., Batra, S. & Singh, A. 2010, ‘A Literature Review of Security Attack in Mobile Ad-hoc Networks’, *International Journal of Computer Applications*, Vol. 9, No. 12, pp. 11-15.

Hightower, J., Vakili, C., Borriello, G. & Want, R. 2001 ‘Design and calibration of the SpotON ad-hoc location sensing system’, University of Washington, Seattle, Washington, United States.
Available at: http://seattle.intel-research.net/people/jhightower/pubs/hightower2001design/hightower2001design.pdf

Hogie, L., Bouvry, P. & Guinand, F. 2006, 'An Overview of MANETs Simulation', *Electronic Notes in Theoretical Computer Science*, Vol. 150, No. 1, pp. 81-101, Elsevier.

Hong, X., Xu, K. & Gerla, M. 2002, 'Scalable Routing Protocols for Mobile Ad Hoc Networks', *IEEE network*, Vol. 16, No. 4, pp. 11-21.

Hu, Y., Johnson, D. & Perrig, A. 2002, 'SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks', *proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002)*, 20-21 June, Callicoon, New York, United States, pp. 3-13.

Hu, Y., Perrig, A. & Johnson D. 2003, 'Packet leashes: a defense against wormhole attacks in wireless network', *proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, 30 March - 3 April, San Francisco, California, United States, Vol. 3, pp. 1976-1986.

Hu, Y., Perrig, A. & Johnson, D. 2002, 'ARIADNE: A Secure On-Demand Routing Protocol for Ad Hoc Networks', *proceedings of the ACM International Conference on Mobile Computing and Networking (MOBICOM 2002)*, 23-28 September, Atlanta, Georgia, United States, pp. 12-23.

Imre, S., Keszei, Cs., Hollos, D., Barta, P. & Kujbus, Cs. 2001, 'Simulation environment for ad-hoc networks in omnet++', *proceedings of the IST Mobile Summit conference 2001*, 9-12 September, Sitges, Barcelona, Spain, pp.135-140.

Inn, E. 2006, 'Mobility-Adaptive Clustering and Network-Layer Multicasting In Mobile Ad Hoc Networks', Ph.D. thesis, National University Of Singapore, Singapore. Available at: http://scholarbank.nus.edu/bitstream/handle/10635/16249/ErII-Thesis.pdf?sequence=1

Izuan, M. & Zukarnain, Z. 2009, 'Performance Analysis of Random-Based Mobility Models in MANET Routing Protocol', *European Journal of Scientific Research*, Vol. 32 , No. 4, pp. 444-454.

Jacobsson, M., Niemegeers, I. & Groot, S. 2010, *Personal Networks: Wireless Networking for Personal Devices*, ISBN 978-0-470-68173-2, (Chapter 3, pp. 23-35), John Wiley and Sons.

Jacquet, P., Muhlethaler, P., Qayyum, A., Laouiti, A., Viennot, L. & Clausen, T. 2003, 'Optimized Link State Routing Protocol (OLSR)', Request for Comments 3626, Internet Engineering Task Force, Network Working Group.
Available at: http://www.ietf.org/rfc/rfc3626.txt

Jhaveri, R., Patel, A., Parmar, J. & Shah, B. 2010, 'MANET Routing Protocols and Wormhole Attack against AODV', *International Journal of Computer Science and Network Security*, Vol. 10, No. 4, pp. 12-18.

Johnson, D. 1994, 'Routing in Ad Hoc Networks of Mobile Hosts', *proceedings of the 1st Workshop on Mobile Computing Systems and Applications*, 8-9 December, Santa Cruz, California, United States, pp. 158-163, *IEEE*.

Johnson, D. & Maltz, D. 1996, ‘Dynamic Source Routing in Ad Hoc Wireless Networks’. In Imielinski, T., Korth, H., *Mobile Computing*, ISBN 0792396979, (pp.153-181), Kluwer Academic Publishers, United States.

Joshi, P. 2011, ‘Security issues in routing protocols in MANETs at network layer’, *Procedia Computer Science*, Vol. 3, No. 1, pp. 954-960.

Kadono, D., Izumi, T., Ooshita, F., Kakugawa, H. & Masuzawa, T. 2010, ‘An ant colony optimization routing based on robustness for ad hoc networks with GPSs’, *Ad Hoc Networks*, Vol. 8, No.1, pp. 63-76, Elsevier.

Kalhor, S., Anisi, M. & Haghighat, A. 2007, ‘A new position-based routing protocol for reducing the number of exchanged route request messages in Mobile Ad-hoc Networks’, *proceedings of the 2nd International Conference on Systems and Networks Communications (ICSNC 2007*), 25-31 August, Cap Esterel, French Riviera, France, pp. 13, IEEE.

Kaplan, E. & Hegarty, C. 2005, *Understanding GPS: principles and applications*, ISBN 1580538940, 2nd Edition, (Chapter 1, pp.1-19), Artech House, Boston, Massachusetts, United States.

Karp, B. & Kung, H. 2000, ‘GPSR: Greedy Perimeter Stateless Routing for Wireless Networks’, *proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 2000*), 6-11 August, Boston, Massachusetts, United States, pp. 243-254.

Khokhar, R., Ngadi, A. & Mandala, S. 2008, ‘A Review of Current Routing Attacks in Mobile Ad Hoc Networks’, *International Journal of Computer Science and Security*, Vol. 2, No. 3, pp. 18-29.

Kim, B., Jung, H., Yoo, J., Lee, W., Park, C. & Ko, Y. 2008, ‘Design and Implementation of Cricket-based Location Tracking System’, *proceedings of the World Academy of Science, Engineering and Technology (WASET)*, 25-27 April, Rome, Italy, Vol. 28, pp. 96-100.

Kirkman, T. 1996, ‘Statistics to Use’.
Available at: http://www.physics.csbsju.edu/stats/

Klein, M. 2003, ‘Dianemu: A java based generic simulation environment for distributed protocols’. Technical Report 2003-7, University of Karlsruhe, Faculty of Informatics, Germany.
Avialble at: http://fusion.cs.uni-jena.de/professur/publications/2003/29-dianemu---a-java-based-generic-simulation

Ko, Y. & Vaidya, N. 2000, ‘Location-Aided Routing (LAR) in mobile ad hoc networks’, *Wireless Network (WINET)*, Vol. 6, No. 4, pp. 307-321, ACM.

Koutsonikolas, D., Das, S., Hu, Y. & Stojmenovic, I. 2010, ‘Hierarchical geographic multicast routing for wireless sensor networks’, *ACM Wireless Networks*, Vol. 16, No. 2, pp.449-466.

Kranakis, E., Singh, H. & Urrutia, J. 1999**,** ‘Compass Routing on Geometric Networks’, *proceedings of the 11th Canadian Conference on Computational Geometry*, 15-18 August, Vancouver, British Columbia, Canada, pp. 51-54.

Krawczyk, H., Bellare, M. & Canetti, R. 1997, 'HMAC: keyedhashing for message authentication', Request for Comments 2104, Internet Engineering Task Force, Network Working Group.
Available at: http://www.ietf.org/rfc/rfc2104.txt

Kwak, B., Song, N. & Miller, L. 2004, 'On the Scalability of Ad Hoc Networks', *IEEE Communications Letters*, Vol. 8, No. 8, pp. 503-505.

Lakshmikanth, G., Gaiwak, A. & Vyavahare, P. 2008, 'Simulation Based Comparative Performance Analysis of Adhoc Routing Protocols', *proceedings of the Region 10 Conference 2008 (TENCON 2008)*, 19-21 November, Hyderabad, India, pp. 1-5, IEEE.

Lee, D., Choi, S., Choi, J. & Jung, J. 2007, 'Location-Aided Secure Routing Scheme in Mobile Ad Hoc Networks', *proceedings of the International Conference Computational Science and Its Applications (ICCSA 2007)*, Part II, 26-29 August, Kuala Lumpur, Malaysia, pp. 131-139.

Lee, D., Jang, B. & Jung, J. 2008, 'An Enhanced Next Hop Selection Scheme for Ad HocWireless Networks', *proceedings of the International Conference on Information Science and Security (ICISS 2008)*, 10-12 January, Seoul, Korea, pp. 66-71.

Lee, S., Belding-Royer, E. & Perkins, C. 2003, 'Scalability Study of the Ad hoc On-Demand Distance Vector Routing Protocol', *Wiley International Journal of Network Management*, Vol. 13, No. 2, pp. 97-114.

Lee, Y., Kim, H., Chung, B., Lee, J. & Yoon, H. 2003, 'On-demand Secure Routing Protocol for Ad Hoc Network using ID based Cryptosystem', *proceedings of the 4th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2003)*, 27-29 August, Chengdu, China, pp. 211-215, IEEE.

Lehnert, J., Gorgen, D., Frey, H. & Sturm, P. 2004, 'A scalable workbench for implementing and evaluating distributed applications in mobile ad hoc networks', *proceedings of the Western Simulation MultiConference (WMC 2004)*, 18-22 January, San Diego, California, United States, pp. 154-161.

Li, H. & Singhal, M. 2006, 'A Secure Routing Protocol for Wireless Ad Hoc Networks', *proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS 2006)*, 4-7 January, Kauai, Hawaii, United States, Vol. 9, pp. 225a - 225a, IEEE.

Li, J., Jannotti, J., De Couto, D., Karger, D. & Morris, R. 2000, 'A scalable location service for geographic ad hoc routing', *proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 2000)*, 6-11 August, Boston, Massachusetts, United States, pp. 120-130.

Li, X., Nayak, A., Ryl, I., Simplot, D. & Stojmenovic, I. 2007, 'Secure Mobile Ad hoc Routing', *proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW 2007)*, 21-23 May, Niagara Falls, Ontario, Canada, pp. 737-742.

Li, Z., Barria, J. & Lent, R. 2008, ‘Evaluation of the LDR protocol in a hybrid sensor networks/MANET architecture’, *proceedings of the Institution of Engineering and Technology Seminar on Target Tracking and Data Fusion: Algorithms and Applications*, 15-16 April, Birmingham, United Kingdom, pp. 161-166.

Lim, T. & Lakshminarayanan, A. 2007, ‘On the Performance of Certificate Validation Schemes Based on Pre-Computed Responses’, *proceedings of the Global Telecommunications Conference (GLOBECOM 2007)*, 26-30 November, Washington, District of Columbia, United States, pp. 182-187.

Lin, T. 2004, ‘Mobile Ad-hoc Network Routing Protocols: Methodologies and Applications’, Ph.D. thesis, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia.
Available at: http://scholar.lib.vt.edu/theses/available/etd-03262004-144048/unrestricted/Tao_PhD_Dissertation.pdf

Lorincz, K. & Welsh, M. 2007, ‘MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking’, *Personal and Ubiquitous Computing*, Vol. 11, No. 6, pp. 489-503.

Mahmoud, A. 2005, ‘Reputed Authenticated Routing for Ad Hoc Networks Protocol (Reputed-ARAN)’, Master thesis, Computer Science Department, The American University in Cairo.
Available at: www.mbifoundation.com/media/18949/Abdalla%20Mahmoud%20-%20Thesis%20Defense.pdf

Mahmudul Islam, M., Pose, R. & Kopp, C. 2008, ‘Security in Ad-Hoc Networks’. In Karmakar, G., Dooley, L., *Mobile Multimedia Communications: Concepts, Applications and Challenges*, ISBN 978-1-59140-766-9, (pp. 296-325), Information Science Reference, Hershey, Pennsylvania, United States.

Malla, D. 2005, ‘Optimal Zonal Protocol for Containing Rebroadcast in Mobile Ad-Hoc Networks’, Master thesis, College of Engineering, Wichita State University.
Available at: http://soar.wichita.edu/dspace/bitstream/handle/10057/743/t05013.pdf

Mandala, S., Ngadi, A. & Abdullah, A. 2008, ‘A Survey on MANET Intrusion Detection’, *International Journal of Computer Science and Security*, Vol. 2, No. 1, pp. 1-11.

Manikandan, K., Satyaprasad, R. & Rajasekhararao, K. 2011, ‘A Survey on Attacks and Defense Metrics of Routing Mechanism in Mobile Ad hoc Networks’, *International Journal of Advanced Computer Science and Applications*, Vol. 2, No.3, pp. 7-12.

Mauve, M., Widmer, J. & Hartenstein, H. 2001, ‘A Survey on Position-Based Routing in Mobile Ad-Hoc Networks’, *IEEE Network*, Vol. 15, No. 6, pp. 30-39.

McCanne, S. & Floyd, S. 1997, ‘The UCB/LBNL/VINT network simulator’, Lawrence Berkeley Laboratory, Berkeley, California.
Available at: http://www.isi.edu/nsnam/ns/

McClean, P. 2000, ‘Intermediate Genetics’.
Available at: http://www.ndsu.edu/pubweb/~mcclean/plsc431/mendel/mendel4.htm

Menaria, S., Valiveti, S. & Kotecha, K. 2010, ‘Comparative study of Distributed Intrusion Detection in Ad-hoc Networks’, *International Journal of Computer Applications*, Vol. 8, No.9, pp. 11-16.

Mizanur Rahman, Sk., Mambo, M., Inomata, A. & Okamoto, E. 2006, ‘An Anonymous On-Demand Position-based Routing in Mobile Ad Hoc Networks’, *proceedings of the International Symposium on Applications and the Internet*, 23-27 January, Mesa/Phoenix, Arizona, United States, pp. 300-306, IEEE.

Mukherjee, A., Bandyopadhyay, S. & Saha, D. 2003, *Location Management and Routing in Mobile Wireless Networks*, ISBN 1580533558, (pp. 1-16 & 117-124), Artech House, London.

Murthy, C. & Manoj, B. 2004, *Ad Hoc Wireless Networks: Architectures and Protocols*, ISBN 013147023X, (pp. 213-214 & 475-490), Prentice Hall Communications Engineering and Emerging Technologies Series, Upper Saddle River, New Jersey, United States.

Murthy, S. & Garcia-Luna-Aceves, J. 1996, ‘An Efficient Routing Protocol for Wireless Networks’, *ACM/Baltzer Mobile Networks and Applications*, special issue on Routing in Mobile Communications Networks, Vol. 1, No. 2, pp. 183-197, Kluwer.

Myers, M., Wikle, T., Helmer, J., Qian, J., Demers, B. & Ranganathan, D. 2006, ‘GPS Tools For Geographers’, Oklahoma State University, Oklahoma, United States. Available at: http://www2.ocgi.okstate.edu/gpstools/overview1.htm

Nagar, V., Singh, Y. & Dhubkarya, D. 2011, ‘Estimation of Reliability and Scalability in Ad-Hoc Multicast Routing Protocol for Sender Network’, *proceedings of the International Conference on Computational Intelligence and Communication Networks (CICN)*, 7-9 October, Gwalior, India, pp. 350-353.

Nanda, S. 2008, ‘Mesh-Mon: a Monitoring and Management System For Wireless Mesh Networks’, Dartmouth Computer Science Technical Report TR2008-619, Ph.D. thesis, Dartmouth College, Hanover, New Hampshire, United States. Available at: http://folk.uio.no/paalee/referencing_publications/ref-xa-nanda-thesis-2008.pdf

Nasipuri, A. & Das, S. 1999, ‘On-Demand Multipath Routing for Mobile Ad Hoc Networks’, *proceedings of the 8th International Conference on Computer Communications and Networks (ICCCN 1999)*, 11-13 October, Boston, Massachusetts, United States, pp. 64-70, IEEE.

Naumov, V. & Gross, T. 2005, ‘Scalability of routing methods in ad hoc networks’, *Elsevier Performance Evaluation*, Vol. 62, pp.193-209.

Othman, M. 1999, ‘Mobile Computing and Communications: An Introduction’, *Malaysian Journal of Computer Science*, Vol. 12, No. 2, pp. 71-78.

Owen, G. & Adda, M. 2009, ‘SOLS: Self Organising Distributed Location Server For Wireless Ad Hoc Networks’, *International Journal of Computer Networks & Communications*, Vol. 1, No. 1, pp. 18-31.

Paik, J., Kim, B. & Lee, D. 2008, 'A3RP: Anonymous and Authenticated Ad Hoc Routing Protocol', *proceedings of the 2nd International Conference on Information Security and Assurance*, 24-26 April, Busan, Korea, pp. 67-72.

Papadimitratos, P. & Haas, Z. 2002, 'Secure Routing for Mobile Ad hoc Networks', *proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, 27-31 January, San Antonio, Texas, United States, pp. 54-62.

Papavassiliou, S., Xu, S. Orlik, P., Snyder, M. & Sass, P. 2002, 'Scalability in Global Mobile Information Systems (GloMo): Issues, Evaluation Methodology and Experience', *Springer Wireless Networks*, Vol. 8, No. 6, pp. 637-648.

Park, V. & Corson, S. 2001, 'Temporally-Ordered Routing Algorithm (TORA), Functional Specification', Internet Draft, Internet Engineering Task Force, MANET Working Group.
Available at: http://tools.ietf.org/id/draft-ietf-manet-tora-spec-04.txt

Park, Y., Lee, W. & Rhee, K. 2007, 'Authenticated On-Demand Ad Hoc Routing Protocol without Pre-shared Key Distribution', *proceedings of the 2007 ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security (BLISS 2007)*, 9-10 August, Edinburgh, United Kingdom, pp. 41-46.

Pei, G., Gerla, M. & Chen, T. 2000, 'Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks', *proceedings of the IEEE International Conference on Communications (ICC 2000)*, 18-22 June, New Orleans, Los Angeles, United States, pp. 70-74.

Perkins, C. & Bhagwat, P. 1994, 'Highly Dynamic Destination-Sequenced Distance Vector (DSDV) for Mobile Computers', *proceedings of the SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications*, 31 August - 2 September, London, United Kingdom, pp. 234-244.

Perkins, C. & Royer, E. 1999, 'Ad hoc on-demand distance vector routing', *proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1999)*, 25-26 February, New Orleans, Los Angeles, pp. 90-100, IEEE.

Perrig, A., Canetti, R., Song, D., Tygar, D. & Briscoe, B. 2004, 'TESLA: multicast source authentication transform introduction', Internet Draft, Internet Engineering Task Force, Multicast Security Working Group.
Available at: http://tools.ietf.org/html/draft-ietf-msec-tesla-intro-04.txt

Pirzada, A. & McDonald, C. 2008, 'Reliable Routing in Ad Hoc Networks Using Direct Trust Mechanisms'. In Cheng, M., Li, D., *Advances in Wireless Ad Hoc and Sensor Networks*, ISBN 0387685650, (pp. 133-159), Springer.

Prakash, V., Kumar, B. & Srivastava, A. 2011, 'Energy Efficiency Comparison of Some Topology-Based and Location-Based Mobile Ad Hoc Routing Protocols', *proceedings of the International Conference on Communication, Computing & Security (ICCCS 2011)*, 12-14 February, Odisha, India, pp. 36-39.

Priyantha, N. 2005, 'The cricket indoor location system', Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States.
Available at: http://nms.lcs.mit.edu/papers/bodhi-thesis.pdf

Razak, S., Furnell, S. & Brooke, P. 2004, 'Attacks against Mobile Ad Hoc Networks Routing Protocols', *proceedings of the 5th Annual Postgraduate Symposium on The Convergence of Telecommunications, Networking & Broadcasting (PGNET 2004)*, 28-29 June, Liverpool, United Kingdom, pp. 147-152.

Razak, S., Furnell, S., Clarke, N. & Brooke, P. 2008, 'Friend-assisted intrusion detection and response mechanisms for mobile ad hoc networks', *Ad Hoc Networks*, Vol. 6, No. 7, pp. 1151-1167.

Razak, S., Samian, N., Ma'arof, M., Furnell, S., Clarke, N. & Brooke, P. 2009, 'A Friend Mechanism for Mobile Ad Hoc Networks', *Journal of Information Assurance and Security*, Vol. 4, pp. 440-448.

Rifa-Pous, H. & Herrera-Joancomarti, J. 2007, 'Secure Dynamic MANET On-demand (SEDYMO) Routing Protocol', *proceedings of the 5th Annual Conference on Communication Networks and Services Research (CNSR 2007*), 14-17 May, Fredericton, New Brunswick, Canada, pp. 372-380.

Riley, G. 2003. 'The georgia tech network simulator', *Proceedings of the 1st SIGCOMM workshop on Models, Methods and Tools for reproducible network research (MoMeTools 2003)*, 25 August, Karlsruhe, Germany, pp. 5-12, ACM.

Riley, G., Fujimoto, R. & Ammar, M. 1999, 'A generic framework for parallelization of network simulations', *proceedings of the 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 1999)*, 24-28 October, College Park, Maryland, United States, pp. 128-135, IEEE.

Rout, A., Sethi, S. & Mishra, D. 2011, 'Optimized Ant Based Routing Protocol for MANET', *proceedings of the International Conference on Communication, Computing & Security (ICCCS 2011)*, 12-14 February, Odisha, India, pp. 21-25.

Santivanez, C., McDonald, B., Stavrakakis, I. & Ramanathan, R. 2002, 'On the Scalability of Ad Hoc Routing Protocols', *proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, 23-27 June, New York, United States, pp. 1688-1697.

Santivanez, C., Ramanathan, R. & Stavrakakis, I. 2001, 'Making link-state routing scale for ad hoc networks', *proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2001)*, 4-5 October, Long Beach, California, United States, pp. 22-32.

Sanzgiri, K., LaFlamme, D., Dahill, B., Levine, B., Shields, C. & Belding-Royer, E. 2005, 'Authenticated Routing for Ad Hoc Networks', *IEEE Journal On Selected Areas In Communications*, Vol. 23, No. 3, pp. 598-610.

Schilling, B. 2005, ‘Qualitative Comparison of Network Simulation Tools’, *Modeling and Simulation of Computer Systems seminar*, 31 January, Institute of Parallel and Distributed Systems, University of Stuttgart, Germany.
Available at: www.ipvs.uni-stuttgart.de/abteilungen/vs/lehre/lehrveranstaltungen/studienprojekte/CUBUS_termine/dateien/schilling.pdf

Schleich, J. 2010, ‘Robust Dominating Set based Virtual Backbones for Wireless Ad hoc Networks’, Ph.D. thesis, Faculty of Science, Technology and Communication, University of Luxembourg, Luxembourg.
Available at: http://www.julienschleich.org/files/Thesis-JulienSchleich.pdf

Seno, S., Budiarto, R. & Wan, T. 2011, ‘A Secure Mobile Ad hoc Network Based on Distributed Certificate Authority’, *Arabian Journal for Science and Engineering*, Vol. 36, No. 1, pp. 245-257.

Sharma, S. & Jena, S. 2011, ‘A Survey on Secure Hierarchical Routing Protocols in Wireless Sensor Networks’, *proceedings of the International Conference on Communication, Computing & Security (ICCCS 2011)*, 12-14 February, Odisha, India, pp. 146-151.

Sivrikaya, F. & Yener, B. 2004, ‘Time Synchronization in Sensor Networks: A Survey’, *IEEE Network*, Vol. 18, No. 4, pp. 45-50.

Song, J., Wong, V. & Leung, V. 2007, ‘Secure position-based routing protocol for mobile ad hoc networks’, *Ad Hoc Networks*, Vol. 5, No. 1, pp. 76-86, Elsevier.

Takagi, H. & Kleinrock, L. 1984, ‘Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals’, *IEEE Transactions on Communications*, Vol. 32, No. 3, pp. 246-257.

Trochim, W. 2001, *The Research Methods Knowledge Base*, ISBN 1931442487, 2nd edition, Atomic Dog Publishing.
Available at: http://www.socialresearchmethods.net/kb/

Vijayakumar, H. & Ravichandran, M. 2011, ‘Efficient Location Management of Mobile Node in Wireless Mobile Ad-hoc Network’, *proceedings of the National Conference on Innovations in Emerging Technology*, 17-18 February, Tamilnadu, India, pp.77-84.

Wang, W. & Ravishankar, C. 2009, ‘Hash-Based Virtual Hierarchies for Scalable Location Service in Mobile Ad-hoc Networks’, *Springer Mobile Networks and Applications*, Vol. 14, No. 5, pp. 625-637.

Wetteroth, D. 2001, *OSI Reference Model for Telecommunications*, ISBN 0071380418, McGraw-Hill Professional Publishing.

Wu, X. 2005, ‘VPDS: Virtual Home Region based Distributed Position Service in Mobile Ad Hoc Networks’, *proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICSCS 2005)*, 6-10 June, Columbus, Ohio, United States, pp. 113-122.

Xenakis, C., Panos, C. & Stavrakakis, I. 2010, ‘A comparative evaluation of intrusion detection architectures for mobile ad hoc networks’, *Computers & Security*, Vol. 30, No. 1, pp. 63-80.

Xu, T. & Cai, Y. 2010, ‘LSR: A Location Secure Routing Protocol for Ad Hoc Networks’, *proceedings of the 5th IEEE Mobile Adhoc and Sensor Systems (MASS 2010)*, 8-12 November, San Francisco, California, United States, pp.176-185.

Yang, S. & Bao, L. 2011, ‘Scalable Mobility Management in Large-Scale Wireless Mesh Networks’, *proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2011)*, 28-31 March, Cancun, Mexico, pp. 1230-1235.

Yau, P., Hu, S. & Mitchell, C. 2007, ‘Malicious Attacks on Ad Hoc Network Routing Protocols’, *International Journal of Computer Research*, Vol. 15, No. 1, pp. 73-100.

Yi, S., Naldurg, P. & Kravets, R. 2001, ‘Security-aware ad hoc routing for wireless networks’, *proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2001)*, 4-5 October, Long Beach, California, United States, pp. 299-302.

Zahariadis, T., Trakadas, P., Maniatis, S., Karkazis, P., Leligou, H. & Voliotis, S. 2009, ‘Efficient detection of routing attacks in Wireless Sensor Networks’, *proceedings of the 16th International Conference on Systems, Signals and Image Processing (IWSSIP 2009)*, 18-20 June, Chalkida, Greece, pp. 1-4.

Zapata, G. 2002, ‘Secure Ad hoc On-Demand Distance Vector Routing’, *ACM Mobile Computing and Communications Review (MC2R)*, Vol. 6, No. 3, pp. 106-107.

Zeng, X., Bagrodia, R. & Gerla, M. 1998, ‘Glomosim: A library for parallel simulation of large-scale wireless networks’, *proceedings of the 12th Workshop on Parallel and Distributed Simulation*, 26-29 May, Banff, Alberta, Canada, pp. 154-161.

Zhou, L. & Haas, Z. 1999, ‘Securing Ad Hoc Networks’, *IEEE Network Magazine*, Vol. 13, No. 6, pp. 24-30.

Zhu, Y., Zhang, J. & Partel, K. 2011, ‘Location-aided routing with uncertainty in mobile ad hoc networks: A stochastic semidefinite programming approach’, *Mathematical and Computer Modelling*, Vol. 53, No. 1, pp. 2192-2203.

Zouridaki, Ch., Mark, B. & Hejmo, M. 2007, ‘Byzantine robust trust establishment for mobile ad hoc networks’, *Telecommunication Systems*, Vol. 35, No. 6, pp. 189-206.

List of Publications Related to This Research

Journals:

- Qabajeh, L., Mat Kiah, M. L. & Qabajeh, M. 2009, ‘A Qualitative Comparison of Position-Based Routing Protocols for Ad-Hoc Networks’, *International Journal of Computer Science and Network*, Vol. 9, No. 2, pp. 131-140. (Refereed publication)

- Qabajeh, L., Mat Kiah, M. L. & Qabajeh, M. 2009, ‘A Scalable and Secure Position-Based Routing Protocol for Ad-Hoc Networks’, *Malaysian Journal of Computer Science*, Vol. 22, No. 2, pp. 99-120. (ISI-indexed publication)

- Mat Kiah, M. L., Qabajeh, L. & Qabajeh, M. 2010, ‘Unicast Position-Based Routing Protocols for Ad-Hoc Networks’, *Acta Polytechnica Hungarica (Journal of Applied Sciences)*, Vol. 7, No. 5, pp.19-46. (ISI-indexed publication)

- Qabajeh, L., Mat Kiah, M. L. & Qabajeh, M. (To be published 2012), ‘A More Secure and Scalable Routing Protocol for Mobile Ad-Hoc Networks’, *Security and Communication Networks*. (ISI-indexed publication)

- Qabajeh, L., Mat Kiah, M. L. & Qabajeh, M. 2011, ‘Secure Unicast Position-Based Routing Protocols for Ad-Hoc Networks’, *Acta Polytechnica Hungarica (Journal of Applied Sciences)*, Vol. 8, No. 6, pp.191-214. (ISI-indexed publication)

Conferences:

- Qabajeh, L., Mat Kiah, M. L. & Qabajeh, M. 2009, ‘A Scalable, Distributed and Secure Routing Protocol for MANETs’, *proceedings of the International Conference on Computer Engineering and Applications (ICCEA 2009)*, 6-8 June, Manila, Philippine, pp. 51-56.

- Qabajeh, L., Mat Kiah, M. L. & Qabajeh, M. 2009, ‘A Scalable Secure Routing Protocol for MANETs’, *proceedings of the International Conference on Computer Technology and Development (ICCTD 2009)*, 13-15 November, Kota Kinabalu, Malaysia, pp. 143-147.

- Qabajeh, L., Mat Kiah, M. L. & Qabajeh, M. 2010, ‘A Scalable and Secure Position-Based Routing Protocol for MANETs’, *proceedings of the Annual International Conference on Network Technologies & Communication (NTC 2010)*, 29-30 November, Phuket Beach Resort, Thailand, pp. N7-N12.

Book Chapters:

- Qabajeh, L., Mat Kiah, M. L. & Qabajeh, M. 2011, ‘A Survey of Position-Based Routing Protocols for Ad-Hoc Networks’. In Gavrilovska, L., Krco, S., Milutinovic, V., Stojmenovic, I., Trobec, R., *Application and Multidisciplinary Aspects of Wireless Sensor Networks: Concepts, Integration, and Case Studies*, ISBN 976-1-84996-509-5, part 1, (pp. 47-83), Springer.