# DESIGN AND DEVELOPMENT OF A ROUTING PROTOCOL FOR MOBILE AD HOC NETWORKS (MANETs)

**Ph.D. THESIS**

**By**

**ANIL KUMAR VERMA**

**(Regn No. 9030351)**

THAPAR UNIVERSITY

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**THAPAR UNIVERSTIY**
**PATIALA – 147004 (INDIA)**
**APRIL 2007**

# DESIGN AND DEVELOPMENT OF A ROUTING PROTOCOL FOR MOBILE AD HOC NETWORKS (MANETs)

**A THESIS**

***Submitted in partial fulfillment of the***

***Requirements for the award of the degree***

***of***

**DOCTOR OF PHILOSOPHY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**


**By**

**ANIL KUMAR VERMA**

**(Regn No. 9030351)**


THAPAR UNIVERSITY


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**THAPAR UNIVERSTIY**
**PATIALA – 147004 (INDIA)**


**APRIL 2007**

Security is like $O_2$, you have it for guaranteed but when you don't, getting it becomes the immediate and pressing priority.

Joseph Nye
Harvard University

Three people can keep a secret only if two of them are dead.

Benjamin Franklin

Privacy is not something that I'm merely entitled to, its' an absolute prerequisite.

Marlon Brando

THAPAR
UNIVERSITY

## CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled " **DESIGN AND DEVELOPMENT OF A ROUTING PROTOCOL FOR MOBILE ADHOC NETWORKS (MANETs)**" in partial fulfillment of the requirements for the award of the Degree of Doctor of Philosophy and submitted in the department of computer science and engineering of Thapar University, Patiala is an authentic record of my own work carried out during a period from July 2003 to March 2007 under the supervision of **Dr. R.C. Joshi**, Professor, Indian Institute of Technology, Roorkee and **Dr. Mayank Dave**, Assistant Professor, National Institute of Technology, Kurukshetra.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute/University.

Anil K. Verma

**(ANIL KUMAR VERMA)**
**Regn No. 9030351**

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

Date: 17.4.07

**(Dr. Mayank Dave)**
Assistant Professor
NIT, Kurukshetra

**(Dr. R.C.Joshi)**
Professor
IIT, Roorkee

...

# ABSTRACT

The desire to be connected anytime, anywhere, anyhow has led to the development of wireless networks, especially in the area of pervasive and ubiquitous computing. The Mobile Ad hoc NETworks (popularly known as MANETs) are no exception. A wireless ad hoc network is a collection of two or more devices/nodes or terminals with wireless communications and networking capability that communicate with each other without the aid of any centralized administrator. They do not have a fixed topology or infrastructure hence they are also known as *infrastructureless networks.* Each node in a wireless ad hoc network functions as a host or a router or both. The network topology is in general dynamic and the communication is via an open medium, which is vulnerable. Thus, sophisticated secure routing protocols are needed to safeguard the data against the unauthorized access. The work presented in this thesis is an effort to address the security issues by proposing a new secure routing protocol, termed as **SANE-DNA**.

We consider a routing protocol, namely, Dynamic Source Routing (DSR). The DSR is an on-demand or reactive routing protocol based on the concept of source routing. A pseudo DNA (Deoxyribo Nucleic Acid) based cryptographic algorithm is used in order to secure the MANETs. The pseudo DNA cryptography is a concept inspired from the field of life science and has been extended to the field of MANETs to secure them. Nowadays, DNA is also being considered as a medium for ultrascale computation and for ultra-compact information storage. In the presented work, one potential key application is used known as DNA-based molecular cryptography systems because it provides a much more compact storage medium, and an extremely small amount of DNA suffices even for huge one-time-pads. We have used one-time-pads for DNA-based cryptography and they are found to be unbreakable. The proposed routing protocol has been verified and validated through various simulation scenarios, using synthetically generated data sets.

Simulation results demonstrate that the algorithm is secure with marginal overhead. The results have been validated based upon RFC4728 for the DSR protocol. The various parameters considered are: Route Acquisition Time, Average End-to-End Delay (or Mean Overall Packet Latency), Routing Overhead and Throughput (or packet delivery ratio).

***Keywords*:** MANET, Routing protocols, Security, DNA cryptography

*To my parents,*
*Nirmala and Jagan Nath Verma*

# ACKNOWLEDGEMENTS

A few sublime human experiences defy expressions of any kind, and a feeling of true gratitude is one of them. I, therefore, find words quite inadequate to express my indebtness to my supervisors **Dr. R. C. Joshi** and **Dr. Mayank Dave** for their virtuous guidance, encouragement and help throughout this work. Their deep insight into the problem and the ability to provide solutions has been immense value in improving the quality of research at all stages. This experience of working with them shall ever remain a source of inspiration and encouragement for me. I learned a great deal from them, not only about research but also matters touching many other aspects that will benefit me in my future endeavors.

My sincere thanks are due to Dr. Abhijit Mukherjee, Director Thapar University, Patiala and Dr. S. C. Saxena, Director, Indian Institute of Technology, Roorkee for providing me the necessary administrative assistance in the completion of the work.

I am extremely grateful to the celebrated authors whose precious works have been consulted and referred in my research work. I also wish to convey my appreciation to fellow research scholars and colleagues who provided encouragement and timely support in the hour of need.

Special thanks are due to my mother Smt. Nirmala Verma whose love and affectionate blessings have been a constant source of inspiration in making this manuscript a reality. I sincerely pay tributes to my father Late Sh. Jagan Nath Verma, who has been a guiding force for all that I have achieved till date. He did not live long enough to see this day in my life.

At the end I express my deep sense of appreciation to my wife Sunita and children Sankalp and Anmol for their cooperation and patience during the completion of my research work.

All the thanks are, however, only fraction of what is due to Almighty for granting me an opportunity and the divine grace to successfully accomplish this assignment.

A.K. Verma

# TABLE OF CONTENTS

**Candidate's Declaration …….…….……. i**

**Abstract ….……..…… ii**

**Dedication …….……. vi**

**Acknowledgements …….……. vii**

**Table of Contents..…….…… ix**

**List of Abbreviations ……...…….. xiii**

**List of Tables..…….. xv**

**List of Figures..……... xvi**

**Chapter 1 Introduction 1**

1.1 Motivation 1

1.2 Statement of the Problem 7

1.3 Organization of the Thesis 8

**Chapter 2 Literature Review and General Consideration 11**

2.1 Introduction 11

2.2 MANET Characteristics 12

2.3 Routing in MANETs 13

2.3.1 Flat Routing 14

2.3.2 Other routing protocols 23

2.4 Network Simulation 25

2.4.1 ns2-Overview 25

2.5 Mobility Models 29

2.6 Simulation results for routing protocols 35

2.7 Simulation results for different Mobility Models 36

2.8 Lessons Learned 38

2.9 Dynamic Source Routing Protocols 39

2.9.1 Route Discovery Phase 40

2.9.2 Route Maintenance Phase 41

2.10 Simulation and Validation of DSR 42

2.10.1 Performance results 44

2.10.2 Characteristics of DSR 46
2.10.3 Advantages and disadvantages of DSR 47
2.11 Some desirable properties of good routing protocols for MANETs 47
2.12 Conclusion 48

**Chapter 3 Security in MANETs 50**
3.1 Introduction 50
3.2 Security goals 51
3.3 Vulnerability of existing protocols 52
3.3.1 Active attacks 52
3.3.2 Passive attacks 53
3.3.3 MANET attack tree 53
3.4 Security mechanisms 55
3.4.1 Link level security 55
3.4.2 Routing/Network level security 57
3.4.3 Key Management 58
3.5 Secure routing protocols 59
3.5.1 ARAN 59
3.5.2 ARIADNE 61
3.5.3 SEAD 64
3.5.4 SRP 64
3.5.5 SAODV 65
3.5.6 SAR 67
3.5.7 SLSP 69
3.5.8 TESLA 69
3.6 Cooperation enforcement in MANETs 71
3.6.1 CONFIDANT 71
3.6.2 CORE 73
3.6.3 SORI 75
3.6.4 OCEAN 76
3.6.5 Nuglets 78
3.7 Conclusion 81

**Chapter 4 DNA Cryptography** **82**

4.1 Introduction 82

4.1.1 Encryption and Decryption 82

4.2 Types of Cryptography 83

4.2.1 Secret Key 84

4.2.2 Public Key Cryptography 85

4.2.3 Key size based tradeoff 86

4.2.4 Threshold Cryptography 86

4.3 DNA Cryptography 87

4.3.1 Pseudo Cryptography method 88

4.3.2 Pseudo DNA Cryptography 89

4.3.3 DNA Cryptosystems using random one-time pads 92

4.3.4 DNA Cryptosystem using one-time pad substitution system 92

4.3.5 Method analysis 94

4.4 Algorithm analysis and evaluation 95

4.4.1 Analysis of pseudo DNA Cryptography 95

4.4.2 Evaluation 95

4.5 Conclusion 98

**Chapter 5 Simulation Results and Discussion** **100**

5.1 Introduction 100

5.2 Strategy used 100

5.3 Simulation parameters 101

5.4 Performance metrics 101

5.5 Published DSR 103

5.6 Published DSR and Implemented DSR comparison 111

5.7 Implemented DSR and SANE-DNA comparison 120

5.8 Discussions 127

5.9 Conclusion 130

**Chapter 6 Conclusion and Scope for Future Work** **131**

6.1 Conclusion 131

6.2 Scope for future Research 134

***Appendix-A*** (commonly used terms in DNA cryptography) **136**

***Appendix-B*** ( source code listing ) **140**

**References** **144**

**Publications** **156**

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ABAM | Associativity-BAsed Multicast routing protocol |
| ABR | Associativity-Based Routing |
| ALARM | Adaptive Location Aided Routing – Mines |
| AMRoute | Adhoc Multicast Routing Protocol |
| AODV | Ad hoc On-demand Distance Vector Routing |
| ARAN | Authenticated Routing for Ad hoc Networks |
| CA | Certification Authority |
| CBRP | Cluster Based Routing Protocol |
| CEDAR | Core Extraction Distributed Ad hoc Routing |
| CGSR | Cluster-head Gateway Switch Routing |
| CONFIDANT | Cooperation of Nodes: Fairness In Dynamic Ad hoc NeTworks |
| CORE | COllaborative Reputation mEchanism. |
| DNA | Deoxyribonucleic acid |
| DoS | Denial of Service |
| DREAM | Distance Routing Effect Algorithm for Mobility |
| DSDV | Destination Sequenced Distance Vector Routing |
| DSR | Dynamic Source Routing |
| FSR | Fisheye State Routing |
| FSR | Fisheye State Routing |
| GPSAL | GPS Ant-Like routing algorithm |
| GSR | Global State Routing |
| HSR | Hierarchical State Routing |
| IMEP | Internet MANET Encapsulation Protocol |
| IP | Internet Protocol |
| KDC | Key Distribution Centre |
| LBM | Location Based Multicast |
| LKH | Logical Key Hierarchy |
| LMR | Lightweight Mobile Routing |
| MAC | Message Authentication Code. |

| | |
|---|---|
| MANET | Mobile Ad hoc Network |
| MANETs | Mobile Ad hoc Networks |
| MAODV | Multicast Ad-hoc On-Demand Distance Vector routing |
| MM | Mobility Model |
| MOCA | Mobile Certification Authority |
| NAM | Network AniMator |
| ns | Network Simulator |
| OCEAN | Observation-based Cooperation Enforcement in Ad hoc Networks |
| OFT | One-way Function Trees |
| PAMAS | Power Aware Multi Access Protocol with Signaling Ad Hoc Networks |
| PARO | Power-Aware Routing Optimization protocol |
| RNA | Ribonucleic acid |
| RWPMM | Random Waypoint Mobility Model |
| SAODV | Secure Ad hoc On demand Distance Vector |
| SAR | Security-Aware ad hoc Routing |
| SEAD | Secure Efficient Ad hoc Distance vector routing protocol |
| SLSP | Secure Link State Routing Protocol |
| SORI | Secure and Objective Reputation-based Incentive |
| SRP | Secure Routing Protocol |
| SSR | Signal Stability Routing |
| TESLA | Timed Efficient Stream Loss-tolerant Authentication. |
| TORA | Temporally Ordered Routing Algorithm |
| URSA | Ubiquitous and RobuSt Access control |
| WRP | Wireless Routing Protocol |
| ZHLS | Zone-based Hierarchical Link State Routing |
| ZRP | Zone Routing Protocol |

# LIST OF TABLES


| | | |
|---|---|---|
| 1.1 | Wireless network classification | 3 |
| 1.2 | Historical development of MANET | 5 |
| 1.3 | Mobile Ad hoc network applications | 6 |
| 2.1 | Basic characteristics of DSDV and WRP | 16 |
| 2.2 | Complexity comparison of DSDV and WRP | 16 |
| 2.3 | Basic characteristics of DSR and AODV | 19 |
| 2.4 | Complexity comparison of DSR and AODV | 20 |
| 2.5 | Basic characteristics of ZRP and ZHLS | 22 |
| 2.6 | Complexity comparison of ZRP and ZHLS | 23 |
| 2.7 | Simulation parameters for simulating selected flat routing protocols | 35 |
| 2.8 | Simulation parameters for comparison of mobility models on different flat routing protocols. | 37 |
| 2.9 | Summary of simulation parameters | 44 |
| 2.10 | Characteristics of DSR | 46 |
| 3.1 | Operational requirements of the surveyed secure Ad hoc routing solutions | 79 |
| 3.2 | Ad hoc routing parameters | 81 |
| 4.1 | Performance of application with plaintext of different contents | 96 |
| 4.2 | Performance of the application with plaintexts of different lengths | 97 |
| 5.1 | Different simulation parameters used for evaluating SANE-DNA | 101 |
| 5.2 | Coefficient of correlation between published DSR and implemented DSR | 119 |
| 5.3 | Simulation results (Route Acquisition Time) | 127 |
| 5.4 | Simulation results (Packet Delay) | 127 |
| 5.5 | Simulation results (Routing Overhead) | 127 |
| 5.6 | Simulation results (Throughput) | 127 |
| 6.1 | Evaluation of SANE-DNA | 133 |

# LIST OF FIGURES


| | | |
|---|---|---|
| 1.1 | An infrastructured network | 4 |
| 1.2 | An infrastructureless network | 4 |
| 2.1 | Routing in MANET | 14 |
| 2.2 | Classification of routing protocols in MANETs | 23 |
| 2.3 | Simplified user's view of NS | 25 |
| 2.4 | Architectural view of NS | 26 |
| 2.5 | Internal mechanism of NS2 for routing in MANETs | 27 |
| 2.6 | Data flows into nam from network data and other sources after pre-processing into the nam trace format. | 28 |
| 2.7 | A sample screen-shot of Network Animator (NAM) being executed on Linux environment for 5 mobile nodes. | 28 |
| 2.8 | A sample screen-shot of the Xgraph utility for the simulation scenario as mentioned in Figure 2.5. | 29 |
| 2.9 | Various categories of Mobility Model in MANETs | 31 |
| 2.10 | Traveling pattern of different mobile nodes in a given simulation area (rectangular shaped) based upon RWPMM. | 32 |
| 2.11 | RPGM Model | 33 |
| 2.12 | Freeway Map | 33 |
| 2.13 | Manhattan map | 34 |
| 2.14 | Throughput of MANET protocols | 35 |
| 2.15 | Packet delivery ratio of MANET protocols | 36 |
| 2.16 | Number of dropped packets | 36 |
| 2.17 | Throughput of DSDV protocol in different mobility models with 50 nodes. | 37 |
| 2.18 | Throughput of DSR protocol in different mobility models with 50 nodes. | 38 |
| 2.19 | Throughput in AODV with 50 nodes for different mobility models. | 38 |
| 2.20 | Propagation of the Route Request (also known as route establishment) showing the building of route entry from the Source ‘S’ to the Destination ‘D’ in a DSR. | 41 |

| | | |
|---|---|---|
| 2.21 | Propagation of the Route Reply containing the route entry from the Destination ‘D’ to Source ‘S’ in a DSR. | 41 |
| 2.22 | Route maintenance in DSR | 42 |
| 2.23 | Packet delivery ratio (or Throughput) | 44 |
| 2.24 | Average end-to-end delay | 45 |
| 2.25 | Routing Overhead | 45 |
| 2.26 | Path Optimality | 46 |
| 3.1 | MANETs Attack Tree | 54 |
| 3.2 | Security requirements for ad hoc network layer and corresponding solutions. | 55 |
| 3.3 | Link Level Security Clouds | 56 |
| 3.4 | Routing/Network Level Security Clouds | 57 |
| 3.5 | Routing Paths in MANET | 59 |
| 3.6 | Route Discovery in ARAN, along with terminologies used | 61 |
| 3.7 | Route Discovery in ARIADNE, along with terminologies used | 63 |
| 3.8 | Route Discovery in SAODV, along with terminologies used | 67 |
| 4.1 | Encryption and Decryption | 83 |
| 4.2 | Secret Key Cryptography | 84 |
| 4.3 | Public Key Cryptography | 85 |
| 4.4 | The Central Dogma of molecular biology | 88 |
| 4.5 | The implementation diagram of the pseudo DNA cryptography method | 91 |
| 4.6 | One-time-pad codebook DNA sequence | 93 |
| 4.7 | Analysis of length of plaintext and cipher text | 98 |
| 4.8 | Analysis of time for encryption and decryption | 98 |
| 5.1 | Basic sequential discrete event simulator algorithm | 100 |
| 5.2 | A model of simulated MA NET system | 102 |
| 5.3 | Graphs showing the output of Route Acquisition Time for Published DSR against different simulation Parameters (a,b,c,d) | 103 |
| 5.4 | Graphs showing the output of Packet Delay for Published DSR against different simulation Parameters (a,b,c,d) | 105 |
| 5.5 | Graphs showing the output of Routing Overhead for Published DSR against different simulation Parameters (a,b,c,d) | 107 |

| | | |
|---|---|---|
| 5.6 | Graphs showing the output of Throughput for Published DSR against different simulation Parameters (a,b,c,d) | 109 |
| 5.7 | Graphs showing the output of Route Acquisition Time for Published DSR vs Implemented DSR against different simulation Parameters (a,b,c,d) | 111 |
| 5.8 | Graphs showing the output of Packet delay for Published DSR vs Implemented DSR against different simulation Parameters (a,b,c,d) | 113 |
| 5.9 | Graphs showing the output of Routing Overhead for Published DSR vs Implemented DSR against different simulation Parameters (a,b,c,d) | 115 |
| 5.10 | Graphs showing the output of Throughput for Published DSR vs Implemented DSR against different simulation Parameters (a,b,c,d) | 117 |
| 5.11 | Graphs showing the output of Route Acquisition Time for Implemented DSR vs SANE-DNA against different simulation Parameters (a,b,c,d) | 120 |
| 5.12 | Graphs showing the output of Packet Delay for Implemented DSR vs SANE-DNA against different simulation Parameters (a,b,c,d) | 122 |
| 5.13 | Graphs showing the output of Routing Overhead for Implemented DSR vs SANE-DNA against different simulation Parameters (a,b,c,d) | 124 |
| 5.14 | Graphs showing the output of Throughput for Implemented DSR vs SANE-DNA against different simulation Parameters (a,b,c,d) | 126 |

# Chapter 1

# Introduction

## 1.1 Motivation

With recent advances in mobile technology and mobile devices, mobile computing has become an important part of our life. People are using wireless networks for their day-to-day work, be it making a phone call or to download news or to see and listen or only listen to their favorite song from various multimedia servers with the help of various devices such as mobile phones, PDAs or a laptop. More services are in the offering in near future [46]. The desire to be connected *anytime, anywhere, anyhow* has led to the development of wireless networks, opening new vista of research in pervasive and ubiquitous computing [100]. This emerging field of mobile and nomadic computing [61] requires a highly secure routing protocol to effectively manage the communication among the peers.

Wireless networks [36,124], in general, refer to the use of infrared or radio frequency signals to share information and resources between devices. Due to basic difference in the physical layer (ISO/OSI model [14]), the wireless devices and networks show distinct characteristics from their wireline counterparts, such as [121,124]:

1. Higher interference results in lower reliability.
2. Low bandwidth and much slower data transfer rate.
3. Highly variable network conditions.
4. Limited computing and energy resources.
5. Device size limitation, and
6. Weaker security.

Apart from these limitations the wireless networks are immensely popular because of the benefits of using wireless technologies, such as [61,62]:

• **Access to more than one technology -** Users can use more than one access technology to service various parts of their network and during the migration phase of their networks, when upgrading occurs on a scheduled basis. It enables a fully comprehensive access technology portfolio to work with existing technologies.

• **Minimal cost -** The inherent nature of wireless is that it doesn't require wires or lines to accommodate the data/voice/video pipeline. Although paying fees for access to elevated areas such as masts, towers, and building tops is not unusual but the associated logistics, and contractual agreements are often minimal as compared to the costs of trenching a cable.

• **Reduced time to revenue -** Companies can generate revenue in less time through the deployment of wireless solutions than with comparable access technologies because a wireless system can be assembled and brought online in a very short span of time.

• **Provides broadband access extension** - Wireless commonly competes and complements existing broadband access. Wireless technologies play a key role in extending the reach of cable, fiber, and Digital Subscriber Link (DSL) markets, and it does so quickly and reliably.

Wireless networks can be categorized in various ways depending upon the chosen criteria for classification, as shown in Table-1.1.

**Table 1.1.** Wireless Network Classification

| S. No. | Criteria | Types |
|---|---|---|
| 1. | Communication coverage area | WAN<br>WMAN<br>WLAN<br>WPAN<br>BAN |
| 2. | Access technology | GSM Networks<br>TDMA Networks<br>CDMA Networks<br>Satellite Networks<br>Wi-Fi (802.11) Networks<br>Hiperlan2 Networks<br>Bluetooth Networks [120]<br>Infrared Networks |
| 3. | Network Applications | Enterprise Networks<br>Home Networks<br>Tactical Networks<br>Sensor Networks<br>Pervasive Networks<br>Wearable Computing<br>Automated Vehicle Networks. |
| 4. | Network Formation and Architecture | Infrastructure Based Networks<br>Infrastructureless Networks |

As mentioned in Table-1.1, based upon the criterion of network formation and architecture, the wireless networks can be subdivided into two classes, Infrastructured and infrastructureless networks [61,62]. These are defined as follows:

a) Infrastructured networks, have fixed and wired gateways. They have fixed base stations connected to other base stations through wires. The transmission range of a base constitutes a cell. A "hand-off" occurs as mobile host travels out of range of one station and into the range of another and thus, the mobile host is able to continue communication seamlessly throughout the network, represented in Figure1.1. The *Cellular networks* fall under this category.

b) Infrastructureless networks, do not have fixed routers and all nodes are capable of movement and can be connected dynamically in an arbitrary

manner. The entire network is mobile, and the individual terminals are allowed to move at will relative to each other, represented in Figure1.2. Mobile Ad hoc Networks *(MANETs)* falls under this category.

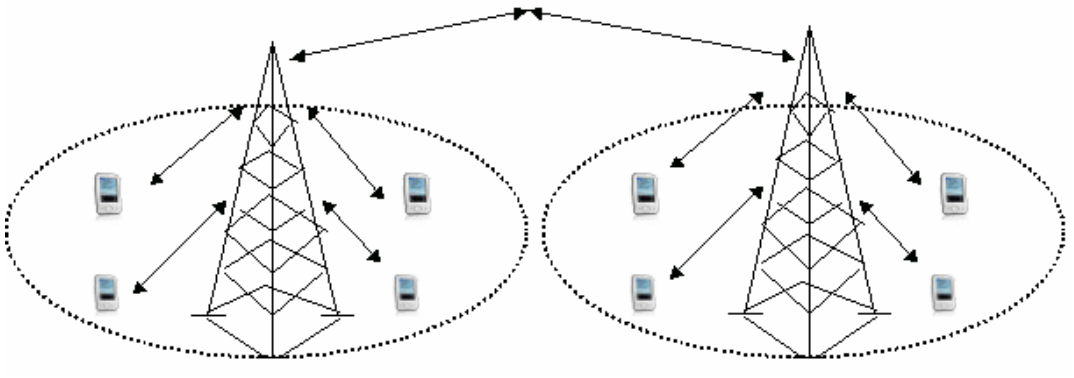


**Figure 1.1.** An infrastructured network

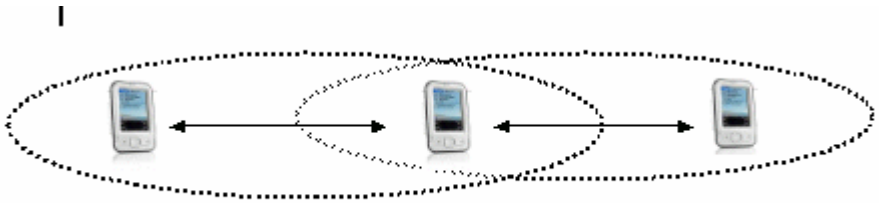


**Figure 1.2.** An infrastructureless network

The basic principle behind ad hoc networking is the multi-hop relaying [36], which traces its root back to 500 B.C. Darius I (522-486 B.C.), the king of Persia, devised an innovative communication system that was used to send messages and news from his capital to the remote provinces of his empire by means of a line of shouting men positioned on the tall structures. The system was faster than normal method of sending the message/news through a messenger. The use of ad hoc voice communication was used in many tribal societies with a string of repeaters of drums, trumpets or horns. In recent times, it was the Department of Defense (DoD), in 1972, initiated a new program on Packet Radio Networks (PRNET) [1] with the intention to create technologies for the battlefield that did not employ the previously deployed infrastructure but were highly survivable. Table 1.2 summarizes the major milestones in the development of MANET [27,49,66,79].

**Table 1.2.** Historical development of MANET

| Date | Generations | Developments |
|---|---|---|
| 1972 | First generation | • PRNET (Packet Radio Networks)<br>• ALOHA (Arial Locations of Hazardous Atmospheres)<br>• CSMA (Carrier Sense Medium Access) |
| 1980 | Second Generation | • SURAN (Survivable Adaptive Radio Networks) |
| Early 1990 | Third generation | • GloMo (Global Mobile Information Systems)<br>• NTDR (Near-term Digital Radio)<br>• Mobile Ad-Hoc Networking Working group was established, 1991. |
| Mid and Late 1990 | | • JTRS (Joint Tactical Radio System), 1996.<br>• IETF published several drafts about Routing protocol of MANET, 2000.<br>• IEEE Workshop on Mobile Ad Hoc Networking and Computing was established, 2000. |
| Future | Fourth generation | • Use of mobile adhoc routers to provide Internet connectivity to mobile users.<br>• Distributive collaborative computing.<br>• Distributed sensing networks.<br>• Disaster recovery networks. |

The benefits of ad hoc networks appeal to applications like conferences, meetings, disaster relief, rescue missions, and battlefield operation. Such scenarios typically lack a central administration or wired infrastructure. Some of the application areas [34,35,39,116] are listed in Table 1.3

**Table 1.3.** Mobile Ad hoc Network Applications

| Applications | Descriptions/Services |
|---|---|
| Tactical networks | Military communication and operations, Automated Battlefields. |
| Sensor networks | Collection of embedded sensor devices used to collect real-time data to automate everyday functions. Data highly correlated in time and space, e.g., remote sensors for weather, earth activities; sensors for manufacturing equipment. Can have between 1000–100,000 nodes, each node collecting sample data, then forwarding data to centralized host for processing using low homogeneous rates. |
| Emergency services | Search-and-rescue operations as well as disaster recovery; e.g., early retrieval and transmission of patient data (record, status, diagnosis) from/to the hospital, replacement of a fixed infrastructure in case of earthquakes, hurricanes, fire, etc. |
| Commercial | E-Commerce, e.g., electronic payments from anywhere (i.e., in a taxi) Dynamic Business environment- access to customer files stored in a central location on the fly provide consistent databases for all agents mobile office, Transmission of news, road conditions, weather, music local ad hoc network with nearby vehicles for road/accident guidance. |
| Home and enterprise | Home/office wireless networking (WLAN), e.g., shared whiteboard-networking application, use PDA to print anywhere, trade shows Personal Area Network (PAN), Body Area Network (BAN). |
| Educational | Set up virtual classrooms or conference rooms applications Set up ad hoc communication during conferences, meetings, or lectures. |
| Entertainment | Multiuse games, Robotic pets, Outdoor Internet access. |
| Location-aware services | Follow-on services, e.g., automatic call forwarding, transmission of the actual workspace to the current location<br>Information services<br>Push, e.g., advertise location-specific service, like- gas stations;<br>Pull, e.g., location-dependent travel guide;<br>Services (printer, fax, phone) availability information; etc. |

Apart from above applications the MANETs are found to be useful for realizing the tetherless computing and opportunistic mobile computing [121].

A MANET can also be defined as a distributed infrastructureless network [112] and mainly relies on individual security solutions from each mobile node and therefore centralized security control is hard to implement [110]. Securing a MANET is a severe problem because of the conjunction of several factors:

- Vulnerabilities: the lack of physical security and the ease of eavesdropping and spoofing leaves much desired gap between the security in wireless communication and the security in standard wireline communication.
- Lack of a priori trust: A MANET consists of set of nodes, which are not part of any organization, therefore the classical security paradigm based on pre-established trust among the parties are not applicable.
- Lack of infrastructure: security solutions comprising of dedicated secure components with predefined roles (such as trusted third party and key servers) cannot be used in this environment.
- Requirement for cooperation: due to lack of dedicated components, such as - routers and servers, the basic network functions and services need to be carried out by a set of ordinary nodes in a distributed fashion. Thus, the routing is affected by the presence of malicious node or the absence of cooperation among the nodes.

## 1.2 Statement of the Problem

The highly dynamic nature of MANETs result in frequent and unpredictable changes in the network topology, which add to the difficulty and complexity to routing among the mobile nodes within the network. Thus, establishing communication among mobile nodes is a great challenge in itself. The applications associated with the field of MANETs, make them an important part of the next generation wireless networks.

In this research work, focus has been put on the strategy to address the security issue because MANETs are generally more vulnerable to information and security threats than fixed-wired networks. The use of open and shared broadcast channels means the nodes with inadequate physical protection are prone to security threats. Further, a lot of emphasis has been given on the routing mechanism and the security area has not been addressed adequately in existing research. Thus, the issue to design and develop an efficient and secure routing protocol is still wide open.

The main objective of the present work can be stated as – *"to design and develop a routing protocol for MANETs*". In order to handle the above problem, the following outline is proposed:

1. Evaluation and Analysis of existing ad hoc routing protocols – The assessment and study of different types of routing protocols will help in better understanding of the basic characteristics and functioning of the protocols. Analysis of some of the routing protocols can be carried through simulation, using synthetically generated data sets. Further, there are various mobility models proposed, it would also be interesting to note the behavior of these protocols when subjected to simulation under these models.

2. Design and development of the proposed routing protocol - Based upon the knowledge so gleaned will act as pavement for improving an existing routing protocols. The new protocol will be proposed after proper verification and validation through simulations. The proposed protocol can be validated against different RFCs proposed by IETF and the verification can be done by taking various performance metrics such as – route acquisition time, average end-to-end delay, routing overhead, and throughput.

## 1.3 Organization of the Thesis

Including this introductory chapter, this thesis contains six chapters.
Chapter 2 presents a literature review and general considerations of routing protocols followed by critical comparison and analysis of different type of routing

protocols for MANETs. The routing protocols can be mainly categorized as: Flat-routing, Hierarchical routing and Location aware routing. The chapter is mainly devoted towards the study of DSR routing protocol, used in conjunction with the proposed technique for secure routing in MANETs.

Chapter 3 discusses the state of the art in MANET security, focusing mainly on the contributions relating to secure routing protocols. The wireless networking and mobile computing hardware is now capable of fulfilling the promise of this technology therefore it is the need of the hour to design and develop routing protocols which can support the performance with protection.

Chapter 4 provides a novel secure routing protocol for secure routing in MANETs. We consider a routing protocol, namely, Dynamic Source Routing (DSR). The DSR is an on-demand or reactive routing protocol based on the concept of source routing. DNA based cryptographic algorithm is used in order to secure the MANET. The DNA cryptography is a concept inspired from the field of life science and has been extended to the field of MANETs to secure them. DNA is being considered as a medium for ultra scale computation and for ultra-compact information storage. In the presented work, one potential key application known as, DNA-based molecular cryptography is used. Its use is also proposed because it also provides a much more compact storage medium, and an extremely small amount of DNA suffices even for huge one-time-pads. We have used one-time-pads for DNA-based cryptography and they are found to be unbreakable. The routing algorithm [125] developed is named as: **SANE-DNA**. The word SANE is of Latin origin and means- *sensible*, (some of the synonyms are- wise, rational, commonsensical) and has been used as an acronym for Securing Ad hoc NEtwork. The word DNA highlights the underlying method used for securing the routing protocol.

Chapter 5 of the thesis highlights the different scenario in which the proposed secure routing protocol has been analyzed by making use of a simulated environment. The chapter mainly deals with the verification of the proposed SANE-DNA routing protocol. We also discuss the various simulation scenarios

used while carrying out the implementation work. Simulation results demonstrate that the algorithm is secure with marginal overhead.

Finally, Chapter 6 summarizes the contribution of this thesis. It also provides a number of useful directions for future research on the presented work.

Apart from above chapters, the thesis also has a list of commonly used terms in DNA cryptography (Appendix-I) and the source coding used in the experimental setup (Appendix-II). In the end, a list of author's research contributions is also provided.

# Chapter 2

# Literature Review and General Considerations

## 2.1 Introduction

Mobile computing [121] is proliferating as devices are becoming smaller, cheaper, and more powerful. By combining mobile devices with wireless communication abilities, the vision of being connected *anytime, anywhere, anyhow* will soon be a reality. New applications arise from mobile entities interacting and collaborating towards a common goal. With cellular phones being widely employed [28] and the mobile Internet emerging into the market place, concepts of dynamic wireless networks that do not depend on expensive infrastructure draws attention to the area of ad hoc networks.

Mobile devices constitute a mobile ad hoc network when they directly and wirelessly communicate with other devices nearby without any fixed infrastructure. The mobile nodes move and thus the network topology changes dynamically and frequently. The absence of any hierarchy, established infrastructure, or centralized administration forces the nodes to control the network on their own.

We quote the definition of a mobile ad hoc network from the charter of the corresponding Internet Engineering Task Force (IETF) [64]:

“A ‘mobile ad hoc network’ (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links-the union of which forms an arbitrary graph. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet".

The ultimate goal of MANETs is to provide secure routing of data resources to mobile users at anytime and from anywhere. In conjunction with the existing routing protocols, providing security for MANETs give rise to significant challenges and performance opportunities [93]. This chapter provides a background on routing and reviews some key approaches for routing in MANETs.

## 2.2 MANET Characteristics

MANETs are new paradigm of networks, offering unrestricted mobility without any underlying infrastructure. Basically, ad hoc network is a collection of nodes communicating with each other by forming a multi-hop network. Following are the characteristics of a MANET [112,121]:

**Dynamic Topologies**

Nodes are free to move arbitrarily. The network topology may change randomly and have no restriction on their distance from other nodes. As a result of this random movement, the whole topology is changing in an unpredictable manner, which in turn gives rise to both directional as well as unidirectional links between the nodes.

**Energy Constrained Operation**

Almost all the nodes in an ad hoc network rely on batteries or other exhaustive means for their energy. The battery depletes due to extra work performed by the node in order to survive the network. Therefore, energy conservation is an important design optimization criterion.

**Bandwidth Constraint**

Wireless links have significantly lower capacity [47] than infrastructures networks. Throughput of wireless communication is much less because of the effect of the multiple access, fading, noise, interference conditions. As a result of this, congestion becomes a bottleneck in bandwidth utilization.

**Limited Physical Security**

MANETs are generally more prone to physical security threats than wireless networks because the ad hoc network is a distributed system and all the security threats relevant to such a system are pretty much present, as a result, there is an increased possibility of eavesdropping, spoofing, masquerading [130], and denial-of-service type attacks.

## 2.3 Routing in MANETs

A routing protocol [115] is the mechanism by which user traffic is directed and transported through the network from the source node to the destination node. Objectives include maximizing network performance from the application point of view - application requirements- while minimizing the cost of network itself in accordance with its capacity. The application requirements are hop count, delay, throughput, loss rate, stability, jitter, cost; and the network capacity is a function of available resources that reside at each node and number of nodes in the network as well as its density, frequency of end-to-end connection (i.e. number of communication), frequency of topology changes (mobility rate). The four core basic routing functionality for mobile ad hoc networks are:

- *Path generation*: which generates paths according to the assembled and distributed state information of the network and of the application; assembling and distributing network and user traffic state information.
- *Path selection*: which selects appropriate paths based on network and application state information.
- *Data Forwarding*: which forwards user traffic along the select route forwarding user traffic along the selected route.

- *Path Maintenance*: maintaining of the selected route.

Consequently routing is bounded by traffic requirements, network capacity and the security requirements, as illustrated in Figure. 2.1

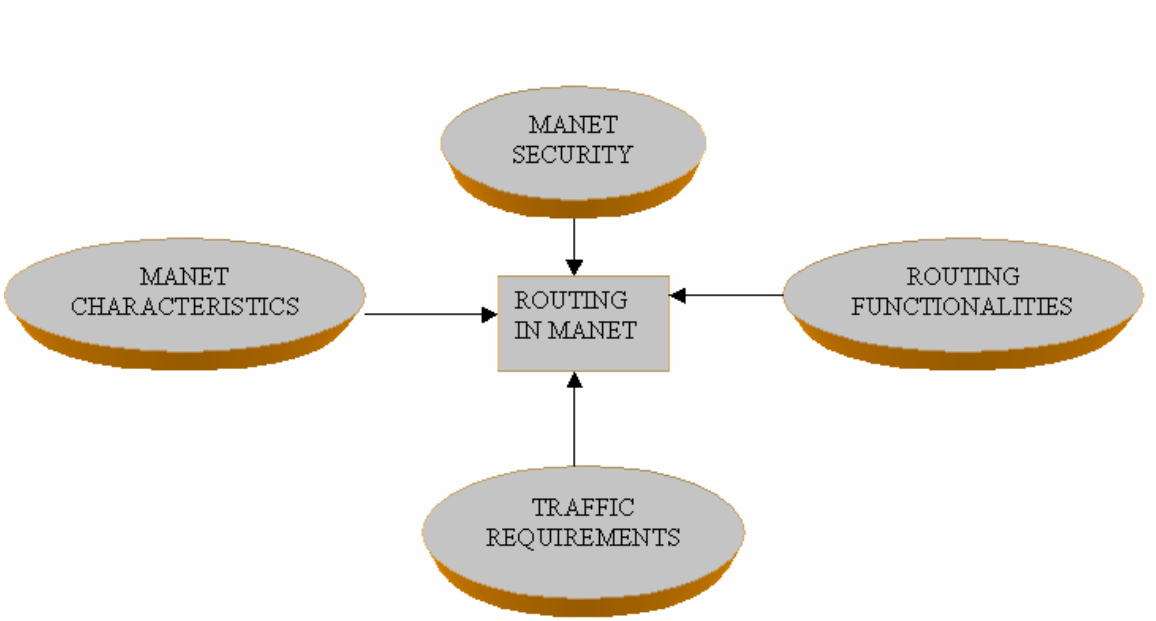



**Figure 2.1** Routing in MANET

Due to its characteristics, other desirable features of ad hoc routing protocol include- fast route establishment, multiple routes selection, energy/bandwidth efficiency and fast adaptability to link changes. Almost all routing systems respond in some way to the changes in network and user traffic state. However, routing systems vary widely in the types of *state* changes to which they respond and the speed of their response. Routing states can be divided into three categories - Static, Quasi Static and Dynamic. Further, each of the three basic routing functions may be implemented in three ways- Centralized, Decentralized and Distributed [36]. The routing protocols can be mainly categorized as: Flat routing, Hierarchical routing and Location aware routing [33,76,90,98].

### 2.3.1 Flat Routing

There are two schemes in flat routing, namely, table-driven (or proactive) routing protocols and on-demand (or reactive) routing protocols [106,140].

**I. Table-Driven Routing (Global/Proactive protocols)**

In proactive routing protocols, the routes to all the destinations (or parts of the network) are determined at the start up, and maintained by using a periodic route update process. In proactive routing protocol each node maintains the information about the other nodes in the tables. Though the number of tables used by the different protocols differ. The various proactive routing protocols differ in the way in which they update the routing information in the tables.

**a) Destination Sequenced Distance Vector (DSDV)**

The DSDV algorithm [32] is a modification of Distributed Bellman Ford algorithm, which guarantees loop free routes. It provides a single path to a destination, which is selected using the distance vector shortest path routing algorithm.

In order to reduce the amount of overhead transmitted through the network, two types of update packets are used. These are referred to as a ‘‘full dump’’ and ‘incremental’ packets. The full dump packet carries all the available routing information and the incremental packet carries only the information changed since the last full dump. The incremental update messages are sent more frequently than the full dump packets. However, DSDV still introduces large amounts of overhead to the network due to the requirement of the periodic update messages, and the overhead grows according to $O(N^2)$. Therefore, the protocol will not scale in large network since a large portion of the network bandwidth is used in the updating procedures.

**b) Wireless Routing Protocol (WRP)**

The WRP protocol [128] also guarantees loops freedom and it avoids temporary routing loops by using the predecessor information. However, WRP requires each node to maintain four routing tables. This introduces a significant amount of memory overhead at each node as the size of the network increases. Another disadvantage of WRP is that it ensures connectivity through the use of *hello* messages. These *hello* messages are exchanged between neighboring nodes

whenever there is no recent packet transmission. This will also consume a significant amount of bandwidth and power as each node is required to stay active at all times (i.e., they cannot enter sleep mode to conserve their power). The Table 2.1 and Table 2.2 outline the basic characteristics and complexity of the two routing protocols discussed above.

**Table 2.1.** Basic Characteristics of DSDV and WRP

| **Protocol** | **Routing Structure** | **Number of Tables** | **Frequency of Updates** | **Hello Message** | **Critical Nodes** | **Characteristic Feature** |
|---|---|---|---|---|---|---|
| DSDV | Flat | 2 | Periodic and as Required | Yes | No | Loop Free |
| WRP | Flat | 4 | Periodic | Yes | No | Loop freedom using predecessor Information |

**Table 2.2.** Complexity Comparison of DSDV and WRP

| **Protocol** | **Convergence Time** | **Memory Overhead** | **Control Overhead** | **Advantages/ Disadvantages** |
|---|---|---|---|---|
| DSDV | O(Diameter of network –1) | O(Number of Nodes in the Network) | O(Number of Nodes in the Network) | Loop free/high overhead |
| WRP | O(Height of the Routing Tree) | O(Number of Nodes in Network)$^2$ | O(Number of Nodes in the Network) | Loop free/memory overhead |

**II. On-Demand Routing (Reactive Protocols)**

In reactive protocols, routes are determined when they are required by the source using a route discovery process. These protocols were designed to reduce the overhead encountered in proactive protocols by maintaining information for active routes only. This means that the routes are determined and maintained for the nodes that are required to send data to a particular destination. Route discovery usually occurs by flooding route request packets through the network. When a node with a route to the destination (or the destination itself) is reached a route reply is sent back to the source node using link reversal if the route request has

traveled through the bi-directional links or by piggy-backing the route in a route reply packet via flooding. Therefore, the route discovery overhead (in the worst case scenario) will grow by O(N+M) when link reversal is possible and O(2N) for unidirectional links (where, N represents the total number of nodes and M represents the total number of nodes in the localized region).

Reactive protocols can be classified into two categories:

i. Source routing, and
ii. Hop-by-Hop routing

In *Source routed* on-demand protocols, each data packets carry the complete source to destination address. Therefore, each intermediate node forwards these packets according to the information kept in the header of each packet. This means that the intermediate nodes do not need to maintain up-to-date routing information for each active route in order to forward the packet towards the destination. Furthermore, nodes do not need to maintain neighbors connectivity through periodic beaconing messages. The major drawback with source routing protocols is that in large networks they do not perform well.

In *hop-by-hop routing* (also known as point-to-point routing), each data packet only carries the destination address and the next hop address. Therefore, each intermediate node in the path to the destination uses its routing table to forward each data packet towards the destination. The advantage of this strategy is that routes are adaptable to the dynamically changing environment of MANETs, since each node can update its routing table when they receiver fresher topology information and hence forward the data packets over fresher and better routes. The disadvantage of this strategy is that each intermediate node must store and maintain routing information for each active route and each node may require to be aware of their surrounding neighbors through the use of beaconing messages.

This following section describes the three protocols along with their performance comparison. The performance metrics represent the worst-case scenario.

**a) Dynamic State Routing (DSR)**

The DSR protocol [43,44,52,129] requires each packet to carry the full address (every hop in the route), from source to the destination. This means that the protocol will not be very effective in large networks, as the amount of overhead carried in the packet will continue to increase as the network diameter increases. Therefore, in highly dynamic and large networks the overhead may consume most of the bandwidth. However, this protocol has a number of advantages over other routing protocols, and in small to moderately size networks (perhaps up to a few hundred nodes), this protocol performs better. An advantage of DSR is that nodes can store multiple routes in their route cache, which means that the source node can check its route cache for a valid route before initiating route discovery, and if a valid route is found there is no need for route discovery. This is very beneficial in network with low mobility, because the routes stored in the route cache will be valid for a longer period of time. Another advantage of DSR is that it does not require any periodic beaconing (or *hello* message exchanges), therefore nodes can enter sleep node to conserve their power. This also saves a considerable amount of bandwidth in the network. A full description of this protocol appears in later text (Refer section 2.9).

**b) Ad hoc On-demand Distance Vector (AODV)**

The AODV routing protocol [31,37,40] is based on DSDV and DSR algorithm. It uses the periodic beaconing and sequence numbering procedure of DSDV and a similar route discovery procedure as in DSR. However, there are two major differences between DSR and AODV. The most distinguishing difference is that in DSR each packet carries full routing information, whereas in AODV the packets carry the destination address. This means that AODV has potentially less routing overheads than DSR. The other difference is that the route replies in DSR carry the address of every node along the route, whereas in AODV the route replies only carry the destination IP address and the sequence number. The advantage of AODV is that it is adaptable to highly dynamic networks. However, node may experience large delays during route construction, and link failure may initiate

another route discovery, which introduces extra delays and consumes more bandwidth as the size of the network increases.

**c) Temporally Ordered Routing Algorithm (TORA)**

Temporally ordered routing algorithm (TORA) [127] is a distributed routing algorithm. The basic underlying algorithm is the one in the family is referred to as link reversal algorithms. TORA is designed to minimize reaction to topological changes. The key concept is that control messages are typically localized to very small set of nodes. It guarantees that all routes are loop free and typically provides many routes to source/destination pair .It provides only the routing mechanism and depends upon Internet MANET Encapsulation Protocol (IMEP) for other underlying functions. Each node has a quintuple associated with it, as represented

- Logical time of link failure,
- The unique ID of the node that defined the new reference level,
- A reflection indicator bit,
- A propagation ordering parameter, and
- The unique ID of the node.

The first three elements collectively represent the reference level. A new reference level is defined each time a node loses its last downstream link due to link failure.

Tables 2.3 and 2.4 outline the basic characteristics and complexity of the three routing protocols discussed in this section.

**Table 2.3.** Basic Characteristics of DSR, AODV and TORA

| **Protocol** | **Multiple Routes** | **Route Metric Method** | **Route Maintained In** | **Route Reconfiguration Strategy** |
|---|---|---|---|---|
| DSR | Yes | Shortest Path or next path available | Route Cache | Erase Route then Source Notification. |
| AODV | No | Freshest and Shortest Path | Route Table | Erase Route then Source Notification or Local |

| | | | | |
|---|---|---|---|---|
| | | | | Route Repair. |
| TORA | Yes | Shortest Path or next path available | Route Table | Link reversal and Route repair |

**Table 2.4.** Complexity Comparison of DSR, AODV and TORA

| **Protocol** | **Time Complexity for Route Discovery** | **Time Complexity for Route Maintenance** | **Advantage** | **Disadvantage** |
|---|---|---|---|---|
| DSR | O(2*Diameter of Network) | O(2*Diameter of Network) | Multiple Routes, Promiscuous overhearing | Scalability problem due to source routing and flooding. |
| AODV | O(2*Diameter of Network) | O(2*Diameter of Network) | Adaptable to highly dynamic topologies. | Scalability Problems and large Delays. |
| TORA | O(2*Diameter of Network) | O(2*Diameter of Network) | Multiple routes | Temporary routing loops. |

**III. Hybrid Protocols**

Hybrid routing protocols combine the basic properties of the two classes of flat routing protocols into one. That is, they are both reactive and proactive in nature. Each group has a number of different routing strategies, which employ a flat or a hierarchical routing structure.

Hybrid routing protocols are new generation of protocols, which are both proactive and reactive in nature. These protocols are designed to increase scalability [68] by allowing nodes with close proximity to work together to form some sort of a backbone to reduce the route discovery overheads. This is mostly achieved by proactively maintaining routes to near by nodes and determining routes to far away nodes using a route discovery strategy. Most hybrid protocols proposed to date are zone-based, which means that the network is partitioned or seen as a number of zones by each node. Other nodes group into trees or clusters. This section describes two such routing protocols proposed for MANETs.

**a) Zone Routing Protocol (ZRP)**

In ZRP [145], the nodes have a routing zone, which defines a range (in hops) that each node is required to maintain network connectivity proactively. Therefore, for nodes within the routing zone, routes are immediately available. For nodes that lie outside the routing zone, routes are determined on-demand (i.e. reactively), and it can use any on-demand routing protocol to determine a route to the required destination. The advantage of this protocol is that it has significantly reduced the amount of communication overhead when compared to pure proactive protocols. It also has reduced the delays associated with pure reactive protocols such as DSR, by allowing routes to be discovered faster. This is because, to determine a route to a node outside the routing zone, the routing only has to travel to a node, which lies, on the boundaries (edge of the routing zone) of the required destination. Since the boundary node would proactively maintain routes to the destination (i.e. the boundary nodes can complete the route from the source to the destination by sending a reply back to the source with the required routing address). The disadvantage of ZRP is that for large values of routing zone the protocol can behave like a pure proactive protocol, while for small values it behaves like a reactive protocol.

**b) Zone-based Hierarchical Link State (ZHLS)**

Unlike ZRP, ZHLS routing protocol [145] employs hierarchical structure. In ZHLS, the network is divided into non-overlapping zones, and each node has a node ID and a zone ID, which is calculated using a GPS. The hierarchical topology is made up of two levels: node level topology and zone level topology, as described previously. In ZHLS location management has been simplified. This is because no cluster-head or location manager is used to coordinate the data transmission. This means there is no processing overhead associated with cluster-head or Location Manager selection when compared to HSR, MMWN and CGSR protocols [18]. This also means that a single point of failure and traffic bottlenecks can be avoided. Another advantage of ZHLS is that it has reduced the communication overheads when compared to pure reactive protocols such as DSR and AODV. In ZHLS, when a route to a remote destination is required (i.e. the destination is in another zone), the source node broadcast a zone level location

request to all other zones, which generates significantly lower overhead when compared to the flooding approach in reactive protocols. Another advantage of ZHLS is that the routing path is adaptable to the changing topology since only the node ID and the zone ID of the destination is required for routing. This means that no further location search is required as long as the destination does not migrate to another zone. However, in reactive protocols any intermediate link breakage would invalidate the route and may initiate another route discovery procedure. The Disadvantage of ZHLS is that all nodes must have a preprogrammed static zone map in order to function. This may not feasible in applications where the geographical boundary of the network is dynamic. Nevertheless, it is highly adaptable to dynamic topologies and it generates far less overhead than pure reactive protocols, which means that it may scale well to large networks. Tables 2.5 and Table 2.6 outline the characteristics and complexity of the two routing protocols discussed above.

**Table 2.5.** Basic Characteristics of ZRP and ZHLS

| **Protocol** | **Routing Structure** | **Multiple Routes** | **Route Metric Method** | **Route Maintained In** | **Route Reconfiguration Strategy** |
|---|---|---|---|---|---|
| ZRP | Flat | No | Shortest Path | Interzone and Intrazone Tables | Route Repair At point of Failure and Source Notification. |
| ZHLS | Hierarchical | Yes , If More than One Virtual Link Exists. | Shortest Path of the Next Available Virtual Link. | Interzone and Intrazone Tables | Location Request. |

**Table 2.6.** Complexity Comparison of ZRP and ZHLS

| **Protocol** | **Time Complexity for Route Discovery** | **Communication Complexity for Route Discovery** | **Advantage** | **Disadvantage** |
|---|---|---|---|---|
| ZRP | Intra:O(Periodic Upadate Interval) / Inter:O(2*Diameter of Network) | O(Number of Nodes in a zone)/ O(Number of nodes in the network + Number of Nodes in the Route Reply path.) | Reduce Retransmission | Overlapping Zones |
| ZHLS | Intra:O(Periodic Upadate Interval) / Inter:O(Diameter of Network) | O(Number of nodes in the network / Number of Zones in the Network)/ O(Number of nodes in the network + Number of Nodes in the Route Reply path) | Reduction of SPF, low CO | Static zone map required |

Figure.2.2 provides a brief outline of the different flat-routing protocols proposed for MANETs.

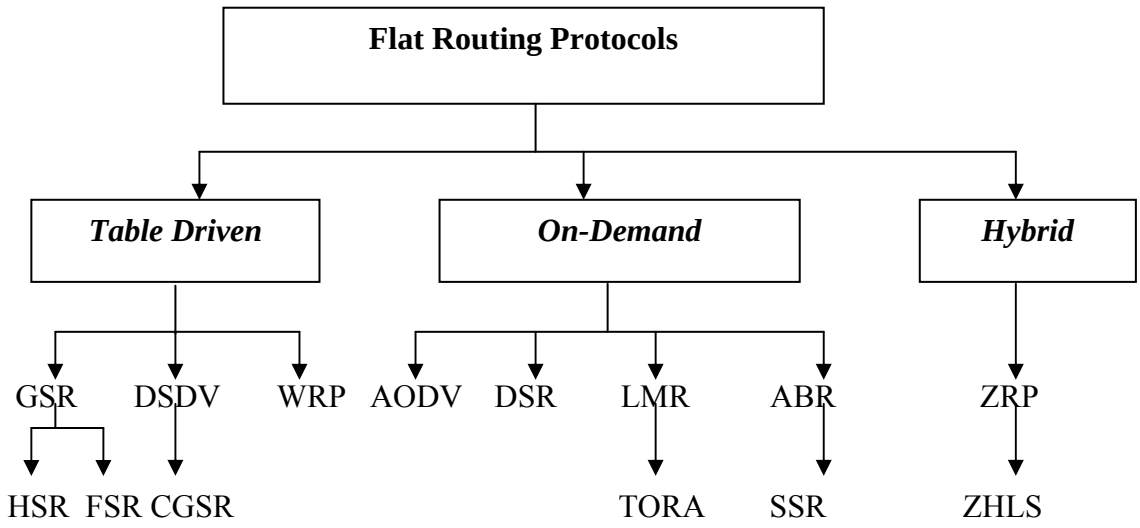


**Figure 2.2.** Classification of Routing Protocols in MANETs

### 2.3.2 Other Routing Protocols

The ***Hierarchical*** routing protocols [36] are used when the network size increases and the flat routing becomes in feasible due to increase in processing overhead. In this approach, the MANET is partitioned into different groups and the nodes are assigned different functions within and outside the group. Some of the routing protocols are: Cluster Based Routing Protocol (CBRP), Core Extraction Distributed Ad hoc Routing (CEDAR) [96], Fisheye State Routing protocol (FSR), Global State Routing protocol (GSR) etc.

The ***Geographical*** routing protocols [73] imply that the hosts participating in the routing process should be aware of their geographic positions. Some of these routing protocols are: Adaptive Location Aided Routing – Mines (ALARM), Distance Routing Effect Algorithm for Mobility (DREAM), GPS Ant-Like Routing Algorithm (GPSAL) etc [142].

There is another category of routing protocols, known as ***Power Aware*** routing protocols [88]; energy optimization is the main criteria to decide the transmission route. These type of routing protocols [67] take into consideration the energy required to transmit a signal, because the energy required is proportional to the square of the distance and transmitting a signal half the distance requires one fourth of the energy [107]. Some of these routing protocols are: Power-Aware Routing Optimization Protocol (PARO), Power Aware Multi Access Protocol with Signaling Ad Hoc Networks (PAMAS) etc.

***Multicast*** routing algorithms such as: On-Demand Associativity-Based Multicast routing protocol (ABAM), Adhoc Multicast Routing Protocol (AMRoute), Multicast Ad-hoc On-Demand Distance Vector routing (MAODV) etc. are used to address the multicasting issues in MANETs and have been extended to geographcally aware mobile nodes, also known as Geographical multicast (better known as ***Geocasting***) routing protcols. Some of the routing protocols are:

Location Based Multicast (LBM), GeoTORA (inspired from TORA, a reactive routing protocol) etc.

Lately, a new kind of completely distributed routing protocols [5] for MANETs have been proposed. These protocols are based upon Swarm intelligence and are inspired from the behavior of social insects [58,60].

## 2.4 Network Simulation

Simulation [6] can be defined as "Imitating or estimating how events might occur in a real situation". It can involve complex mathematical modeling, role-playing without the aid of technology, or combinations. The importance of simulation lies in the consideration of realistic conditions that change as a result of behavior of others involved and thus we can anticipate the sequence of events or the final outcome. Different simulators such as ns2, GloMoSim, OPNET etc., are being used by researchers in order to evaluate the routing protocols. We have used ns2 for the evaluation of the proposed routing protocol as the same is an open source, freely available and the programming languages used are C++, Tcl and OTcl.

### 2.4.1 ns2-Overview

The Network Simulator (ns) [42,72] is an event driven network simulator developed at UC Berkeley that simulates variety of IP networks. It implements network protocols such as Transmission Control Protocol and User Datagram Protocol, traffic source behavior such as File Transfer Protocol, Telnet, Web, Constant Bit Rate and Variable Bit Rate, queue management mechanism, routing algorithms and more. ns also implements multicasting and some of the MAC layer protocols for LAN simulations. The ns project is now a part of the VINT project that develops tools for simulation results display, analysis and converters that convert network topologies generated by well-known generators to ns formats. Currently, ns (version 2) written in C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT) is available.

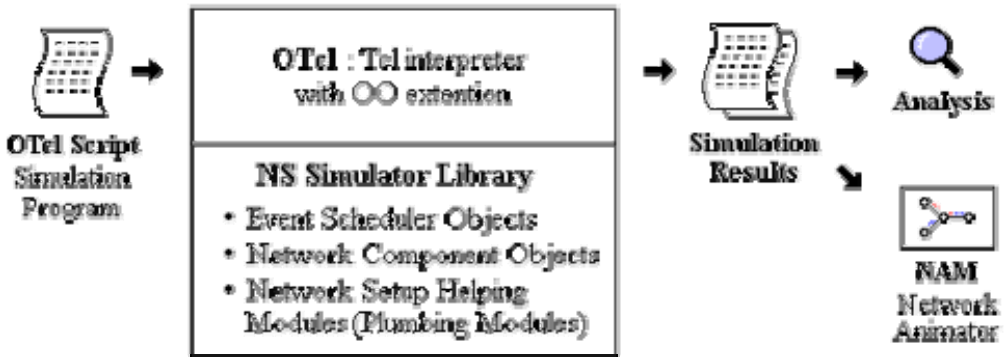
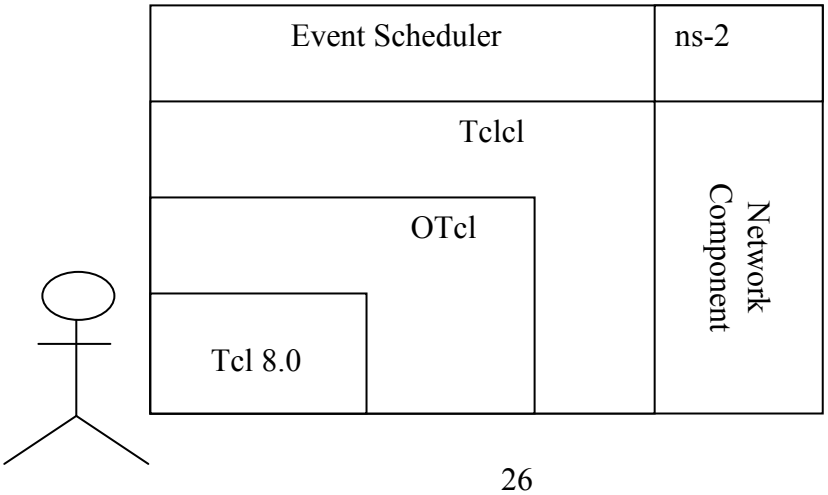



**Figure 2.3.** Simplified User's View of ns

Figure 2.3, presents a simplified user's view, ns is Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network setup module libraries.

To setup a simulation network, an OTcl script is written and to simulate it the script is executed which initiates an event scheduler and the network topology is setup using the network objects, controlling the traffic sources and the time to start and stop the transmitting of packets.

Figure 2.4 shows the general architecture of ns. In this figure, a general user (not an ns developer) can be thought of standing at the left bottom corner, designing and running simulations in Tcl using the simulator objects in the OTcl library. The event schedulers and most of the network components are implemented in C++ and available to OTcl through the OTcl linkage that is implemented using tclcl. The whole thing together makes the ns, which is a OO extended Tcl interpreter with network simulator libraries.



User

**Figure 2.4.** Architectural view of ns

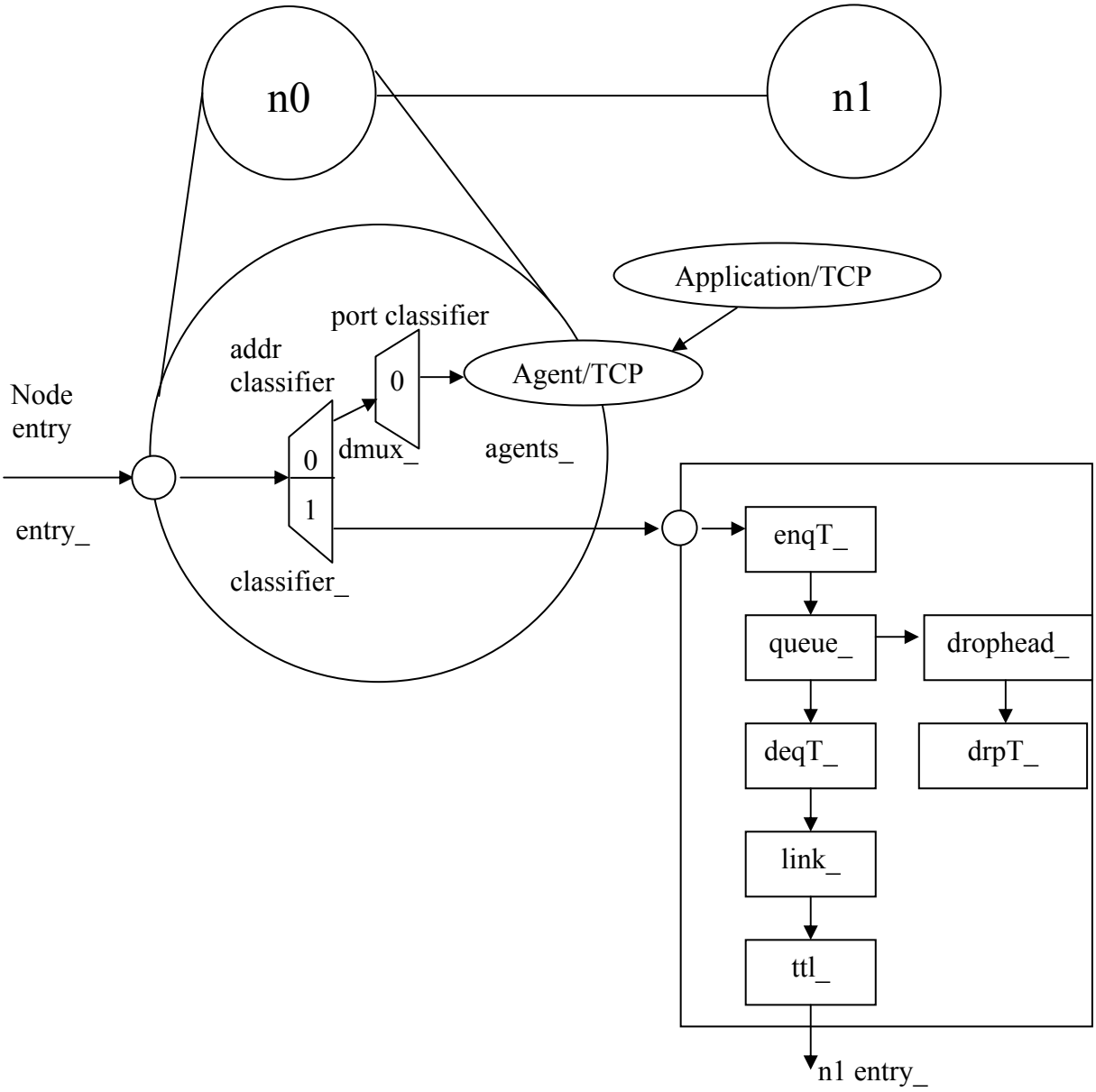



**Figure 2.5.** Internal mechanism of NS2 for routing in MANETs

The biggest advantage of network animator (NAM) is that it provides a graphical user interface (GUI) for the different simulation environment according to the parameters specified by the user. The Xgraph utility generates the graphical output of the input data (or trace files).

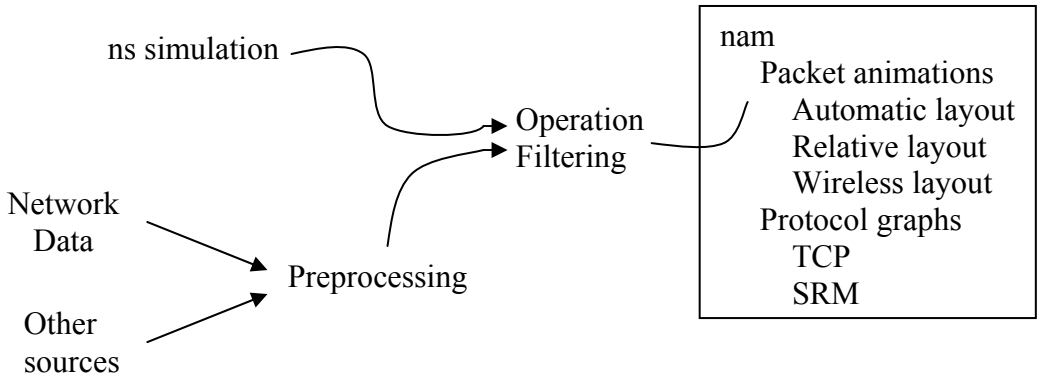



**Figure 2.6.** Data flows into nam from network data and other sources after pre-processing into the nam trace format.

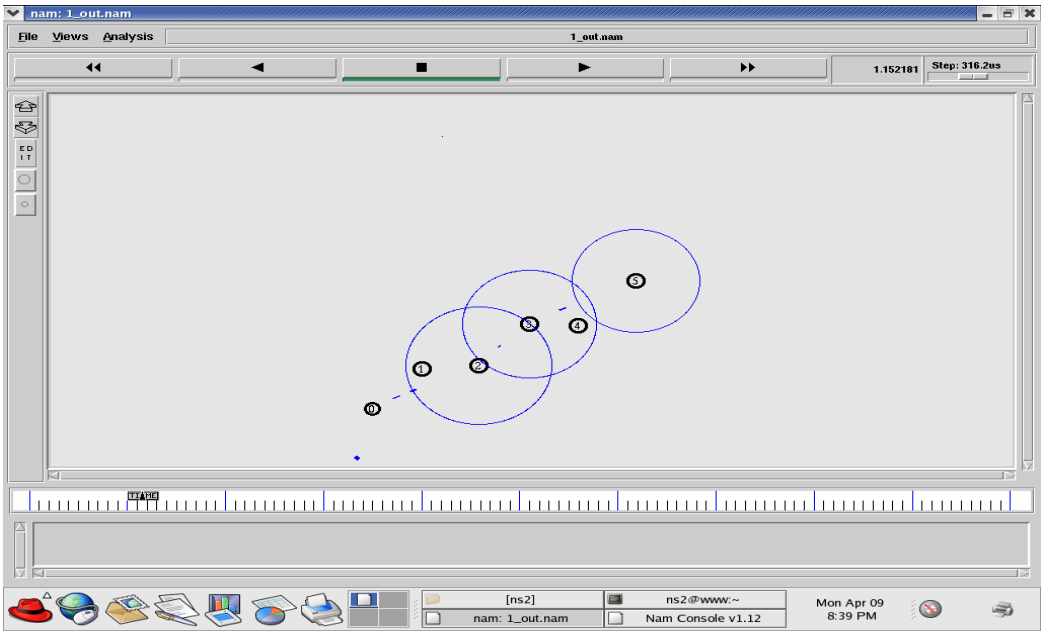



**Figure 2.7.** A sample screen-shot of Network Animator (NAM) being executed on Linux environment for 6 mobile nodes.

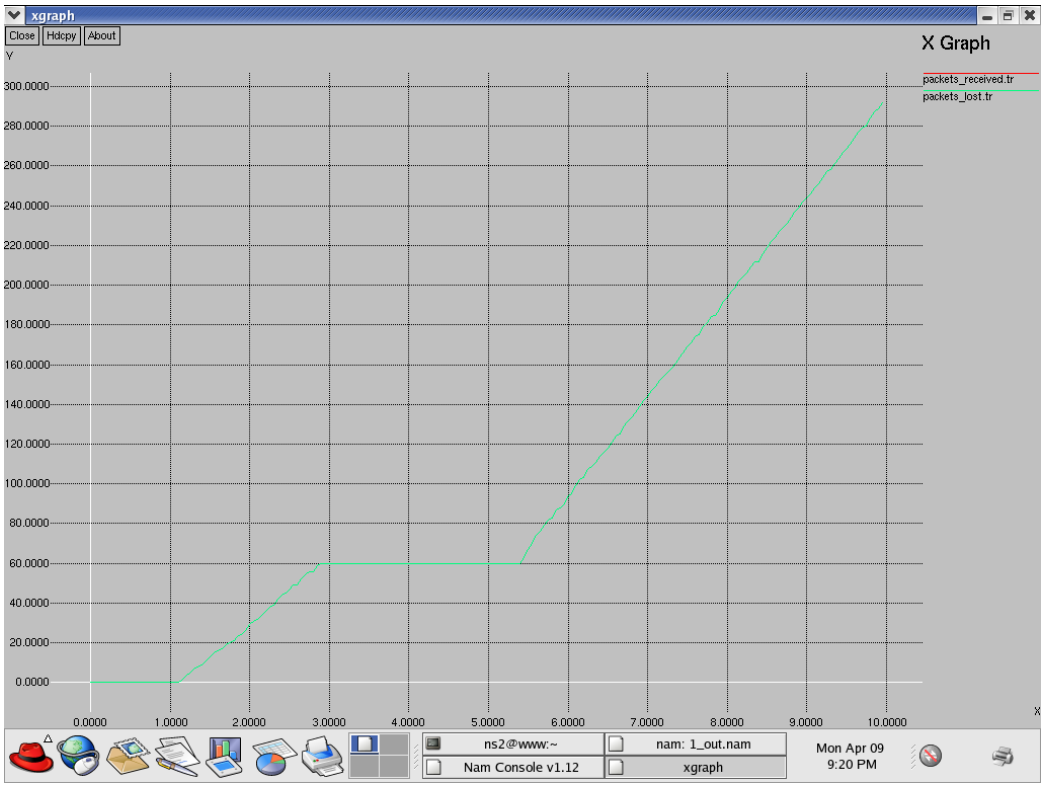



**Figure 2.8.** A sample screen-shot of the Xgraph utility for the simulation scenario as mentioned in Figure 2.5.

## 2.5 Mobility Models

To evaluate the performance of protocol in MANET, the protocol should be tested under realistic conditions such as – transmission range, data traffic, movement of mobile users (nodes) etc. There have been a wide variety of mobility models (MM) proposed and it is expected the MM should attempt to mimic the movement of real mobile nodes, the changes in speed and direction must occur in reasonable time slots. The MM can further be classified as [53,126]:

- Entity Mobility Model (EMM), and
- Group Mobility Model (GMM).

There are seven different categories of EMM, as defined below:

- Random Walk Mobility Model is a simple mobility model based on random directions and speeds.

- Random Waypoint Mobility Model includes pause times between changes in destination and speed.
- Random Direction Mobility Model is the model that forces mobile nodes to travel to the edge of the simulation area before changing direction and speed.
- Boundless Simulation Area Mobility Model converts a 2D regular simulation area into a torus-shaped simulation area.
- Gauss-Markov Mobility Model uses one tuning parameter to vary the degree of randomness in the mobility pattern.
- Probabilistic Version of the Random Walk Mobility Model utilizes a set of probabilities to determine the next position of a mobile node.
- City Section Mobility Model is a simulation area that represents streets within a city.

The five different categories of GMM are:

- Exponential Correlated Random Mobility Model uses a motion function to create movements.
- Column Mobility Model, the set of mobile nodes form a line and are uniformly moving forward in a particular direction.
- Nomadic Community Mobility Model, a set of mobile nodes moves together from one location to another.
- Pursue Mobility Model is a GMM where a set of mobile nodes follow a given target.
- Reference Point Group Mobility Model, the group movements are based upon path traveled by a logical center.

Further, as per [55], the various categories of MM are illustrated below:

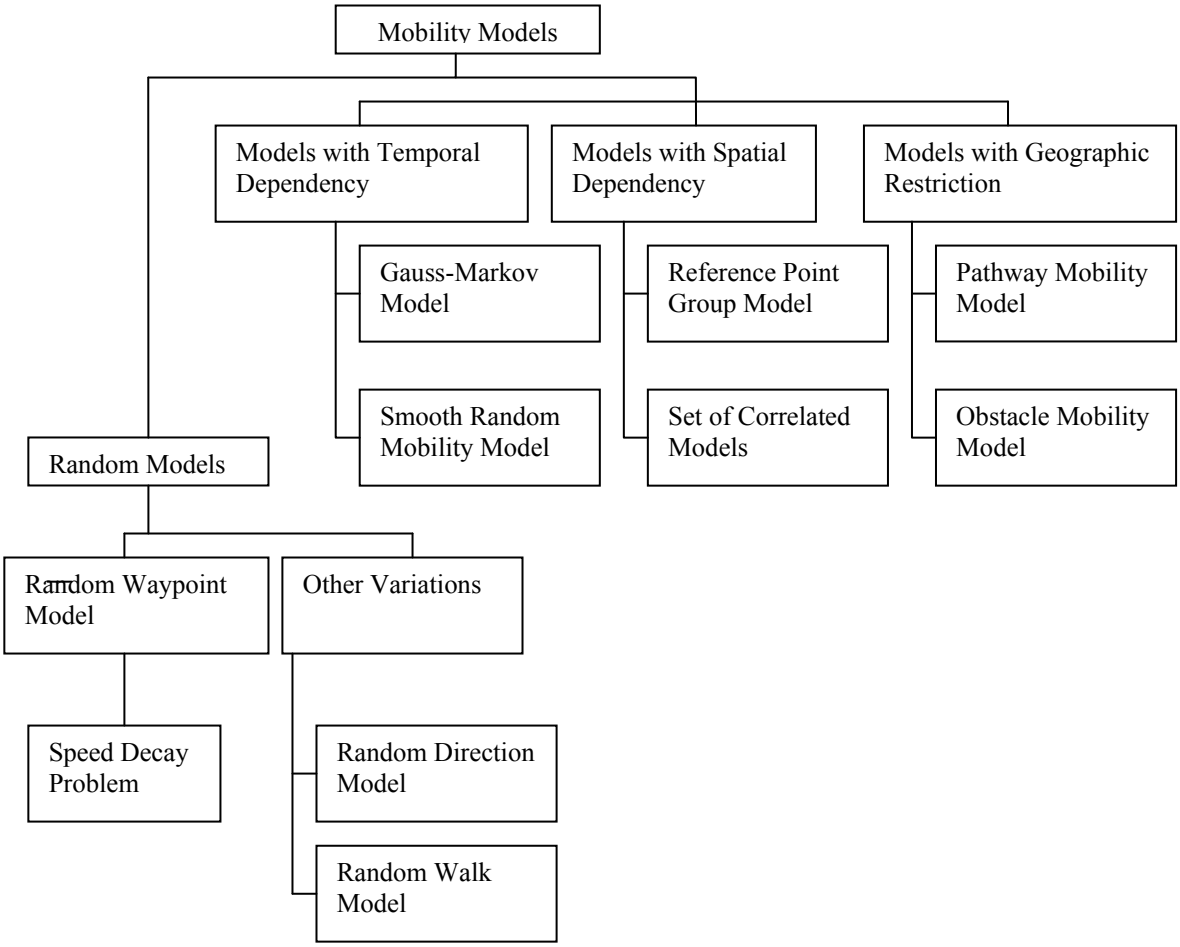



**Figure 2.9.** Various Categories of Mobility Model in MANETs

*A. Random Waypoint Mobility Model*

The Random Waypoint Mobility Model [25,27] is extensively used in simulation studies of MANET. In this mobility model a node selects its destination and its speed. The node keeps moving until it reaches its destination at that speed. A mobile node begins the simulation by waiting a specified pause-time. After this time it selects a random destination in the area and a random speed distributed uniformly between 0 m/s and $V_{max}$. After reaching its destination point, the mobile node waits again pause for time seconds before choosing a new waypoint and speed.

The mobile nodes are initially distributed over the simulation area. This distribution is not representative to the final distribution caused by node movements. To ensure a random initial configuration for each simulation, it is necessary to discard a certain simulation time and to start registering simulation results after that time.

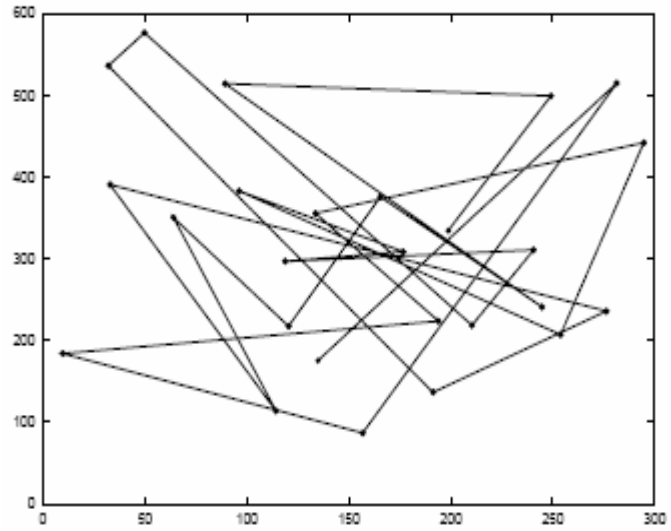



**Figure 2.10.** Traveling pattern of different mobile nodes in a given simulation area (rectangular shaped) based upon RWPMM.

**Characteristics of Random Waypoint**

The Random Waypoint model uses the concepts of *epoch* and *pause* making it a little bit more similar to realistic user mobility model [121]. It is widely accepted mainly due to its simplicity of implementation and analysis.

This section presents a very brief introduction about some of the MMs considered for different simulation scenarios.

*B. Reference Point Group Mobility Model*

In this model [126], each group has a logical center (group leader) that determines the groups motion behavior. Initially, each member of the group is uniformly distributed in the neighborhood of the group leader. Subsequently, at each instant, every node has a speed and direction that is derived by randomly deviating from that of the group leader.

**Applications:** Group mobility can be used in military battle field communications where the commander and soldiers form a logical group.

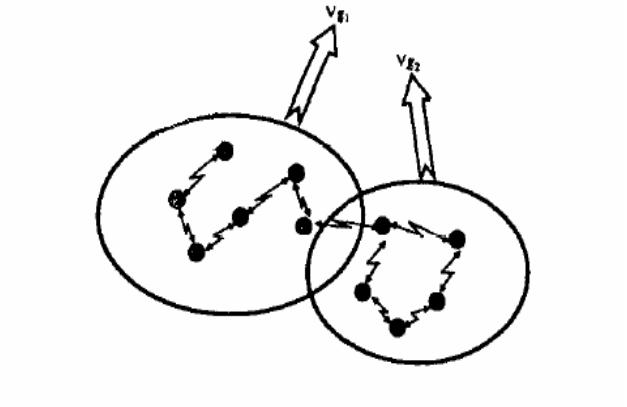


**Figure 2.11.** RPGM Model

*C. Freeway Mobility Model*

In this model [126] we use maps. There are several freeways on the map and each freeway has lanes in both directions.

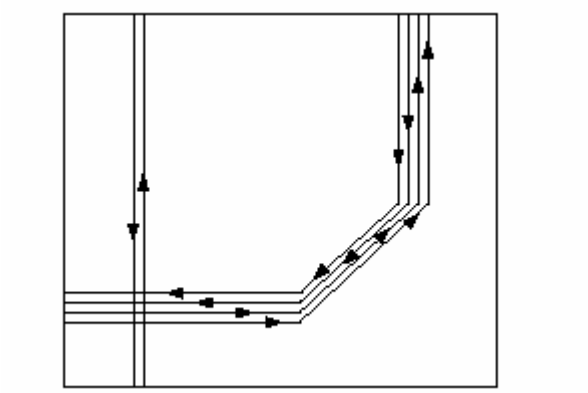


**Figure 2.12.** Freeway Map

The differences between Random Waypoint and Freeway are the following:

- Each mobile node is restricted to its lane on the freeway.
- The velsocity of mobile node is temporally dependent on its previous velocity.
- If two mobile nodes on the same freeway lane are within the safety distance, (SD), the velocity of the following node cannot exceed the velocity of preceding node.

**Applications:** It can be used in exchanging track status or tracking a vehicle [83] on a freeway.

*D. Manhattan Mobility Model*

The Manhattan model [126] is used to emulate the movement pattern of mobile nodes on streets defined by maps, also termed as City Section Mobility Model.

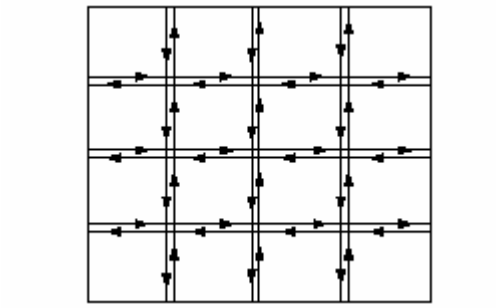


**Figure 2.13.** Manhattan map

The map is composed of a number of horizontal and vertical streets. Each street has two lanes for each direction (north and south direction for vertical streets, east and west for horizontal streets). The mobile node is allowed to move along the grid of horizontal and vertical streets on the map. At an intersection of a horizontal and a vertical street, the mobile node can turn left, right or go straight. This choice is probabilistic: the probability of moving on the same street is 0.5, the probability of turning left is 0.25 and the probability of turning right is 0.25. However, it differs from the Freeway model in giving a node some freedom to change its direction.

**Applications:** It can be useful in modeling movement in an urban area where a pervasive computing service between portable devices is provided.

## 2.6 Simulation Results for Different Routing Protocols

**Table 2.7.** Simulation Parameters for simulating selected flat routing protocols

| **Parameter** | **Value** |
|---|---|
| Number of nodes | 6 |
| Transmitter Range | 250 m |
| Simulation time | 10 sec |
| Pause Time | 0.1 sec |
| Environment Size | 670x670 m |
| Packet size | 200 bytes |
| Traffic Type | Constant Bit Rate |
| Packet Rate | 4 packet/sec |

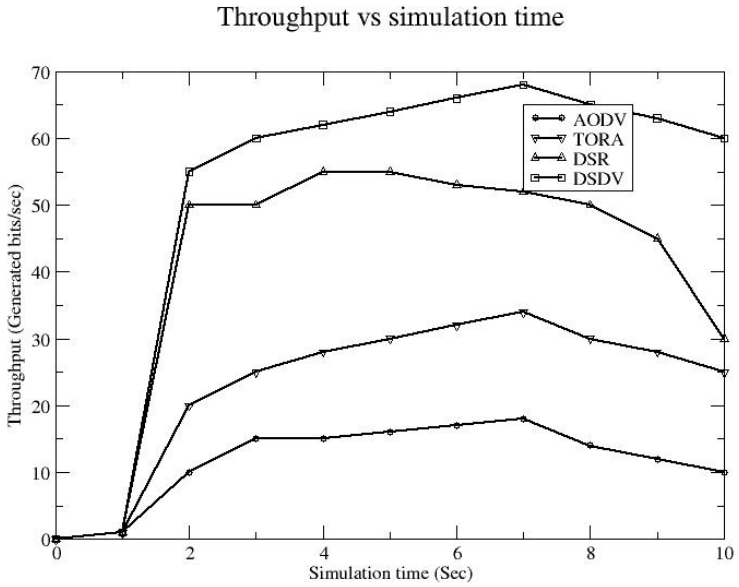



**Figure 2.14.** Throughput of MANET protocols

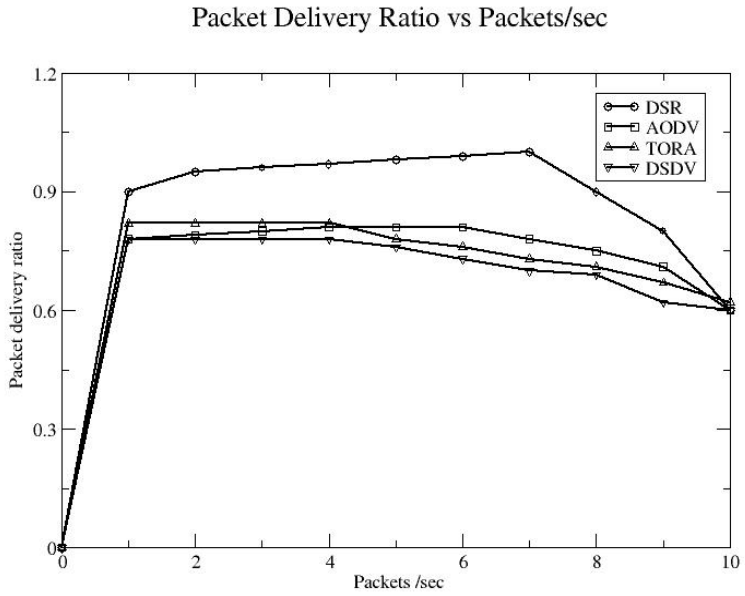



**Figure 2.15.** Packet delivery ratio of MANET protocols

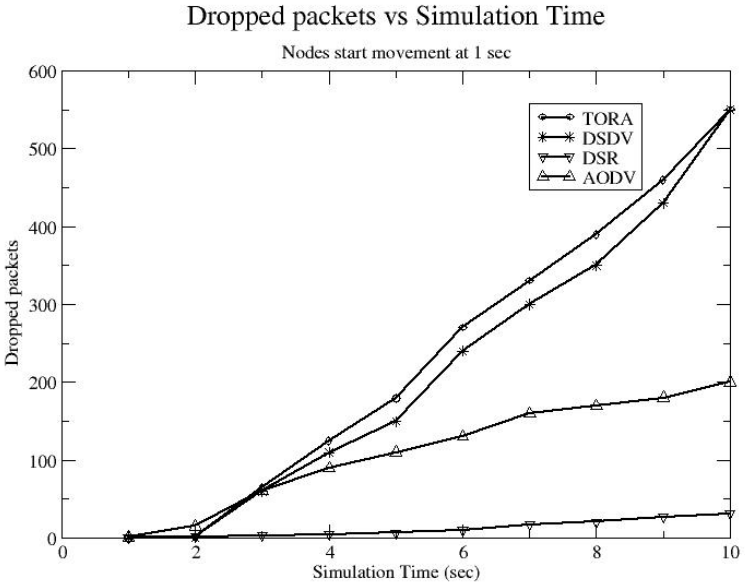



**Figure2.16.** Number of dropped packets

## 2.7 Simulation Results for Different Mobility Models

In this section, we consider different mobility models and the protocols are simulated under the following parameters:

**Table 2.8.** Simulation parameters for comparison of mobility models on different flat routing protocols.

| Parameter | Value |
|---|---|
| Number of nodes | 40 |
| Transmitter Range | 250 m |
| Simulation time | 900 sec |
| Pause Time | 0.1 sec |
| Environment Size | 1000x1000 |
| Packet Size | 64 bytes |
| Traffic Type | Constant Bit Rate |
| CBR sources | 20 |
| Packet rate | 4 packets/sec |

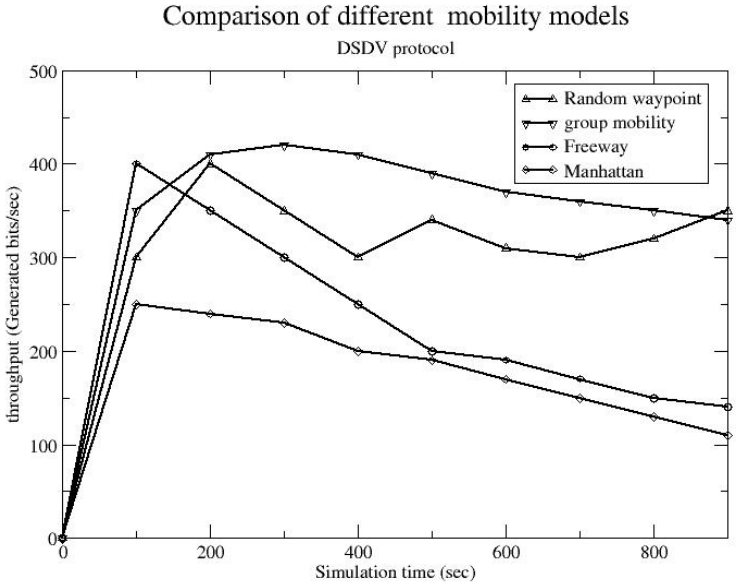



**Figure 2.17.** Throughput of DSDV protocol in different mobility models with 50 nodes.

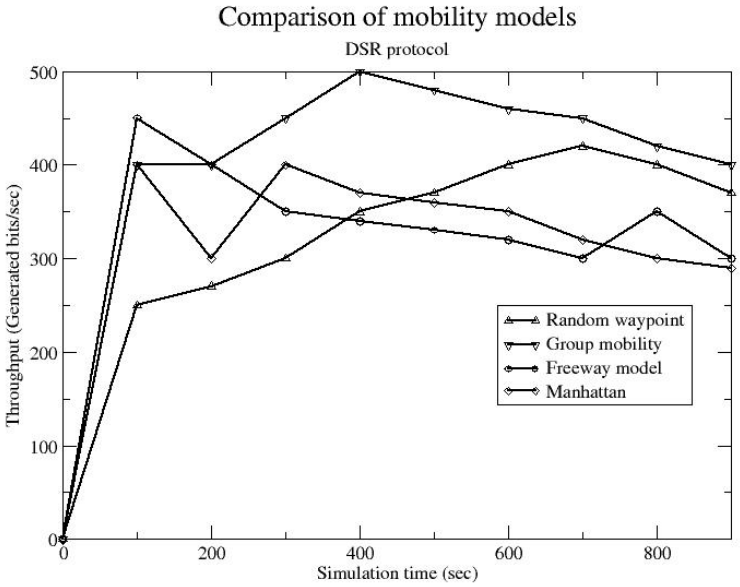



**Figure 2.18.** Throughput of DSR protocol in different mobility models with 50 nodes.

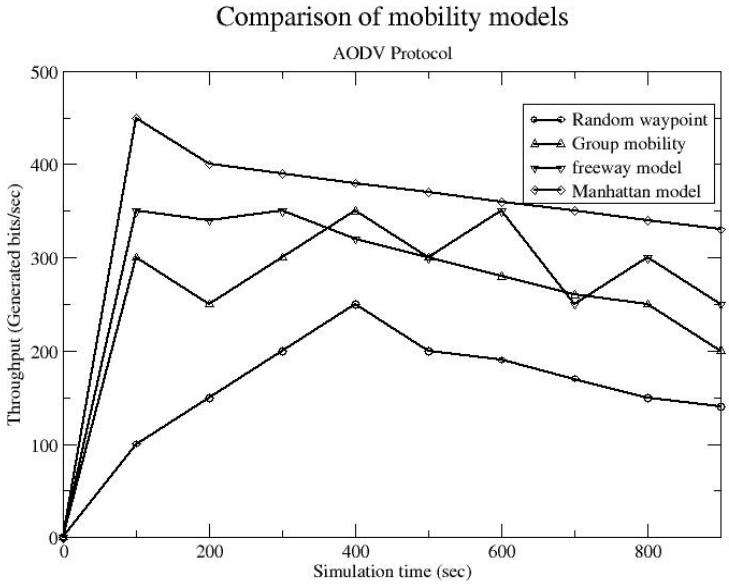



**Figure 2.19.** Throughput in AODV with 50 nodes for different mobility models.

## 2.8 Lessons Learned

While simulating and evaluating different routing protocols and their performance under the influence of different mobility models, we learned a great deal and would like to share our experience.

In our study the on-demand (reactive) protocols have a lower communication overhead because the roots are built only when required and there are no periodic updates required. Therefore, they do not incur substantial traffic and also consume less power.

The first packet latency is more (because the route is established, on demand) and the route to every other node in an ad hoc network is not available.
The storage requirements of on-demand routing protocols are also usually lower than their table-driven counterparts. All these factors are indicative that they use a lesser resources (power and storage) hence can provide better scalability.
It is amply clear that the mobility model chosen for simulation, although other network parameters remain unchanged, influences the behavior of a routing protocol.

## 2.9 Dynamic Source Routing (DSR) Protocol

Dynamic Source Routing (DSR) was developed at Carnegie Mellon University [43,44,52]. It is a direct descendant of the source routing scheme used in bridged LANs. This protocol is designed to restrict the bandwidth consumption by control packets as it eliminates the periodic table-update by the control packets. As compared with other on-demand routing protocols, it is a *beacon-less* and therefore does not require periodic *hello* packet (*beacon)* transmission, usually used by a node to inform its presence to the neighbors. The basic approach of this protocol is briefly described as under:

The sender of a packet determines the complete sequence of nodes through which the node has travel. The sender of the packet explicitly mentions the list of all nodes in the packet's header, identifying each forwarding 'hop' by the address of the next node to which to transmit the packet on its way to destination host. In this protocol the nodes don't need to exchange the Routing table information periodically and thus reduces the bandwidth overhead in the network. Each Mobile node participating in the protocol maintains a *routing cache,* which contains the list of routes that the node has learnt. Whenever the node finds a new route it adds the

new route in its routing cache. Each mobile node also maintains a sequence counter 'request id' to uniquely identify the requests generated by a mobile host. The pair < source address, request id > uniquely identifies any request in the ad hoc network. The protocol does not need transmissions between hosts to work in bi-direction. The main phases in the protocol are – *Route Discovery phase* and *Route Maintenance phase.*

### 2.9.1 Route Discovery Phase

Router discovery allows any host to dynamically discover the route to any destination in the Ad Hoc network. In DSR, a source initiates a route discovery process when the source wants to send a packet to a destination to which it doesn't have a valid route. The Source, if it has the valid route in its routing cache then it uses it otherwise it sends a route request packet by broadcasting it to the neighbors. The route request packet contains the source address; request- id and a route record in which the sequence of hops traversed by the request packet before reaching the destination are noted down. A node upon getting a Route request packet does the following:

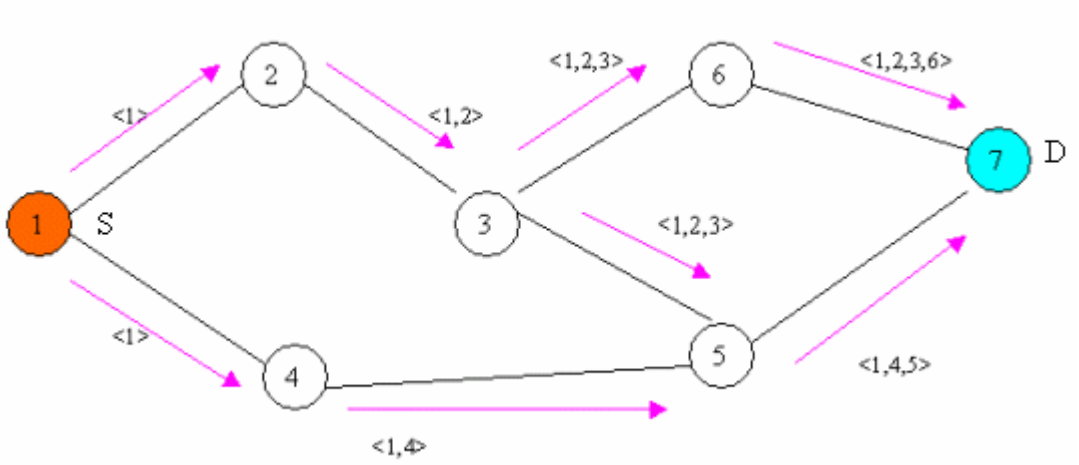
---

It checks to see if it has the pair <initiators address, request id> in its list of recently seen requests if so discard the packet.

1. Otherwise, if this host's address is already present in the route record of the request packet then it discards the packet. This eliminates the looping problem
2. Otherwise, if the destination the source is looking for matches with its address then it sends the route reply packet to the initiator containing the list of nodes the request packet has traversed before it reached the destination.
3. Otherwise, it appends its own address to the route request packet and rebroadcasts it. The route request travels the network until it reaches the destination node.

---

Any node forwards the route reply packet by using a route in its route cache if it has one for the initiator node or by using the node reverses the route in the reply packet to which node it need to send the reply packet.




Path 1:1-2-3-5-7
Path 2:1-2-3-6-7
Path 3: 1-4 –5-7

**Figure 2.20.** Propagation of the Route Request (also known as route establishment) showing the building of route entry from the Source 'S' to the Destination 'D' in a DSR.

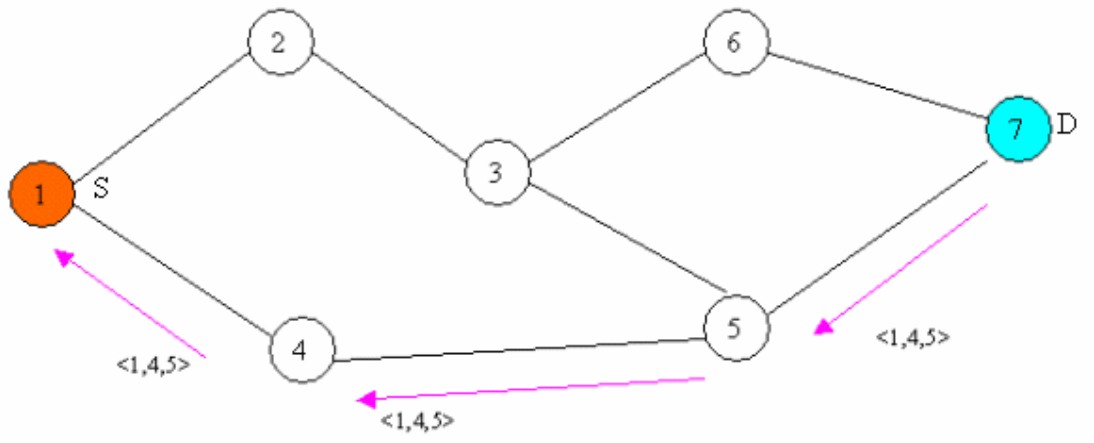



**Figure 2.21.** Propagation of the Route Reply containing the route entry from the Destination 'D'. to Source 'S' in a DSR.

### 2.9.2 Route Maintenance Phase

Route maintenance is a procedure of monitoring the correct operation of route in use. The host that uses the route does this maintenance. Since the nodes do not

exchange any routing information in this protocol the route maintenance procedure monitors the operation of the route and informs the source of any errors. Any host if it detects that its neighboring node, which is the next hop for a route, is not working then the node sends an *error packet* containing its address and the address of the hop not working. A node upon receiving the route error packet removes the hop in error from its routing cache. Acknowledgements are used to verify the correct operation of the route. The route maintenance can be provided by using either hop-to-hop or by using end-to-end acknowledgements. In case of hop-to-hop acknowledgements the hop in error is indicated in the route error packet. But in case of end-to-end acknowledgements the source node assumes that the last hop of the route to the destination is error.

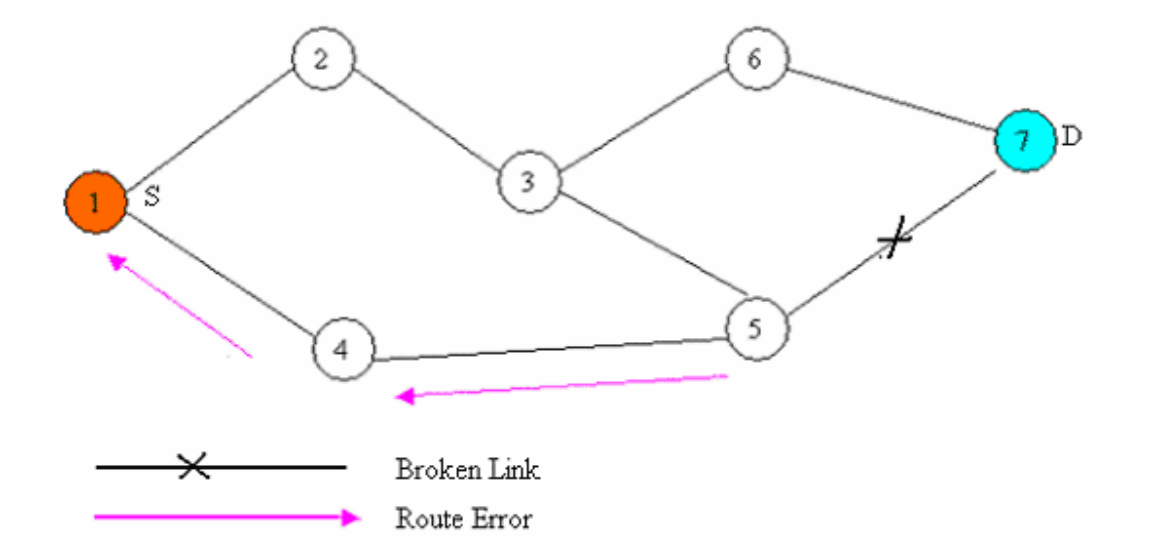



**Figure2.22.** Route maintenance in DSR

## 2.10 Simulation and validation of DSR

Here we describe the movement and communication patterns used for study and evaluation of the DSR protocol. The DSR was written in Tcl/Tk script and simulated using the ns-2 simulator.

A total of 100 nodes are simulated for 500s over a network space of 2200m x 600m. The traffic pattern is modeled as 40 CBR sources with data sent in 512-byte packets at 2 packets/s. Such a scenario allows a reasonably timed simulation, while

stressing the protocol with a sufficiently high load without causing congestion. The nodes move according to the random way point model (RWPM), with a pause time of 0s corresponds to continuous motion, while a pause time of 500s (length of simulation) corresponds to no motion. Each node can move at a maximum speed of 20m/s. Each pause time is simulated with 5 movement scenarios, each generated by using a different seed and we plot the mean of each performance metric over these 5 runs.

The protocol is evaluated based on the following metrics:

i) *Packet delivery ratio*: It can be defined as the ratio of number of data packets delivered to the destination with respect to the number of data packets generated by the traffic (CBR, in this case) source. It is also known as 'throughput' of the routing algorithm.

This metric is important as it describes the loss rate as seen by the protocol, which in turn affects the maximum throughput that a network can support. This metric characterizes both the correctness and completeness of the routing protocol.

ii) *Average end-to-end delay:* This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer time.

This metric determines how fast the algorithm delivers the data packets to the destination.

iii) *Routing overhead*: The total number of routing packets transmitted during the simulation. For packets sent over multiple hops, each transmission of the packet (each hop) counts as one transmission.

This metric is also an important as it measures the scalability of the protocol, the degree to which the protocol can function in a congested or low-bandwidth environments and also shows its efficiency in in terms of consuming node battery power.

iv) *Path optimality*: The difference between the numbers of hops a packet took to reach its destination and the length of the shortest path that physically existed through the network when packet was originated. In the absence of congestion or other noise, path optimality measures the ability of the routing protocol to efficiently use the network resources by selecting the shortest path from a source to destination.

**Table 2.9.** Summary of simulation parameters

| Parameter | Setting |
|---|---|
| Mobility model | Random way point |
| Traffic Model | 40 Constant Bit Rate (CBR) sources |
| Network space | 2200m x 600m |
| Number of nodes | 100 |
| Maximum node speed | 20 m/s |
| Packet sending rate | 2 packets/s |
| Data payload | 512 bytes |

### 2.10.1 Performance results

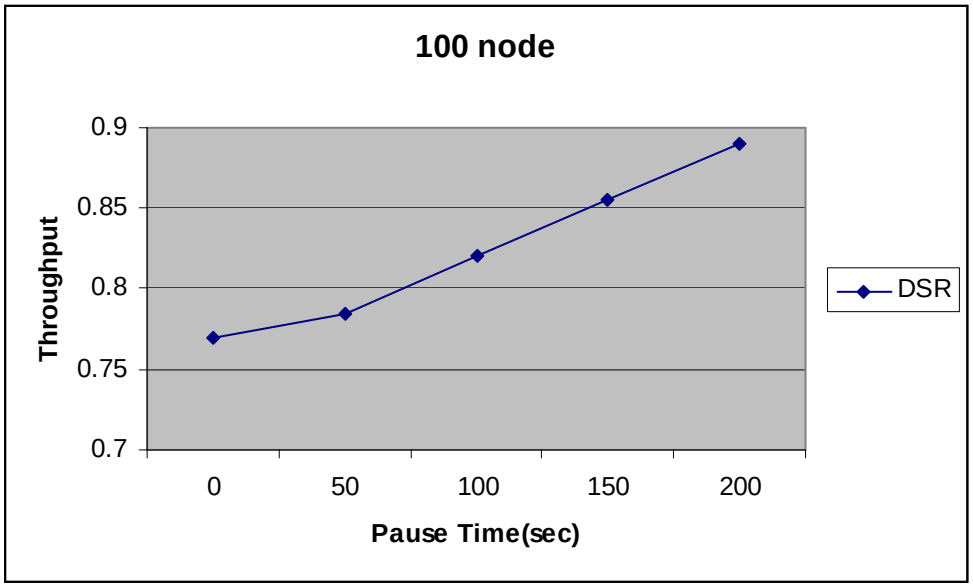



**Figure 2.23.** Packet delivery ratio (or Throughput)

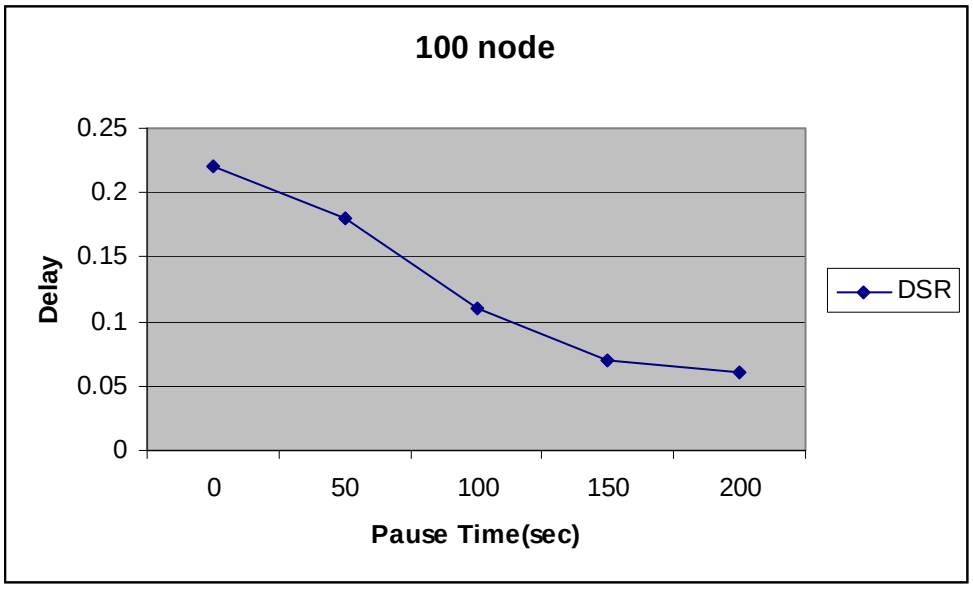



**Figure 2.24.** Average end-to-end delay

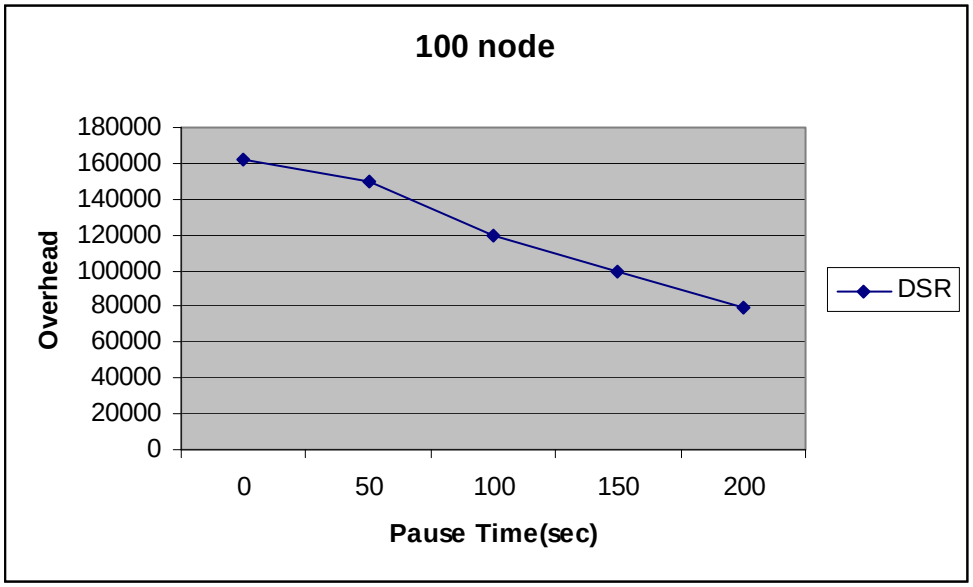



**Figure 2.25.** Routing Overhead

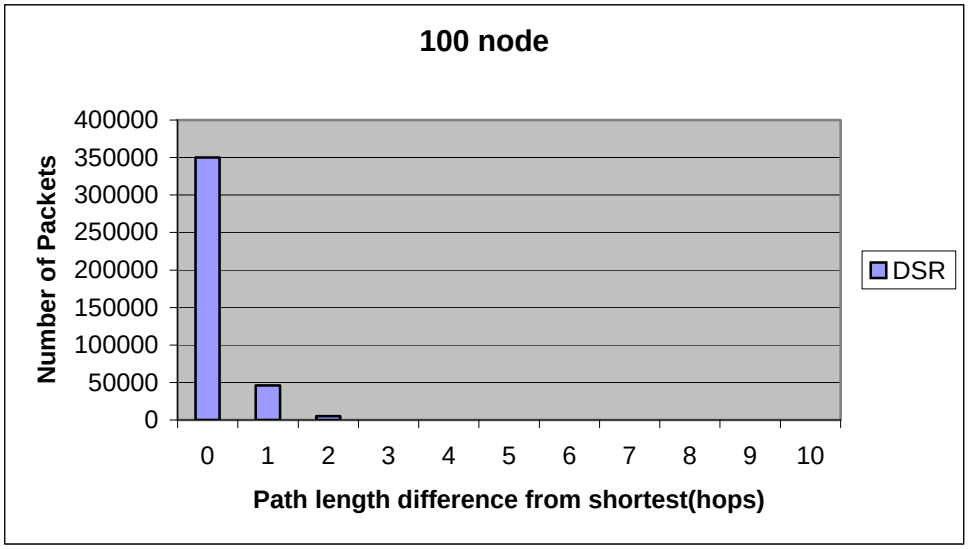



**Figure 2.26.** Path Optimality

### 2.10.2 Characteristics of DSR

The characteristics of the DSR are summarized in Table2.10:

**Table 2.10.** Characteristics of DSR

| **Property** | **Description** |
|---|---|
| *a) Basic* | |
| i) Routing Structure Type | Flat |
| ii) Multiple Routes | Yes |
| iii) Beacons | No |
| iv) Route Metric Method | Shortest Path or next available in route |
| v) Route Maintained In | Route Cache |
| vi) Route Configuration strategy | Erase route and then source notification |
| *b) Complexity* | |
| i) Time Complexity (Route discovery) | O (2 * Diameter of the network) |
| ii) Time Complexity (Route maintenance) | O (2 * Diameter of the network) |
| iii) Communication complexity (Route discovery) | O (2 * Number of nodes in the network) |
| iv) Communication complexity (Route maintenance) | O (2 * Number of nodes in the network) |

### 2.10.3 Advantages and Disadvantages of DSR

The route is established only when required (on-demand) and hence there is no need to find routes to all other nodes in the ad hoc network. It relieves the need to periodically flood the network with table update message usually required in a table-driven approach. The intermediate nodes also utilize the route cache information efficiently to reduce the control overhead and it performs equally well in static and low-mobility environments. The disadvantage of this protocol is that the route maintenance mechanism does not locally repair a broken link and stale route cache information could result in inconsistency during the route reconstruction phase. Further, the connection setup delay is higher than in table-driven protocols.

## 2.11 Some Desirable Properties of Good Routing Protocol for MANETs

The routing protocol performance issues in MANET centers around two areas. The first area concerns the limitation of the environment (wireless, limited bandwidth, battery powered, security) [17,18,19] and the other concerns the many ways in which data communication may take place leading to the desirable qualitative properties of MANET routing protocols, include the following:

1. Distributed operation: Decentralized nature of MANET requires that it execute its operation in a distributed fashion, a routing protocol should be able to support the same.
2. Loop-freedom: To ensure proper message delivery and efficient network operation, a protocol must be loop free.
3. Demand-based operation: Instead of assuming a uniform traffic distribution within the network (and maintaining routing between all nodes at all times), the routing algorithm should be capable to adapt to the traffic pattern on a demand or need basis. If this is done intelligently, it can utilize network energy and bandwidth resources more efficiently, at the cost of increased route discovery delay.
4. Security: A MANET routing protocol is vulnerable to many forms of attack. It is relatively simple to snoop network traffic, replay transmissions, manipulate

packet headers, and redirect routing messages, within a wireless network without appropriate security provisions. While these concerns exist within wired infrastructures and routing protocols as well but maintaining the physical security of the transmission media is harder in practice with MANETs. Sufficient security protection of the routing protocol operation is desired.

5. "Sleep" period operation: In order to conserve energy or some other need to be inactive, nodes of a MANET may stop transmitting and/or receiving (even receiving requires power) for arbitrary time periods. A routing protocol should be able to accommodate such sleep periods without overly adverse consequences.
6. Unidirectional link support: The presence of a unidirectional link in a MANET has a higher probability than in a wired network. In situations where a pair of unidirectional links in opposite directions forms the only means of communication between two ad hoc zones, the unidirectional link support would prevent unnecessary partitioning of the network.

In addition to above, a MANET routing protocol should function effectively over a wide range of networking contexts--from small, collaborative, ad hoc groups to a larger multi hop [50,51,69] networks.

## 2.12 Conclusion

Routing in MANETs has its share of challenges, mainly due to their inherent characteristics. In this chapter, we have critically reviewed the various categories of routing protocols in general and the flat routing protocols in particular by considering few protocols of each type. A brief description of the simulator (ns-2) is provided followed by the different Mobility Models used in MANET simulations. The Random Way Point Mobility Model is a popular mobility model used by many researchers for MANET simulation. The chapter highlights the analysis of various routing protocols based upon different simulation scenarios and a comparison between different mobility models is also carried through the simulations. The chapter discusses the DSR routing protocol and different simulation scenarios validate the correctness of the Tcl/Tk script written to implement in ns-2 (Version 2.30, Release date: September 2006) on a Linux

platform using a P-IV (1.7 GHz)/512MB/40GB computer. Finally, desirable properties of good routing protocols are listed. The rest of the thesis describes the security aspects of MANETs and the proposed cryptographic method used to secure the DSR protocol.

# Chapter 3

# Security in MANETs

## 3.1 Introduction

Security in a MANET is an essential component for basic network functions like packet forwarding and routing. The network operation can be easily jeopardized if countermeasures are not embedded into basic network functions at the early stages of their design. Unlike conventional networks [22,27,56,77,97], the ad hoc networks carry out basic support functions like - packet forwarding, routing, and network management all of the available nodes without the support of dedicated nodes and also the data travels through the open medium.

As opposed to dedicated nodes of a wired network, the nodes of an ad hoc network cannot be trusted for the correct execution of critical network functions. Further, when tamper-proof hardware and strong authentication infrastructure(s) [78] are not available, for example, in an open environment where a common authority that regulates the network does not exist, any node of an ad hoc network can endanger the reliability of basic functions like routing. The correct operation of the network requires not only the correct execution of critical network functions by each participating node but it also requires that each node perform a fair share of the functions. The latter requirement seems to be a strong limitation for wireless mobile nodes, which have to save power for their operation, so that they can 'live' on the network for a longer time period.

Due to the lack of a priori trust, classical network security mechanisms based on authentication and access control cannot cope with selfishness and cooperative security schemes seem to offer the only reasonable solution. In a cooperative security scheme, node misbehavior can be detected through the collaboration

between the numbers of nodes, assuming that a majority of nodes do not misbehave.

## 3.2 Security Goals

Security services include the functionality required to provide a secure networking environment. The main security services can be summarized as follows:

- Authentication: This service [8,9] verifies a user's identity and assures the recipient that the message is from the source that it claims to be from. Firstly, at the time of communication initiation, the service assures that the two parties are authentic, that each is the entity it claims to be. Secondly, it must assure that a third party does not interfere by impersonating one of the two legitimate parties for the purpose of authorized transmission and reception. Authentication can be provided using encryption along with cryptographic hash functions, digital signatures [134] and certificates. Details of the construction and operation of digital signatures can be found in RFC2560 [102].
- Confidentiality: This service ensures that the data/information transmitted over the network is not disclosed to unauthorized users. Confidentiality can be achieved by using different encryption techniques such as only legitimate users can analyze and understand the transmission. This is comprehensively covered in Chapter 4.
- Integrity: The function of integrity control is to assure that the data is received in verbatim as sent by authorized party. The data received contains no modification, insertion or deletion.
- Access Control: This service limits and controls the access of such a resource, which can be a host system or an application.
- Availability: This involves making the network services or resources available to the legitimate users. It ensures the survivability of the network despite malicious incidences.

## 3.3 Vulnerability of Existing Protocols

Malicious and selfish nodes are the ones that fabricate attacks [70,141] against physical, link, network, and application-layer functionality. Current routing protocols are exposed to two types of attacks:

### 3.3.1 Active Attacks

Active attacks are the attacks that are performed by the malicious nodes that bear some energy cost in order to perform the threat. These attacks can be classified into further following types.

- Impersonation: Since current ad hoc routing protocols do not *authenticate* routing packets a malicious node can launch many attacks in a network by masquerading as another node ( known as *spoofing*). Spoofing occurs when a malicious node misrepresents its identity in order to alter the vision of the network topology that a benign node can gather.

- Modification: Existing routing protocols assume that nodes do not alter the protocol fields of messages passed among nodes. Malicious nodes can easily cause traffic subversion and denial of service by simply altering the fields of the packet: such attacks compromise [38] the *integrity* of routing computations.

- Fabrication: The notation “fabrication” is used when referring to attacks performed by generating false routing messages. Such kind of attacks can be difficult to identify as they come as valid routing constructs, especially in the case of fabricated routing error messages, which claim that a neighbor can no longer be contacted.

- Wormhole Attack: A more subtle type of active attack is the creation of a tunnel [65] (or wormhole) in the network between two colluding malicious nodes linked through a private network connection. This exploit allows a node to short-circuit the normal flow of routing messages creating a virtual vertex cut in the network that is controlled by the two colluding attackers.

- Denial of Service: This active attack [48] aims at obstructing or limiting access to a certain resource. The resource can be a specific node or service or the whole network. The nature of ad-hoc networks, where several routes exist between nodes and routes are very dynamic gives ad hoc a built-in resistance to Denial of Service attacks, compared to fixed networks.

### 3.3.2 Passive Attacks

Passive attacks are mainly due to lack of cooperation with the purpose of saving energy. A selfish node that wants to save battery life for its own communication can endanger the correct network operation by simply not participating to the routing protocol or by not executing the packet forwarding (this attack is also known as the *black hole attack*). Current ad hoc routing protocols do not address the selfishness problem [91].

### 3.3.3 MANET Attack Tree

Having discussed the security issues in MANETs, the attacks [26] can be classified as shown in Figure 3.1.

Active attacks

- Incorrect forwarding
  - No forwarding
    - Data packets
    - Routing packets
      - Error packets
      - Route request packets
      - Route reply packets
  - Too slow
  - Replay
  - Changing of packet (before forwarding)
    - Route change
      - Silent route change
      - Route salvaging although no error has been observed (DSR specific)
    - Data manipulation
  - Forwarding messages to partners for analysis
- Denial of service
  - Bogus routing information
    - Replay of old routing information
    - 'Black hole routes'
  - Distorting routing information
  - Cause overload
    - Sending route updates at short intervals
    - Sending route requests at short intervals
- Lack of error messages, although an error has been observed
- Gathering information
  - Unusual traffic attraction
    - Advertising many very good routes
    - Choose a very short reply time, so the route will be prioritized

Passive attacks (eavesdropping)

- Gathering information
  - Use promiscuous mode to listen to traffic destined for other nodes

**Figure 3.1.** MANETs Attack Tree

## 3.4 Security Mechanisms

Security Mechanisms for wireless ad-hoc networks should aim to provide all the security services listed above and prevent any of the attacks mentioned. Further, due to the lack of the infrastructure in the ad-hoc networks, the typical wired-network implementation of the methods is not possible. Along with the general issues listed above, there are also other specific key issues and challenges for providing security in ad-hoc networks [87,91].

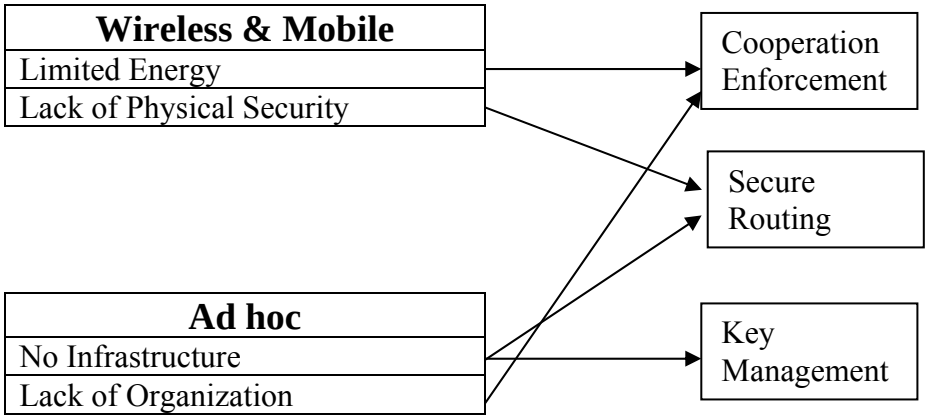



**Figure 3.2.** Security requirements for ad hoc network layer and corresponding solutions.

### 3.4.1 Link Level Security

Link Level security specifies securing the data transfer through the link between the two nodes. In wireless environment the links are susceptible to attacks where eavesdropper can intercept data packets. Physical barriers such as walls and room doors provide no barrier to wireless radio packets. The security measures that are required to implement the link level security specifies that there has to the trust relation between the two nodes [21,30].

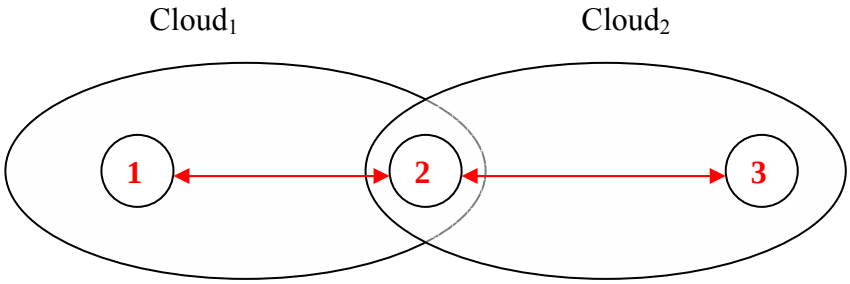



**Figure 3.3.** Link Level Security Clouds

The main characteristic feature of the link level security clouds are given below:

- They are always implemented link wise. As shown in Figure 3.3, it is very well shown that the $Cloud_1$ is the link level security cloud responsible for the secure data transmission between node1 and node2 and the $Cloud_2$ is the link level security cloud responsible for the secure data transmission between the node2 and node3.
- The other characteristic feature of the link level security clouds is that they are always implemented independently, that is, if one cloud fails then it may or may not affect the working of the other cloud.
- Although the security clouds are always implemented independently but indirectly they may affect each other. If the node common between the two nodes is compromised then the both the clouds are affected. The node may have been compromised due to the weak keys or other flaw that may have crept in. Then in that scenario the failure of the one cloud will affect the working of another cloud indirectly.

The security implemented at the link level is through the security clouds or any other terminology but it should be specific enough that nobody is able to *see* the data being transferred between the node1 and node2. For implementing such a scenario there has to be a security association between 1 and 2. The general security association that can be implemented through the keys is encryption and decryption. This process can be a part of the symmetric key cryptography or it may be part of the asymmetric key cryptography. So, for implementing link level security measures the trust infrastructure should be in place first.

Another way to implement the link level security is the use of the certification authority [82]. This approach carries additional overhead and the other disadvantage is the compromising of the certification authority, as a result the whole implementation fails.

### 3.4.2 Routing/Network Level Security

The routing within ad hoc networks is more vulnerable to attack as each device itself acts as a router. An attacker can pose as a member node and incorrectly route packets to achieve the attack, such as Denial of Service (DoS) attacks. Thus, implementation of secure routing protocol is one of the challenges within ad hoc network. The routing/network level security is responsible for the secure data transmission from the source to the destination, thus the data remains secure through the whole path of transmission.

The main difference between the Link level security measures and the network/routing level security measures is that the later has to check that the data being modified by the various nodes during the network operation is modified correctly and that each and every node is participating in the network operation for the secure data transfer between the source and the destination.

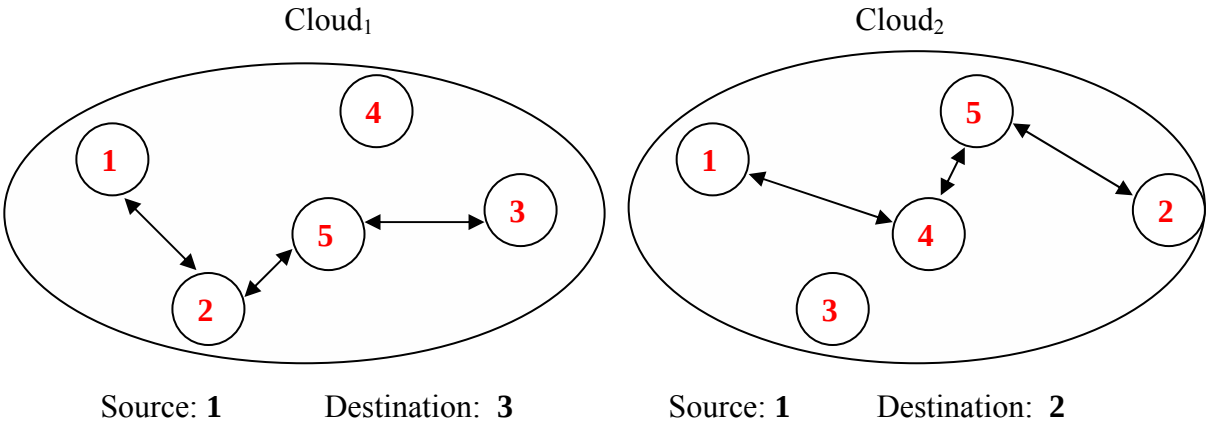



**Figure 3.4.** Routing/Network Level Security Clouds

The main characteristics of routing/network level security cloud are:

- The routing/network level security cloud is unique for the whole network.
- The routing/network level security cloud supervises the link level security clouds as well as the data manipulation being done at every node.
- The main overhead of the routing/network level security cloud is to detect the malicious nodes that may modify or inject the new packets leading to the consumption of the resources (such as – power) of nodes.
- The other challenge for the routing/network level security cloud is to detect the selfish nodes to ensure that all the nodes are working in the proper way. The selfish node wants to save battery life for its own communication and can endanger the correct network operation by simply not participating in the routing protocol or by not forwarding packets.
- Misbehavior/intrusion detection [144] and response is the main responsibility of the cloud that has to go hand in hand for the proper implementation of the routing/network level security.

The different approaches being used for the implementation of the routing/network level security are – IPSec, Self-Issued certificates.

### 3.4.3 Key Management

Key management [24,25,111] is also another aspect for security of MANETs. The purpose of key management is to provide secure procedures for handling cryptographic keying material. The task of key management includes key generation, key distribution and key maintenance. In MANETs, the computational load and complexity for key management are strongly subject to restriction by node's available resources and the dynamic nature of network topology. A number of key management schemes have been proposed for MANETs. They can be categorized as Symmetric (such as- one-way function trees (OFT) and Logical Key Hierarchy (LKH) and Asymmetric (such as – Ubiquitous and Robust Access Control (URSA) and Mobile Certification Authority (MOCA).

General network security implementation of keys involves a trusted authority. Given the lack of infrastructure for ad hoc networks, it is generally not possible to

have a fixed trusted authority. Different trust models have been proposed, such as – Centralised trust model, Web-of-trust model, Decentralized trust model and Hybrid Trust Model.

## 3.5 Secure Routing Protocols

Security has become a primary concern in MANETs. The characteristics of MANETs pose both challenges and opportunities in achieving security goals. Various proposed routing protocols are described along with the features in terms of strengths and weaknesses they posses.

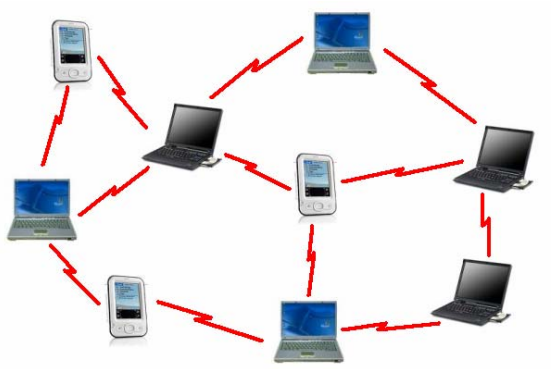


**Figure 3.5.** Routing Paths in MANET

### 3.5.1 ARAN

The ARAN [20] secure routing protocol is an on-demand routing protocol that detects and protects against malicious actions carried out by third parties and peers in the ad hoc environment. ARAN introduces *authentication*, message *integrity* and *non-repudiation* as part of minimal security policy for the ad hoc environment and consists of a preliminary certification process, a mandatory end-to-end authentication stage and an optional second stage that provides secure shortest paths.

#### I. Brief Operation

ARAN requires the use of a trusted certificate server (T): before entering in the ad hoc network, each node has to request a certificate signed by T. The certificate contains the IP address of the node, its public key, a timestamp of when the certificate was created and a time at which the certificate expires along with the

signature by T. All nodes are supposed to maintain fresh certificates with the trusted server and must know T's public key.

The goal of the first stage of the ARAN protocol is for the source to verify that the intended destination was reached. As with any secure system based on cryptographic certificates, the key revocation issue has to be addressed in order to make sure that expired or revoked certificates do not allow the holder to access the network. In ARAN, when a certificate needs to be revoked, the trusted certificate server T sends a broadcast message to the ad hoc group that announces the revocation. Any node receiving this message re-broadcasts it to its neighbors. Revocation notices need to be stored until the revoked certificate would have expired normally. Any neighbor of the node with the revoked certificate needs to reform routing as necessary to avoid transmission through the now un-trusted node. This method is not failsafe.

In some cases, the un-trusted node that is having its certificate revoked may be the sole connection between two parts of the ad hoc network. In this case, the non-trusted node might not forward the notice of revocation for its certificate, resulting in a partition of the network, as nodes that have received the revocation notice will no longer forward messages through the un-trusted node, while all other nodes depend on it to reach the rest of the network. This only lasts as long as the un-trusted node's certificate would have otherwise been valid, or until the un-trusted node is no longer the sole connection between the two partitions. At the time that the revoked certificate should have expired, the un-trusted node is unable to renew the certificate, and routing across that node ceases. Additionally, to detect this situation and to hasten the propagation of revocation notices, when a node meets a new neighbor, it can exchange a summary of its revocation notices with that neighbor; if these summaries do not match, the actual signed notices can be forwarded and re-broadcasted to restart propagation of the notice.
Various Terminologies used are also depicted in Figure 3.6.

| | | | |
|---|---|---|---|
| $K_{A+}$ | *Public-key of node A.* | $N_A$ | *Nonce issued by node A.* |
| $K_{A-}$ | *Private-key of node A.* | $IP_A$ | *IP address of node A.* |
| $\{d\}K_{A+}$ | *Encryption of data d with key KA+.* | **RDP** | *Route Discovery Packet identifier.* |
| $\{d\}K_{A-}$ | *Data d digitally signed by node A.* | **REP** | *REPly packet identifier.* |
| $cert_A$ | *Certificate belonging to node A.* | **RREQ** | *Route REQuest* |
| **T** | *Timestamp.* | **RREP** | *Route REPly* |
| **E** | *Certificate expiration time.* | **ERR** | *ERRor packet identifier.* |
| **brdcast** | *Broadcast* | | |

$A \rightarrow brdcast: [RDP, IP_x, cert_A, N_A, t] K_{A-}$

$B \rightarrow brdcast: [[RDP, IP_x, cert_A, N_A, t] K_{A-}] K_{B-}, cert_B$

$C \rightarrow brdcast: [[RDP, IP_x, cert_A, N_A, t] K_{A-}] K_{c-}, cert_c$

$X \rightarrow D: [REP, IP_a, cert_x, N_A, t] K_{x-}$

$D \rightarrow C: [[REP, IP_a, cert_x, N_A, t] K_{x-}] K_{D-}, cert_D$

$C \rightarrow B: [[REP, IP_a, cert_x, N_A, t] K_{x-}] K_{c-}, cert_c$

**Figure 3.6.** Route Discovery in ARAN, along with terminologies used

**II. Features**

i. The ARAN protocol protects against exploits using *modification*, *fabrication* and *impersonation*

ii. The ARAN protocol uses of asymmetric cryptography makes it a very costly protocol to use in terms of CPU and energy usage.

iii. ARAN is not immune to the *wormhole* attack

### 3.5.2 ARIADNE

ARIADNE [138] is an on-demand secure ad hoc routing protocol based on DSR that withstands node compromise and relies only on highly efficient *symmetric* cryptography. ARIADNE guarantees that the target node of a route discovery process can authenticate the initiator, that the initiator can authenticate each

intermediate node on the path to the destination present in the RREP message and that no intermediate node can remove a previous node in the node list in the RREQ or RREP messages.

**I. Brief Operation**

As for the SRP [16] protocol, ARIADNE needs some mechanism to bootstrap authentic keys required by the protocol. In particular, each node needs a shared secret key ($K_{S, D}$) is the shared key between a source S and a destination D) with each node it communicates with at a higher layer, an authentic TESLA [11] key for each node in the network and an authentic "Route Discovery chain" element for each node for which this node will forward RREQ messages. Various terminologies used are depicted in Figure. 3.7.

**II. Features**

i. ARIADNE provides point-to-point *authentication* of a routing message using a message authentication code (MAC) [10] and a shared key between the two parties.

ii. For authentication of a broadcast packet such as RREQ, ARIADNE uses the TESLA broadcast authentication protocol.

iii. Selfish nodes are not taken into account.

iv. ARIADNE copes with attacks performed by *malicious* nodes that modify and fabricate routing information, with attacks using impersonation and, in an advanced version, with the wormhole attack

v. ARIADNE is protected also from a flood of RREQ packets that could lead to the cache poisoning attack [135].

vi. ARIADNE is immune to the wormhole attack [139] only in its advanced version: using an extension called TIK (TESLA with Instant Key disclosure) that requires tight clock synchronization between the nodes, it is possible to detect anomalies caused by a wormhole based on timing discrepancies.

| | | | |
|---|---|---|---|
| **Brdcast** | *Broadcast* | $K_{AB}$ **and** $K_{BA}$ | *MAC keys shared between A and B* |
| $MAC_{KAB}$ **(M)** | *computation of the message authentication code (MAC) of message M with the MAC key KAB* | **REQ** | *REQuest* |
| **REP** | *REPly* | $H_n$ | *One way hash function executed n times.* |
| **id** | *Unique id* | $t_i$ | *time interval* |
| **S** | *Initiator node* | **D** | *Target node* |
| | | $M_A$ | *MAC address of node A* |

***S:*** $h_0=MAC_{KSD}(REQ, S, D, id, t_i)$

**S -> * : < REQ, S, D,id,ti,h0,(),()>**

**A:** $h_1=H[A,h_0]$

$M_A=MAC_{KAti}(REQ, S, D, id, t_i, h_1,(A),())$

**A -> * : <REQ, S, D,id,ti,**$\underline{h_1}$**,(**$\underline{A}$**),(**$\underline{M_A}$**)>**

**B:** $h_2=H[B,h_1]$

$M_B=MAC_{KBti}(REQ, S, D, id, t_i,h_2,(A,B),(M_A))$

***B -> * : <REQ, S, D,id,ti,***$\underline{h_2}$***,(A,***$\underline{B}$***),(M_A,***$\underline{M_B}$***)>***

**C:** $h_3=H[C,h_2]$

$M_C=MAC_{KCti}(REQ, S, D, id, t_i,h_3,(A,B,C),(M_A,M_B))$

***C -> * : <REQ, S, D,id,ti,***$\underline{h_3}$***,(A,B,***$\underline{C}$***),(M_A,M_B,***$\underline{M_c}$***)>***

***D:*** $MD=MAC_{KDS}(REP,D,S,ti,(A,B,C), (M_A,M_B,M_c))$

***D-> C:< REP,D,S,ti,(A,B,C), (M_A,M_B,M_c),***$\underline{M_D}$***,()>***

***C->B: < REP,D,S,ti,(A,B,C), (M_A,M_B,M_c),M_D,(***$\underline{K_{Cti}}$***)>***

***B->A: < REP,D,S,ti,(A,B,C), (M_A,M_B,M_c),M_D,(K_Cti,***$\underline{K_{Bti}}$***)>***

***A->S: < REP,D,S,ti,(A,B,C), (M_A,M_B,M_c),M_D,(K_Cti K_Bti,***$\underline{K_{Ati}}$***)>***

**Figure 3.7.** Route Discovery in ARIADNE, along with terminologies used

### 3.5.3 SEAD

Hu *et al*. presented a *proactive* secure routing protocol based on the Destination-Sequenced Distance Vector protocol (DSDV)[32]. In a proactive (or periodic) routing protocol nodes periodically exchange routing information with other nodes in attempt to have each node always know a current route to all destinations.

**I. Brief Operation**

SEAD [137] authenticates the sequence number and metric of a routing table update message using hash chains elements. In addition, the receiver of SEAD routing information also authenticates the sender, ensuring that the routing information originates form the correct node. The source of each routing update message in SEAD must also be authenticated, since otherwise, an attacker may be able to create routing loops through the *impersonation* attack.

**II. Features**

i. SEAD deals with attackers that *modify* routing information broadcasted during the update phase of the DSDV-SQ protocol: in particular, routing can be disrupted if the attacker modifies the sequence number and the metric field of a routing table update message.
ii. SEAD makes use of efficient *one-way hash chains* rather than relaying on expensive asymmetric cryptography operations.
iii. SEAD assumes some mechanism for a node to distribute an authentic element of the hash chain that can be used to authenticate all the other elements of the chain.
iv. SEAD does not cope with *wormhole* attacks.

### 3.5.4 SRP

The Secure Routing Protocol (SRP) [3,4,95] was designed as an extension compatible with a variety of existing *reactive* routing protocols. SRP combats attacks that disrupt the route discovery process and guarantees the acquisition of correct topological information.

**I. Brief Operation**

SRP allows the initiator of a route discovery to detect and discard bogus replies. SRP relies on the availability of a *security association* (SA) between the source node (S) and the destination node (T). The SA could be established using a hybrid key distribution based on the public keys of the communicating parties. S and T can exchange a secret symmetric key ($K_{S,T}$) using the public keys of one another to establish a secure channel. S and T can then further proceed to mutual authentication of one another and the authentication of routing messages.

**II. Features**

i. SRP copes with non-colluding *malicious* nodes that are able to modify (corrupt), replay and fabricate routing packets.

ii. Assuming that the neighbor discovery mechanism maintains information on the binding of the medium access control and the IP addresses of nodes, SRP is proven to be essentially immune to IP spoofing.

iii. In case of the Dynamic Source Routing (DSR) protocol [9], SRP requires including a 6-word header containing unique identifiers that tag the discovery process and a message authentication code (MAC) computed using a keyed hash algorithm.

iv. The basic version of SRP suffers from the route cache poisoning attack.

v. SRP suffers from the lack of a validation mechanism for route maintenance messages

vi. SRP is not immune to the wormhole attack: two colluding malicious nodes can misroute the routing packets on a private network connection and alter the perception of the network topology by legitimate nodes.

### 3.5.5 SAODV

The Secure Ad hoc On Demand distance Vector (SAODV) [81,143] protocol is an extension of the AODV protocol. The Secure AODV scheme is based on the assumption that each node possesses certified public keys of all network nodes.

**I. Brief Operation**

The originator of the routing control packet appends its RSA signature and the last element of a hash chain to the routing packets. A packet transverse the network, intermediate nodes cryptographically authenticate the signature and the hash value. The intermediate nodes generate the $k^{th}$ element of the hash chain, with k being the number of transverse hops, and place it in packet.

The SAODV protocol gives two alternatives for ROUTE REQUEST and ROUTE REPLY messages. In the first case when a ROUTE REQUEST is sent, the sender creates a signature and appends it to packet. Intermediate nodes authenticate the signature before creating or updating the reverse route to the host. The reverse rout is stored only when the signature is verified. When the node reaches the destination, the node signs the ROUTE REPLY with its private key and send it back. The intermediate nodes again verify the signature .The signature of the sender is again stored with the along with the route entry.

| | | | |
|---|---|---|---|
| **RREQ** | *Route Request* | ***brdcast*** | *Broadcast* |
| $Seq_s$ | *Sequence No issued by s* | $K_{A-}$ | *Private-key of node A* |
| $\{d\}K_{A-}$ | *Data d digitally signed by node A* | $Seq_A$ | *Sequence number issued by A.* |
| ***S*** | *Initiator node* | ***D*** | *Target node* |
| $h_n$ | *Hash chain function operated n times.* | ***brdcast*** | *Broadcast* |
| ***RREP*** | *Route Reply* | $oldseq_A$ | *Old sequence number issued by A* |
| ***Id*** | *Unique Id for the packet* | ***N*** | *Nonce* |

$S \rightarrow brdcast: \quad ((RREQ,id, S, seq_s, D, oldseq_D, h_0, N)_{KS-}, 0, h_N)$

$A \rightarrow brdcast: \quad ((RREQ,id, S, seq_s, D, oldseq_D, h_0, N)_{KS-}, 1, h_{N-1})$

$B \rightarrow brdcast: \quad ((RREQ,id, S, seq_s, D, oldseq_D, h_0, N)_{KS-}, 2, h_{N-2})$

$C \rightarrow brdcast: \quad ((RREQ,id, S, seq_s, D, oldseq_D, h_0, N)_{KS-}, 3, h_{N-3})$

$D \rightarrow C: \quad ((RREP, D, seq_D, S, lifetime, h'_0, N)_{KD-}, 0, h'_N)$

$C \rightarrow B: \quad ((RREP, D, seq_D, S, lifetime, h'_0, N)_{KD-}, 1, h'_{N-1})$

$B \rightarrow A: \quad ((RREP, D, seq_D, S, lifetime, h'_0, N)_{KD-}, 2, h'_{N-2})$

$A \rightarrow S: \quad ((RREP, D, seq_D, S, lifetime, h'_0, N)_{KD-}, 3, h'_{N-3})$

**Figure 3.8.** Route Discovery in SAODV, along with terminologies used

**II. Features**

i. Ownership of certified public keys enables intermediate enable intermediate nodes to authenticate all in-transit routing packets.

ii. The protocol operates mainly by using the new extension message with the AODV protocol.

iii. The SAODV can be used to protect the route discovery mechanism of the AODV by providing security features like integrity, authentication and non-repudiation.

### 3.5.6 SAR

Security-Aware Ad-Hoc Routing (SAR) [33,117] is the generalized framework for any on-demand ad-hoc routing protocol. SAR requires that nodes having same

trust level must share a secret key. SAR augments the routing process using hash digests and symmetric encryption mechanisms. The signed hash digests provide message integrity while the encryption of packets ensures their confidentiality.

**I. Brief Operation**

SAR when implemented on AODV protocol adds two additional fields to the ROUTE REQUEST packet and one additional to the ROUTE REPLY packet. The first field added to the ROUTE REQUESTPACKET is the security requirement field and is set by the sender .It indicates the preferred level of trust for the path to the destination. The Second field added to is the security guarantee that signifies the maximum level of security provided by the discovered paths. If the security requirement field has an integer representation then the security guarantee field will be minimum of all security levels of the participating nodes in the path. If the security requirement field is represented in vectors then the security guarantee field value id computed by ANDing the security requirement values of the participating nodes in the path. The value thus computed is copied into additional security guarantee field of the ROUTE REPLY packet and sent back to the sender. This value is also copied into the routing table of nodes in the reverse path, to preserve the security information with reference to cashed paths

**II. Features**

i. SAR uses security information to dynamically control the choice of routes installed in the routing table.
ii. SAR enables applications to selectively implement a subset of security services based on the cost-benefit analysis.
iii. The routes discovered by SAR may not always be the shortest between any two communicating entities in terms of hop-count. However these routes have quantifiable guarantee of the security.
iv. SAR will find the optimal route if all the nodes on the shortest path satisfy the security requirements.
v. SAR may fail to find the route if the ad hoc network does not have a path on which all nodes on the path satisfy the security requirements in spite of being connected.

### 3.5.7 SLSP

Secure Link State Routing Protocol (SLSP) [4,16,94] provides a secure proactive topology discovery [104] and can be used as either as a stand-alone protocol or as a part of hybrid routing framework when combined with a reactive protocol.

**I. Brief Operation**

To function effectively without central key management authority, SLSP enables each node to periodically broadcast its public key to nodes within its zone. In addition each node also broadcasts signed HELLO messages containing its medium access control address and IP address pair to its neighbors. The distribution of medium access control address strengthens the scheme by forbidding nodes from spoofing at the data link layer [71,122].

To achieve theses goals a Neighbor Lookup Protocol (NLP) is made an integral part of SLSP. The NLP is responsible for the tasks like - maintaining a mapping of MAC and IP layer addresses of the node's neighbors and identifying potential discrepancies such as - the use of multiple IP addresses by a single data-link interface.

The rate of incoming control packets helps in discarding nodes which maliciously seek to exhaust network resources.

**II. Features**

i. SLSP can operate in the networks of recurrently changing topology and memberships.
ii. SLSP is resilient against individual attackers and is capable of altering its range between local and network wide topology discovery.
iii. SLSP employs a round robin servicing mechanism to provide the assurance the benign control traffic will be relayed even under clogging DoS attacks.

### 3.5.8 TESLA

Timed Efficient Stream Loss-tolerant Authentication (TESLA) [11] is an efficient broadcast authentication protocol with low communication and computation

overhead. It can scale to large numbers of receivers, can tolerate packet loss, and uses loose time synchronization between sender and receivers.

**I. Brief Operation**

For secure broadcasting, a sender chooses a random initial key KN and generates a one-way key chain by repeatedly computing the one-way hash function H on the starting value $K_{N-1} = H[K_N]$ , $K_{N-2} = H[K_{N-1}]$, . . ., $K_0 = H[K_1]$. In general, $K_i = H[K_{i+1}] = H_{N-i}[K_N]$ where $H_i[x]$ is the result of applying the function H to x, for i times. The sender node predetermines a schedule at which it discloses each key of its one-way key chain. Keys are disclosed in the reverse order from generation, i.e. $K_0$, $K_1$, $K_2$, . . . ,$K_N$ then the MAC computed using the key $K_i$ is added to the packet. When the packet reaches the receiver, it checks the security condition of the key disclosure. If the key $K_i$ used to authenticate the packet was not disclosed, then it buffers the packet and waits for the sender to disclose $K_i$, while using an already disclosed key to authenticate the buffered packets. However, if the key is already disclosed, then receiver will discard the packet.

**II. Features**

i. TESLA mainly uses purely symmetric cryptographic functions, however, it achieves asymmetric properties from clock synchronization and delayed key disclosure.
ii. For secure authentication, either the receiver or the sender must buffer some messages.
iii. TESLA authenticates the initial packet with a digital signature, which is too expensive for wireless nodes, and disclosing a key in each packet requires too much energy for sending and receiving.
iv. TESLA is vulnerable to DoS attacks as malicious nodes can create buffer overflow state in the receiver while it waits for the sender to disclose its keys.

## 3.6 Cooperation Enforcement in Mobile Ad hoc networks

For the avoidance of the selfish nodes' effects on the cooperation functions of a MANET and the consolidation of the network robustness, a class of methods, referred to as cooperation enforcement methods, is considered more appropriate. These, recently introduced, distributed, and lightweight methods contribute to the trust establishment between MANET nodes without prior knowledge of the nodes' behavior. They apply to the network layer of a MANET, and their primary goal is to protect or enforce the two elementary functions of this layer: *routing* and *packet forwarding*.

### 3.6.1 CONFIDANT

Buchegger and Le Boudec [109] proposed a scheme, called CONFIDANT, designed as an extension to an on-demand routing protocol, such as the DSR. CONFIDANT facilitates monitoring and reporting for a route establishment that avoids the misbehaving nodes. It is based on the assumption that the packets of misbehaving nodes are not forwarded by fair nodes. If, however, a node was incorrectly accused or turns out to be a repentant and no longer malicious, re-integration into the network is possible.

**I. Architecture-mechanism**

CONFIDANT employs four functional components relying on each node, which include: a monitor, reputation records for first-hand and trusted second-hand observations about routing and forwarding function of other nodes, trust records to control the trust that is given to received warnings and a path manager to take routing decisions that avoid malicious nodes. The term reputation is used to evaluate routing and forwarding behavior according to the network protocol, whereas the term trust is used to evaluate participation in the protocol. Nodes monitor their neighbors and change reputations accordingly. Specifically, a node can detect selfish behavior of the next node in the source route either directly, by promiscuously sensing the transmission of the next node, or indirect, by observing routing protocol misbehavior. The *Monitor* component registers these deviations. As soon as a specific misbehavior occurs, the *Reputation System* is called and the

*Trust Manager* sends ALARM messages. Outgoing ALARMS are generated by the node itself after having experienced, observed, or received a report on malicious behavior of another node. They convey warnings of malicious nodes presence. The recipients of the ALARM messages, so called friends, are maintained in a friends list. Incoming ALARMs that originate from *"strangers"* are checked for trustworthiness before triggering a reaction. The disadvantage here is the requirement of a pre-existed trust relationship. If there is sufficient evidence that the node reported in the ALARM is malicious, the information is sent to the *Reputation System*. This manages a table consisting of entries corresponding to nodes and their ratings. A rating is modified if two conditions coincide:

- There is sufficient evidence of malicious behavior, and
- A misbehavior occurs a number of times, exceeding a threshold to rule out coincidences.

The ranking of a node is changed according to a rate function. This function features the greatest weight for own experience, a smaller weight for observations captured in the neighborhood and the smallest weight to the reported, second-hand, experiences. If the ranking of a node has deteriorated so much as to fall out of a tolerable range, the *Path Manager* is activated. This component excludes routes containing misbehaving nodes and isolates them, ranks the paths in a cache, and forwards an ALARM about this node.

**II. Features**

The first version of CONFIDANT was vulnerable to rumor spreading phenomena. In a recent enhancement, this problem has been addressed through a Bayesian model that classifies and excludes the liars. In this enhanced version, both positive and negative reputations are used to calculate a *"cooperation factor"*. This factor consists of the frequency of misbehavior in relation to the cumulative activity of the node. The positive and negative experiences collected by a node should reveal the same sort of information for a node as what is gathered by the other nodes. Every node *i* keep a cooperation factor of every other node *j*, expressed as *Rij*. This factor is expressed as a function of $\alpha$ and $\beta$, whereas, $\alpha$ and $\beta$ is the number of misbehaviors and regular behaviors, respectively. These numbers are updated

based on recent experiences. A recommendation is accepted if it is compatible (in the Bayesian model), that is, if the recommended *RV* is not completely different. This technique reduces the impact of the false accusations. For a misbehaving node, it is hard to know the entries of its reputation in other nodes or to modify its reputations. However, it is still possible to alter the values of $\alpha$ and $\beta$ or to change its identity. Only identities generated with cryptographic means can reduce this threat. The Bayesian approach reduces the impact of tampering with $\alpha$ and $\beta$. If values are not compatible with each other the algorithm will just ignore them. Evil nodes could only change the values with a small amount that is tolerable by the system.

### 3.6.2 CORE

This scheme, introduced by Michiardi and Molva in [92], relies on the DSR routing protocol. It stimulates node collaboration through monitoring of the cooperativeness of nodes and a reputation mechanism.

#### I. Architecture-mechanism

It uses first and second-hand experiences, combined by a specialized function, and is used by the *Watchdog* mechanism [109] for the evaluation of other nodes' behavior. If the observed behavior is different than the outcome of this function then the rating of the observed node is altered. Each node of the network monitors the behavior of its neighbors, with respect to the requested function, and collects observations about the execution of that function. These observations are recorded to the *Reputation Table* (RT), maintained by each node. Each row of the table corresponds to a neighbor node and consists of four entries, regarding the monitored function: the unique id of the node, a collection of recent (first-hand) observations made on the node's behavior, a list of the recent second-hand *RVs* provided by other nodes, and the *RV* evaluated for the monitored function. Thus, each node maintains one RT for each monitored function. Finally, a global RT is used to combine the different *RVs* calculated for the different functions.

CORE differentiates the *RVs* between *subjective reputation* ([-1, 1]), *indirect reputation* (positive reports by others), and *functional reputation* (e.g, when packet

forwarding has greater effect than routing), which are weighted to provide a combined *RV*. The formula used to evaluate the *RV* avoids false detections by using an aging factor that gives more relevance to past observations. However, such an approach is vulnerable to an attack where a node can build up a good reputation before misbehaving. The *RVs* evaluated for each entry of the RT vary. A positive *RV* is decremented along time. So, if a node enters in an idle mode, its reputation has to be decreased, even if during the active time (i.e., when communicates) it cooperates to the network operation. Reputation is decreased until it reaches a null value, which corresponds to a neutral behavior. Furthermore, if the monitored function provides a reply message (e.g., the Route Reply of the DSR), reputation information can also be gathered about non-adjacent nodes. In this case, only positive ratings are assigned to the nodes that participate to this function.

**II. Features**

The CORE scheme is immune to attacks performed using the mechanism itself: no negative ratings are spread, and, thus, it is impossible for a node to maliciously decrease another node's reputation. However, two or more nodes may collude (i.e., send positive rating messages) in order to increase their reputation. To prevent such phenomena, the CORE implicitly provides some protection, since subjective reputation has more impact (i.e., weight) than the indirect. CORE allows MANETs nodes to gradually isolate misbehaving nodes: when the reputation assigned to a neighboring node falls below a predefined threshold, the service provided to this node is interrupted. Misbehaving nodes can, however, be reintegrated in the network if they increase on purpose their reputation, by cooperating to the network operation. CORE does not discriminate malfunction and misbehaving nodes. Additionally, a second chance mechanism is not consolidated, as in OCEAN(refer section 3.6.4), and, hence, a malfunctioning node can't rebuild its reputation when it recovers from temporal problems. Finally, the CORE mechanism assumes that every node will assign the same weights to the same functions. This might not be the case, since in MANETs the devices are equipped with different resources and provide discrete services, and, hence they prefer to use difference levels of importance on functions.

### 3.6.3 SORI

The Secure and Objective Reputation-based Incentive Scheme for Ad-hoc Networks, introduced in [121], focuses on the packet forwarding function.

#### I. Architecture-mechanism

SORI [3,4,99,121], consists of three basic components: *Neighbor Monitoring*, *Reputation Propagation* and *Punishment*. A promiscuous mode is assumed, and a node is capable of overhearing the transmissions of its neighbors and to maintain a neighbor node list. Each neighbor's forwarding function is linked with two parameters. The *RFN(X*) (Request-for-Forwarding) is used to indicate the total number of packets that node *N* has transmitted to *X* for forwarding. The index *HFN(X)* (Has-Forwarded) corresponds to the total number of packets that have been forwarded by *X* and noticed by *N*. Given *RFN(X)* and *HFN(X)*, *N* creates a record, called local evaluation record. This record, denoted by *LERN(X)*) for the neighbor *X*, contains a confidence metric that is used to depict how confident the node *N* is for its judgment on the reputation of *X*. The more the packets transmitted to *X* for forwarding, the higher the confidence about the trustworthiness of *X*.

#### II. Features

SORI combine features of the fist-hand schemes and those that use reputation spreading. In SORI the nodes exchange reputation information only with their neighbors. This way a no-cooperative node will be punished by all of its neighbors (who share the reputation information about its misbehavior), instead of just the ones who are directly affected by this node. Each node *N* periodically updates its *LERN(X)* for each neighbor node *X* based on the current values of *RFN(X)* and *HFN(X)*. The updated record is broadcast to the neighborhood if the ratio *RFN(X)*/*HFN(X)* has been significantly changed. Node *N* uses his own *LERN(X)* and the respective values of its neighbors to calculate its overall evaluation record of *X*, denoted as *OERN(X)*. To do so, it takes into account the credibility of the nodes, which contribute to the calculation of the reputation. This makes it difficult for an attacker to test multiple identities, trying to impersonate one identity that it can be used in order to improve its reputation. If the *OERN(X)* is lower than a

predefined threshold, node *N* takes a punishment action by probabilistically dropping the packets originated from *X*. This mechanism, as mentioned in, is designed to treat generously the nodes that do not intentionally drop packets. In, a complementary security mechanism is proposed to deal with a node that uses the following attacks:

- impersonation of an adjacent node's id, ranked with a good reputation, in order to send more packets, and
- impersonating a distant node's id, ranked with a good reputation, to broadcast fake observation information in order to boost its reputation. This mechanism is based on a one-way hash chain and Message Authentication Codes (MACs). Finally, SORI take no countermeasures to prevent collusion.

### 3.6.4 OCEAN

The Observation-based Cooperation Enforcement in Ad hoc Networks, proposed in [108], introduces an intermediate layer that resides between the network and the MAC layers. This layer helps the nodes to make intelligent routing and forwarding decisions. It is designed on top of the DSR, but its principles can be applied to other routing protocols, as well. OCEAN relies only on first-hand observations.

#### I. Architecture-mechanism

Every node maintains ratings for each neighboring node and monitors their behaviors through promiscuous observations. Positive or negative events are recorded through the reaction of a neighbor that is expected to forward a packet. Rating is initialized to a neutral value. Due to empirical studies, the absolute value of a decrement is chosen to be bigger than the value of an increment. When the rating of a node drops below a threshold, called *faulty threshold* the node is added to a faulty list. This list is constructed using the node's personal experiences and is attached (as a field called *avoid-list*) to the Route Request (RREQ) message of the DSR protocol in order to be flooded. A route is rated good or bad, based on whether the next hop in the route belongs to the *avoid-list*. The receiver of an RREQ decides to drop it or to further process it (through relaying or a Route Reply), if the intersection of the *avoid-list* and the DSR route in the RREQ packet

is void. In this way, each node along a route, makes its own decision about the trustworthiness of other nodes, and has control only over routes that it belongs to. Every node rejects the data packets arrived from the nodes belonging to its faulty list. Thus, misbehaving nodes are eventually isolated. However, a *Second-Chance* mechanism is used to allow nodes that misbehaved in the past to become operational again. After a certain period, a misbehaved node is excluded from the faulty list and assigned with a neutral rating. OCEAN uses a different policy to deal with nodes that don't participate in the route discovery process. This policy requires no tamper-proof hardware or a central server. Each node measures the behavior of its neighbors by directly interacting with them. Nodes track the forwarding balance with their neighbors by maintaining one counter, called *chip count,* per node. The counter increases when requesting a node to forward a packet and decreases with an incoming request from that node. Assume that a node *B* didn't participate on the establishment of route with a source node *A*. If *B* demands from *A* to forward its packets, then, *A* will punish *B* and reject its requests, as long as the chip count for *B* is low. This policy is considered unfair for nodes belonging to the perimeter of the MANET, since they are not frequently required to forward messages on behalf of others. Penalizing these nodes might cause the network to shrink. To overcome such phenomena, the OCEAN introduces a *chip accumulation rate* (CAR) parameter, which expresses the rate at which all *chip count* in the network are increased per unit time. Thus, the forwarding of the packets sent by circumferential nodes is enforced, even at a reduced rate. CAR can't be adjusted easily and there no mechanism to prevent a node to change it at will.

**II. Features**

Promiscuous mode of operation does not always provide sufficient evidence on the trustworthiness of a monitored node. A monitored node may not be able to relay the packet due to the low quality of the wireless link. Additionally, other reasons, such as network interface restart, or low battery, might affect the relay task. Thus, nodes should incorporate the logic to discriminate the cases where other nodes malfunction and misbehave, and not to faulty punish low capacity nodes. The introduced *Second-Chance* mechanism was designed to overcome the potential

problems that might be observed due to the absence of such intelligence. Hacker nodes might take advantage of the *avoidlists* of the OCEAN, which are included on the RREQs, and tamper these lists to perform wormhole attacks. Simulations showed that OCEAN performs well under the presence of such attacks as long as the network topology is not static. The *faulty threshold* reflects the speed and accuracy of misbehavior detection. A low value adds nodes quicker to a faulty list. High values might decrease the detection speed. Detection speed is important for the models that use first-hand observations, since the evaluation of a new joined node takes place from scratch. In contrast, schemes that use second-hand reputations obtain trust indexes for remote nodes that eventually will become adjacent, and thus, operate proactively. Simulations have showed that with a low *faulty threshold*, OCEAN performs better than a generic scheme that uses second-hand information. This is because OCEAN is more resilient to rumor spreading. However, OCEAN is sensitive to the tuning of the *faulty threshold* parameter; second-hand schemes perform better over a broader range of tunings. Additionally, it is not effective in reducing the throughput of misbehaving nodes. Finally, OCEAN takes no countermeasures to prevent collusion.

### 3.6.5 Nuglets

In Buttyan and Hubaux [74] present two important issues targeted specifically at the ad hoc networking environment: first, end users must be given some incentive to cooperate in the network operation (especially to relay packets belonging to other nodes); second, end users must be discouraged from overloading the network. The solution presented in their paper consists of the introduction of a virtual currency (known as *Nuglets*) used in every transaction. Two different models are described: the Packet Purse Model and the Packet Trade Model. In the Packet Purse Model, each packet is loaded with nuglets by the source and each forwarding host takes out nuglets for its forwarding service. The advantage of this approach is that it discourages users from flooding the network, but the drawback is that the source needs to know exactly how many nuglets it has to include in the packet it sends. In the Packet Trade Model, each packet is traded for nuglets by the intermediate nodes: each intermediate node buys the packet from the previous node on the path. Thus, the destination has to pay for the packet. The direct advantage of

this approach is that the source does not need to know how many nuglets need to be loaded into the packet. On the other hand, since the packet generation is not charged for, malicious flooding of the network cannot be prevented. There are some further issues that have to be solved: concerning the Packet Purse Model, the intermediate nodes are able to take out more nuglets than they are supposed to; concerning the Packet Trade Model, the intermediate nodes are able to deny the forwarding service after taking out nuglets from a packet.

Table 3.1 and Table 3.2 present a comparison and summary of different routing security [84] mechanism based upon different parameters.

**Table 3.1.** Operational requirements of the surveyed secure ad hoc routing solutions.

| **Protocol** | **Requirements** | **Security Mechanism** | **Attacks Prevented** | **Comments** |
|---|---|---|---|---|
| SEAD | Clock synchronization, or a shared secret between each pair of nodes | One-way hash chains | Prevents an attacker from forging better metrics or sequence numbers in routing update packets | Used with DSDV<br>- Designed to protect routing update packets<br>- Does not prevent an attacker from tampering other fields or from using the learned metric and sequence number for sending new routing updates |
| ARIADNE | Clock synchronization and the existence of a shared secret between each pair of nodes. Also, an authentic TESLA key for each node in the network and an authentic route discovery chain element for each node for which this node will forward route requests. TESLA keys are distributed to the participating nodes via an online key distribution center | One-way hash chains | Prevents attackers from tampering un compromised routes consisting of uncompromised nodes<br>- Immune to wormhole attack | Used with DSR<br>- Provides a strong defense against attacks that modify and fabricate routing information<br>- Prone to selfish node attack |

| Protocol | Requirements | Security Mechanism | Attacks Prevented | Comments |
|---|---|---|---|---|
| SAR | Key distribution or secret sharing mechanism | Quality of Protection (QoP) metric | Uses sequence numbers and timestamps to stop replay attacks in routing update packets | Used with AODV<br>- Route discovered may not be the shortest route in terms of hopcount, but it is always secured<br>- Defends against modification and fabrication attacks |
| SRP | Existence of a security association between each source and destination node. Malicious nodes do not collude within one step of the protocol process. | Secure certificate server | Defends against attacks that disrupt the route discovery process and guarantees to identify the correct topological information | Used with DSR, ZRP<br>- Lack of validation mechanism for route maintenance messages<br>- Prone to wormhole attacks and invisible node attacks |
| ARAN | Online trusted certification authority. Each node knows *a priori* the public key of the CA | Secure certificate server | Provides network services like authentication and non-repudiation | Used with AODV, DSR<br>- Heavy asymmetric cryptographic computation<br>- Prone to wormhole attack if accurate time synchronozation is not available |
| CONFIDANT | Nodes cannot change their identifier to get rid of their reputation rating. Pre-defined lists of friendly nodes. | Monitor<br>- Reputation System<br>- Path Manager<br>- Trust Manager | Attacks on packet forwarding and routing are defended efficiently | Used with DSR<br>- Detection based reputation system has few limitations<br>- Vulnerable to spoofing and sybil attacks |
| TESLA | Distribution of initial hash chain element | One-way Hash Chain | Uses loose time and delayed time synchronization | Vulnerable to DoS attacks as malicious nodes can create buffer overflow<br>State and accurate time synchronization is not easy to obtain |

**Table 3.2.** Ad hoc Routing Parameters.

| **Proposed Solutions** | **Routing Approaches** | **Loop Freedom** | **Routing Metric** | **Shortest Path Identification** | **Intermediate Nodes Allowed to Reply to Route Requests** |
|---|---|---|---|---|---|
| ARAN | On-demand | Yes | None | Optional | No |
| SAR | On-demand | Depends on the selected security requirement | A security requirement | No | No |
| SRP | On-demand | Yes | Distance | No | Optional |
| SEAD | Table-Driven | Yes | Distance | No | No |
| ARIADNE | On-demand | Yes | Distance | No | No |
| CONFIDANT | On-demand | Yes | Path Reliability | Depends | No |

## 3.7 Conclusion

In this chapter, we have presented some of the proposed secure routing protocols for mobile ad hoc networks. The comparison between the surveyed secure routing protocols indicate that the design of a secure ad hoc routing protocol constitutes a challenging research problem since already existing generic solutions, cannot be successfully applied. An additional difficulty in designing a secure protocol lies in the application scenario for which the secure protocol is designed and how well it can handle the different scenarios than the one it has been explicitly designed for [89]. A flexible secure ad hoc routing solution should take into account the performance-security trade-off associated with an application and dynamically achieve the required equilibrium. We believe that security should be an important and integral part of any ad hoc and wireless networking routing solution. In order to guarantee successful employment a solution should have realistic operational requirements based on the application domain.

# Chapter 4

# DNA Cryptography

## 4.1 Introduction

The word “cryptography” [7,105,136] is derived from Greek and when literally translated, means “secret writing.” Before the advent of digital communications, cryptography was used primarily by the military for the purposes of espionage. With the advances in modern communication, technology has enabled businesses and individuals to transport information at a very low cost via public networks such as the Internet. This development comes at the cost of potentially exposing the data transmitted over such a medium. Therefore, it becomes imperative for businesses to make sure that sensitive data is transferred from one point to another in an airtight, secure manner over public networks. Cryptography can help us achieve this goal by making messages unintelligible to all but the intended recipient.

Traditionally, cryptography was a government monopoly with very little cross-fertilization between governments [105]. The Cryptographic algorithms were designed and evaluated by government experts and details were kept secret. Governments trusted their procedures and hence trusted the cryptography. Although the cryptographic details were kept secret, this secrecy was not relied on for the security of the communications. The change over recent decades is that cryptography has become a necessary tool for a wide commercial market.

### 4.1.1 Encryption and Decryption

Encryption refers to the transformation of data in "plaintext" form into a form called "ciphertext," which renders it almost impossible to read without the knowledge of a "key," which can be used to reverse this transformation [45]. The recovery of plaintext from the ciphertext requires the key, and this recovery process is known as decryption. This key is meant to be secret information and the privacy of the ciphertext depends on the cryptographic strength of the key.

The process of turning back the *Cipher Text* into *Plain Text* is known as *Decryption*. (Based on ISO 7498-2 standards, we can refer them as *Encipher & Decipher*). This concept is depicted in the Figure 4.1. Further, in case of MANETs the most simple and common method for encryption is monoalphabetic substitutions and that can be easily broken by use of Genetic algorithm and Tabu Search algorithms [2].

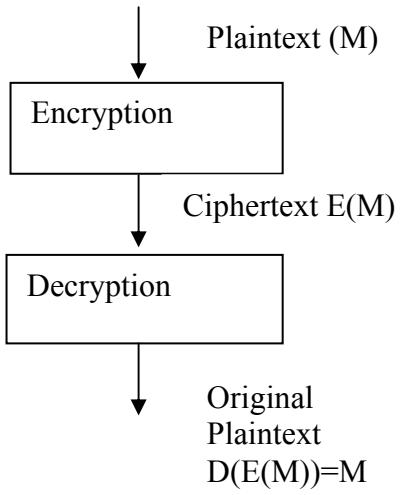



**Figure 4.1.** Encryption and Decryption

## 4.2 Types of Cryptography [7,23]

Cryptography can be used to protect sensitive and valuable information against criminals or malicious hackers. Historically, four groups, the military, diplomats, diarist and lovers, have used and contributed to the art of cryptography [136].

### 4.2.1 Secret Key [13]

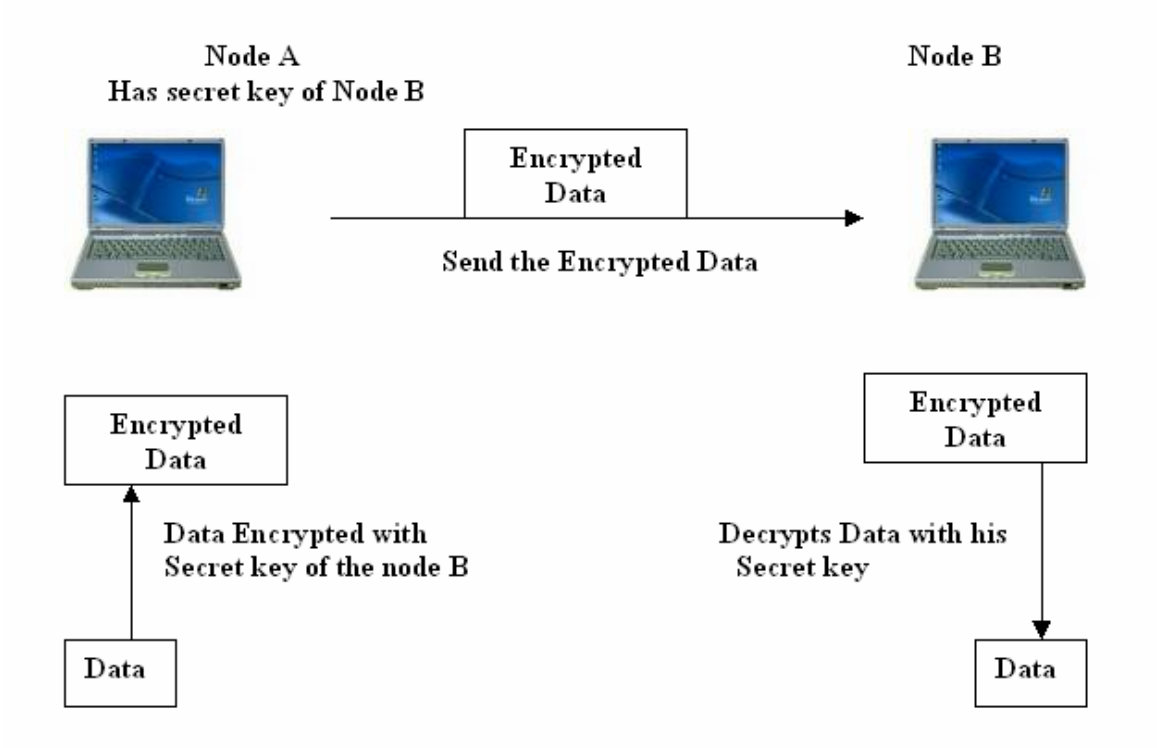



**Figure 4.2.** Secret Key Cryptography

This crypto-system uses the same key for both encryption and decryption (this is also referred to as "symmetric" cryptography). Both the sender and the receiver need to have the same key in order to communicate successfully [41].

The Advantages of Secret Key Cryptography are:

- Very fast relative to public key cryptography,
- Considered secure, provided the key is relatively strong,
- The ciphertext is compact (that is, encryption does not add much excess "baggage" to the ciphertext), and
- Widely used and very popular.

The Disadvantages of Secret Key Cryptography are:

- The administration of the keys can become extremely complicated,
- A large number of keys is needed to communicate securely with a large group of people,
- Non-repudiation is not possible, and
- The key is subject to interception by hackers.

### 4.2.2 Public Key Cryptography

This crypto-system uses one key for encryption and another key for decryption (also known as "asymmetric" cryptography [12]). Each user has two keys – one public key, which is revealed to all users, and one private key, which remains a secret. The private key and the public key are mathematically linked. Encryption is performed with the public key and decryption is performed with the private key. Examples: RSA, Elliptic Curve Cryptography (ECC).

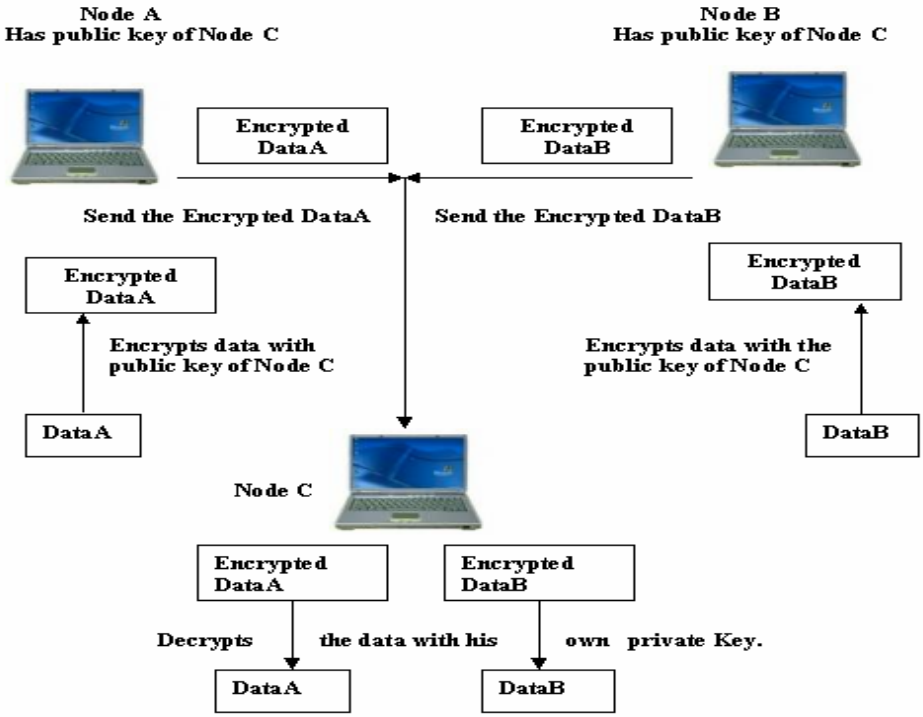



**Figure 4.3.** Public Key Cryptography

The advantages of Public Key Cryptography are:

- Considered very secure,
- No form of secret sharing is required, thus reducing key administration to a minimum,
- Supports non-repudiation, and
- The number of keys managed by each user is much less compared to secret key cryptography.

The disadvantages of Public Key Cryptography are:

- Much slower compared to secret key cryptography, and
- The cipher text is much larger than the plaintext, relative to secret key cryptography.

### 4.2.3 Key Size Based Tradeoff

Often, people associate the size of the key [86] to the amount of security being applied. While cryptography does make e-commerce possible by protecting electronic information from prying eyes, the effectiveness of this protection partly depends on the cryptographic size of the key. If the algorithm is inherently strong, then it can be assumed that the larger the key size for the ciphers, the harder it is for a hacker to perform an attack (i.e., a brute-force attack) on the cipher text [123]. But applying the right level of security through key size is also somewhat dependent on the value of the data being transferred. The higher the value, the less risks the user is willing to take, the higher the level of security they will want to apply. Often, this means larger keys but larger keys leads to lower levels of performance. Simply put, it takes longer to communicate and in an e-commerce situation, larger keys can severely degrade server performance. There are, therefore, trade-offs, which are traditionally made between the level of security and other factors, like performance. These are issues, which must be addressed in the successful implementation of a cryptosystem or even a good security policy.

### 4.2.4 Threshold Cryptography

Threshold cryptography involves the sharing of the key by working in the distributed manner or it may involve to redundantly split the message into $n$ pieces such that with $t$ or more pieces the original message can be recovered. The former

approach specifies the to have the distributed architecture in the hostile environment. The later approach ensures secure message transmission between two nodes over *n* multiple paths.

The various threshold schemes generally involve the procedures of - Key generation, encryption, share generation, share verification and share combining algorithms. Further, share generation for data confidentiality and integrity is the basic requirement of such type of schemes.

Threshold models can be broadly divided into two schemes :

- Single secret sharing threshold e.g. Shamir's t-out-of-n scheme based on Lagrange's interpolation, and
- Threshold sharing functions e.g. geometric based threshold .

These schemes are being used to implement threshold variants of RSA, El Gamal, and Diffie-Hellman cryptographic algorithms that have characteristic, E (x + y) = E (x) * E (y), called *homomorphism*.

Threshold Cryptography has many applications, some of them are - Document authorization/signing or verification in organizations, voting system for allowing access to system resources, E-commerce transactions, Distributed online certification authority, Key distribution in computer networks. Further, it can be implemented in various applications in a MANET, such as coordinating efforts of military attacks using wireless devices in the battlefield or in disaster-struck area, wireless connectivity of various home appliances, and establishing communication among laptops, PDAs and other wireless devices at conferences.

## 4.3 DNA Cryptography

It was in 1950s that the helical structure of DNA [17,54] was discovered by Watson and Crick [133], but it was in 1994 that Leonard Adleman, University of South California solved NP complete problems using algorithms inspired by DNA. In DNA cryptography [57] the messages to be encrypted, known as the plain text are transformed by a function that is parameterized by key. The output of the

encryption process known as the *ciphertext* (or *cryptogram*) is then transmitted often through a messenger or radio. In the following section, the pseudo DNA cryptography method is discussed and has been analysed and evaluated to be suitable for secure communication.

### 4.3.1 Pseudo Cryptography Method

Cryptography by pseudo random numbers [29,85] is being extended into this field. This method does not really use DNA sequences (or oligos), but only the mechanisms of the DNA function; therefore, the method is a kind of pseudo DNA cryptography methods. The cipher/ decipher process of the method is based on the central dogma of the molecular biology, and the process is similar to the DNA transcription, splicing and RNA translation of the real organisms. Fig. 2 represents the central dogma of molecular biology [3,4].

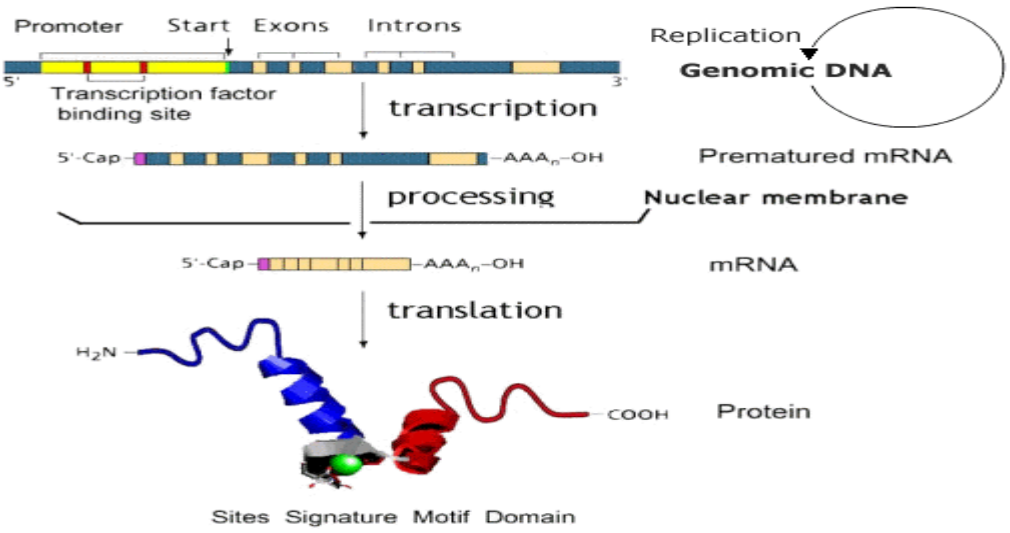



**Figure 4.4.** The Central Dogma of molecular biology [75]

The transcription, splicing and translation processes are briefly explained below:

**Transcription and Splicing:** a DNA segment that constitutes a gene is read, starting from the promoter (starting position) of the DNA segment. The non-coding areas (intron) are removed according to certain tags, and the remaining coding areas (extron) are rejoined and caped. Then the sequence is transcribed into a

single stranded sequence of mRNA (messenger RNA). The mRNA moves from the nucleus into the cytoplasm.

**Translation:** the mRNA sequence is translated into a sequence of amino acids as the protein is formed. During translation, the ribosome reads the fragment starting from certain three-bases, and then the ribosome reads three bases (a codon) at a time from the mRNA and translates them into one amino acid; there are also certain ending three-bases to sign the end of the translation.

Essentially, in the transcription and the splicing steps, introns are cut out, and extrons are kept to form mRNA, which will perform the translation work. In the translation process, codons are translated into the amino acids according to the genetic code table [80].

### 4.3.2 Pseudo DNA cryptography

The pseudo DNA cryptography [17] method consists of two steps, similar to the translation/splicing and transcription processes. To make the cipher text difficult to decipher, some changes are suggested to the original splicing process. Originally, their starting and ending codes characterize the intron, which makes the guess about introns relatively easy. So, another scheme, in which the start codes and the pattern codes specify the introns can be deployed. The pattern codes are non-continuous patters, which define which parts of the frame to be removed, and which parts to be kept. This makes the guess on introns difficult, since they are now spaced introns. Since the starting and pattern codes can determine the length of the introns, the ending codes are not necessary.

Suppose Alice be the sender of the information and Bob be the receiver. From Alice's point of view, the information is stored in the binary form, and can be transformed into DNA form (A for 00, C for 01, G for 10, T for 11). Alice also knows the starting codes (codes that indicate the begin of the intron) and pattern codes (codes that define which parts of the frame to be removed, and which parts to be kept) of the introns, so she knows where are the introns in the DNA form of

the information and which parts should be removed. Therefore, the two steps can be described briefly as below:

The DNA form of information is scanned by Alice to find out the introns; she records the introns places, and cut out the introns according to the specified pattern. So that the DNA form of data is translated into the mRNA form of data. Alice translates the mRNA form of data into protein form of data according to the genetic code table (61 codons to 20 amino acids).

The protein form of data is then transferred to Bob. The starting and pattern codes of the introns, the places of the introns, the removed spaced introns, and the codon-amino acids mapping of the protein are the keys to decrypt the protein form of data, and they are transferred to Bob through a secure channel (or they are encrypted by Bob's public key, and transferred to Bob).

On Bob's side, he receives keys through the secure channel from Alice (or uses public key protocol to communicate with Alice to receive keys). When he received the protein form of data and the keys, Bob uses the keys to recover mRNA form of data from protein form of data, and then recover DNA form of information, in the reverse order as Alice encrypt the information. He can then recover then binary form of information, and finally gets what Alice sent him. The scheme of the method is illustrated in the Figure 4.5.

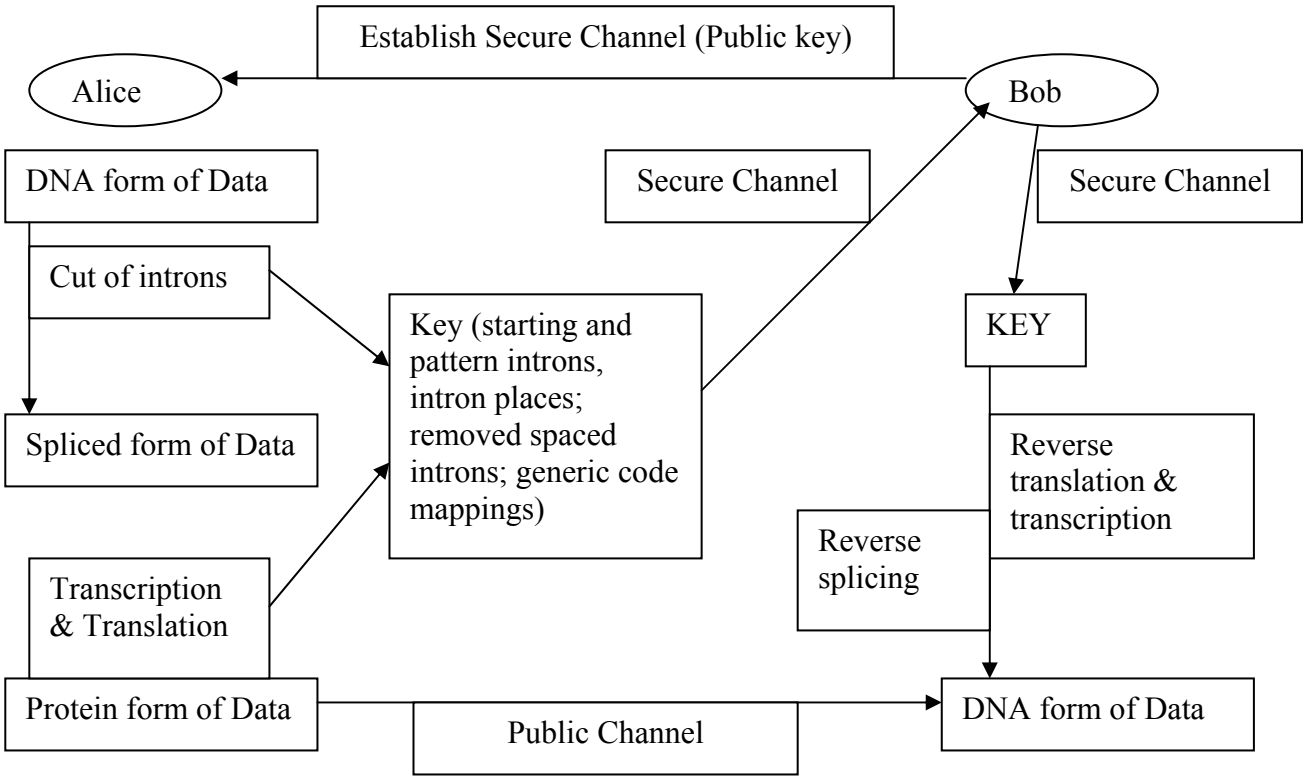



**Figure 4.5**. The implementation diagram of the pseudo DNA cryptography method

Generally, such a scheme can be formulated as below (M and C are for plaintext and cipher text, respectively)

- Encryption key EI = (the starting and pattern codes of the introns, the places of the introns)
- Encryption key E2 = (the codon-amino acids mapping)
- Decryption key $D_1 = E_2$
- Decryption key $D_2 = E_1$,
- $C' = E_1 (M)$, $C = E_2 (C')$
- $M' = D_1 (C)$, $M = D_2 (M')$

The scheme is principally a symmetric key algorithm, except that the sender initially has only part of the keys, and he generates the rest part of the keys. It is obvious that such a scheme is essentially a 2 step substitution process, though they are substitutions in a general sense (not letter-by-letter substitution).

### 4.3.3 DNA Cryptosystems Using Random One-Time-Pads

One-time-pad encryption uses a codebook of random data to convert plaintext to ciphertext. Since the codebook serves as the key, if it were predictable (i.e. not random), then an adversary could guess the algorithm that generates the codebook, allowing decryption of the message. No piece of data from the codebook should ever be used more than once. If it were, then it would leak information about the probability distribution of the plaintext, which would result in increasing the efficiency of an attempt to guess the message. This class of cryptosystem using a secret random one-time-pad is the only cryptosystem known to be absolutely unbreakable.

First, assemble a large one-time-pad in the form of a DNA strand, which is randomly assembled from short oligonucleotide sequences, then isolated and cloned. These one-time-pads are assumed to be constructed in secret, and we further assume that specific one-time-pads is shared in advance by both the sender and receiver of the secret message. This assumption requires initial communication of the one-time-pad between sender and receiver, which is facilitated by the compact nature of DNA.

### 4.3.4 DNA Cryptosystem using one-time-pad Substitution system

A substitution one-time-pad system uses a plaintext binary message and a table defining a random mapping to ciphertext. The input strand is of length $n$ and is partitioned into plaintext words of fixed length. The table maps all possible plaintext strings of a fixed length to corresponding ciphertext strings, such that there is a unique reverse mapping.

Encryption by substituting each occurs plaintext DNA word with a corresponding DNA cipher word. The mapping is implemented using a long DNA pad consisting of many segments, each of which specifies a single plain-text word to cipher word mapping. The plaintext word acts as a hybridization site for the binding of a primer, which is then elongated. This results in the formation of a plaintext-ciphertext word-pair. Further, cleavage of the word-pairs and removal of the plaintext portion must occur.

An ideal one-time-pad library would contain a huge number of pads and each would provide a perfectly unique, random mapping from plaintext words to cipher words. The structure of an example pad [5] is given in Fig. 4.6.

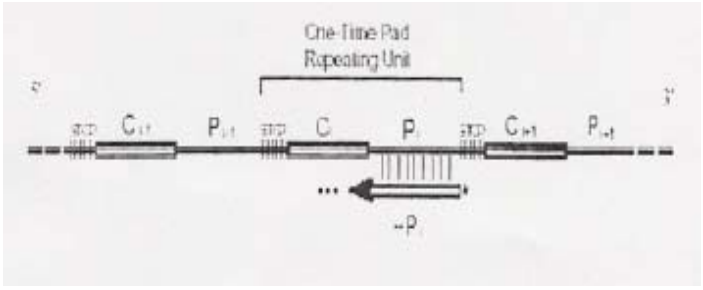



**Figure. 4.6.** One-time-pad codebook DNA Sequences

The repeating unit is made up of:

- One sequence word, $C_i$, from the set of cipher or codebook-matching words;
- One sequence word, $P_i$, from the set of plaintext words; and
- A polymerase "stopper" sequence.

The stopper sequence prohibits extension of the growing DNA strand beyond the boundary of the paired cipher word. Each $P_i$ includes a unique subsequence, which prevents frequency analysis attacks by mapping multiple instances of the same message plaintext to different ciphertext words. Further, this prefix could optionally be used to encode the position of the word in the message.

The one-time-pad consists of a DNA strand of length $n$ containing $d = n/L_1+L_2+L_3$ copies of the repeating pattern: a plaintext word of length $L_1$ ; a cipher word of length $L_2$ ; and stopper sequence of length $L_3$. The word length grows logarithmically in the total pad length; specifically for fixed integer constants $C_1$ ,$C_2$ ,$C_3 > 1$.

The length may be defined as –

$L_1 = C_1 \log_2 n;$

$L_2 = C_2 \log_2 n$ , and

$L_3 = C_3$ ;

Each repeat unit specifies a single mapping pair, and no codebook word or plaintext word will be used more than once on any pad. Therefore. given a cipher word $C_i$, we are assured that it maps to only a single plaintext word $P_i$ and vice versa. Stopper sequences consist of a run of identical nucleotides that act to halt strand copying by DNA polymerase given a lack of complementary nucleotide triphosphate in the test tube.

### 4.3.5 Method Analysis

The experimental feasibility depends upon the following factors:

- The size of the lexicon, which is the number of plaintext-ciphertext word-pairs,
- The size of each word,
- The number of DNA one-time-pads that can be constructed in a synthesis cycle.
- The length of each message that is to be encrypted.

If for experimental reasons, a small lexicon is required, then the words used could represent a more basic set such as ASCII characters, resulting in a lexicon size of 128. It is estimated that in a single cloning procedure, we can produce $10^6$ to $10^8$ different one-time-pad DNA sequences. It is important to note that the choice of word encodings must guarantee an acceptable Hamming distance between sequences such that the fidelity of annealing is maximized.

The entire construction process can be repeated to prepare greater numbers of unique pads. Construction of the libraries of codebook pads can be approached using segmental assembly procedures used successfully in gene library construction projects and DNA word encoding methods used in DNA computation. We can set the constants $C_1$ and $C_2$ large enough so that the probability of getting repeated words on a pad of length $n$ is acceptably small.

## 4.4 Algorithm Analysis and Evaluation

### 4.4.1 Analysis of Preudo DNA Cryptography

Let us suppose that the DNA form of information D has a length *n.* There are *k* introns, and their average length is *m.* Then the mRNA form of information D' have the length *n-k*m.* Since one codon (consists of 3 nucleic acids) generally can be translated into one amino acid, the protein form of information D" have the length of *(n-k*m)/3.* Both the encryption and decryption process have the time complexity of *O (n).*

As there are 61 coding Codons, and only 20 amino acids, thus on an average 3 codons to be mapped on to the same amino acid, so the different number of mRNA for the protein form of data is about $3^{(n - k * m)/3}$. In reverse transcription process there are $2^{(n - k * m)}$ places to insert introns. Now, even if the introns are known, the time complexity of recovering DNA form data from mRNA form is $0(k*2^{(n-k*m)})$. If spaced intron scheme is used, the time complexity can be $0(2^k \text{ x } 2^{(n-k*m)})$. So the total time complexity can be at least $0(3^{(n-k*m)} \text{ x } 2^{(n-k*m)})$. Further by careful control, if *k*m* is around *0.35n* (= $log_2 3/ (log_2 3+3)*n$), then the complexity above is $O (2^n)$. The smaller the *k*m,* such time complexity can be increased, but more partial information would be revealed, which makes it venerable to other attacks. Since *n* is generally of the 1Mb scale, such a brute force attack method is unpractical.

### 4.4.2 Evaluation

The encryption/ decryption algorithm has been performed in C-language under Linux environment A P-IV (1.7 GHz)/512MB/40GB system was used for carrying out the work.

Each of the encryption/ decryption process is performed ten times and the average time is considered and for checking the robustness of the system, we have selected four different plain texts with increasing size.

The program is designed for a simple sender-receiver system. On the sender side, an initial key is required (starting introns and pattern codes) which the user himself

generates. The user first translates the plaintext into DNA form of information using conversion program. Processes of central dogma – splicing, transcription and translation are also simulated including necessary padding for compatibility reason. Now, the starting and pattern codes of the introns along with the places of introns, removed spaced introns and the codon-amino acid are added into the key file and the enciphered information is also created. These two files are then sent to receiver through two different channels, the enciphered file through public channel and the key file through secure channel.

At the receiver side, the enciphered information and the key file are received from two different channels. The key information in the key file is used to decipher the received information. Reverse translation, reverse transcription and reverse splicing processes are applied using the respective program and the information stored in the key file. After this process the receiver gets the DNA form of information and then the plain text can be easily recovered in order to know what the sender has sent to the receiver.

We have used highly divert plaintext, which include short-text, long-text purely alphabetical and text combining alphabets in order to test the performance of the program. Table 4.1 shows the different texts considered:

**Table 4.1.** Performance of application with plaintext of different contents

| **Dataset** | **Description** | **Number of Different Characters** | **Recovery of Plaintext** |
|---|---|---|---|
| Test1 | Only alphabetical and digital characters | 52 | Yes |
| Test2 | Only non-alphabetical and non-digital characters | 100 | Yes |
| Test3 | Combination of Characters | 75 | Yes |
| Test4 | Combination of Characters | 125 | Yes |

Further, each plaintext has the length 10 times that of the former one, starting from 10 and then number bits needed to store in ASCII format is calculated, which is eight times the length of plain text. As observed

For cipher text, the length and the relative bits are also calculated. Since, one amino acid can be represented by a 3-letter sequence, and for unique 64 amino acids, we require 6 bits and only 2 bits are needed for each letter in cipher text. As a result, the number of bits needed for each of cipher-text is double the length of cipher text.

We have also introduced the redundancies (tags and separators). The actual key information is dependent on size of the starting and pattern codes of the introns but its size is roughly less than half of the size including redundancy (refer Table 4.2). The encryption and decryption times are also listed, which points towards the efficiency of the algorithm, as shown in Figure 4.7 and Figure 4.8.

**Table 4.2.** Performance of the application with plaintexts of different lengths

| **Dataset** | **Length of Plaintext (Bits)** | **Length of Cipher Text (Bits)** | **Size of Key (with redundancy)** | **Encryption Time (ms)** | **Decryption Time (ms)** |
|---|---|---|---|---|---|
| Test1 | 10 (80) | 51 (102) | 100 | 254 | 428 |
| Test2 | 100 (800) | 396 (792) | 708 | 258 | 431 |
| Test3 | 1000 (8000) | 3521 (7042) | 5211 | 298 | 483 |
| Test4 | 10000 (80000) | 36212 (72424) | 48132 | 1292 | 1398 |

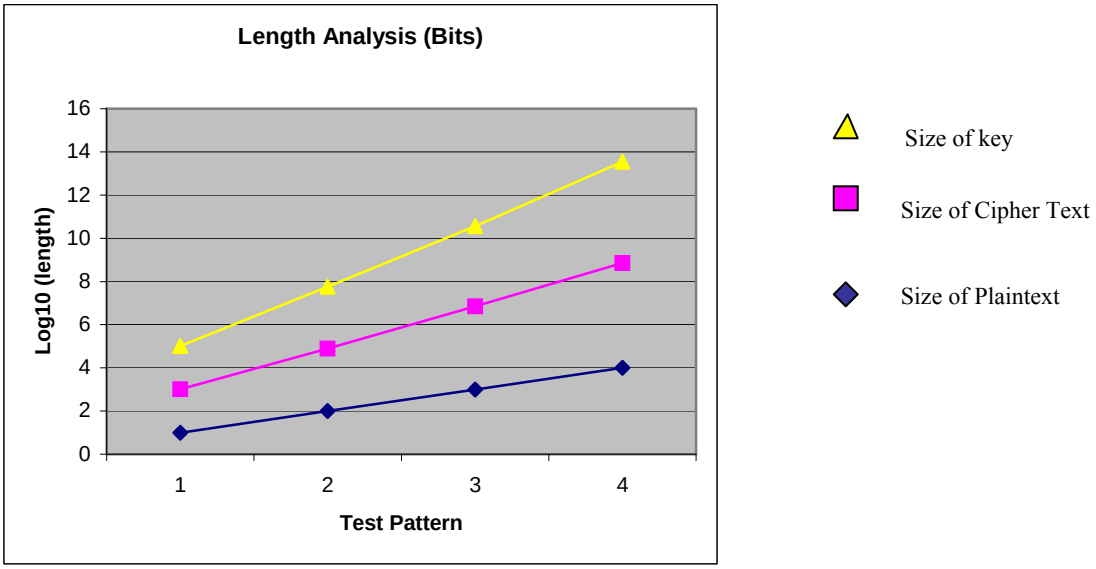



**Figure 4.7.** Analysis of Length of plaintext and cipher text.

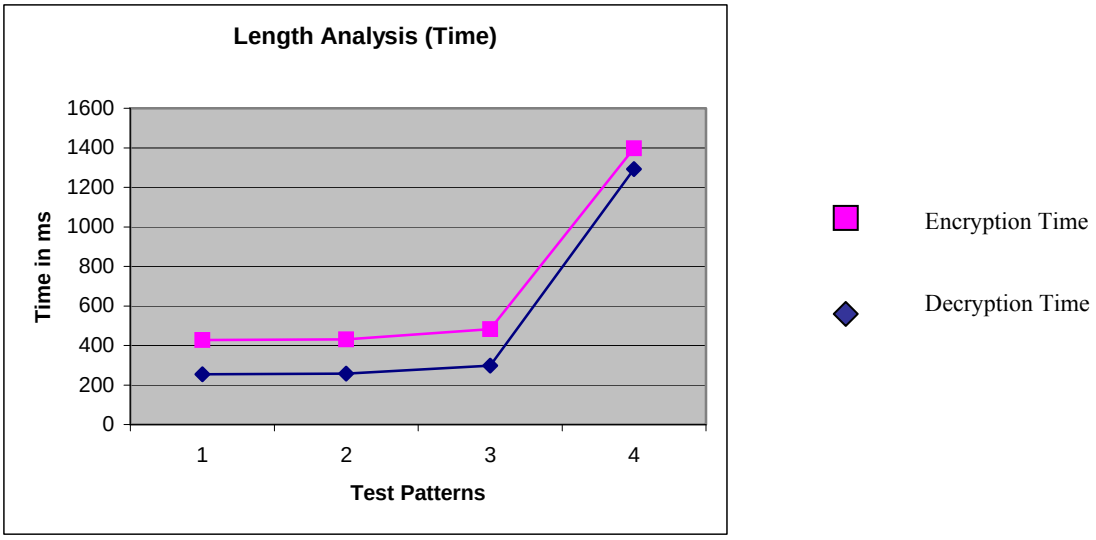



**Figure 4.8.** Analysis of Encryption and Decryption Time

## 4.5 Conclusion

We have proposed a pseudo DNA cryptography method, based on central dogma of molecular biology. The method simulates the transcription and translation process of the central dogma; it also adds some artificial features to make the resulting cipher texts difficult to break. The simulated analysis shows that this method is powerful against certain attacks [118], especially against brute force

attacks. The experiments not only show the power of such a method, but also reveal that this method is very efficient in computation. The encryption and decryption time is very much affordable to be employed in area of MANETs.

# Chapter 5

# Simulation Results and Discussion

## 5.1 Introduction

The proposed secure routing protocol has been implemented by using pseudo DNA cryptography. We have used ns2, a discrete event simulator to verify and validate our results.

---

**While** Not Empty (*EventQueue*) **Do**

dequeue(*m*) /* earlier event from EventQueue */

update(*clock*)

simulate(*m*)

enqueue() /* enqueue any events produced */

**EndWhile**

---

**Figure 5.1.** Basic Sequential Discrete Event Simulator Algorithm

## 5.2 Strategy used

We correlate the published DSR [103] and the implemented DSR (writing the DSR script in Tcl/Tk). Then, we modified the written Tcl/Tk script to take into account the pseudo DNA cryptography method, as discussed in chapter 4. The modified Tcl/Tk script implements the proposed SANE-DNA protocol. For all the simulations carried, a warm up time of 1000 s has been taken into account before the start of the actual simulation, as suggested by Tracy camp et al. [126]. A total of 5 simulation runs have been performed for each metric and the results are discussed below.

## 5.3 Simulation Parameters

The different parameters considered for the simulation are shown in Table 5.1.

**Table 5.1.** Different simulation parameters used for evaluating SANE-DNA.

| Parameter | Value |
|---|---|
| Transmission Range | 250 m |
| Maximum Velocity ($v_{max}$) | 20 m/s |
| Simulation Time | 600 sec |
| Pause Time | 0 to 600 (interval time of 100 sec) |
| Environment Size | 1200 m x 300 m |
| Packet Size | 512 bytes |
| Link Bandwidth | 2 Mbps |
| Traffic Type | CBR (Constant Bit Rate) |
| Packet Rate | 2 packets/sec |

## 5.4 Performance Metric

The following performance metrics are used for evaluating, SANE-DNA, the proposed secure routing protocol.

- **Route Acquisition Time:** The time it takes a source node to find a route to a destination node.

- **Average End-to-End Delay or Mean Overall Packet Latency:** This implies the delay a packet suffers between leaving the sender application and arriving at the receiver application.

- **Routing Overhead:** The total number of routing packets transmitted during the simulation.

- **Throughput or packet delivery ratio:** The ratio between the number of packets sent out by the sender application and the number of packets correctly received by the corresponding peer application.

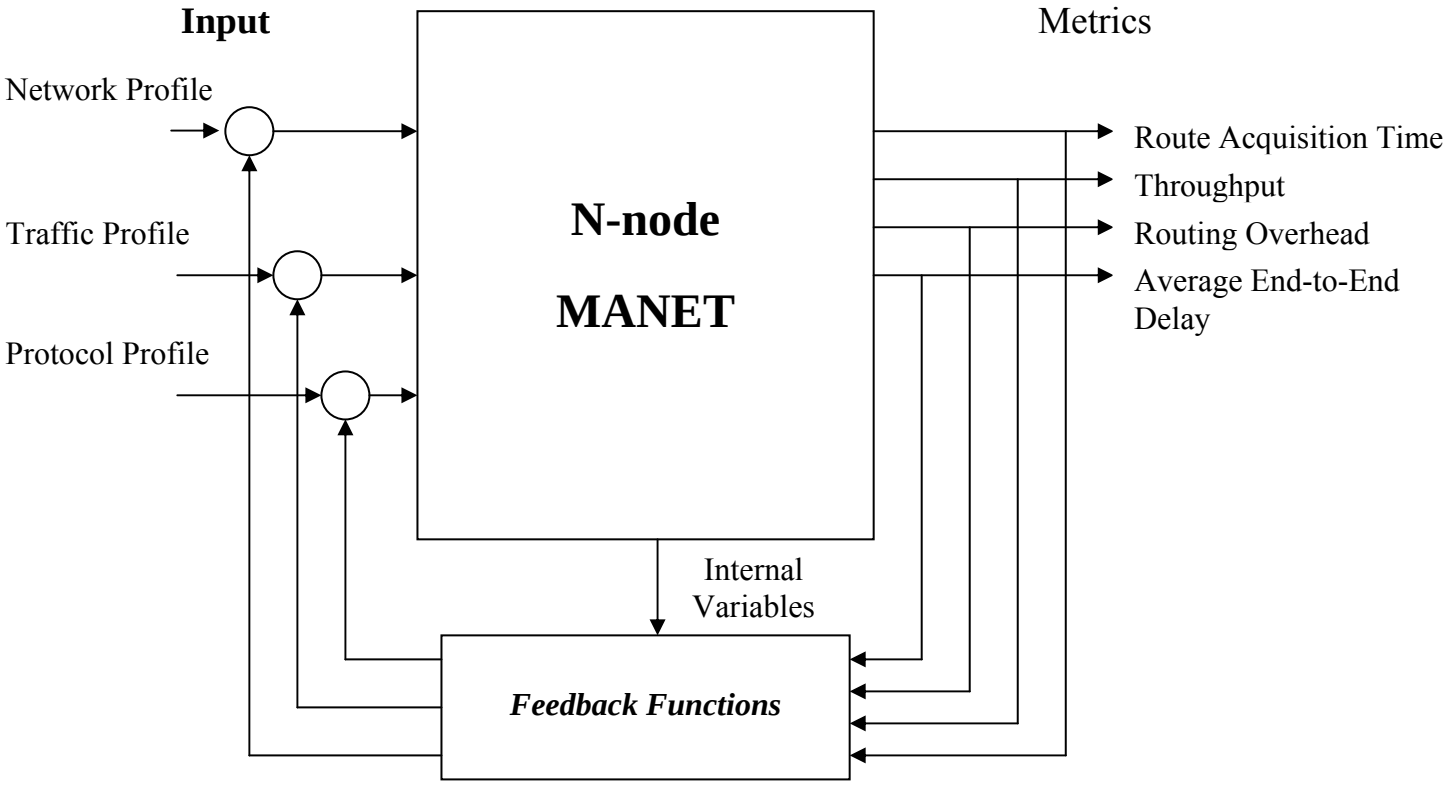



**Figure 5.2.** A model of simulated MANET System

## 5.5 Published DSR

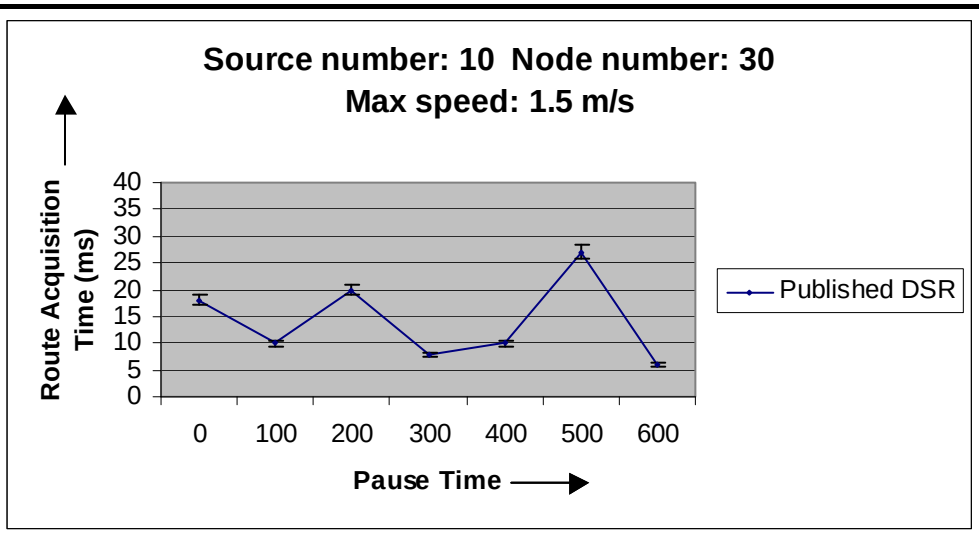



Figure (a)

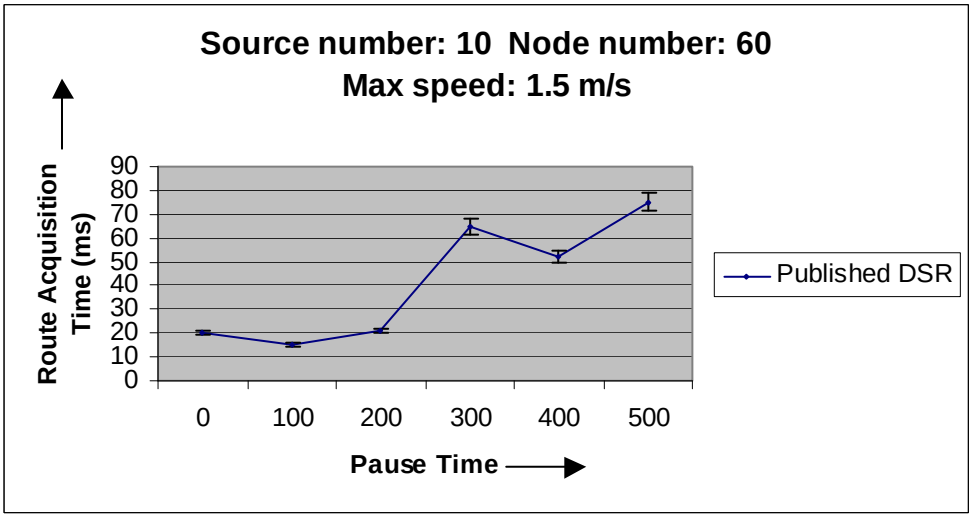



Figure (b)

**Figure 5.3.** Graphs showing the output of Route Acquisition Time for Published DSR against different simulation Parameters

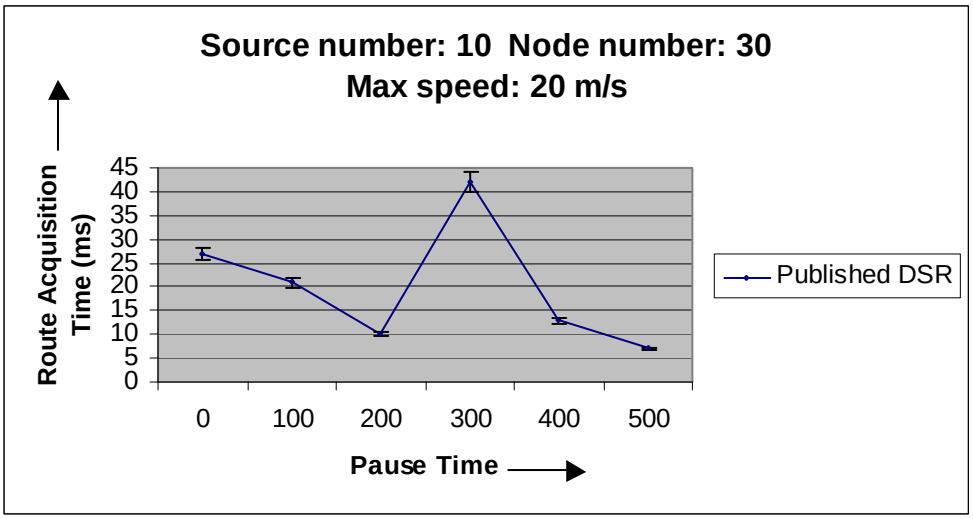



Figure (c)

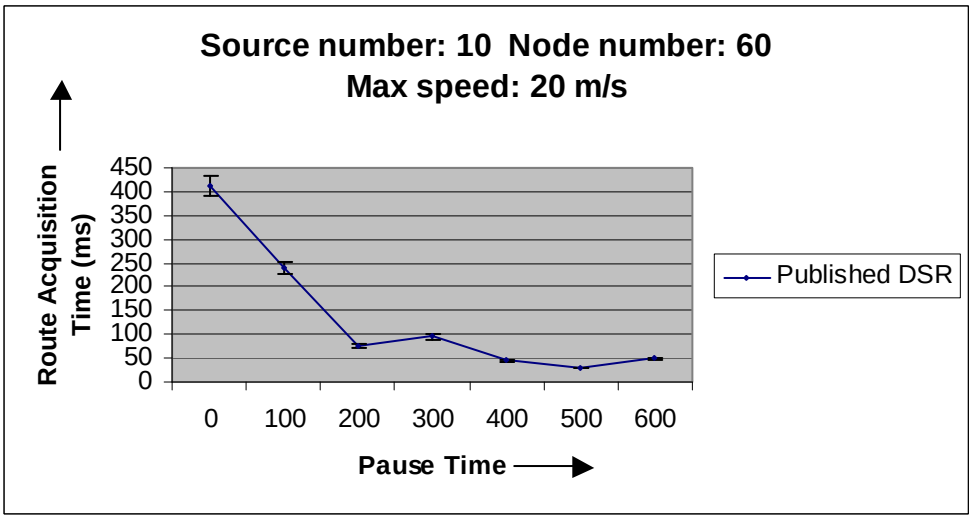



Figure (d)

**Figure 5.3.** Graphs showing the output of Route Acquisition Time for Published DSR against different simulation Parameters

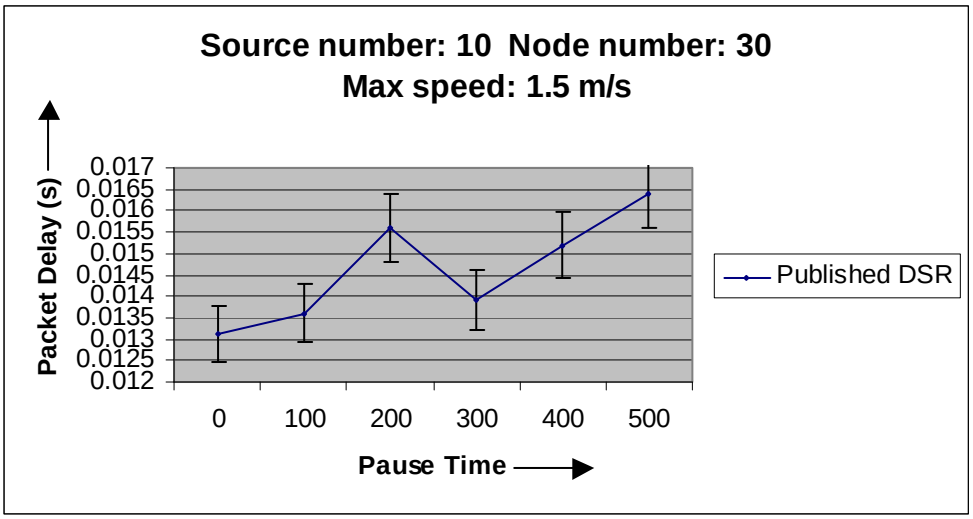



Figure (a)

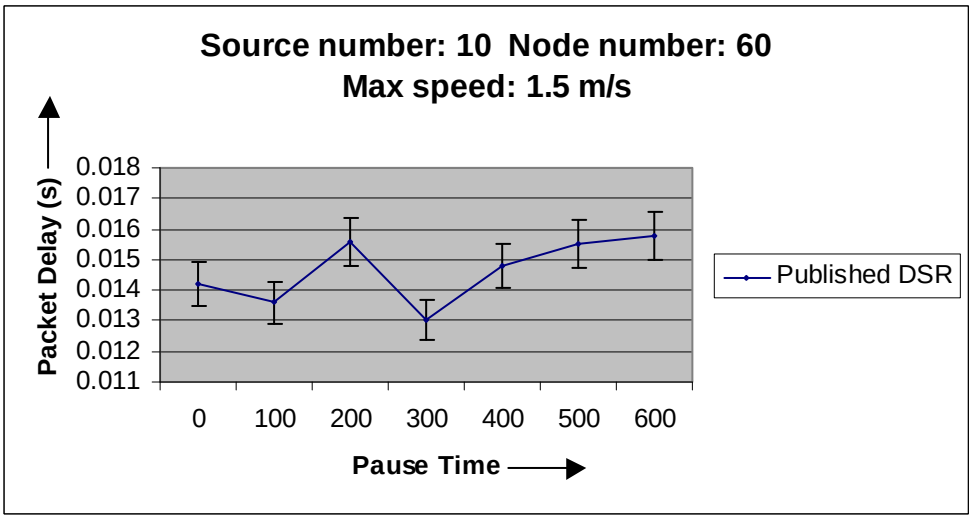



Figure (b)

**Figure 5.4.** Graphs showing the output of Packet Delay for Published DSR against different simulation Parameters

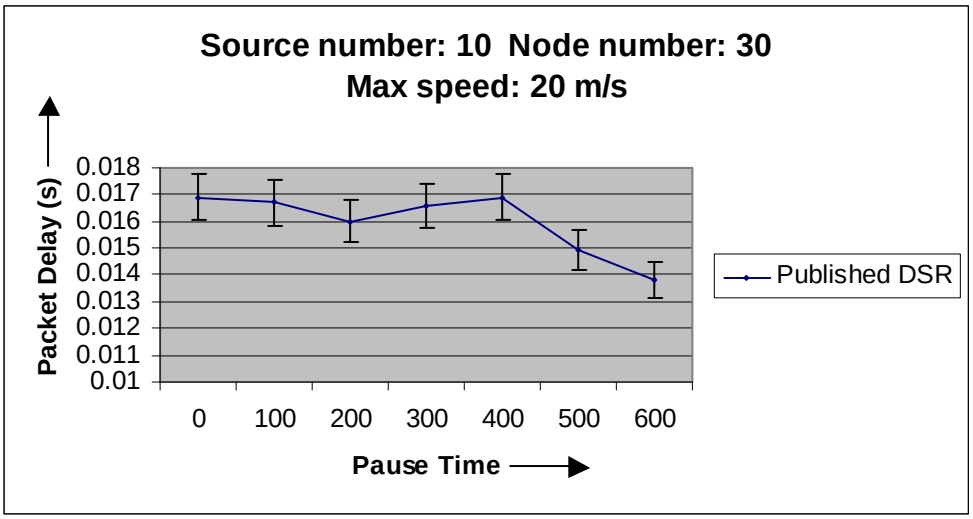



Figure (c)

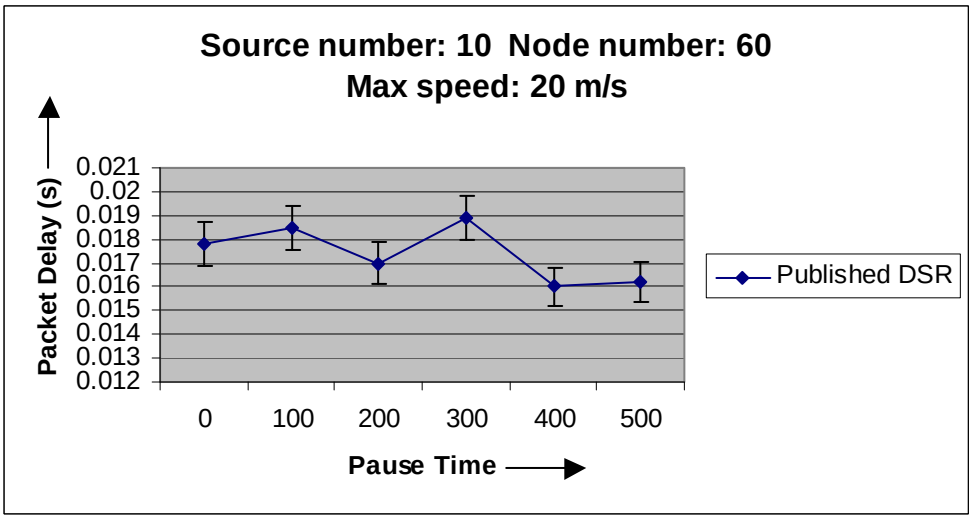



Figure (d)

**Figure 5.4.** Graphs showing the output of Packet Delay for Published DSR against different simulation Parameters

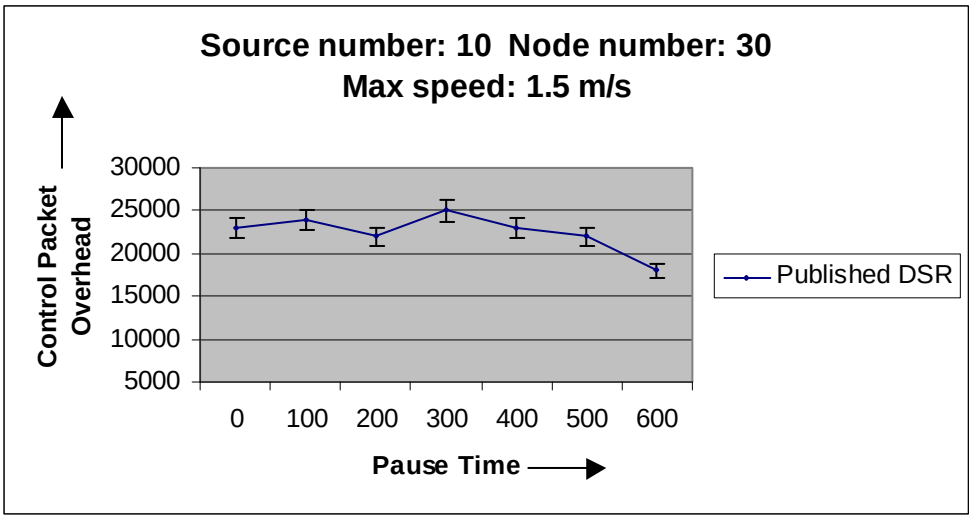



Figure (a)

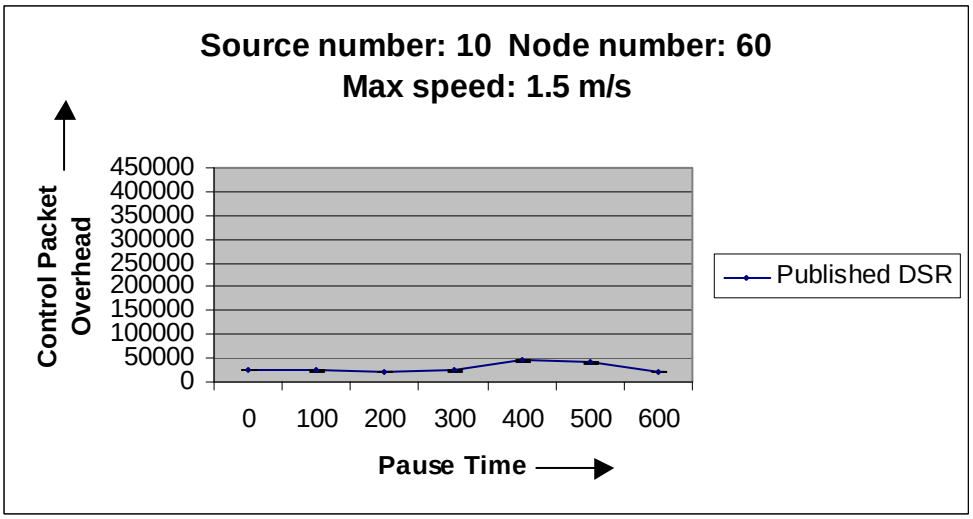



Figure (b)

**Figure 5.5.** Graphs showing the output of Routing Overhead for Published DSR against different simulation Parameters

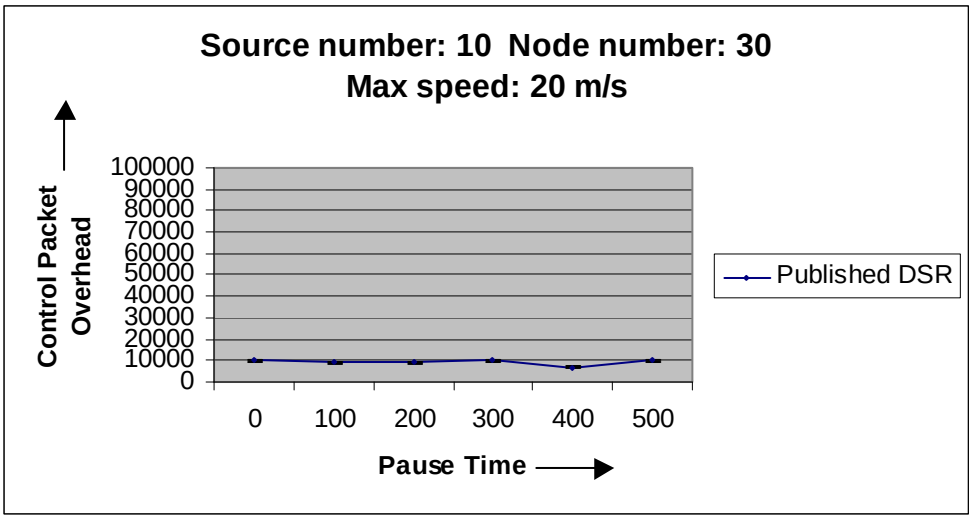



Figure (c)

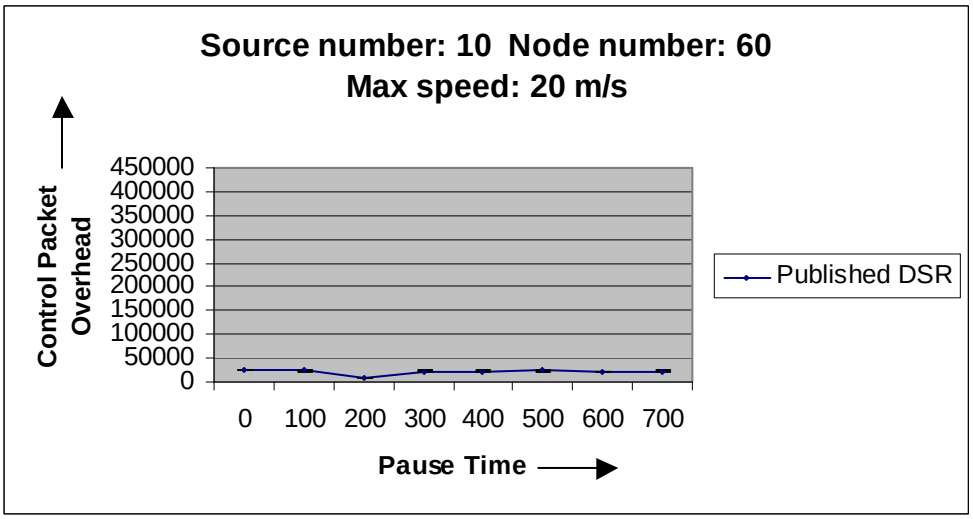



Figure (d)

**Figure 5.5.** Graphs showing the output of Routing Overhead for Published DSR against different simulation Parameters

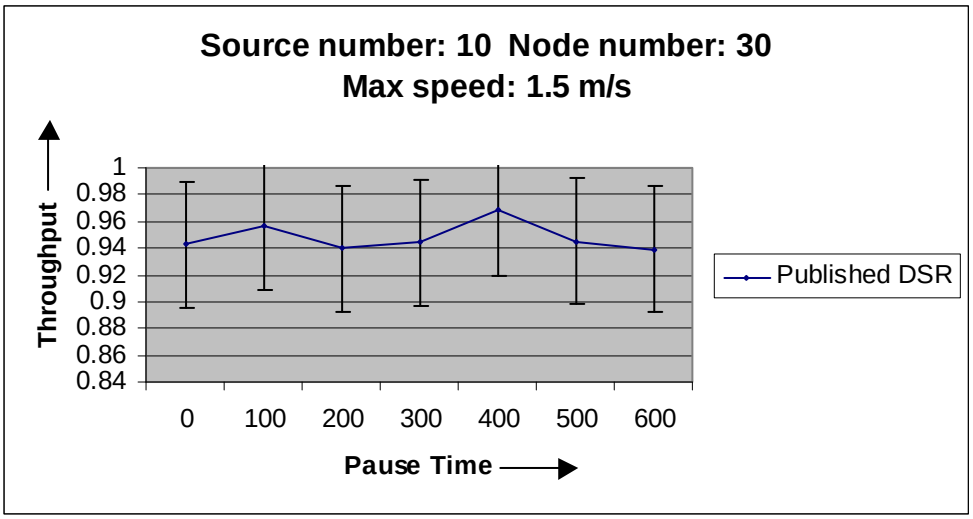



Figure (a)

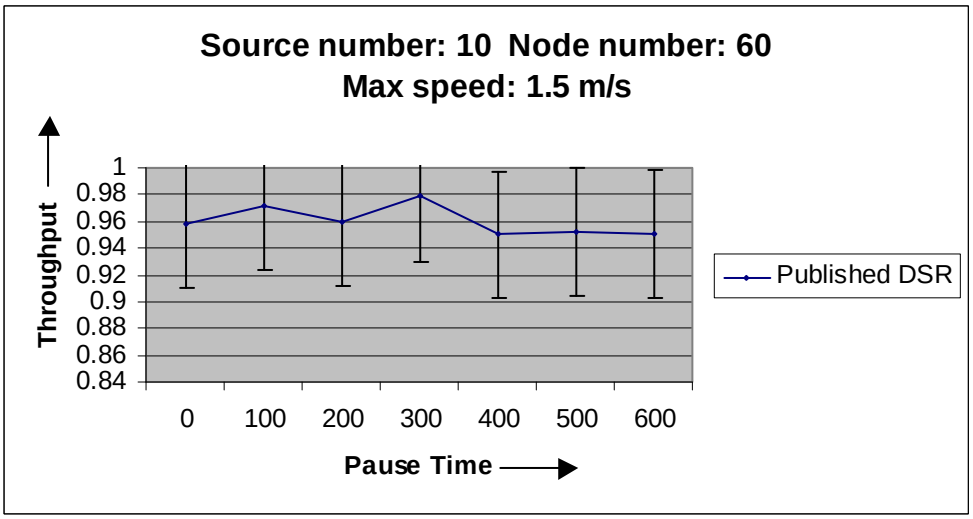



Figure (b)

**Figure 5.6.** Graphs showing the output of Throughput for Published DSR against different simulation Parameters

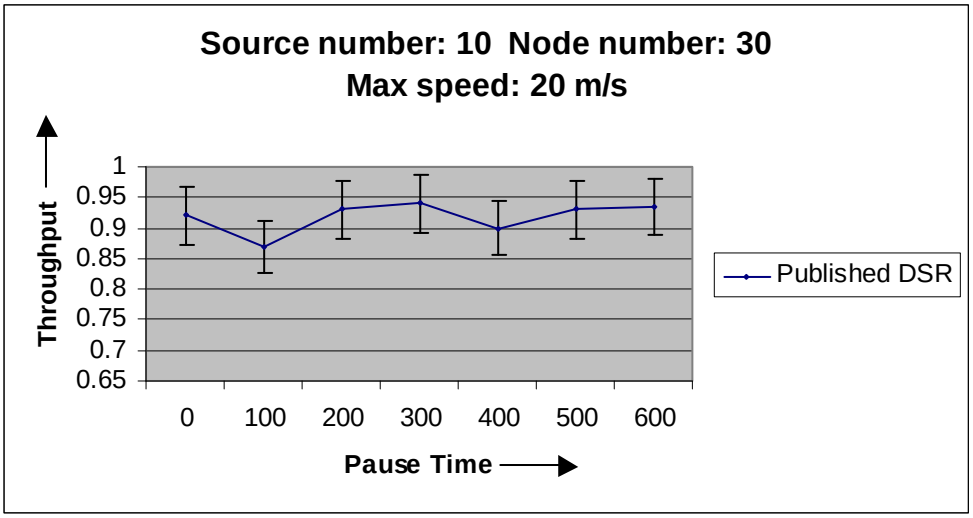



Figure (c)

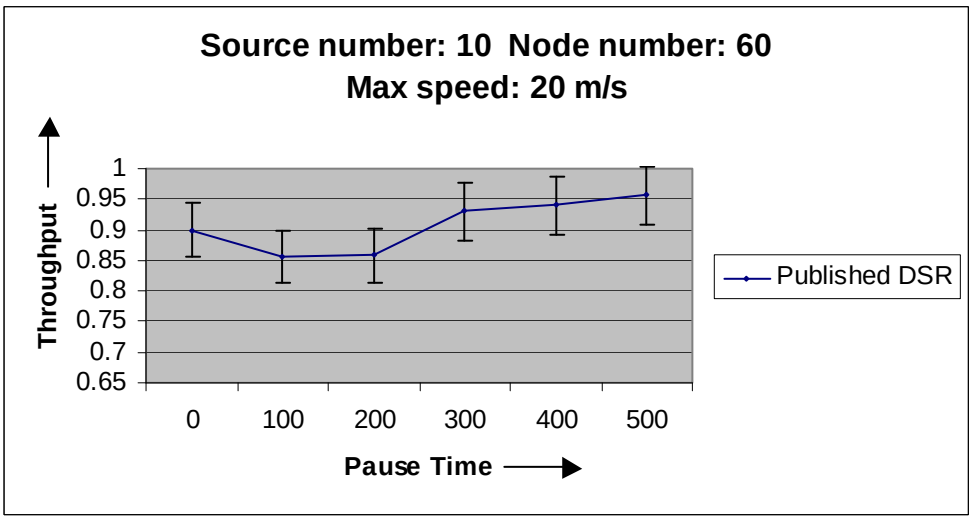



Figure (d)

**Figure 5.6.** Graphs showing the output of Throughput for Published DSR against different simulation Parameters

## 5.6 Published DSR and Implemented DSR comparison

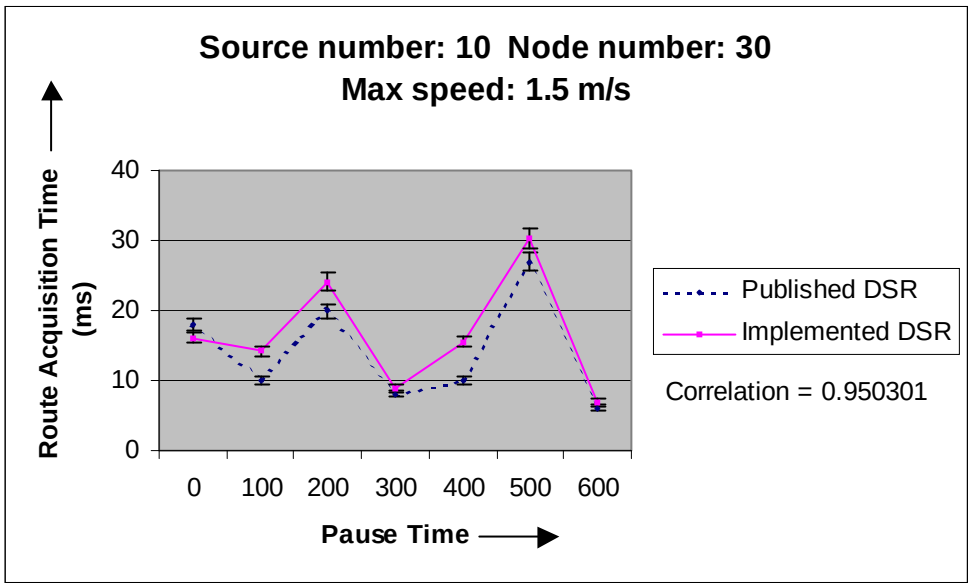



Figure (a)

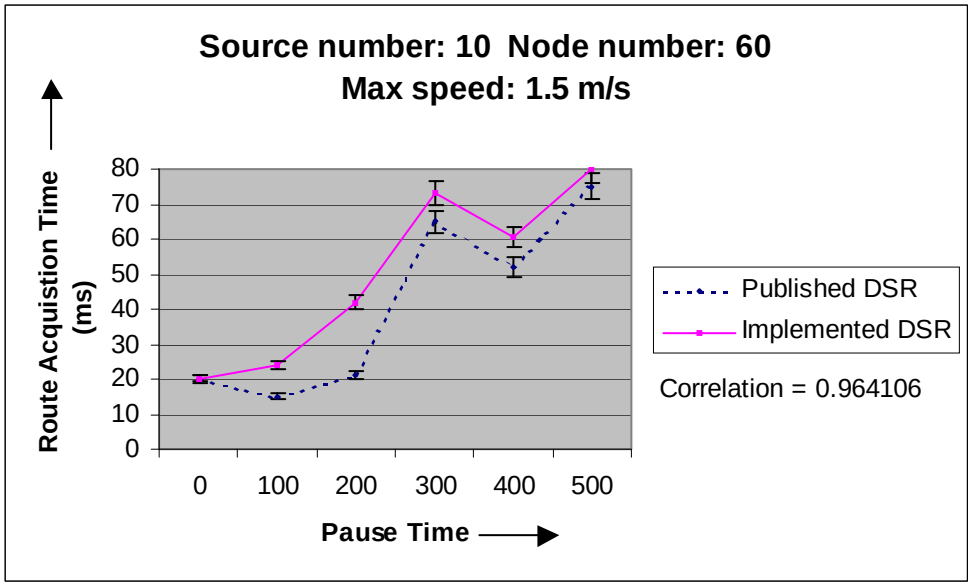



Figure (b)

**Figure 5.7.** Graphs showing the output of Route Acquisition Time for Published DSR vs Implemented DSR against different simulation Parameters

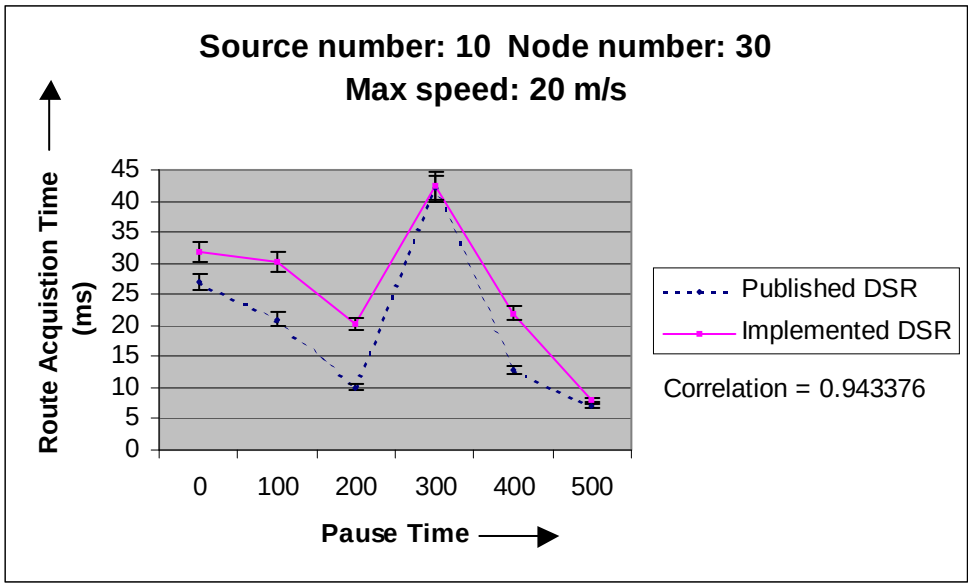



Figure (c)

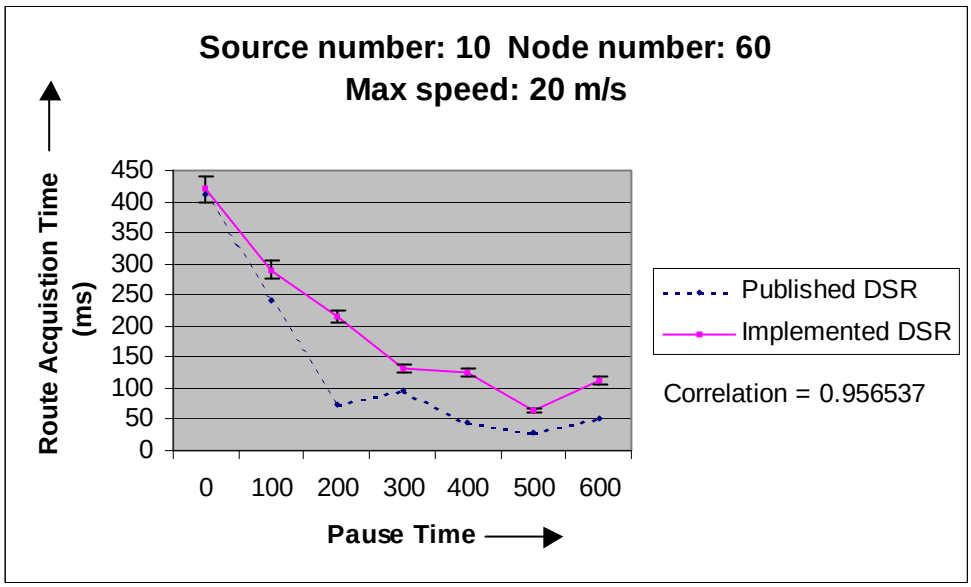



Figure (d)

**Figure 5.7.** Graphs showing the output of Route Acquisition Time for Published DSR vs Implemented DSR against different simulation Parameters

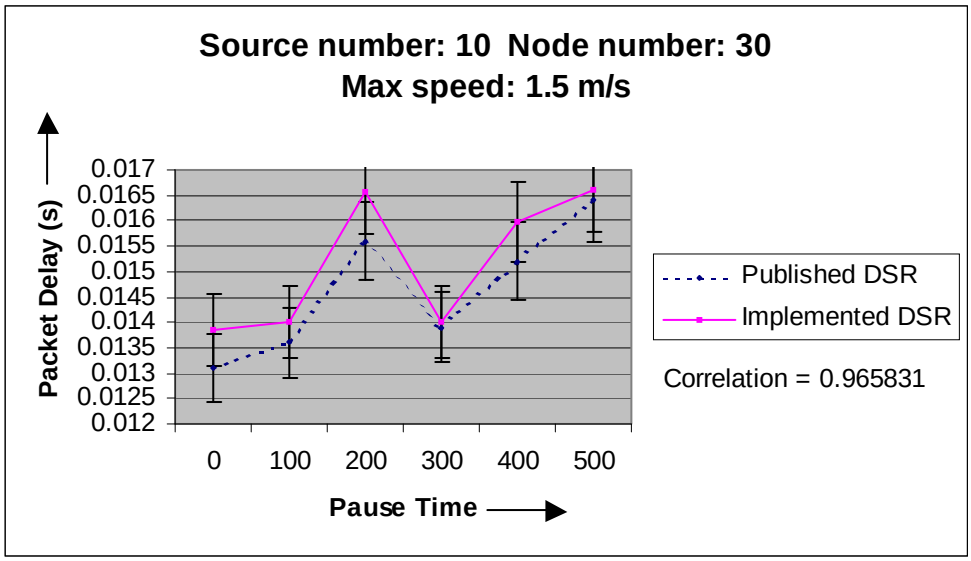



Figure (a)

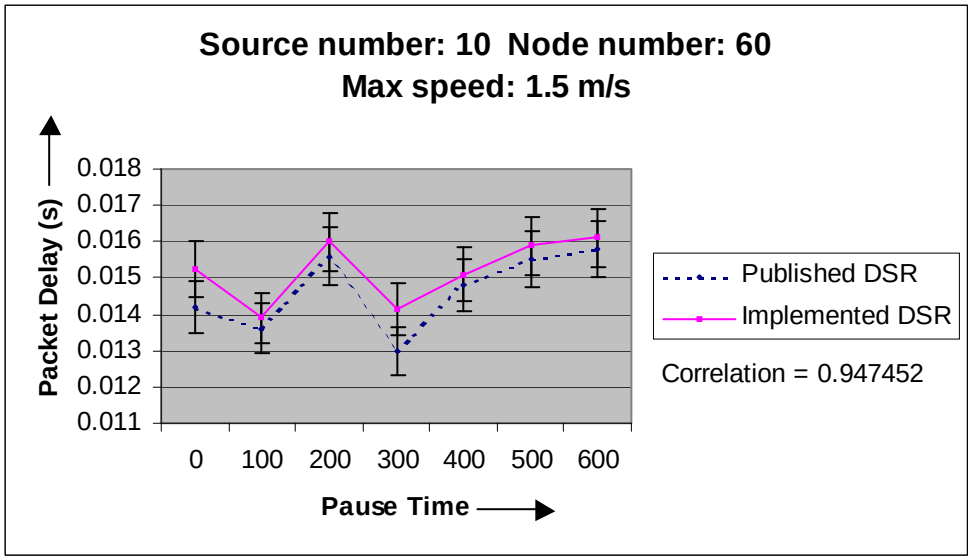



Figure (b)

**Figure 5.8.** Graphs showing the output of Packet Delay for Published DSR vs Implemented DSR against different simulation Parameters

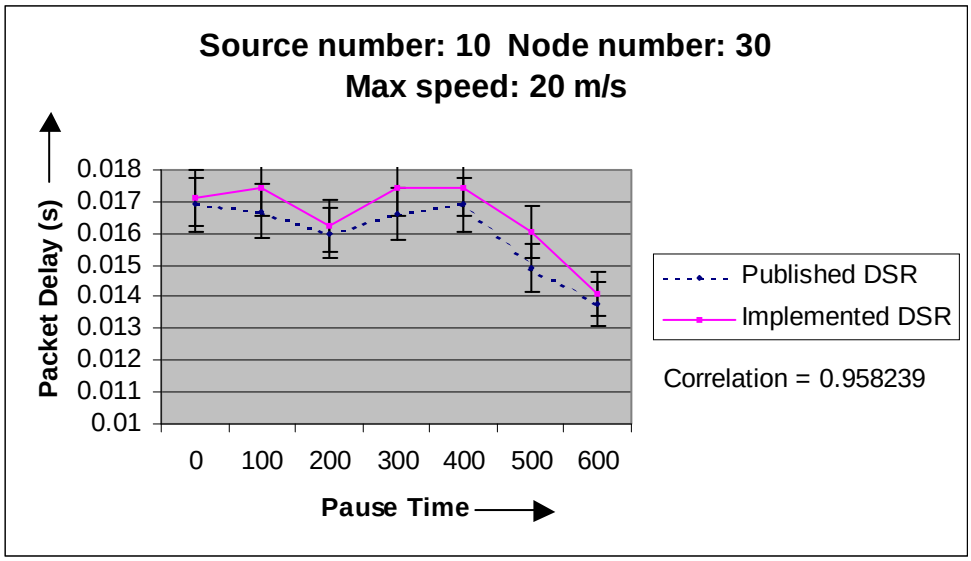



Figure (c)

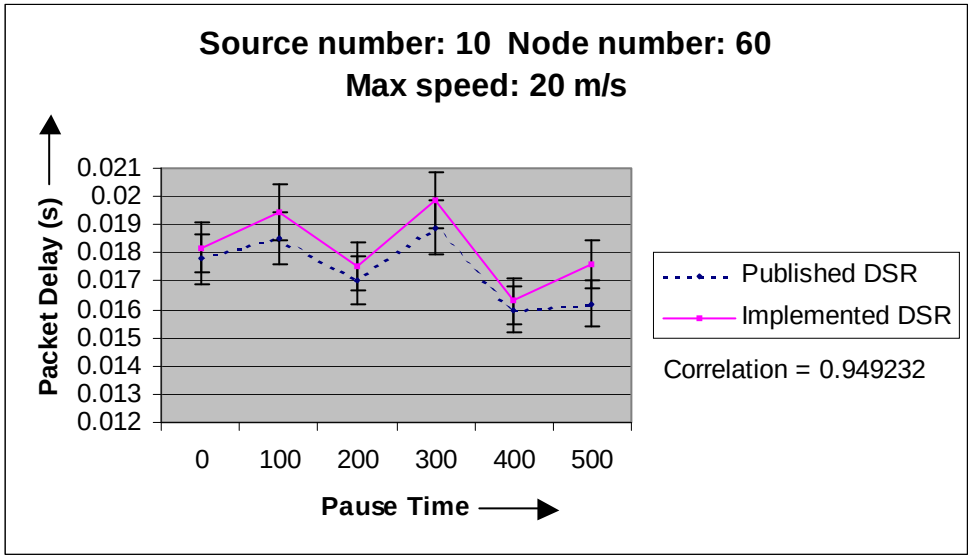



Figure (d)

**Figure 5.8.** Graphs showing the output of Packet Delay for Published DSR vs Implemented DSR against different simulation Parameters

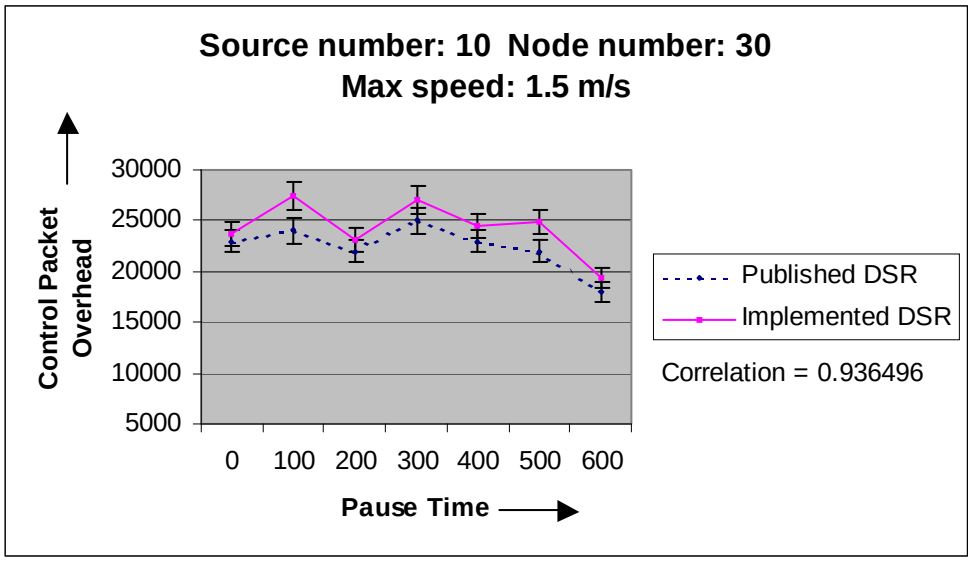



Figure (a)

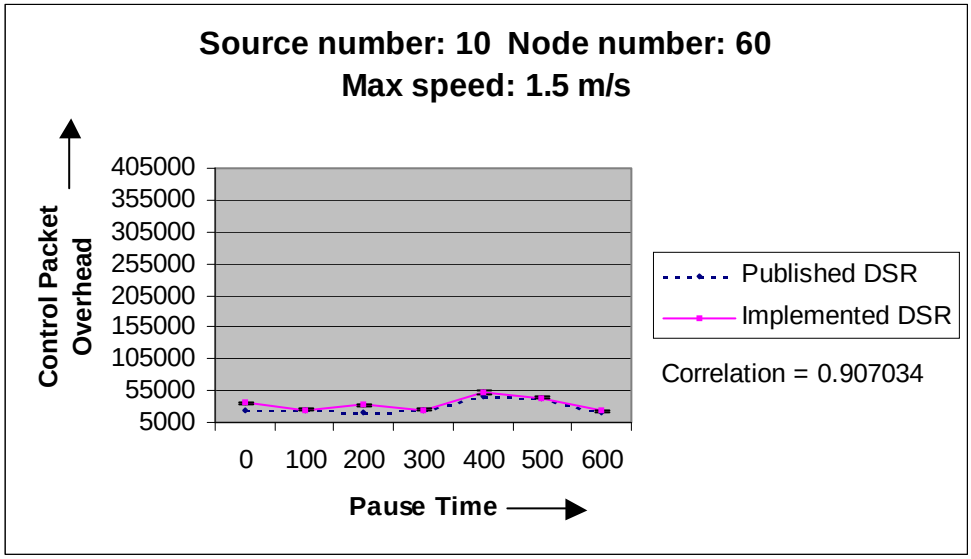



Figure (b)

**Figure 5.9.** Graphs showing the output of Routing Overhead for Published DSR vs Implemented DSR against different simulation Parameters

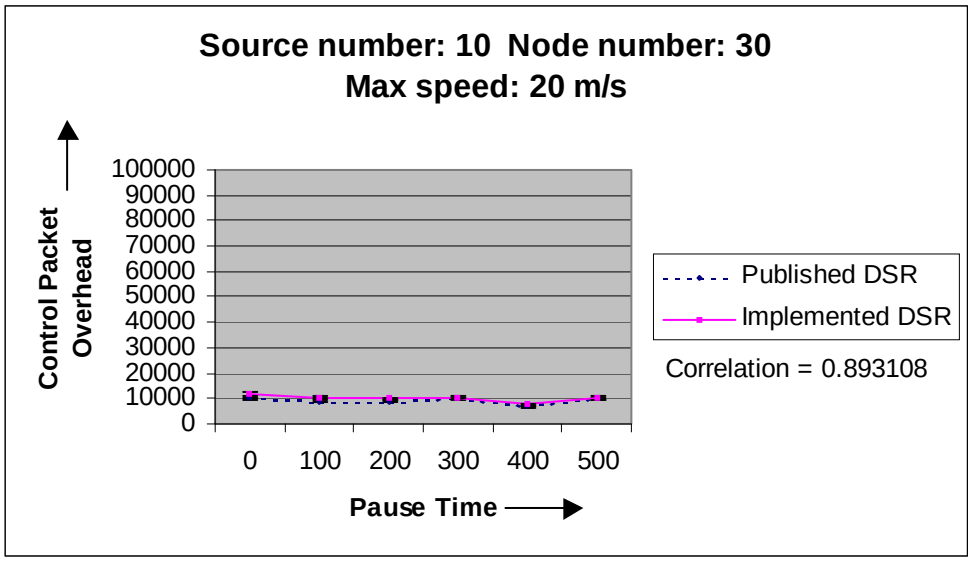



Figure (c)

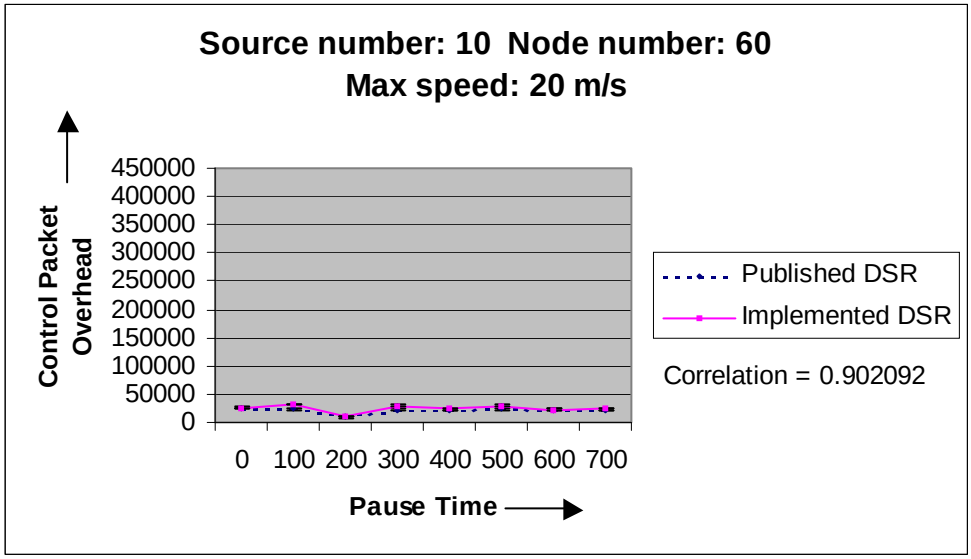



Figure (d)

**Figure 5.9.** Graphs showing the output of Routing Overhead for Published DSR vs Implemented DSR against different simulation Parameters

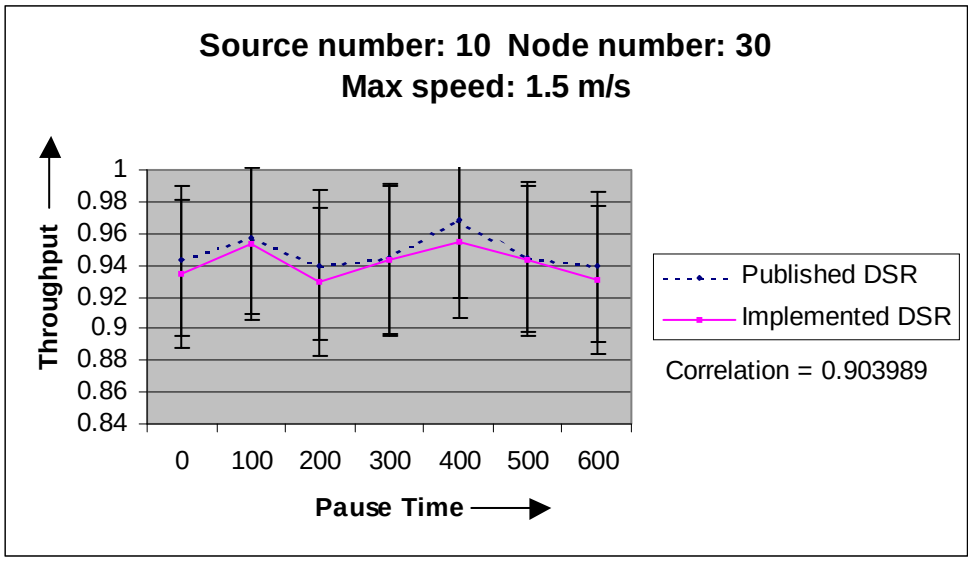



Figure (a)

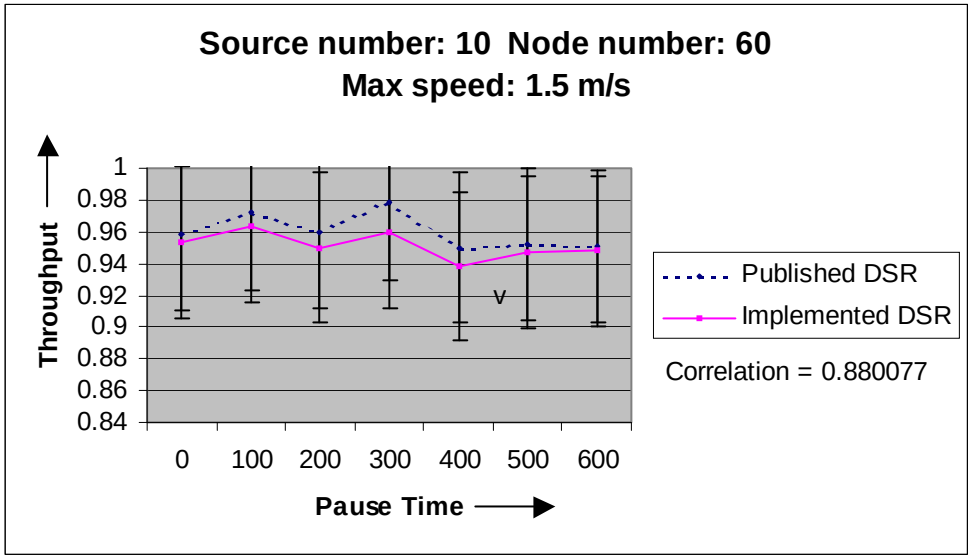



Figure (b)

**Figure 5.10.** Graphs showing the output of Throughput for Published DSR vs Implemented DSR against different simulation Parameters

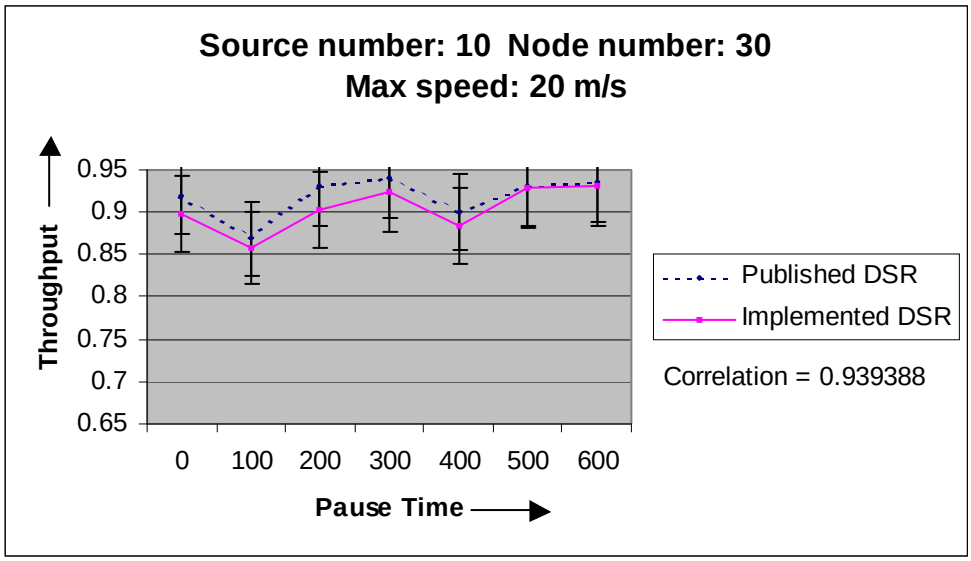



Figure (c)

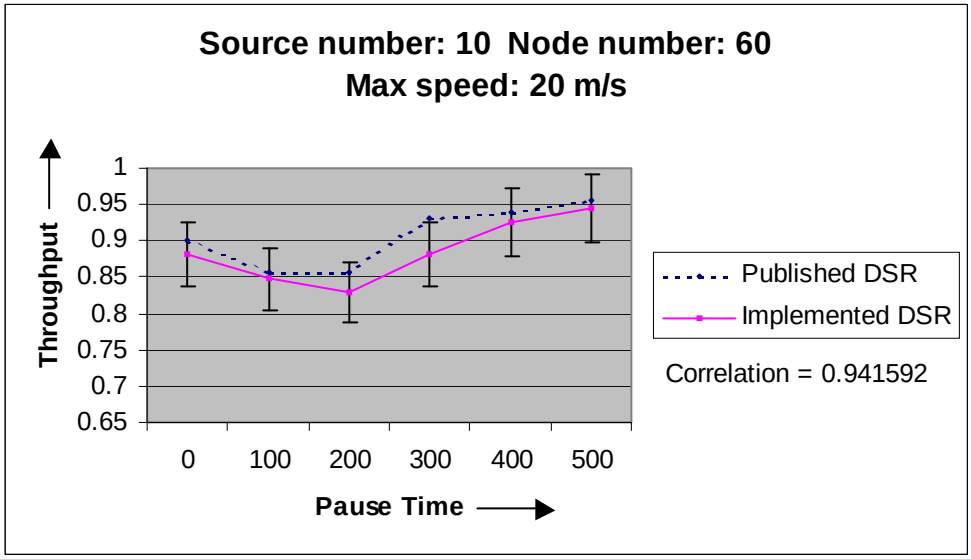



Figure (d)

**Figure 5.10.** Graphs showing the output of Throughput for Published DSR vs Implemented DSR against different simulation Parameters

**Table 5.2.** Coefficient of correlation between published DSR and implemented DSR

| Parameters Metrics | Source no. = 10 Node no. = 30 Max. speed = 1.5 m/s | Source no. = 10 Node no. = 60 Max. speed = 1.5 m/s | Source no. = 10 Node no. = 30 Max. speed = 20 m/s | Source no. = 10 Node no. = 60 Max. speed = 20 m/s |
|---|---|---|---|---|
| Route Acquisition Time (ms) | 0.950301 | 0.964106 | 0.943376 | 0.956537 |
| Packet Delay | 0.965831 | 0.947452 | 0.958239 | 0.949232 |
| Control Packet Overhead | 0.936496 | 0.907034 | 0.893108 | 0.902092 |
| Throughput | 0.903989 | 0.880077 | 0.939388 | 0.941592 |
| Σ | 3.956617 | 3.698669 | 3.734111 | 3.949353 |

→

↓ Mean Coefficient of Correlation = 15.33875/16 = 0.9586875

## 5.7 Implemented DSR and SANE-DNA comparison

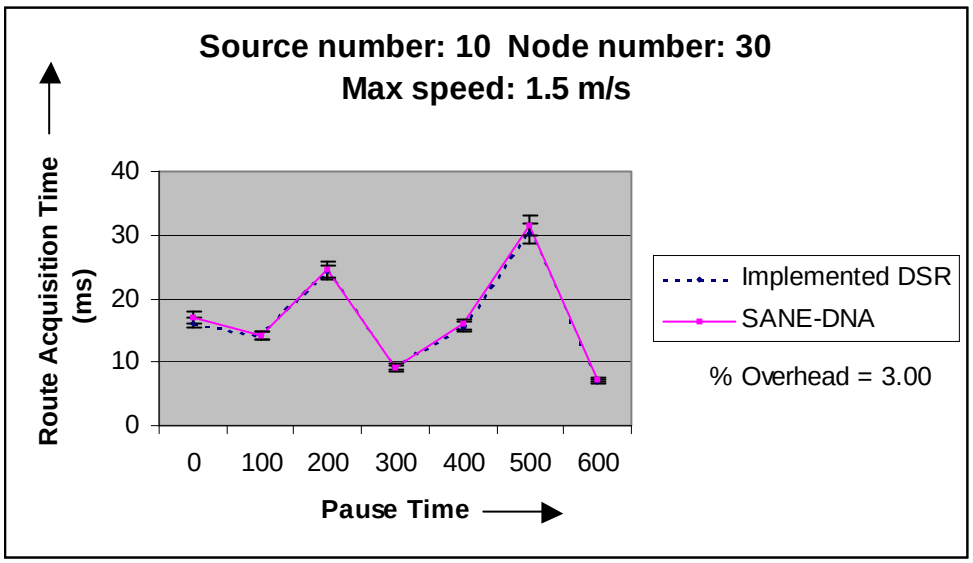



Figure (a)

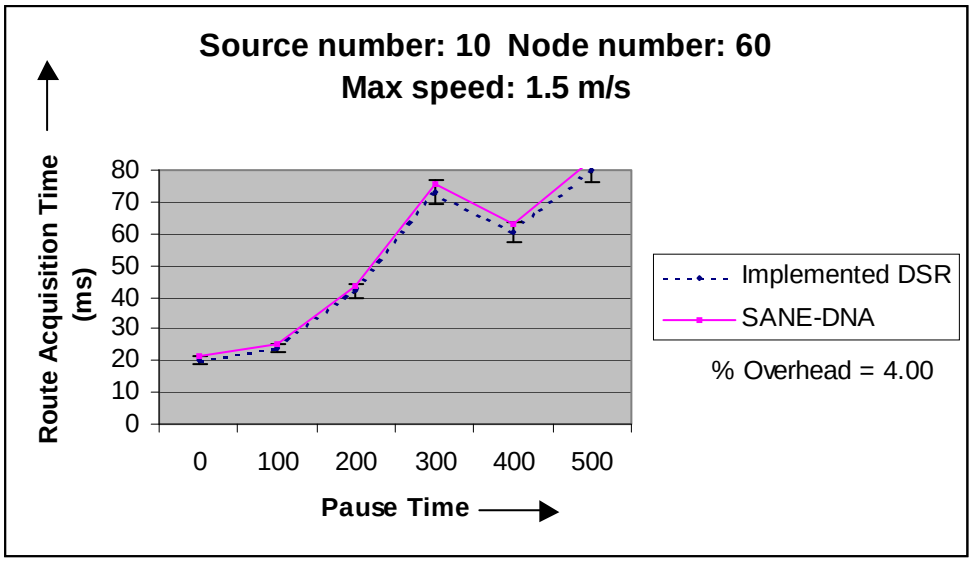



Figure (b)

**Figure 5.11.** Graphs showing the output of Route Acquisition Time for Implemented DSR vs SANE-DNA against different simulation Parameters

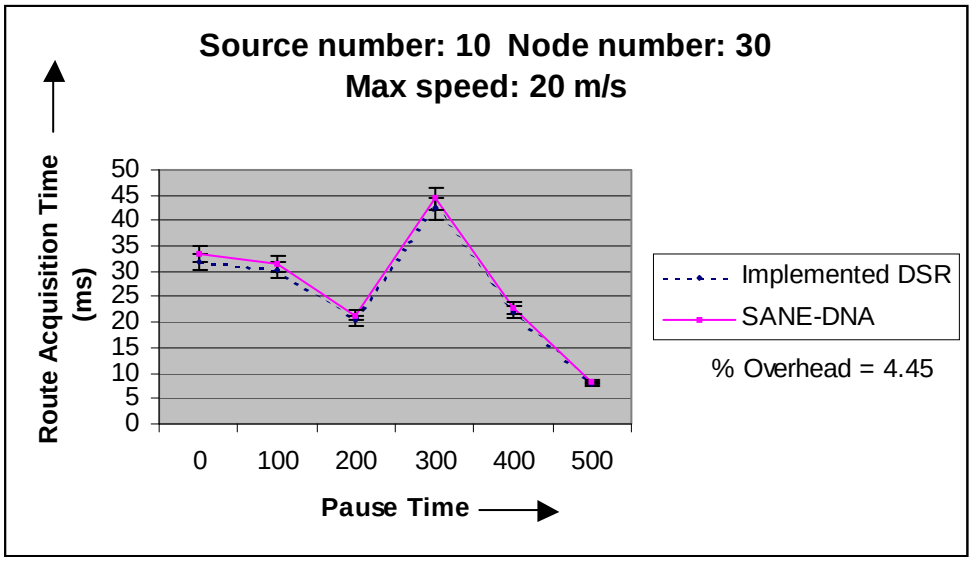



Figure (c)

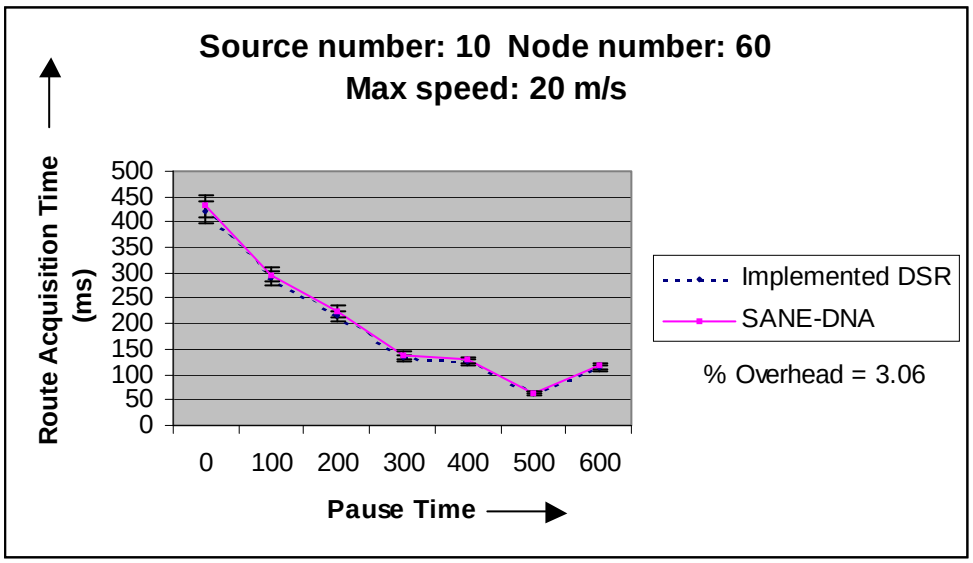



Figure (d)

**Figure 5.11.** Graphs showing the output of Route Acquisition Time for Implemented DSR vs SANE-DNA against different simulation Parameters

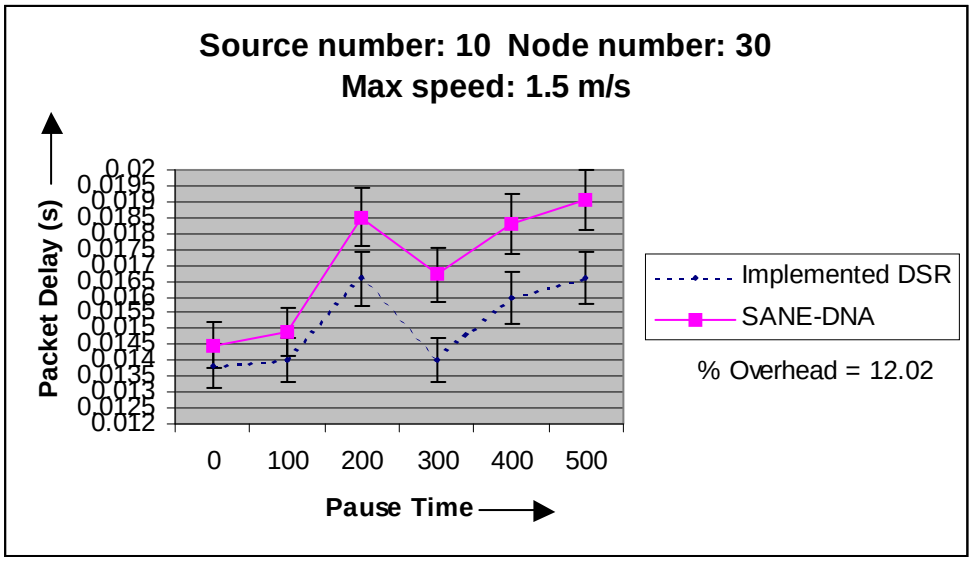



Figure (a)

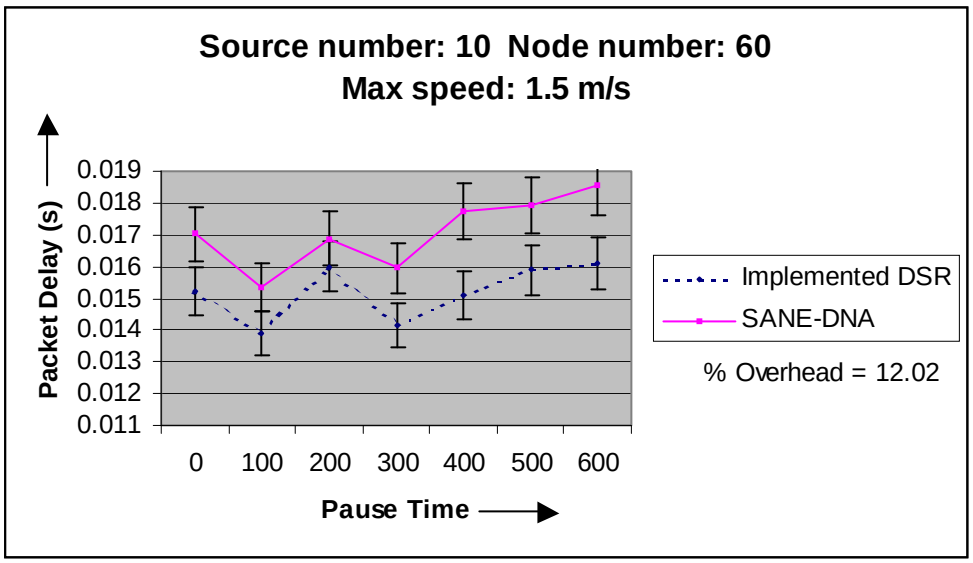



Figure (b)

**Figure 5.12.** Graphs showing the output of Packet Delay for Implemented DSR vs SANE-DNA against different simulation Parameters

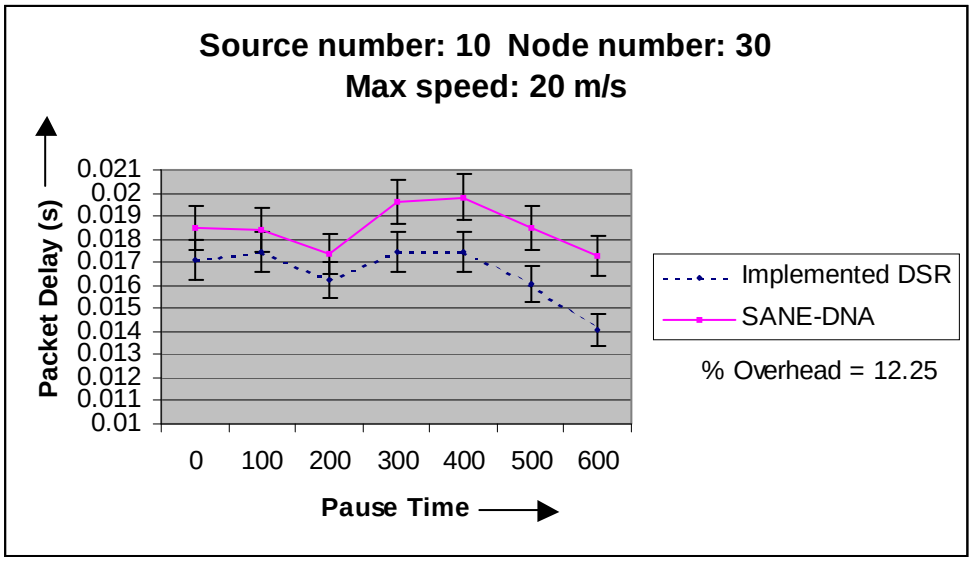



Figure (c)

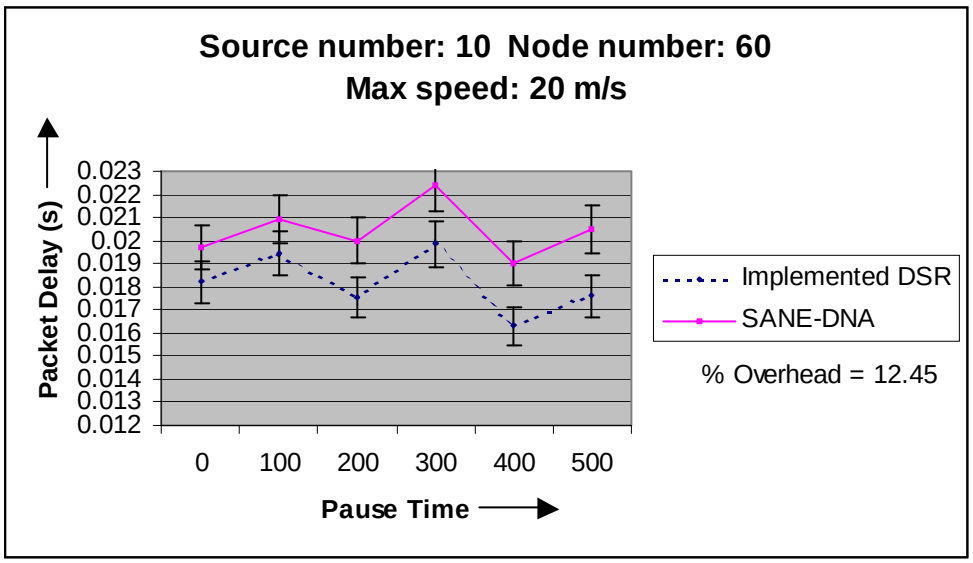



Figure (d)

**Figure 5.12.** Graphs showing the output of Packet Delay for Implemented DSR vs SANE-DNA against different simulation Parameters

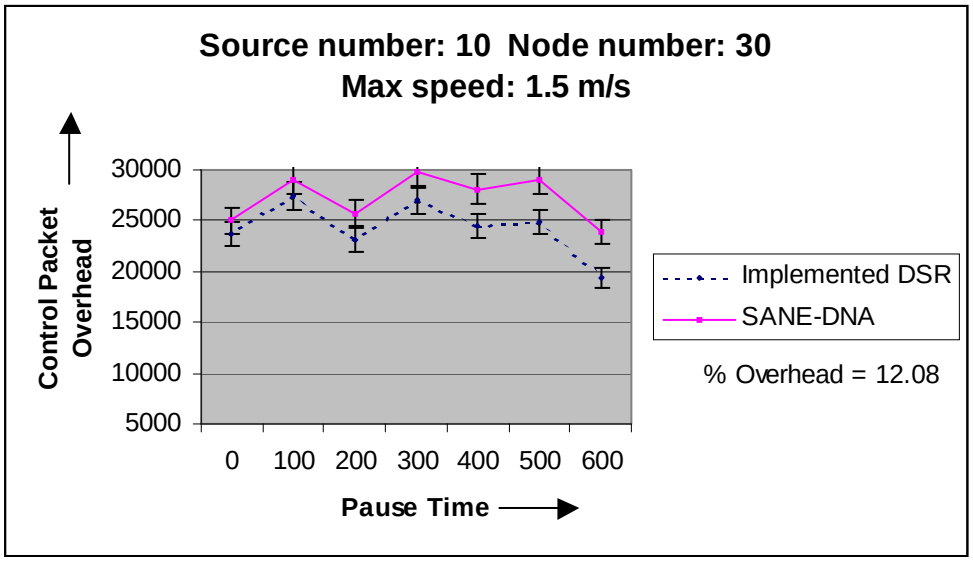



Figure (a)

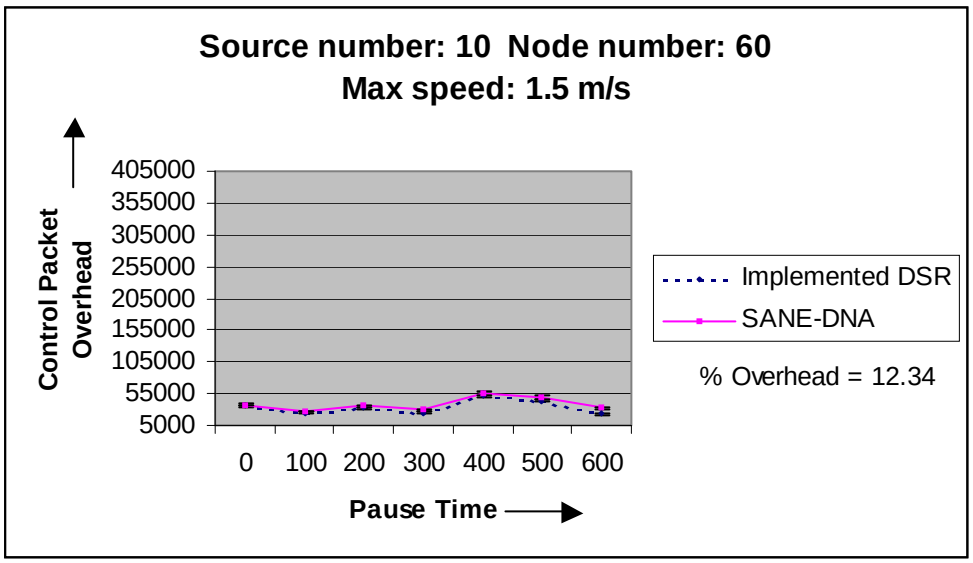



Figure (b)

**Figure 5.13.** Graphs showing the output of Routing Overhead for Implemented DSR vs SANE-DNA against different simulation Parameters

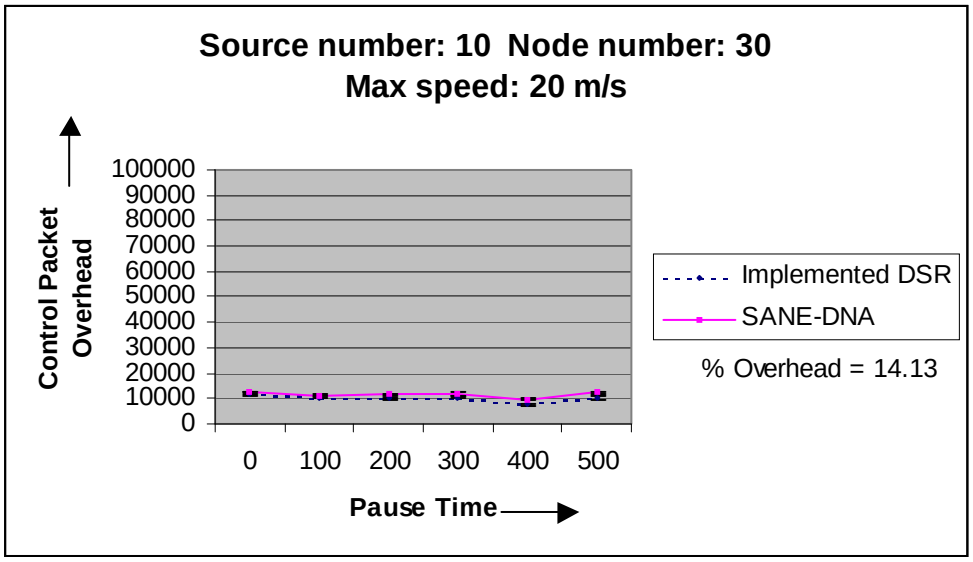



Figure (c)

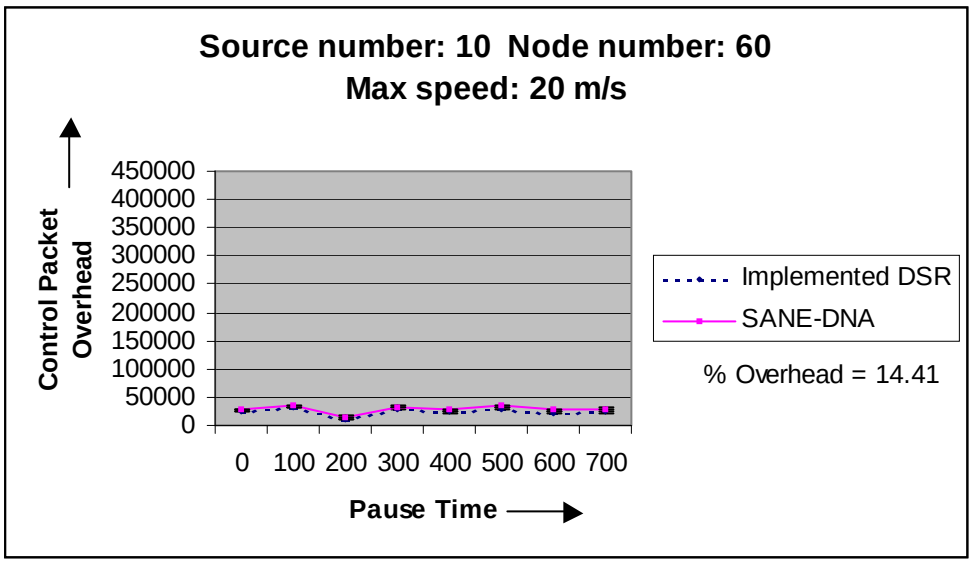



Figure (d)

**Figure 5.13.** Graphs showing the output of Routing Overhead for Implemented DSR vs SANE-DNA against different simulation Parameters

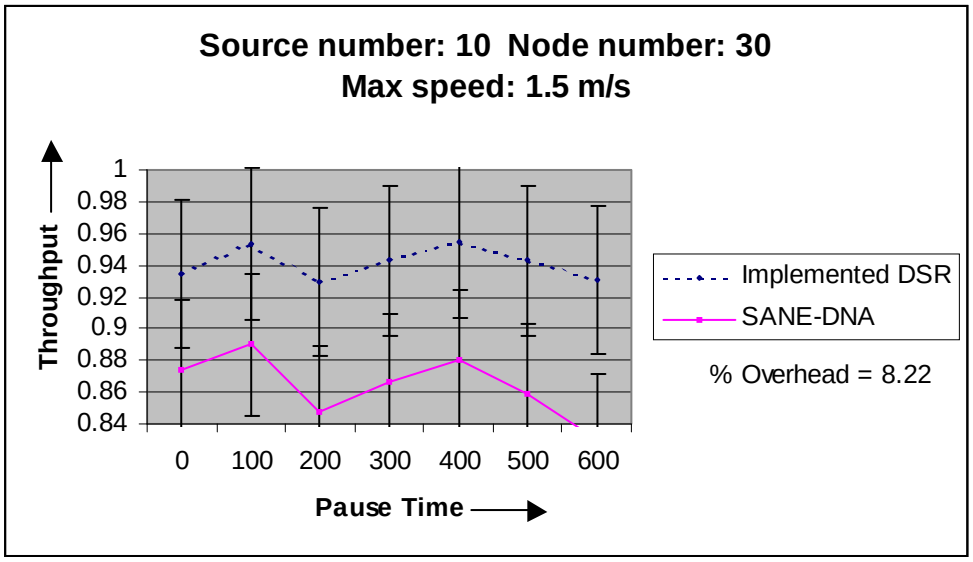



Figure (a)

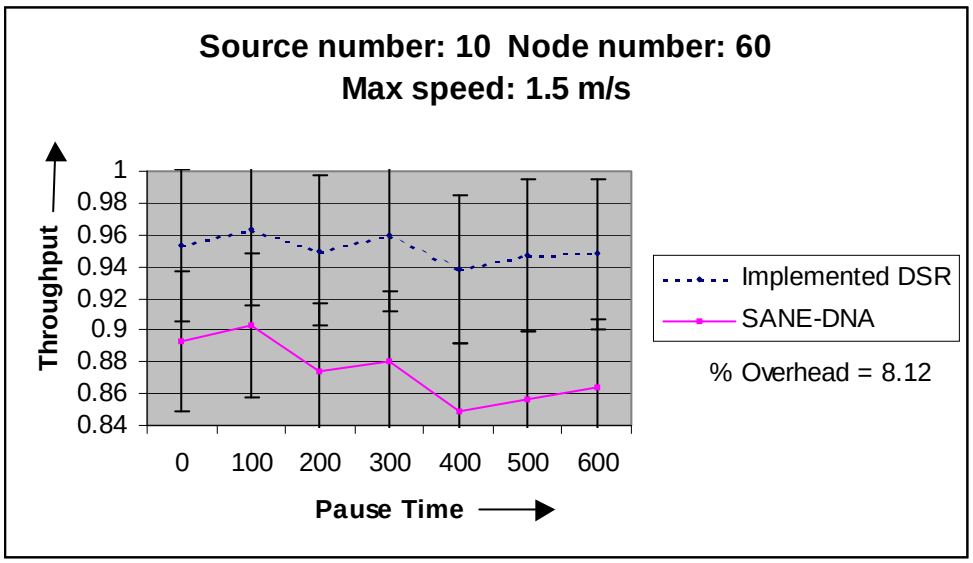



Figure (b)

**Figure 5.14.** Graphs showing the output of Throughput for Implemented DSR vs SANE-DNA against different simulation Parameters

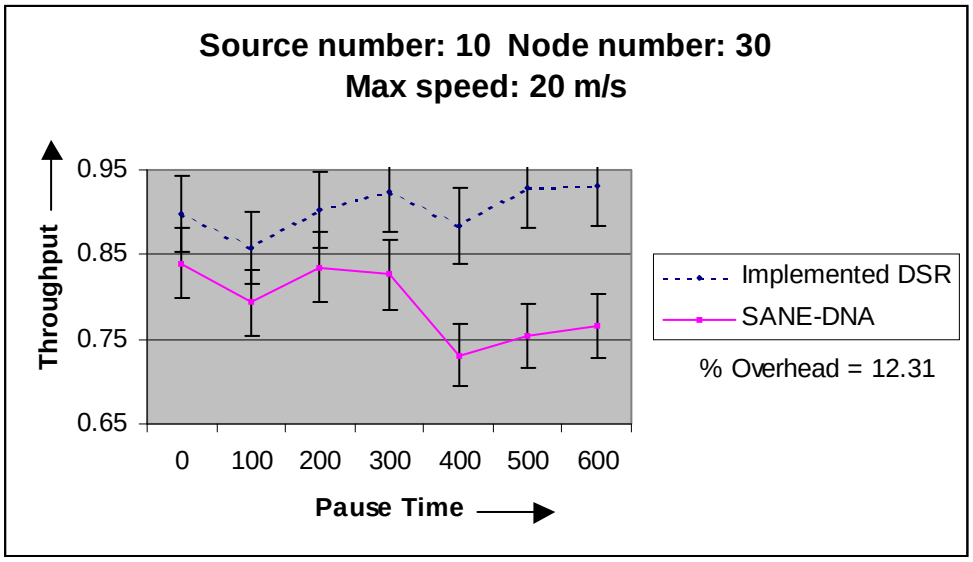



Figure (c)

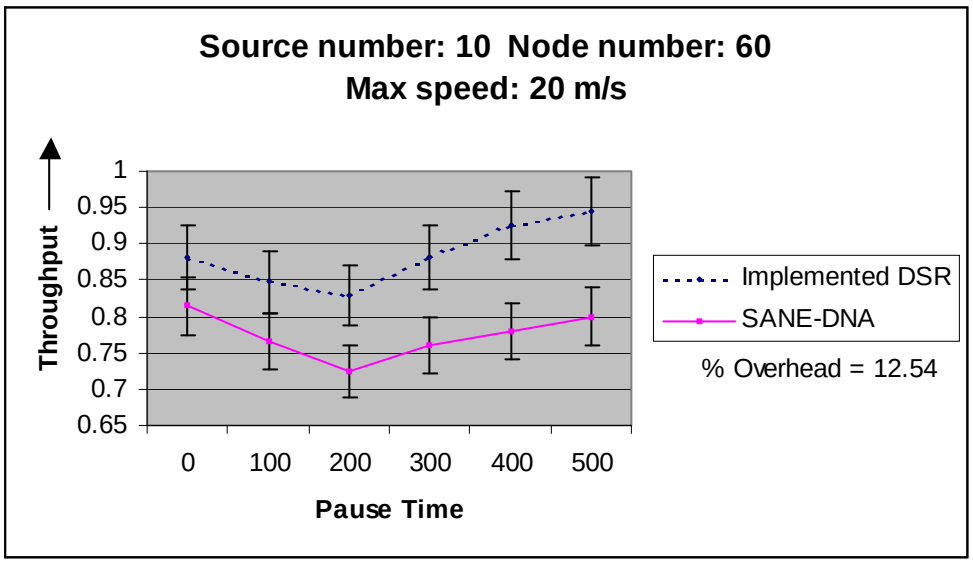



Figure (d)

**Figure 5.14.** Graphs showing the output of Throughput for Implemented DSR vs SANE-DNA against different simulation Parameters

**Table 5.3.** Simulation results (Route Acquisition Time)

| | ***Source = 10, Node = 30*** | | ***Source = 10, Node = 60*** | |
|---|---|---|---|---|
| | Correlation | %OH | Correlation | %OH |
| 1.5 m/s | 0.950301 | 3.00 | 0.964106 | 4.00 |
| 20 m/s | 0.943376 | 4.45 | 0.956537 | 3.06 |

**Table 5.4.** Simulation results (Packet Delay)

| | ***Source = 10, Node = 30*** | | ***Source = 10, Node = 60*** | |
|---|---|---|---|---|
| | Correlation | %OH | Correlation | %OH |
| 1.5 m/s | 0.965831 | 12.02 | 0.947452 | 12.02 |
| 20 m/s | 0.958239 | 12.25 | 0.949232 | 12.45 |

**Table 5.5.** Simulation results (Routing Overhead)

| | ***Source = 10, Node = 30*** | | ***Source = 10, Node = 60*** | |
|---|---|---|---|---|
| | Correlation | %OH | Correlation | %OH |
| 1.5 m/s | 0.936496 | 12.08 | 0.907034 | 12.34 |
| 20 m/s | 0.893108 | 14.13 | 0.902092 | 14.41 |

**Table 5.6.** Simulation results (Throughput)

| | ***Source = 10, Node = 30*** | | ***Source = 10, Node = 60*** | |
|---|---|---|---|---|
| | Correlation | %OH | Correlation | %OH |
| 1.5 m/s | 0.903989 | 8.22 | 0.880077 | 8.12 |
| 20 m/s | 0.939388 | 12.31 | 0.941592 | 12.54 |

## 5.8 Discussion

The MANET environment chosen for the above simulation is rectangular shaped in order to force the use of longer routes between the nodes than would occur in a square space with equal node density.

The coefficient of correlation has been shown in each simulation for different metrics indicating the behavior of the implemented DSR. The overall coefficient value comes out to be 0.958, which shows that the implemented DSR is in close proximity with the published DSR protocol. Thus, validating the Tcl/Tk script of the implemented DSR.

When the SANE-DNA protocol is simulated over the implemented DSR, the percentage overhead is calculated in order to study the behavior of SANE-DNA protocol. The percentage overhead is defined as:

$$\%Metric_{Overhead} = \left| \frac{Metric_{SANE-DNA} - Metric_{ImplementedDSR}}{Metric_{ImplementedDSR}} \right| \times 100$$

Where, Metric can be - Route Acquisition Time, Packet Delay, Routing Overhead Throughput, as the case may be.

The simulation result leads us to the following observations:

- The Route Acquisition Time$_{overhead}$ is increased, indicating minimal interference of the proposed protocol, SANE-DNA with the normal functioning of the DSR protocol (3% to 4%).
- The End-to-End Delay$_{overhead}$ of the SANE-DNA is increased over the DSR protocol due to the extra time required to communicate the public key and the encryption and decryption process taking place at the source and destination nodes (around 12%).
- A Routing Overhead$_{overhead}$ of the SANE-DNA protocol increases because of the extra information being sent as encrypted message (12% to 14%).
- The Throughput$_{overhead}$ of the SANE-DNA and the DSR protocol is almost constant, ranging from 8% to 12.5% because the number of packets lost while reaching the destination is almost same in both the cases.

Now, let us observe the behavior of SANE-DNA with other security goals, considering them one by one, as listed below:

- Impersonation

  DNA cryptography does not allow the attacker to do impersonation as before transmitting the data a secure channel is established between the source and destination that itself implies the destination before hand knows from whom he is going to get the data.

- Modification

  Before transmitting the data a secure channel is established between the source and destination this protects the data to be modified by anyone. Before sending the data it is encrypted by using the public key of the sender and the data can be only modified if the destination‘s private key is compromised.

- Fabrication

  DNA cryptography algorithm does not allow the message to be fabricated. Destination can also choose to accept or reject the packet based on the prior behavior of the node. The node, which injects the invalid routing packets, can also be penalized for routing misbehavior [113,114], as done in the Watchdog mechanism.

- Wormhole Attack

  DNA cryptography algorithm takes care of the packet security as the transmission can only start if the secure channel has been established, as the sender generally knows the route before the transmission starts. So the sender may also implement a kind of feedback mechanism to know about which packets are successfully delivered.

- Denial of Service

  DNA cryptography algorithm also limits the denial of service attack to the minimum as the destination can limit its number of connections as it can accept or reject the connections before establishing any route with any node. So, it can limit the connections depending upon the resources available with the node.

## 5.9 Conclusion

The proposed SANE-DNA protocol has been verified and validated through simulation by synthetically generated data sets. The confidence factor of the entire simulation process is 0.95. The DSR can be secured with marginal overhead within tolerable limits.

# Chapter 6

# Conclusion and Scope for Future Work

## 6.1 Conclusions

Wireless mobile ad hoc networks present difficult challenges to routing protocol designers. Mobility, constrained bandwidth, and limited power cause frequent topology changes. It also has its share of security vulnerabilities. The very basic nature of the mode of communication is the main concern because anything that moves over the open air medium is susceptible to be picked up by unauthorized access. For any mission critical or organizationally sensitive information, ad hoc networks add an element of insecurity. The most important and vital element is to route the information among the network in a secured manner. Secure routing [119] in ad hoc networks is the main focus of our research. DNA cryptography has been proposed and simulated for slow moving nodes (1.5m/s) and fast moving nodes (20 m/s) along with the variations in number of nodes. We summarize our contributions are as follows:

1. We conducted a performance evaluation of various routing protocols of different types, mainly focusing on the flat-routing protocols. The routing protocols were analyzed in diverse network scenario to assess their relative strength and weaknesses. Our results provided meaningful indications to protocol designers in this area.

2. We investigated the effect of various mobility models on working of different flat-routing protocols. It is determined that the choice of mobility model does, in fact, affect the relative performance of different routing protocols.

3. We performed simulation of the DSR protocol. Our study results indicate that DSR may be considered as one of the best routing protocol for providing secure routing because there are no periodic beacons, thus resulting in a lesser overhead during communication.

4. We introduced a novel secure routing protocol, termed as **SANE-DNA.** The proposed protocol is based upon pseudo DNA cryptography method using one-time-pads. The scheme has been illustrated for a DSR protocol and could easily be adopted for other on-demand routing protocols for providing integrity, non-repudiation and confidentiality. The proposed algorithm has been evaluated with different network parameters under a simulated environment. As per our knowledge, this is the first reported work for securing ad hoc networks using pseudo DNA cryptography. We are also exploring the possibility of submitting it to IETF MANET working group.

5. To establish the effectiveness of the proposed protocol, the same has also been compared with existing secure routing protocols, as shown in Table 6.1.

**Table 6.1** Evaluation of SANE-DNA

| **Performance Parameters** | **ARIDANE** | **ARAN** | **SEED** | **SRP** | **SAODV** | **SAR** | **SLSP** | **SANE-DNA** |
|---|---|---|---|---|---|---|---|---|
| **Type** | Reactive | Reactive | Proactive | Reactive | Reactive | Reactive | Proactive | Reactive |
| **MANET Protocol** | DSR | DSR/AODV | DSDV | DSR/ZRP | AOD V | AODV | ZHLS | DSR |
| **Encryption** | Sym | Asym | Sym | Sym | Asym | Sym/Asym | Asym | Sym |
| **Synchronization** | Yes | No | Yes | No | No | No | No | Yes |
| **Trust Authority** | KDC | CA | CA | CA | CA | CA/KDC | CA/KDC | No |
| **Authentication** | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Confidentiality** | No | Yes | No | No | No | Yes | No | Yes |
| **Integrity** | Yes | Yes | No | Yes | Yes | Yes | No | Yes |
| **Non-repudiations** | No | Yes | No | No | Yes | Yes | Yes | Yes |
| **Anti-Spoofing** | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes |
| **DoS Attack** | Yes | No | Yes | Yes | No | No | Yes | Yes |

## 6.2 Scope for Future Research

This work opens new avenues for future research. The research can be extended in several directions and some of them are summarized below:

1. The proposed secure routing protocol, SANE-DNA, is simulated under different conditions of the MANET. We have only performed the simulation studies. It would be interesting to note the performance of the proposed protocol for different parameters such as - average number of edges, average link down time, average density, average number of link change route, average length of path etc. Further, in order to have a better understanding of the strategy it can be implemented on the real-world test bed, such as - WHYNET, established in south California. WHYNET is a multi-institutional effort to build an interconnected test bed of wireless networks for use by the academic community.

2. The proposed secure routing protocol, SANE-DNA, is implemented on a reactive routing protocol, the DSR. The DNA cryptography algorithm can further be extended on other categories of routing protocols such as- proactive routing protocols, hybrid routing protocols, geographical routing protocols, power-aware routing protocols etc.

3. The random way point mobility model (RWPMM) is used in the presented work to obtain simulation results by varying different network parameters. The RWPMM can further be improved upon its accuracy through steady-state initialization and the results can then be compared. Further, the mobility model used in the simulation is RWPMM, which is suitable for a certain scenario. It would be interesting to note the behavior of the proposed secure routing protocol as simulated by different mobility model such as- Random walk, Random point group mobility model or the Manhattan model, which are useful for simulating other real-world scenarios.

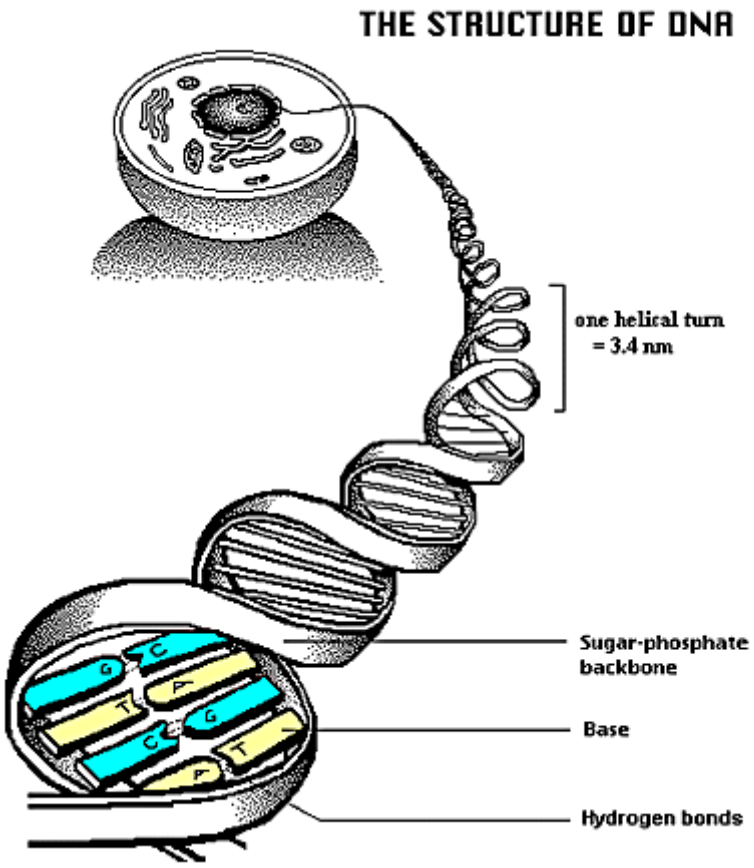
4. It is a common belief and proven fact that the formal systems are helpful in verifying and validating the security mechanisms for any secure routing

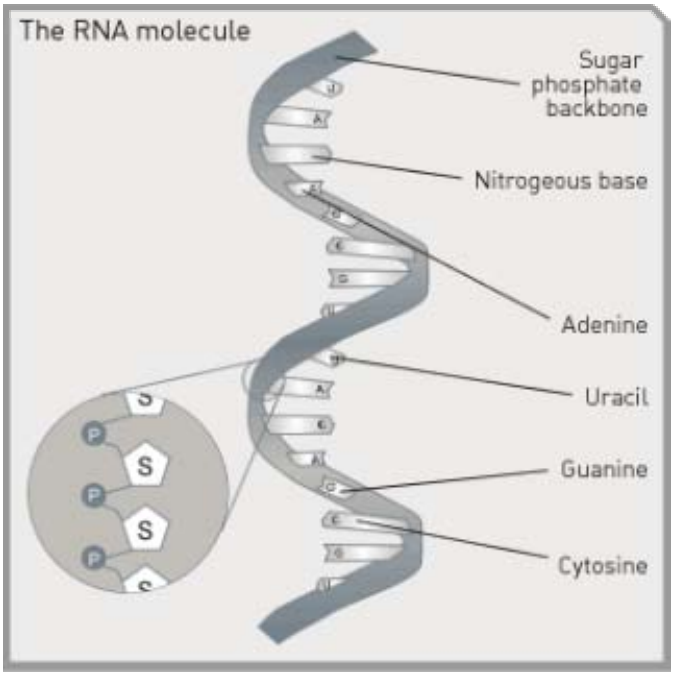
protocol. The same can be applied on the presented work by using - Verisim or SPIN Model Checker or $S^3A$ (Spi calculus Specification Symbolic Analyser) or BAN logic, to name a few.

5. In the presented work, the selfish nodes are not dealt with; it would be interesting to note the behavior of a routing protocol capable of handling both selfish and malicious nodes using DNA cryptography.

## *APPENDIX- A*

**Terminologies used in DNA Cryptography** [133]

| | |
|---|---|
| DNA | Revolution in life science started with the discovery of structure of DNA [131] in 1953 by Watson and Crick. DNA has blueprint of all life forms. It contains the genetic instructions specifying biological developments.   |

| | |
|---|---|
| RNA | RNA [132] serves as the template for translation of genes into proteins, transferring amino acids to the ribosome to form proteins and also translating the transcript into proteins. There are three forms of RNA namely, mRNA , tRNA and rRNA.<br>  <br> |
| mRNA | Messenger RNA<br>Carries information from DNA to the ribosome sites of protein synthesis in the cell. |
| rRNA | Ribosomal RNA<br>Is a component of the ribosomes , the protein synthetic factories in the cell. |
| tRNA | Transfer DNA<br>Is a small RNA chain of about 74-93 nucleotide that transfers a specific amino acid to a growing chain. |
| Codon | The gene sequence inscribed in DNA , and in RNA , is composed of tri-nucleotide units called codons.<br>e.g. the string UUUAAACCC contains the codons UUU, AAA and CCC .Each of which specifies one amino acid. Therefore it |

| | |
|---|---|
| | represents a protein sequence , three amino acids long. |
| Oligonucleotide | Are short sequences of nucleotides (DNA/RNA) typically with twenty or fewer base. |
| Purine | Adenine and Guanine are called Purine. |
| Pyrimidine | Cytosine and Thymine are called Pyrimidine. |
| Polymerase | Is an enzyme that assists in DNA replication. It acts as a catalyst for the production of existing DNA or RNA |
| Lexicon | Is a list of words together in the additional word-specific information i.e. a dictionary. |
| Hamming Distance | Between two strings of equal length is the number of positions for which the corresponding symbols are different.<br>e.g. For the binary strings 1011101 and 1001001 the Hamming distance is two. |
| Introns | Introns are sections of DNA colinear to the RNA sequence that will be spliced out after transcription, but before the RNA is translated. |
| Exons | An exon is any region of DNA within a gene that is transcribed to the final messenger RNA (mRNA) molecule. |
| Adenine | Adenine is one of the two purine nucleobases used in forming nucleotides of the nucleic acids. In DNA, adenine binds to thymine via two hydrogen bonds to assist in stabilizing the nucleic acid structures. In RNA, which is used in the cytoplasm for protein synthesis, adenine binds to uracil. |
| Guanine | guanine is a derivative of purine, consisting of a fused pyrimidine-imidazole ring system with conjugated double bonds. |
| Cytosine | Cytosine is one of the 5 main nucleobases used in storing and transporting genetic information within a cell in the nucleic acids DNA and RNA. It is a pyrimidine derivative. |
| Thymine | Thymine, also known as 5-methyluracil, is a pyrimidine nucleobase. In RNA thymine is replaced with uracil in most cases.In DNA, thymine(T) binds to adenine (A) via two |

| | |
|---|---|
| | hydrogen bonds to assist in stabilizing the nucleic acid structures. |
| Uracil | Uracil is a pyrimidine which is common and naturally occurring.Found in RNA, it base pairs with adenine and is replaced by thymine in DNA. |

# *APPENDIX-B*

## 1. Source Side Procedures

### 1.1 Main Procedure

- *Char * procedure_sourceside_main(char * buffer,char * publickey)*

*1.*

*start // start of procedure_sourceside_main();*

*2.*

*declare char dna[100], mrna[100], keys[100],protein [100], enc_data[100];*

*// declare strings to store the intermediate representations.*

*3.*

*dna = procedure_convert_data_into_dna_form (buffer);*

*// this procedure will convert the data stored in the buffer to the dna form.*

*4.*

*mrna= procedure_convert_dna_into_mrna_form (dna);*

*//this procedure will convert the dna form to mrna form*

*5.*

*keys= generate_keys ( );// get the public key by establishing a secure channel*

*6.*

*protein= procedure_convert_mrna_into_protein_form (mrna,keys);*

*//this procedure will convert the mrna form into the protein form by employing the*

*//transcription and translation process.*

*7.*

*enc_data=procedure_encrypt_protein_form(protein,publickey);*

*//this procedure will convert the protein form into the encrypted data form by using*

*//the public key of the sender.*

*8.*

*return enc_data;*

*9.*

end *// end of procedure_sourceside_main();*

**1.2 Child Procedures**

- *char * procedure_convert_data_into_dna_form (char * buffer)*

*//used to convert data into the dna form by using (A for 00, C for 01, G for 10, T for 11)*

- *char * procedure_convert_dna_into_mrna_form (char * dna)*

*//used to convert the dna form to mrna form, as the source knows the starting //codes (codes that indicate the begin of the intron) and pattern codes (codes that //define which parts of the frame to be removed, and which parts to be kept) of the //introns, so the source knows where are the introns in the DNA form of the //information and which parts should be removed. This procedure scans the DNA //form of information to find out the introns and records the introns places, and //cut out the introns according to the specified pattern.*

- *char * generate_keys (void )*

//used to generate the keys which are the starting and pattern codes of the introns, //the places of the introns, the removed spaced introns, and the codon-amino acids //mapping of the protein.

- *char * procedure_convert_mrna_into_protein_form (char * mrna,char * keys )*

*//used to convert the mrna formmof information to the protein formusing the keys.*

- *char * procedure_encrypt_protein_form(char * protein, char *publickey)*

*//used to encrypt the data i.e. the protein form and the keys that are to be send to //the destination side through a secure channel.*

## 2. Destination Side Procedures

### 2.1 Main Procedure

- *Char * procedure_destination side_main (char * buffer, char * keys, char * privatekey)*

*1.*

*start  // start of procedure_sourceside_main();*

*2.*

*declare char dna[100], mrna[100], protein[100], received_data[100];*

*// declare strings to store the intermediate representations.*

*3.*

*protein=procedure_decrypt_data (buffer,privatekey);*

*4.*

mrna=procedure_convert_data_into_mrna_form(protein,keys);

*5.*

*dna =  procedure_convert_data_into_dna_form (mrna);*

6.

received_data = procedure_convert_into_digital_form(dna);

7.

return received_data;

8.

*end // end of procedure_sourceside_main();*

**2.2 Child Procedures**

- *char * procedure_decrypt_data(char * buffer,char * privatekey)*

//used to convert the data received form the source into its original protein form by //decrypting it with private key , if the server has send the data by encrypting it with //the //public key.

- *char * procedure_convert_data_into_mrna_form(char * protein, char * keys)*

//used to convert the protein form of data into the mrna form using the keys send by //the //source.

- *char * procedure_convert_data_into_dna_form (char * mrna);*

*// used to convert the mrna form of data into the dna form by using the //information where are the introns in the information.*

- *char * procedure_convert_into_digital_form(char * dna);*

//used to convert the dna form of information into the digital form consisting of 1's //and 0's.

# REFERENCES


[1] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," *In Proc. IEEE*. 75(1), pages 56–73, January 1987.

[2] A. K. Verma, Mayank Dave and R C Joshi, "Genetic Algorithm and Tabu Search Attack on the Mono-Alphabetic Substitution Cipher in Ad-hoc Networks," *International J. of Computer Science, NY (USA),* 3(3), 134-137, 2007.

[3] A. K. Verma, Mayank Dave and R C Joshi, "Secure Data Sharing in Mobile Adhoc Networks," *J. International Review on Computers and Software (IRECOS), ISSN 1828-6003* (Peer reviewed and accepted).

[4] A. K. Verma, Mayank Dave and R C Joshi, "Secure Routing in Mobile Networks: A Review," *International J. of Systemics, Cybernetics and Informatics (IJSCI), ISSN 0973-4864* (Peer reviewed and accepted).

[5] A. K. Verma, Mayank Dave and R C Joshi, "Applying Distributive Computing In Mobile Ad hoc Networks (MANETs)," peer reviewed and accepted for publication in *J. of Punjab Academy of Sciences.*

[6] A. Law and W. Kelton, "Simulation Modeling and Analysis," *McGraw-Hill*, 2000.

[7] A. Menezes, P.Oorschot and S. Vanstone, "Handbook of Applied Cryptography," *CRC Press*, 1996.

[8] A. Perrig, R. Canetti, D. Song and J. D. Tygar, "Efficient and Secure Source Authentication for Multicast," *In Proc. of NDSS 2001.*

[9] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," *In Proc. of IEEE Symposium on Security and Privacy,* 2000.

[10] A. Perrig, Y-C Hu, and D. B. Johnson, "Wormhole Protection in Wireless Ad Hoc Networks," *Technical Report TR01-384*, Dept. of Computer Science, Rice University.

[11] A. Perrig, R. Canetti, "The TESLA Broadcast Authentication Protocol", Internet Draft, 2000.

[12] A. Salomaa, "Public-Key Cryptography," *Springer-Verlag*, 1996.

[13] A. Shamir, "How to Share a Secret," *Communications of ACM,* 1979.

[14] A. Tenenbaum, "Computer Networks," *PH PTR*, 2003.

[15] Abdul Sahid Khan, Madhavan Mukund, and S.P. Suresh, "Genetic verification of security protocols", *In Proc. of SPIN'05*, Springer LNCS 3639, pages 221-235, August 2005.

[16] Asad Amir Prizada and Chris McDonald, "Secure Routing Protocols for Mobile Ad hoc Wireless Networks," *In Proc. 2nd Workshop on Internet, Telecommunication and Signal Processing," WITSP* 2003, Australia, pages 57-80, 2003.

[17] Ashish Gehani, Thomas LaBean and John Reif "DNA-Based Cryptography," *In Proc. 5th DIMACS workshop on DNA Based Computers,* MIT Cambridge, 1999.

[18] B. Anand and S. Papavassiliou, "A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks," *International J. of Network Management*, vol. 11, pages 387-395, 2001.

[19] B. Bellur and R. G. Ogier, "A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks," *In Proc. IEEE INFOCOM '99*, New York, March 1999 .

[20] B. Dahill, B. N. Levine, E. Royer, and C. Shields, "ARAN: A secure Routing Protocol for Ad Hoc Networks," *UMass Tech Report 02-32*, 2002.

[21] B. K R. Yahalom and T.Beth (1994). "Trust relationships in secure systems- a distributed authentication perspective" *Computer System,* 7 (1), pages 45-73.

[22] B. S. Bakshi, P. Krishna, D. K. Pradhan, and N. H. Vaidya, "Improving Performance of TCP over Wireless Networks," in *Proceedings of International Conference on Distributed Computing Systems,* May 1997.

[23] B. Schneier, *Applied Cryptography,*Wiley, 1996.

[24] B. Wu, J. Wu, E.B. Fernandez, M. Ilyas, and S. Mangliveras, "Secure and Efficient Key Management in Mobile Ad-hoc Networks," *J. of Network and Computer Applications*, 2005.

[25] B. Wu and J. Wu, "A Survey of Key Management in Mobile Ad Hoc Networks," *Handbook of Research on Wireless Security,* J. Zheng, Y. Zhang, and M. Ma (Eds.), Idea Group Inc. 2007.

[26] B. Wu and J. Wu, "A Survey on Attacks and Countermeasures in Mobile Ad hoc networks," *Wireless/Mobile Network Security,* Y. Xiao, X. Shen, and D.Z. Du(Eds.), Springer 2006.

[27] Bdale Garbee, "Thoughts on the Issues of Address Resolution and Routing in Amateur Packet Radio TCP/IP Networks," *In Proc. ARRL Amateur*

*Radio 6th Computer Networking Conference*, pages 56–58. American Radio Relay League, August 1987.

[28] Bin Xie, Anup Kumar, Dharma Agrawal and S. Srinivasan, "Secured Macro/Micro-Mobility Protocol for Multi-hop Cellular IP," *J. of Pervasive and Mobile Computing*, Volume 2, No. 2, pages 111-136, 2006.

[29] Borujeni, S.E., "Cryptography by pseudo random number generator", *In Proc. First IEEE symposium on Intelligent systems*, pp.244-247, 2002.

[30] Bussard Laurent, Refik Molva,"Establishing trust with privacy," *12th International Workshop on Security Protocols*, LNCS Volume 3957, pages 199-209, April 2004.

[31] C. E. Perkins and E. M. Royer, "The Ad Hoc On-Demand Distance-Vector Protocol (AODV)," In Ad Hoc Networking, C. E. Perkins (Ed.), pages. 173–219, Addison-Wesley, 2001.

[32] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *In Proc. of SIGCOMM 1994.*

[33] C. E. Perkins, "Ad Hoc Networking," *Addison-Wesley Longman,* 2000.

[34] C. K. Toh, "A Novel Distributed Routing Protocol to Support Ad-Hoc Mobile Computing," *In Proc. of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications*, pages 480–486, March 1996.

[35] C. K. Toh, "Ad Hoc Mobile Wireless Networks: Protocols and Systems," *Prentice Hall Englewood Cliff,* NJ 07632,2002.

[36] C. Murthy and B. Manoj, "Ad hoc Wireless Networks: Architectures and Protocols," *Prentice Hall PTR*, 2005.

[37] C. Perkins, "Ad hoc On Demand Distance Vector (AODV) Routing," *Internet draft, draft-ietfmanet- aodv-00.txt.*

[38] C´edric Adjih, Thomas Clausen, Anis Laouiti, Paul M¨uhlethaler, and Daniele Raffo. "Securing the OLSR Routing Protocol With or Without Compromised Nodes in the Network," *Technical Report INRIA RR- 5494* HIPERCOM project, INRIA Rocquencourt, February 2005.

[39] Carlos Cordeiro and Dharma P Agrawal, "Adhoc and Sensor Networks, Theory and Applications, " *World Scientific Publishing,* 2006.

[40] Charles E. Perkins and Elizabeth M. Royer, "Ad-Hoc On Demand Distance Vector Routing," *In Proc. 2nd IEEE Workshop on Mobile Computing*

*Systems and Applications*, pages 90–100, IEEE Computer Society, February 1999.

[41] Claude E. Shannon, ”Communication Theory of Secrecy Systems,” *J. Bell System Technical*, vol.28-4, pages 656--715, 1949.

[42] CMU Monarch Group, “CMU Monarch extensions to the NS-2 simulator,” http://monarch.cs.cmu.edu/cmu-ns.html

[43] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing,” in *Ad Hoc Wireless Networks, Mobile Computing,* T. Imielinski and H. Korth (Eds.), Chapter 5, pp. 153–181, Kluwer Academic Publishers, 1996.

[44] D. B. Johnson, D. A. Maltz, and J. Broch, “DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks,” *In Ad Hoc Networking, C. E. Perkins (Ed.)*, pages 139–172, Addison-Wesley, 2001.

[45] D. Boneh, C. Dunworth and R. Lipton, "Breaking DES Using a Molecular Computer", *Technical report, Department of Computer Science*, Princeton University, 1995.

[46] D. P. Agrawal and Qing-An Zeng, “Introduction to wireless and Mobile Systems,” *Brooks/Cole, 2005.*

[47] D. De Couto, J. Li, C. Blake, H. Lee, and R. Morris, “Capacity of Wireless Ad Hoc Wireless Networks,” in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom’01),* Rome, Italy, pp. 61–69, July 2001.

[48] Daniel B. Faria and David R. Cheriton, “DoS and Authentication in Wireless Public Access Networks,” *In Proc. of the 1st ACM Workshop on Wireless Security”* (WiSe’02), September 2002.

[49] Daniel M. Frank, “Transmission of IP Datagrams over NET/ROM Networks,” *In ARRL Amateur Radio 7th Computer Networking Conference*, pages 65–70. American Radio Relay League, October 1988.

[50] David A. Maltz, Josh Broch and David B. Johnson, “Experiences Designing and Building a Multi-Hop Wireless Ad Hoc Network Testbed,” *Technical Report CMU-CS-99-116, School of Computer Science*, Carnegie Mellon University, Pittsburgh, Pennsylvania, March 1999.

[51] David A. Maltz, Josh Broch, and David B. Johnson, “Lessons From a Full-Scale MultiHop Wireless Ad Hoc Network Testbed,” *IEEE Personal Communications*, 8(1): pages 8-15, February 2001.

[52] David B. Johnson and David A. Maltz, “Dynamic source routing in ad hoc wireless networks,” *In Mobile Computing,* Imielinski and Korth (Eds), chapter 5, pages153–181, Kluwer Academic Publishers, 1996.

[53] David B. Johnson, "Validation of Wireless and Mobile Network Models and Simulation," *In Proc. DARPA/NIST Workshop on Validation of Large-Scale Network Models and Simulation* Fairfax, VA, May 1999.

[54] Donald Nixon, "DNA and DNA Computing in Security Practices – Is the Future in Our Genes?*" GSEC Assignment Version 1.3*.

[55] Fan Bai and Ahmed Helmy "A survey of mobility models in Wireless Ad hoc networks," *http://nile.usc.edu/important/chapter1.pdf* ,last accessed on February 15,2007.

[56] Galvin Holland and Nitin Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," *In Proc. 5th International Conference on Mobile Computing and Networking (MobiCom'99)*, pages 219–230. ACM, August 1999.

[57] Gheorghe Paun, "Computing with Bio-Molecules (Theory and Experiments)," *Springer- Verlag Singapore Pte. Ltd*, May 1998.

[58] Gunes, M., Sorges, U., Bouazisi, I., "ARA – The Ant-Colony Based Routing Algorithm for MANETs," Proc. ICPP Workshop on ad Hoc Networks (IWAHN 2002), IEEE Computer Society Press (2002), 79-85.

[59] H. Luo and S. Lu, "Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks," *UCLA-CSD-TR-200030*

[60] H. Matsuo and K. Mori, "Accelerated Ants Routing in Dynamic Networks," in Proc. Intl. Conf. On Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Pp. 333-339, Aug.2001.

[61] I. Chlamtac and J. Redi, "Mobile Computing: Challenges and Opportunities," *In Encyclopedia of Computer Science, 4th Edn,*D. Hemmendinger, A. Ralston, and E. Reilly (Eds), International Thomson Publishing, 1998.

[62] I. Chlamtac, M. Conti, and J. Liu, "Mobile Ad Hoc Networking: Imperatives and Challenges," *J. Ad Hoc Networks*, vol. 1, no. 1, pages13 – 64, 2003.

[63] IEEE 802.11b-1999 Supplement to 802.11-1999,*Wireless LAN MAC and PHY specifications*: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band.

[64] Internet Engineering Task Force (IETF) www.ietf.org

[65] Issa Khalil, Saurabh Bagchi, Ness B. Shroff, " MOBIWORP: Mitigation of the wormhole attack in mobile multihop wireless networks ", *to appear J. Ad Hoc Networks (Elsevier),* available February 3,2007.

[66] Ivan Howitt, Wayne Manges, Teja Kuruganti, Glenn Allgood, Jose Gutierrez, James M. Conrad, "Wireless Industrial Sensor Networks: Framework for QoS Assessment and QoS Management," *Transactions of the Instrumentation, Systems, and Automation Society (ISA),* 45(3), July 2006.

[67] Ivan Stojmenovic and Xu Lin, "Power aware localized routing in wireless networks," *IEEE Transactions on Parallel and Distributed Systems***,** Vol. 12, No. 11,pages 1122-1133, November 2001.

[68] Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, pages.1369–1379, Aug. 1999.

[69] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols," *In Proc. 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobi-Com'98)*, pages 85–97, 1998.

[70] James Heather, Gavin Lowe, and Steve Schneider, " How to Prevent Type Flaw Attacks on Security Protocols," *In Proc. of the 13th IEEE Computer Security Foundations Workshop (CSFW 13)*, pages 255-268, July 2000.

[71] Jesse Walker, "Unsafe at any key size: an analysis of the WEP encapsulation," *IEEE Document802.11-00/362*, Oct 2000. Web page online available at http://www.drizzle.com/~aboba/IEEE/ last accessed on 4 October 2002.

[72] K. Fall, K. Varadhan and the VINT project. (2003) http://www.isi.edu/nsnam/ns/ns-documentation.html

[73] K. Sundrani, L.K. Awasthi and M.C. Govil, "A Novel Scheme for Location Management in Adhoc Networks," *ACCST research journal*, 3(3), pages 199-204, 2005.

[74] L. Buttyan and J.-P. Hubaux, "Nuglets: A Virtual Currency to Stimulate Cooperation in Self- Organized Ad Hoc Networks," *Technical Report DSC/2001/001*, Swiss Federal Institute of Technology, Lausanne, 2001.

[75] Leonard Adleman, "Molecular Computation of Solutions to Combinatorial Problems," Science, 266:1021-1024, November 1994.

[76] M. Abolhasan, T Wysocki and E Dutkiewicz, "A review of routing protocols for mobile ad hoc networks," *J. Ad Hoc Networks 2 (2004)*, pp. 1-22.

[77] M. C. Govil, D. Gopalani, R. Jain, R. Ladha and S. Sharma, "Evolution of TCP over wireless links," *In Proc. National Symposium on emerging trends in Broadband Communication*, April 8-9, 2006.

[78] M. Reiter and S. Stybblebine, "Authentication Metric Analysis and Design," *ACM Transactions on Information and System Security,* 1999.

[79] M. Weiser, "The Computer for the Twenty-First Century," *Scientific American*, pages 94–100, September 1991.

[80] M. Yarus, "RNA-ligand chemistry: a testable source for the genetic code," *RNA 6 (2000)*, pages 475–487, 2000.

[81] M. Zapata, "Secure Ad hoc On-demand Distance Vector (SAODV)," Internet Draft, draft-guerrero-manet-saodv-01.txt, 2002.

[82] Manel Guerrero Zapata and N. Asokan. *"Securing Ad Hoc Routing Protocols". In Proc. of the ACM Workshop on Wireless Security* (WiSe 2002), September 2002.

[83] Maria Fazio, ClaudioEnrico Palazzi, Shirshanka Das, Mario Gerla*,"* Facilitating Real-time Applications in VANETs through Fast Address Auto-configuration," *In Proc. 3rd IEEE CCNC International Workshop on Networking Issues in Multimedia Entertainment, (CCNC/NIME 2007), IEEE Communications Society*, Las Vegas, USA.

[84] Mohammed O. Pervaiz, Mihaela Cardei, and Jie Wu, "Routing Security in Ad Hoc Wireless Networks," *Network Security*, Scott Huang, David MacCallum, and Ding Zhu Du (Eds.), springer, 2005.

[85] N. Kang, "A pseudo DNA cryptography method," http:// www.comp.nus.edu.sg/~ningkang/reports/DNAcryptography.doc, last accessed on March 27, 2005.

[86] National Institute of Standard and Technology, Security requirements for cryptographic modules," FIPS 140-1, Jan. 1994.

[87] Nikola Milanovic, M. Malek, A. Davidson and V. Milutinovic, "Routing and security in Mobile Ad Hoc Networks," *IEEE Computer Society,* pages 61-65, Feb. 2004.

[88] P. J. Wan, G. Calinescu, X.Y. Li, and O. Frieder, "Minimum energy broadcast routing in static ad hoc wireless networks," *In Proc. of the IEEE Conference on Computer Communications (INFOCOM)*, pages 1162-1171, 2001.

[89] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks," *In Proc. of MOBICOM 1999.*

[90] P. Kuosmanen, "Classification of Ad Hoc Routing Protocols," *Finnish Defence Forces, Naval Academy*, Finland.

[91] P. Michiardi and R. Molva, "Simulation-based Analysis of Security Exposures in Mobile Ad Hoc Networks," *In Proc. of European Wireless Conference, 2002.*

[92] P. Michiardi and R. Molva, "Core: A COllaborative REputation mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," *In Proc. of IFIP Communication and Multimedia Security Conference 2002.*

[93] P. Michiardi and R. Molva, "Game Theoretic Analysis of Security in Mobile Ad Hoc Networks," *Institut Eurecom Research Report RR-02-070*, April 2002.

[94] P. Muhlethaler, P. Jacquet, and A. Qayyum, "Optimized Link State Routing Protocol," Internet Draft, draft-ietf-manet-olsr-00.txt, November 1998.

[95] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," *In Proc. SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, 2002.

[96] P. Sinha, R. Sivakumar, and V. Bharghavan, "CEDAR: a Core-Extraction Distributed Ad hoc Routing Algorithm," *IEEE Journal on Selected Areas in Communications, 17,* 8, August 1999.

[97] P. Zerfos, Z. Fu, K. Xu, H. Luo, S. Lu, L. Zhang, and M. Gerla, "On TCP Performance in Multihop Wireless Networks," in *Proceedings of InfoCom,* 2003.

[98] Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek, and Mikael Degermark, "Routing Protocols for Mobile Ad-hoc Networks—A Comparative Performance Analysis," *In Proc. of the Fifth International Conference on Mobile Computing and Networking (MobiCom'99), ACM*, August 1999.

[99] Q. He,D.Wu. and P.Khosla,"SORI: A Secure and Objective Reputation based Incentive Scheme for Ad-hoc Networks," *In Proc. IEEE WCNC2004, Mar. '04.*

[100] R. G. T. Anderson, B. Bershad, and D. Wetherall, "A System Architecture for Pervasive Computing," *In Proc. 9th ACM SIGOPS European Workshop*, pp. 177–182, Kolding, Denmark, September 2000.

[101] Rajwinder Singh, Navdeep kaur and A.K. Sarje,"Secure Itineraries Framework for Mobile Agent Systems," *LNCS 4332*, Pages 361-364, 2006.

[102] RFC 2560 www.ietf.org/rfc/rfc2560.txt, last accessed on March 25,2007.

[103] RFC 4728 www.ietf.org/rfc/rfc4728.txt, last accessed on March 25,2007.

[104] Richard G. Ogier, Fred L. Templin, and Mark G. Lewis, "Topology dissemination based on reverse-path forwarding (TBRPF)," *IETF RFC3684* - Experimental Standard, February 2004.

[105] Richard Walton, "Cryptography and Trust ," *Information Security Technical Report II* (Elsevier) , pp 68-71,2006.

[106] Robert Castaneda and Samir R. Das, "Query Localization Techniques for On-demand Routing Protocols in Ad Hoc Networks," *In Proc. of the 5th International Conference on Mobile Computing and Networking (MobiCom'99)*. ACM, August 1999.

[107] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang, and Satish K. Tripathi, "Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks," *IEEE Personal Communications*, 4(1): pages 36–45, February 1997.

[108] S. Bansal and M. Beker,"Observation-Based Cooperation Enforcement in Ad-hoc Networks," *Technical Report, Stanford University, '03.*

[109] S. Buchegger and J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol," in *Proc. of MobiHoc 2002.*

[110] S Buchegger & J-Y Le Boudec, "Nodes Bearing Grudges: Towards Security, Fairness and Robustness in Mobile Ad hoc Networks," *In Proc. 10th IEEE Euromicro Workshop on Parallel, Distributed and Network based Processing,* pages 403-410, January 2002

[111] S. Capkun, L. Buttyan, and J-P Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks," *In Proc. of ACM International Workshop on Wireless Security, WiSe 2002.*

[112] S. Corson and J. Macker, "Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations," *RFC 2501*, Jan. 1999.

[113] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *In Proc. of MOBICOM 2000.*

[114] S P Alampalayam and A Kumar, "Security Models for Routing Attacks in Mobile Ad hoc Networks," *IEEE 58th Vehicular Technology Conference*, vol 3, pages 2122-2126, Oct 2003.

[115] S. Ramanathan and Streenstrup M., "A Survey of Routing Techniques for Mobile Communication Networks," *ACM/Baltzer Mobile Networks and Applications, special issue on Routing in Mobile Communication Networks*, Vol. 1, No. 2, pages 89-104, 1996.

[116] S. Tapaswi, Ramesh Joshi, "Environment Monitoring using wireless Sensor Networks, " *In Proc. National Conference on Issues and Trends in Wireless Networks (IT-WiNS 2004)*, pages 244-249, December 2004.

[117] *S. Yi, P. Naldurg, R. Kravets,* "Security Aware Ad hoc Routing for wireless Networks," *Report No. UIUCDCS-R-2002-2290,* UIUC, 2002.

[118] Skoudis, Ed, Counter Hack, "A Step-by-Step Guide to Computer Attacks and Effective Defenses," *Prentice Hall, Upper Saddle River,* New Jersey, pages 351-358, 2002.

[119] Siddhartha Gupta, Mukesh Singhal, "Secure Routing in mobile wireless ad hoc networks," J. Ad Hoc Networks 1 (2003) 151-174.

[120] "Specification of the Bluetooth System," Bluetooth Special Interest Group, Version 1.1, February 22, 2001, http://www.bluetooth.com/pdf/Bluetooth_11_Specifications_Book.pdf.

[121] Stefano Basagni, Marco Conti, Silvia Giordano, Ivan Stojmenovic (Eds.), "Mobile Ad Hoc Networking," *IEEE Press,* Wiley Interscience. 2004.

[122] Stubblefield, Loannidis, and Rubin, "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP," *AT&T Labs Technical Report*, 2001.

[123] T. Jakobsen and L.R. Knudsen, "Attacks on block ciphers of low algebraic degree," *J. Cryptology*, 14: pages197-210, 2001.

[124] T. S. Rappaport, "Wireless Communications: Principles and Practice, 2/E", *Prentice Hall PTR,* 2002.

[125] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford "Introduction to Algorithms," Second Edition, *The MIT press,* 2001.

[126] Tracy Camp, Jeff Boleng and V Davies, "A survey of Mobility Models for Ad Hoc Network Research", http://toilers.mines.edu last accessed on February 15, 2007.

[127] V.Park and S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Ver.1 Functional Specification," IETF draft, 2001.

[128] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang, "MACAW: A media access protocol for wireless LAN's," *In Proc. of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, pages 212–225, August 1994.

[129] Vangelis Angelakis Apostolos Traganitis, "Extending the Dynamic Source Routing (DSR) Protocol to Deal with Node Selfishness in Ad Hoc Networks," *http://www.ics.forth.gr/~angelak/files/AngelakTragani_DSR_selfishness.pdf,* last accessed on January 24, 2007

[130] Vikram Goyal, Shyam K. Gupta, Anand Gupta, "Malafide Intension and its mapping to Privacy Policy Purposes for Masquerading," *In Proc. 10th International Database Engineering and Applications Symposium (IDEAS 2006),* pages 311-312, 2006.

[131] www.accessexcellence.org/RC/VL/GG/structure.html

[132] www.makingthemodernworld.org.uk/learning_modules/biology /01.TU.03/illustrations/01.IL.09.gif

[133] www.wikipedia.org

[134] W. Mehuron, "Digital Signature Standard (DSS)," U.S. Department of commerce, *National Institute of Standards and Technology (NIST)*, Information Technology Laboratory (ITL). FIPS PEB 186,1994

[135] Wensheng Zhang, Guohong Cao, "Defending against cache consistency attacks in wireless ad hoc networks", *To appear J. Ad Hoc Networks* (Elsevier), available February 6,2007.

[136] William Stallings "Cryptography and network security: principles and practice," Prentice *Hall Inc*. Upper Saddle River, NJ.

[137] Y-C Hu, D. B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," *In Proc. Of 4th IEEE Workshop on Mobile Computing Systems and Applications.*

[138] Y.-C. Hu, A. Perrig , and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad hoc Networks," *In Proc. 8th ACM Int'l. Conf. Mobile Computing and Networking (Mobicom'02)*, Atlanta, Georgia, pages 12-23.

[139] Y. C. Hu, A. Perrig, and D. B. Johnson, "Wormhole detection in wireless ad hoc networks," *Tech. Rep. TR01-384,* Department of Computer Science, Rice University, June 2002.

[140] Yih-Chun Hu and David B. Johnson, "Implicit Source Routing in On-Demand Ad Hoc Network Routing," *In Proc. of the 2nd Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, pages 1-10, October 2001.

[141] Yih-Chun Hu, Adrian Perrig, David B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," *In Proc. ACM workshop on Wireless security*, 2003.

[142] Young-Bae Ko and Nitin Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *In Proc. 4th International Conference on Mobile Computing and Networking (MobiCom'98)*, pages 66–75. ACM, October 1998.

[143] Zapata, M. G., "Secure ad-hoc on-demand distance vector (SAODV) routing," *IETF MANET, internet draft (Work in progress)*, draft-guerrero-manet- saodv-00.txt, 2001.- accessed 10/10/2006.

[144] Zhang, W. lee, Y. Huang, "Intrusion detection techniques for mobile wireless networks," *ACM-Kluwer MONET*, vol.9, No.5, 2003

[145] Zygmunt J. Haas and Marc R. Pearlman, "The Performance of Query Control Schemes for the Zone Routing Protocol," *In Proc. of the ACM SIGCOMM '98 Conference*, pages 167–177, September 1998.

# PUBLICATIONS

**International Journal(s)**

1. A K Verma, Mayank Dave and R C Joshi, "Genetic Algorithm and Tabu Search Attack on the Mono-Alphabetic Substitution Cipher in Ad-hoc Networks," *International J. of Computer Science, NY (USA),* 3(3), 134-137, 2007.

2. A K Verma, Mayank Dave and R C Joshi, "Secure Data Sharing in Mobile Adhoc Networks," *J. International Review on Computers and Software (IRECOS), ISSN 1828-6003* (Peer reviewed and accepted).

3. A K Verma, Mayank Dave and R C Joshi, "Secure Routing in Mobile Networks: A Review," *International J. of Systemics, Cybernetics and Informatics (IJSCI), ISSN 0973-4864* (Peer reviewed and accepted).

**National Journal(s)**

4. A K Verma, Mayank Dave and R C Joshi, "Applying Distributive Computing In Mobile Ad hoc Networks (MANETs)," peer reviewed and accepted for publication in J. of Punjab Academy of Sciences (India).

**International Conference(s)**

5. A K Verma, Mayank Dave and R C Joshi, "Securing Ad hoc Networks with DNA Cryptography" at IEEE International conference on Computers and Devices for Communication (CODEC06), pp. 781-786, Dec. 18-20, 2006.

6. A K Verma, Mayank Dave and R C Joshi, "Ad hoc Networks – A Novel
solution for Value Creation in Modern Economy for NextGen Enterprise," South-East Asian Regional Computer Confederation (SEARCC'06), pp. 122-131, Colombo, Oct. 10-11,2006.

7. A K Verma, Mayank Dave and R C Joshi "Performance Evaluation of MANET Protocols: A simulation study" at XXVIIIth General Assemby of International Union of Radio Sciences (URSIGA05), pp. 13, New Delhi, Oct. 23-29, 2005.

8. A K Verma, Mayank Dave and R C Joshi "DNA Cryptography: A Biological Approach for Generating Ciphertext using DNA Computing" at 3rd International conference on Computer Science and its Applications (ICCSA-2005), pp. 165-172, San Diego, California, USA, June 27-29, 2005.

9. A K Verma, Mayank Dave and R C Joshi "Issues for routing in Mobile Ad hoc Networks" at IEEE International conference on Computers and Devices for Communication (CODEC04) pp. 213, Jan. 1-3, 2004.

**National Conference(s)**

10. A K Verma, Mayank Dave and R C Joshi "Performance Evaluation of MANET Protocols" at National Conference on Issues and Trends in Wireless Networks (IT-W*i*NS04), pp. 250-255, Dec. 17-18, 2004.

11. A K Verma, Mayank Dave and R C Joshi, "Software Engineering in support for MANET", at National Conference in Software Engineering – Principles and Practice (SEPP04), pp. 34-36, Mar 5-6,2004.

12. A K Verma, Mayank Dave and R C Joshi, "Routing Protocol Performance issues in MANET", at Advances in Computer

Communication Networks (CCN2003), IIT, Roorkee, pp.17-21, Feb 7-8, 2004.

13. A K Verma, Mayank Dave and R C Joshi, "Classification of Routing Protocols in MANET", at National Symposium on Emerging Trends in Networking & Mobile Communication (NSNM-2003), pp. 132-139, Sept 5 – 6, 2003.

**Technical Article**

14. A K Verma, "Mobile Ad hoc Networks (MANETs): An Introduction", in TTC Newscaster (a quarterly newsletter of Thapar Technology Campus), pp. 13-14, April 2004.

**Under Review (International Journals)**

A K Verma, Mayank Dave and R C Joshi, "SANE-DNA: A Novel Secure Routing Protocol for MANETs", *J. Ad hoc Networks (Elsevier)* Uploaded March 25, 2007.

A K Verma, Mayank Dave and R C Joshi, "Secure Routing in Adhoc Networks using DNA cryptography," *WSEAS Transactions on Computers.* Transaction-ID: 2007-261.

A K Verma, Mayank Dave and R C Joshi, "Formal Methods for Verification of Security Protocols in Adhoc Networks," *Emerald J. of information management and computer security.* Uploaded Dec 13, 2006.