Optimising Routing and trustworthiness of Ad Hoc Networks Using Swarm Intelligence

A thesis submitted in partial fulfilment of the requirements of the degree of Doctor of Philosophy (PhD) to:
Electronic and Computer Engineering
School of Engineering and Design
Brunel University
United Kingdom

By:

Saman Hameed Amin

Supervised By:

Professor Hamed Al-Raweshidy

March 2014

Optimising Routing and trustworthiness of Ad Hoc Networks Using Swarm Intelligence

A thesis submitted in partial fulfilment of the requirements of the degree of Doctor of Philosophy (PhD) to:
Electronic and Computer Engineering
School of Engineering and Design
Brunel University
United Kingdom

By:

Saman Hameed Amin

Supervised By:

Professor Hamed Al-Raweshidy

March 2014

Abstract

This thesis proposes different approaches to address routing and security of MANETs using swarm technology. The mobility and infrastructure-less of MANET as well as nodes misbehavior compose great challenges to routing and security protocols of such a network. The first approach addresses the problem of channel assignment in multichannel ad hoc networks with limited number of interfaces, where stable route are more preferred to be selected. The channel selection is based on link quality between the nodes. Geographical information is used with mapping algorithm in order to estimate and predict the links' quality and routes life time, which is combined with Ant Colony Optimization (ACO) algorithm to find most stable route with high data rate. As a result, a better utilization of the channels is performed where the throughput increased up to 74% over ASAR protocol.

A new smart data packet routing protocol is developed based on the River Formation Dynamics (RFD) algorithm. The RFD algorithm is a subset of swarm intelligence which mimics how rivers are created in nature. The protocol is a distributed swarm learning approach where data packets are smart enough to guide themselves through best available route in the network. The learning information is distributed throughout the nodes of the network. This information can be used and updated by successive data packets in order to maintain and find better routes. Data packets act like swarm agents (drops) where they carry their path information and update routing information without the need for backward agents. These data packets modify the routing information based on different network metrics. As a result, data packet can guide themselves through better routes.

In the second approach, a hybrid ACO and RFD smart data packet routing protocol is developed where the protocol tries to find shortest path that is less congested to the destination. Simulation results show throughput improvement by 30% over AODV protocol and 13% over AntHocNet. Both delay and jitter have been improved more than 96% over AODV protocol. In order to overcome the problem of source routing introduced due to the use of the ACO algorithm, a solely RFD based distance vector protocol has been developed as a third approach. Moreover, the protocol separates reactive learned information from proactive learned information to add more reliability to data routing. To minimize the power consumption introduced due to the hybrid nature of the RFD routing protocol, a forth approach has been developed. This protocol tackles the problem of power consumption and adds packets delivery power minimization to the protocol based on RFD algorithm.

Finally, a security model based on reputation and trust is added to the smart data packet protocol in order to detect misbehaving nodes. A trust system has been built based on the privilege offered by the RFD algorithm, where drops are always moving from higher altitude to lower one. Moreover, the distributed and undefined nature of the ad hoc network forces the nodes to obligate to cooperative behaviour in order not to be exposed. This system can easily and quickly detect misbehaving nodes according to altitude difference between active intermediate nodes.

To My Family

ACKNOWLEDGMENTS

I am grateful to the creator who made every living thing of water, without his blessing and mercy; this thesis would not have been possible.

I would like to express my deepest gratitude to my parents, my sisters and my brother, without their tremendous support, help and prayers I would not have made it through my PhD degrees.

It is my greatest pleasure to thank my supervisor, Professor Hamed Al-Raweshidy, whose encouragement, guidance and support enabled me to develop this work. I am thankful to my colleagues at the WNCC and friends at Brunel University. I would like to show my gratitude to all the staff at Brunel University as well.

Finally, I would like to express gratitude to those who helped me and supported me to finalize my thesis; especially my Facebook group members, without them, I could not have written this thesis.

Declaration

This is to certify that:

i. The thesis comprises only my original work towards the PhD except where indicated.

ii. Due acknowledgement has been made in the text to all other material used.

Saman Hameed Amin

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Meaning
ABC	Ant Based Control algorithm
ABED	Ant-based Adaptive Trust Evidence Distribution
ACK	Acknowledgement
ACO	Ant Colony Optimization
AHCPN	Ad Hoc Cognitive Packet Network
ANFIS	Adaptive Neuro-Fuzzy Inference System
AODV	Ad hoc On Demand Distance Vector
AOTDV	Multipath trust routing protocol
ARA	Ant Colony Based Routing Algorithm
Ariadne	A Secure On-Demand Routing Protocol for Ad Hoc Networks protocol
AS	Ant System
ASAODV	Adaptive Secure Ad hoc On Demand Distance Vector
ATM	Automatic Multilevel Thresholding
CBR	Constant Bit Rate
CM	Cognitive Map
CORE	COllaborative REputation mechanism to enforce node cooperation in mobile ad hoc networks
CPN	Cognitive Packet Network
DAG	Directed Acyclic Graph
DELAR	Device-Energy-Load Aware Relaying
DP	Dumb Packets
DSDV	Destination-Sequenced Distance Vector routing protocol
DSR	Dynamic Source Routing
GBR	Gradient Based Routing protocol
GM	Gauss Markov mobility
GPS	Global Positioning System
HODVM	Hybrid On-demand Distance Vector Multi-path
IWD	Intelligent Water Drop
MAC	Medium Access Control
MAC PDU	MAC Protocol Data Unit
MANET	Mobile ad hoc network
MB	Mailbox
MIMO	Multiple-Input Multiple-Output
MPR	multipoint re1ay nodes
NP-hard	non-deterministic polynomial-time hard
OLSR	Optimized Link State Routing protocol

OSPF	Open Shortest Path First
PDTMRP	Power-aware Dual-Tree-based Multicast Routing Protocol
PLCP	Physical Layer Convergence Protocol
PoLiQ	Power-Aware Link Quality estimation
POSANT	Position Based Ant Colony Routing algorithm
PSO	Particle Swarm Optimization
QoS	Quality of Service
RERR	Route Error message
RFD	River Formation Dynamics
RFD	River Formation Dynamics
RNN	Recurrent Neural Network
RREP	route reply message
RREQ	route request message
RRS	Robust Reputation System
RWP	random waypoint mobility
SAODV	Secure AODV Protocol
SEAD	Secure Efficient Distance vector routing for mobile wireless ad hoc networks
SI	Swarm Intelligence
SINR	Signal-to-Interference and Noise Ratio
SISF	Short InterFrame Space
SLSP	Secure Link State Routing Protocol
SMART	Smart data packet routing protocol
SP	Smart Packets
SSCH	Slotted Seeded Channel Hopping
ST-AODV	Simple Trust model for Ad hoc On Demand Distance Vector protocol is introduced
SWAH	Spatial Wireless Ad Hoc
TC	topology control
TORA	Temporally-Ordered Routing Algorithm
TSP	Travelling Salesman Problem
WNR	Wireless Networked Robotics
WSN	Wireless Sensor Network

Chapter One

Routing and Security in Ad Hoc Networks

1.1 Introduction

The great technological development in the field of designing mobiles and portable devices has led to a broader request for communications services. The more advance in technology, the more demands for more services. Mobile ad hoc network (MANET) is a distributed autonomous system where multiple mobile devices communicate with each other, directly or indirectly, without any predefined infrastructure. Despite the revolutionary development in communication technology, mobile ad hoc networks are still facing several challenges due to wireless communication media, nodes power limitation, no infrastructure and others. Routing algorithm should jointly optimize different performance metrics of the network. Routing and security protocol in such networks can be considered most challenging problems.

Artificial swarm intelligence is a distributed self-organized system used to solve many problems. Swarm intelligence[1] is inspired from the collective behaviour observed in natural systems such as insect societies. Ant Colony Optimization (ACO) is an artificial ant algorithm that mimics the behaviour of the real ants and is widely used to solve optimization problems. ACO makes use of groups of independent artificial ants who are able to communicate with each other through an indirect pheromone-based communication. Ants work cooperatively and share their solutions with each other using pheromone intensity on each link. This work usually leads to good solutions to problems. River Formation Dynamics (RFD) is another type of swarm intelligence which simulates different natural behaviour. The main inspiration behind RFD is how the water forms rivers in nature. In nature, when rain falls on the mountains it joins together forming rivers and moves towards lower lands which eventually fall into the sea. As these water drops move, they change the environment by eroding the surface by an amount proportional to the decreasing slope of the surface and deposits carried sediments on a flatter

[1] Throughout this thesis, swarm intelligence is mainly used to refer to artificial swarm intelligence.

surface. The speed and direction of moving water drops depend on the gradient of the surface. In this way, the altitude of the surface is changed and water drops creates rivers. Progressively, new water drop will reinforce the best shortest path which will be considered as an optimal path between the place where it is raining and the sea.

Swarm intelligence uses different types of mobile agents in order to optimize network performance. These autonomous agents have the capability to search, maintain, cooperate and adapt network parameters. Swarm intelligence is motivating techniques which is used to address MANETs routing and security problems.

1.2 Motivation

The characteristic of mobile ad hoc networks, like dynamic nature, infrastructure-less, and wireless communication media, adds extra complexity to routing protocol in order to find the optimal path with satisfactory quality of services between source and destination nodes. Channel assignment in such a network is challenging issue especially if the number of available interfaces is limited. The problem of finding a stable path considering link qualities between nodes requires re-investigation.

The dynamic nature of mobile ad hoc network demands the efficient use of every packet in the network in a way that incorporates in enhancing the routes between sources and destinations. Most routing protocol use dedicated control packet to collect network status and exchange routing information. Data packet usually routed upon the information gathered by control packets. An interesting research question arises:

Can data packets route themselves and search for the optimal path to their destinations?

Giving data packets the ability to collect route information, and incorporate with control packet in the process of finding optimal path could enhance network performance. Consequently, this will give them the ability to move and search for their own destinations.

Generally, wireless networks are more prone to security attacks than wired networks. Nodes cooperation is the main principle in ad hoc networks in order to deliver data packets especially to out of communication range nodes. Selfish behaviours as well as malicious misbehaviour have enormous impact on routing algorithms. These kinds of attack could ruin the routing algorithm and degrade network performance. Monitoring nodes behaviour could help in detecting misbehaviour nodes. Enhancing monitoring capabilities requires an efficient technique to detect misbehaving nodes. Detecting extra features could enhance any monitoring system especially if the features are independent and could be detected by any node in the network. This will increase the security as malicious act will be exposed.

1.3 Aim and Objectives

The main aim of this research is to design and develop intelligent routing and security protocols based on swarm intelligence for mobile ad hoc networks. The goals of this research are addressed through the following objectives:

1- Efficiently utilize multiple channels where the number of wireless interfaces is limited. Ant Colony Optimization is used to collect network statues and control the routing algorithm. At the same time, selecting a reliable path by predicting route life time from link qualities. Geographical information is used to predict link states, incorporated with mapping algorithm to reflect end to end geographical structure of routes.

2- Introducing the use of the RFD algorithm in mobile ad hoc routing. A combination of ACO and RFD is used to design a smart routing protocol where data packets try to find the shortest path to their destinations that is less congested.

3- Implementing a routing protocol that is solely based on the RFD algorithm and separating reactive and proactive collected information into two separate records. Data packets will be routed on most recent information which resides in the reactive records to add more reliability to data routing.

4- Reduce the power consumed to send each packet to its destination by detecting nodes remaining battery power and select a route with higher remaining battery power. At the same, the sediment carried by the drops will contain selected route information which reflects the consumed power through the route.

5- Design a reputation and trust based security routing protocol. The protocol is based on smart data packet routing protocol where trust and reputation values are converted to altitudes and used to select nodes with higher trust levels. Detecting misbehaviours and building trust values is by monitoring packet altitudes as well as the ratio of packet forwarding. Reputation information is shared between nodes within the network.

1.4 Main contributions

The contributions of this work are summarized as following

1-The design of geographical multi-channel multi interface ant based routing protocol. This routing protocol makes use of mapping method and uses local geographic information to build an end to end route quality predictor. The protocol addresses the limitation in number of interface cards and tries to select most stable route. The protocol predicts and tries to setup a route before the occurrence of route failure.

2- The work introduces the use of river formation algorithm in MANETs where a smart data packet protocol is developed. The protocol shows a novel way of converting data packets to data agents. Data packets are smart enough to guide themselves through best available route in the network. This approach uses distributed swarm learning approach which will minimize convergence time by using smart data packets. This decreases the number of control packets in the network as well as it permits learning using data packets which in turn provides better reaction to changes in the network environment. The learning information is distributed throughout the nodes of the network. This information can be used and updated by successive data or agent packets in order to maintain and find better routes. The protocol shows that learning could be carried out without the need for backward agents. Moreover, the number of bytes that is added to data packet in order to convert it to smart data packet is minimized. Route maintenance is carried out by data packets themselves. The proposed solution creates a multipath protocol where this further provides a mechanism of load balancing across the network.

3- The work proposed three different approaches to use River Formation Dynamics in ad hoc network routing. Firstly, a hybrid ACO and RFD algorithm is introduced. The ACO is only used at the initialization to set up multi-path routes to destination, while the RFD mainly used as a base algorithm for the routing protocol. Secondly, a pure and solely RFD based routing algorithm is introduced, whereas a proactive and reactive information is separated into different tables to reliably route data packet. Finally, a power aware routing protocol is developed based on the RFD algorithm, without the separation of routing information, where power characteristic is converted to altitude parameters. Packets are forwarded according to the remaining power in the nodes. The protocol also searches for the shortest and non-congested path to the destination. At the same time, the sedimentation process uses the power information of the path and added sediment to the route.

4- A secured routing protocol based on RFD algorithm is developed, where reputation and trust is used to detect misbehaving nodes. The mapping and management of monitoring information is based on RFD algorithm. The technique introduces extra features that could be monitored to detect misbehaving nodes. This feature is adapted from basic idea of the RFD algorithm where water drops move only toward the sea. This allows any monitoring node in-between two interacted nodes to determine if the routing decisions made are right or not. Moreover, this feature could be detected directly, unlike the usual monitoring of the forwarding rate of packets which needs time and has uncertainty problem.

1.5 Thesis Outline

This thesis is organised into nine chapters.

Chapter two is an introduction to the ad hoc networks. The aim of the chapter is to provide a technical background on routing and security protocols used in mobile ad hoc networks. The chapter starts with a brief introduction to mobile ad hoc network and its properties. Different types of routing protocols are discussed throughout the chapter. Finally, security algorithms are discussed and mainly focusing on reputation and trust security systems.

Chapter three provides a review of swarm intelligence techniques. The aim is to supply sufficient information to understand swarm intelligence, including explanation of different type of swarm intelligence relative to the work presented in this thesis. Ant colony optimization is explained, flowed by brief implementation of ant colony optimization in ad hoc network routing protocols. River formation dynamics is then explained. An example is given to show the process of river formation dynamic. The chapter also includes a brief review of intelligent water drop algorithm.

Chapter four presents multi-channel multi-interface routing algorithm for mobile ad hoc network where each node is equipped with limited number of interface cards. The protocol is based on ACO algorithm and uses geographical information to predict links and routes qualities. The chapter also describes the simulation model and the output results and compare the results of proposed protocol with an existing ant based multi-channel multi-interface routing protocol.

Chapter five introduces smart data packet routing protocol. The protocol is based on hybrid ACO and RFD algorithms. The proposed routing model is explained showing the changes to the basic RFD algorithm in order to fit with ad hoc network. The advantages of using RFD algorithm over other protocol are highlighted describing current bottlenecks in other protocols. The result is being compared with two protocols, AODV and AntHocNet. Different protocol parameters are being tested and their effect on the protocol performance is studied.

Chapter six introduces RFDManet; an RFD solely based routing protocol. The routing tables are duplicated into two sets, reactive and proactive sets of tables. These sets are updated and used by different parts of the proposed protocol. The first set is reactive set of tables, which are updated frequently and there for contains recent information of the network status, while the second set is produced as a result of information diffusion and may contain expired information. Different sets of tests are carried out to test the performance of the protocol, and compared to AODV and AntHocNet protocols.

In chapter seven, a power and congestion aware routing protocol is proposed. The protocol is solely based on the RFD algorithm. The protocol combines the idea of previous two chapters and adds power awareness to the protocol. The protocol uses single set of tables for both reactive in proactive information. The protocol is compared to AODV in order to show the performance of the protocol.

In chapter eight, a reputation and trust system is used to build a secure routing protocol called RFDTrust which is based on the smart data packet protocol. The protocol uses the features of RFD algorithm to add more robustness against malicious attacks. The security model is described and the process of information gathering and sharing is explained. At the end of the chapter, various simulations are run to evaluate the proposed secured protocol.

Chapter nine summarize the research finding and contributions. The chapter also gives an insight for future research.

Chapter Two

Routing and Security in Ad Hoc Networks

2.1 Introduction

Mobile Ad hoc Networks are used to provide wireless communication without a predefined infrastructure or centralized administration. The absence of infrastructure in mobile ad hoc network and the dynamic movement of its nodes present real challenges to routing and security algorithms in such networks. Nodes between the source and the destination nodes act as routers and forward packets according to the routing algorithm. A number of algorithms have been proposed to address routing problems in ad hoc networks [1-4]

In this chapter, different types of routing algorithms and mechanisms ad hoc networks will be highlighted. Moreover the problem of channel assignment in multi-channel multi-interface networks will be explained. This chapter also includes a discussion on different types of attacks on routing algorithms in mobile ad hoc networks. Reputation and trust systems will be discussed as a method to detect misbehaviour nodes in MANETs.

2.2 Mobile ad hoc networks

The rapid growth and development of wireless mobile devices has led to a tremendous increase in research and enhancements on routing protocols for wireless mobile ad hoc networks. The sharing media and the movement of mobile nodes in wireless networks add many restrictions to these networks such as the limitation of bandwidth, power limitation, security issues, and others. In spite of these limitations, mobile ad hoc networks paradigms provide the only means of mobile communication for decentralized and infrastructure-less networks.

Mobile ad hoc networks are autonomous self-organized infrastructure-less networks where any node can move freely in the network and can communicate with other nodes in the network directly or indirectly. Nodes within the transmission range of the node are connected directly to the node. Other nodes that are out of range will communicate through intermediate nodes which act like routers. Mobile ad hoc networks permit the instantaneous deployment of decentralized networks without per-existing infrastructure. Mobile ad hoc networks offer suitable solution for emergency situation such as flooding and earthquakes as they require minimal configuration and could be deployed rapidly. Mobile ad hoc networks find its way in military and battlefield application where the application environments are highly dynamic in nature. Another application of mobile ad hoc network is in swarm robotic area. Figure 2.1 shows an example for a mobile ad hoc network.

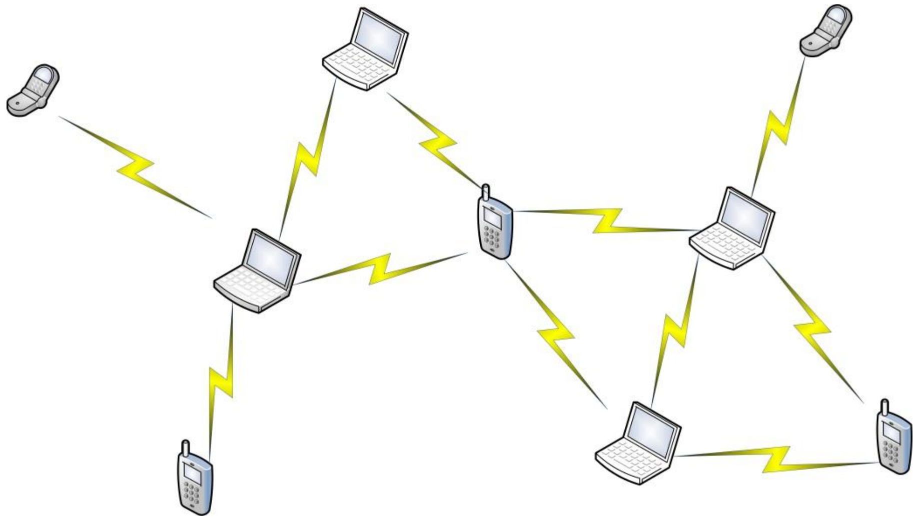


Figure 2.1 Mobile ad hoc network

2.3 A brief review of network security

Both wired and wireless networks are vulnerable to security attacks. Security is an essential component for wireless networks to function properly in the presence of attacks. The characteristics of mobile ad hoc networks pose more challenges in achieving security goals such as access control, authentication, data confidentiality, data integrity, availability and privacy. Attacks on any network could be divided into two main types: passive and active attacks. Passive attacks are based on monitoring and observing data packets without disrupting the network operation such as eavesdropping and traffic analysis attacks. On the other hand, active attacks try to disrupt the network operation through information interruption, modification, or fabrication. Examples of active attacks are message replay, message modification, and denial of service attacks [5].

2.4 Properties of MANETs

MANETs share many properties with wireless networks; however it has some characteristics that distinguish them from other wireless networks.

- *Mobility and dynamic topology*: As explained earlier, MANETs consists of several mobile nodes that communicate directly or indirectly with each other. Nodes within MANETs may move arbitrarily. The movement of the nodes as well as the possibility of some node failures lead to dynamic topology.

- *Autonomous and infrastructure-less:* MANET networks are self-configuring networks whereas control and administration of the network is distributed, and do not rely on any infrastructure. The distributed nature of MANETs makes them better protected against single failure points.

- *Wireless medium and link quality*: Nodes within MANET network communicate wirelessly. The wireless medium has a major impact on link quality. Node distance effects link quality. The effect of fading, multi-path cancellation as well as interference could degrade link quality in wireless medium. Since nodes share the same medium, there is increasing possibility for congestion as the number of nodes and traffic increases. Link quality could become unpredictable due to node movement and limited transmission range of the nodes.

- *Multi-hop communications*: Nodes in MANETs not only act as end users, but they also act as routers. When a node needs to communicate with other nodes within is transmission range, it may communicate directly with them. However, if destination node is not within its transmission range of the same node, intermediate nodes will act as routers and forward packets towards their destinations.

- *Limited energy resource:* Nodes in MANETs are powered by batteries and therefore have limited energies. Reducing communication related power consumption is an important factor in designing routing algorithms. Other limitations of MANETs' nodes are node processing power, memory, etc.

- *Limited Physical Security*: The wireless medium of ad hoc networks adds more risk to MANET. The transmission media is shared which makes it vulnerable to security attacks.

2.5 Routing in mobile ad hoc networks

The main goal of routing algorithms is to find the best available path from the source node/s to the destination/s to meet the given Quality of Service (QoS) requirements. Routing algorithm faces enormous challenges in mobile ad hoc networks, as the variation of network topology of MANETs caused by the movement of the nodes adds extra complexity to the optimization method. Due to this mobility of the nodes, link qualities may vary over time. This requires certain updates to cope with the quality of service. Routing algorithms could be categorized according to different criteria such as the method of route establishment to collect topology information (reactive, proactive or hybrid), or on which type of route information is carried by a packet which effects the method of routing packets in routers (source routing or distance vector routing), or the kind of information that is used to route packets (geographic routing or link state routing). Or the entity which is responsible for making route decisions (ordinary routing protocols or cognitive routing protocols).

2.6 Proactive, reactive and hybrid routing algorithms

Routing protocols that depend on network topology are categorized into three different types, reactive, proactive and hybrid protocols. The main differences of these routing protocols depend on when the information about network topology is collected and how to maintain the routes. The reactive routing protocols, sometimes called on demand routing protocols, only collects information when a route is required, when a source needs to exchange data with a destination, and there is no valid route to carry the information to the destination. On-demand routing protocol does not maintain up-to-date information about network topology all the time, rather they only request this information when a route is required. The route discovery process usually consists of flooding route request messages throughout the network. These types of protocols can be mainly divided into two categories, source routing protocols like Dynamic Source Routing (DSR) [6] and distance vector routing protocols like Ad hoc On Demand Distance Vector (AODV) routing protocol [7]. Source routing protocols suffer from routing overhead especially in large networks. The frequent route breaks due to the mobile nature of mobile ad hoc networks requires the reinitiating of route discovery process, in both types, which adds more delay and queuing to the network. On the other hand, proactive routing protocols are table driven routing protocols where all nodes are required to have complete knowledge of the network and therefore routing tables are periodically updated. Examples of proactive routing protocols are Optimized Link State Routing protocol (OLSR) [8] and Destination-Sequenced Distance Vector routing protocol (DSDV) [9]. Proactive routing protocols have low route setup latency. However, such routing protocols may not efficiently cope with the dynamic and rapid change of the network topology of mobile ad hoc networks. Since nodes in ad hoc networks are moving, links between nodes may become unavailable. Moreover, the quality of the links may change as these nodes are moving. Convergence to a new stable route after such a movement as well as maintaining route tables are costly in terms of resource usage and may lead to some delays in the network. Hybrid routing protocols tries to address the problem in both reactive and proactive by combining features from both reactive and proactive protocols into a hybrid protocol. An example of hybrid routing protocol is zone routing [10]. The main drawback of hybrid routing protocol is the high resource usage.

2.6.1 Ad-hoc On-Demand Distance Vector routing

C.E. Perkins and E. Belding-Royer introduced Ad hoc On Demand Distance Vector routing protocol [7]. AODV is a reactive protocol and route discovery initiates on request whenever there is a need for a new route and a route is not available at the source node. AODV uses hop-by-hop routing and it is considered as an on demand version of Destination Sequenced Distance Vector protocol. AODV uses several types of control packets in order to establish a route. Route discovery starts up when a node requires a route to a destination that is not one of its neighbours; usually the hello message is used in AODV to discover neighbour nodes. It starts by broadcasting a route request message (RREQ). This RREQ message contains the destination IP address, last known sequence number for that destination, as well as its own IP address and current sequence number.

When an intermediate node receives a RREQ message, it first creates a reverse route entry pointing to the originating node throughout the previous broadcasting sender node. The node checks if it has a valid route to the destination node based on the last known sequence number for that destination. If there is no route to the destination then the node checks the packet sequence number and hop count. If the packet sequence number is higher than the last received packet, or if the sequence numbers are equal, but the hop count is less than the existing hop count in the routing table, it will broadcast the RREQ message. On the other hand, if the node had a route to the destination, or the node is the destination node, it will send back a route reply message (RREP). The node adds the destination sequence number to the RREP packet and unicasts it back to the original source node. Upon receiving RREP message, the intermediate node creates a forward route entry for the destination node. After the first broadcasting, the original source node will wait for a time period equal to *network traversal time*. If no acknowledgment is received throughout this period, the source node may send another broadcast message until the *maximum number of retries* is reached. When the source node receives RREP, it starts sending data packets.

When link break in active route occurs the node will broadcast back Route Error (RERR) message to its neighbours. Upon receiving RERR, the intermediate node will check if the RERR affects it route table. If the route table is affected than the node broadcasts RERR for the effected route. When the original source receives a RERR it invalidates the route to the destination and starts the process of route discovery.

AODV uses HELLO messages to discover its neighbours. HELLO messages are limited to one hop and are sent periodically every Hello_interval milliseconds.

2.6.2 Dynamic Source Routing

David B. Johnson and David A. Maltz introduced DSR in 1996 [6]. This is another type of on demand ad hoc routing protocol which uses source routing method, as it is clear from its name. Unlike hop to hop routing, source routing protocol adds the complete route path to the packet which gives the source node complete control on how the packet moves in the network. According to DSR protocol, when a node needs to communicate with a destination, it first checks its routing table for a valid route to that destination. If no route is available, route discovery process starts. Route discovery consists of flooding the network with route request messages. Each node visited by the RREQ will add a copy of its address to the route record field in the packet before forwarding it to the next node. As a result the packet will have a full route path in its record when it arrives at the destination node. Upon receiving the RREQ at the destination node, it will reply with a route reply message RREP. The RREP will carry the discovered route which contains the addresses of all visited nodes by the RREQ message. An advantage of source routing is when an intermediate node receives a packet; it can extract information about routes to other intermediate nodes from the packet. This can minimize the need for route discovery as well as providing a backup route in case of route failure.

2.6.3 Optimized Link State Routing

OLSR protocol was introduced by P. Jacquet, *et al*. and it is a table driven proactive routing protocol [8]. The inspiration method behind OLSR is a wired protocol known as Open Shortest Path First (OSPF). Each node in OLSR uses hello messages to discover its two hop neighbours. HELLO message contains a set of neighbours' addresses. These HELLO messages allow a node to detect its neighbour upon receiving them. Using the information carried by these messages a node can learn about two hops neighbours. Based on the information carried by HELLO messages, each node can select its multipoint re1ay MPR nodes. It should be noted that whenever a node sends a HELLO message it marks the MPR node addresses with a MPR link status. MPR nodes are a set of nodes that allow a node to had links to all its two hop nodes through these MPR nodes. The MPR set need not be optima, however is should be small set. Each node maintains a table for its neighbours (neighbour table) as well as another table that contains addresses of its neighbour nodes that have selected it as an MPR (MPR selector table). If a node does not have an empty MPR selector table, it will broadcast a topology control (TC) message. Each TC message contains a set or a subset of addresses of nodes that have selected the sender as MPR node. These TC messages are broadcasted periodically. The time period between sending TC messages changes and an earlier transmission of TC message occur if the topology selector table changed. Link information data base is created by parsing the information in the TC messages. Each node will build a routing table based on link information database.

2.7 Geographic routing protocols

In this method of routing, the algorithm of routing protocol is based on geographic information, like position, speed and direction of movement [11]. Each node should be equipped with a device that locates its position. Most geographical routing protocols assume that a node is equipped with Global Positioning System (GPS). The routing algorithm makes use of this information by finding the best forwarding node based on this information. When a node knows the position of its neighbour and the packet destination position, it can select the forward node in a direction of the destination based on some geographic criteria.

2.8 Power aware routing

Nodes in mobile ad hoc networks are often powered on battery, therefore minimizing power consumption in such a network will increase their lifetime. Network lifetime is the time period from the starting of the network until a node in the network runs out of energy. Power consumption in any mobile node may be related to communication or others like processing power consumption and movement in robotic networks. Power aware routing addresses communication power consumption.

Most non power aware routing protocols try to maximize network performance at the expense of the power. Usually they don’t consider the increase in power consumption. The shortest path is not always the best path so power aware protocols try to find routes that will minimize power consumption [12].

Many methods have been proposed to address power consumption problems [13, 14] . Five significant power aware metrics that can be used to determine the quality of a route are [15, 16]:

1. ***Minimize energy consumed / packet***

2. ***Maximize time to Network Partition***

3. ***Minimize variance in node power levels***

4. ***Minimize cost / packet***

5. ***Minimize maximum node cost***

Power aware routing protocols are classified into two types: activity based protocols and connectivity based protocols. Activity based protocols are the type of protocols that try to control the activity of sending and forwarding packets in order to minimize power. These protocol controls how packets are routed and how routing decision are made [178].

In activity based protocol the protocol tries to maximize network lifetime and minimizing the power consumption for each delivered packet. Maximizing network lifetime protocols usually try to load balance the network and increase the probability of using nodes with higher remaining energy. Minimizing power for delivered packet protocols try to select paths that consume less power while delivering the packet from the source to the destination.

Connectivity based protocols are those protocols which either try to control the transmission power of a node or control the sleep time of a node to save energy. Protocols that try to minimize the transmission power, in order to cover specific nodes with some desired signal to noise ratio, cause connectivity problem in the network which should be considered when designing such protocols. The other type of connectivity based protocols try to turn off the transmitter for a specific amount of time, which also may lead to connectivity problems.

2.9 Cognitive routing

Unlike ordinary routing protocols that have been described in previous subsections, where routers have full control on forwarding packets and making route decisions, the cognitive packet network introduces different infrastructure where routing capability is moved from routers to the packets themselves.

Ordinary routing protocols give the full control of packet forwarding to the routers. Routers usually contain routing tables. These routers use dedicated control messages to build the routing table and respond to network status changes. In these networks, data packets are dump packets, they delegate routing and forwarding decisions to the routers. However, packets in cognitive packet networks can be smart. Cognitive packet networks use other control packets to distribute information across the network. Smart packets are responsible for exploring the network and they have the ability to route themselves in the network [17, 18]. Smart packets use routers as buffers to store and exchange information with other packets as well as the routers serving as processing units to run executable code carried by packets. In general, the smart packet format consisted of four fields as shown in Figure 2.2. The header field contains information about packet type, quality of service requirement, source address, destination address, etc. Cognitive Map (CM), which is used as a buffer to carry packet information is used to compute routing based on the packets QoS needed. Executable code field used to run specific codes on the routers. Finally, the payload field [17].

Dumb packets are guided using source routing method and the source node selects a path based on the best path discovered by smart packets.

Header	DATA	Cognitive Map	Executable Code

Figure 2.2 Representation of a CP [17]

Erol Gelenbe, et.al. introduced Cognitive Packet Network (CPN) [17],where intelligent packets have been used to route themselves in the network. In his work, the routing decision is made by Random Neural Networks. Smart Packets contain extra fields for Cognitive Map and Executable Code. Each router should process the content of the packet and a specific program interpreter is required to execute the codes

Cognitive Packet Networks is considered as another approach to solve problems of routing [6]. The process of selecting the next node is based on online sensing and monitoring of quality of service. Recurrent Neural Network (RNN) or other adaptive algorithm is used as a learning engine in CPN. Each node is equipped with a buffer named Mailbox (MB) which is used to exchange information between packets. A schematic diagram of CNP is shown in Figure 2.3.

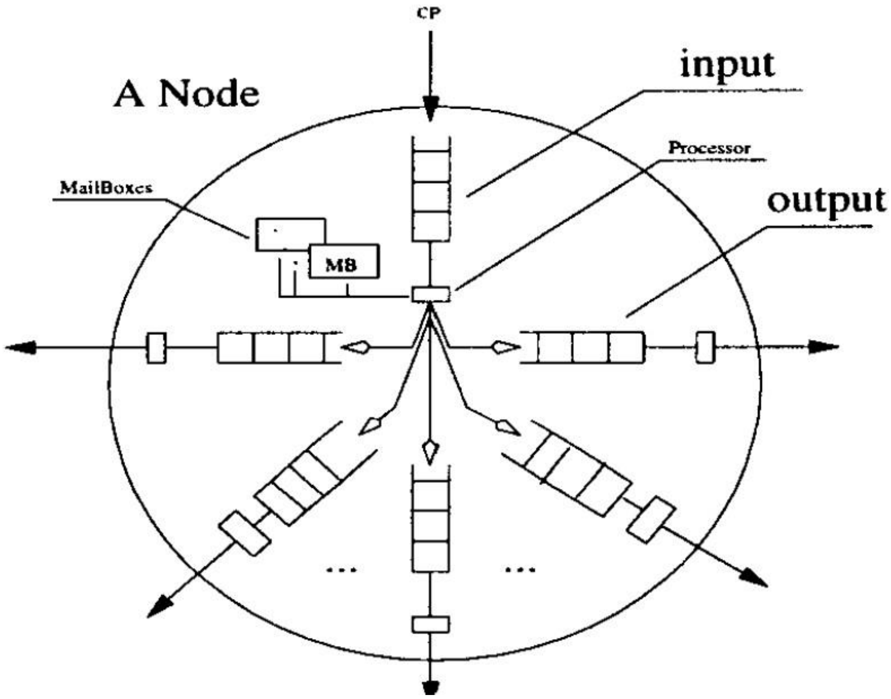
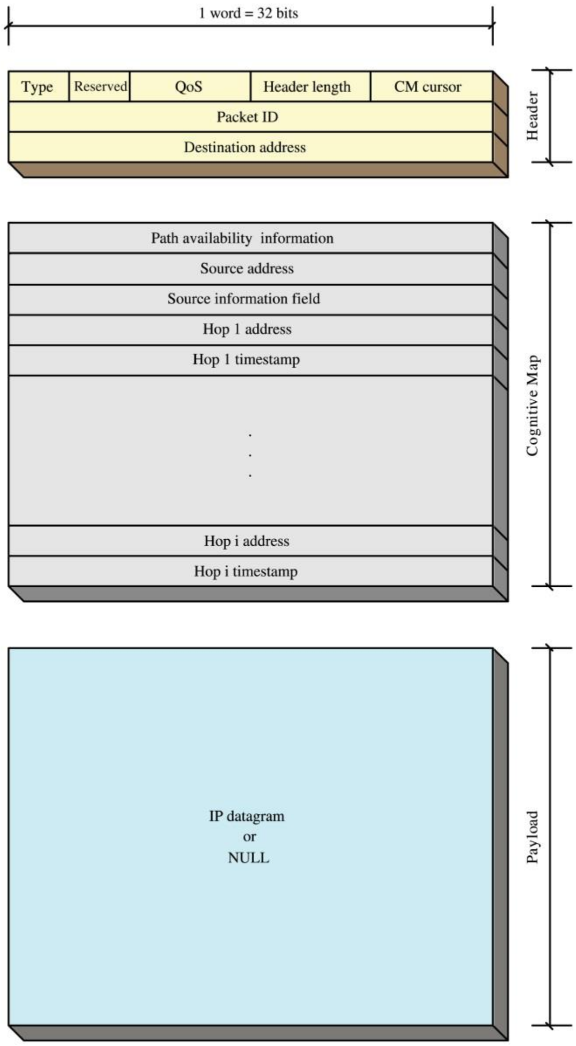


Figure 2.3 Contents of a CPN Node [12]

As the cognitive packets move in the network, they learn to move according to the required QoS. The packets in CPN can be categorized into three different types, Smart Packets (SPs), Dumb Packets (DPs) and Acknowledgement (ACKs). Smart Packets are responsible for the exploring and learning process in the network. When an SP reaches its destination, an ACK packet is sent to the source through the inverse route recorded by the smart packet. This ACK packet contains the routing information which will be delivered to the source node in order to route dumb packets using the source routing method [17-19].



.

Figure 2.4 Packet format in AHCPN [20]

Ad hoc CPN (AHCPN) [20] is an implementation of CPN in ad hoc networks where the algorithm tries to decrease power consumption. The algorithm tries to minimize the number of broadcasting in the network in order to save power. Whenever possible, smart packet will be unicasted. Smart packets check nodes energy in their search for route between source and destination. At the destination, the acknowledgment packet will carry the information of a route where nodes have higher battery energy. In order to decrease the packet overhead, the execution code in the smart packet has been removed. This will minimize the size of smart packets. Figure 2.4 shows the content of AHCPN packet. The first 32 bits identify the type of the packet, the desired QoS, header and cognitive map length, and cognitive map cursor is an index used to show the position of the node that is sending in the cognitive map. The second 32 bits are used to uniquely identify the packet. The last field in the header is the destination address. The cognitive map has a variable length. The first field is the path availability information which gives the probability that the route from the current node to the destination is available. Source and intermediate node’s addresses are also recorded in the cognitive field. A list of arrival times for intermediate nodes and departure time for the source is also recorded. Finally, the payload field contains the IP datagram.

CPN suffers from high overheads as the amount of control information added to the packet is high. CPN uses Random Neural Networks which adds more computation and resource usage to the network, as well as a vast amount of information (neural network statues) should be carried by the packets. CPN protocol infrastructure is completely different from the well-known IP layered approach of routing infrastructure. These and other factors led to less interest in CPN.

2.10 Channel assignment design strategies

The IEEE 802.11 specification offers many non-overlapping working channels. The 802.11b for example supports 3 non-overlapped channels and the 802.11a supports 12 non-overlapped channels [21]. Most of 802.11 based protocols are typically configured to operate on a single channel, although they support many non-overlapped channels. Using a single channel in an ad hoc network will effectively reduce the capacity due to the interference.

Different researches have been proposed to increase the capacity of wireless ad hoc networks. Multiple radio interfaces [22, 23], directional antennas [23], Multiple-Input Multiple-

Output (MIMO) techniques [24, 25], and others have been used to enhance the capacity of ad hoc networks.

The use of directional antennas can increase the power efficiency; however this will introduce a major challenge to Medium Access Control (MAC) protocol due to the angular reduction in range. Each node will need to discover its neighbours and know their directions in order to detect and exchange information with them [22]. The MIMO technique may use multiple antennae in the transmitter or the receiver or both of them. The transmitter antennae are simultaneously used to transmit parallel data streams (transmit diversity). The receiver antennae are used to receive the same information which will produce receiver diversity. The MIMO system can increase the throughput; however one major problem is the complexity [26].

Using multiple radio interfaces and assigning different channels to different radio interface will increase the capacity of ad hoc networks as this provides less interference between the nodes [27]. However this approach introduces the problem of channel assignment. The main problem of multi-channel multi-interface ad hoc network is the problem of finding efficient algorithm to assign channels to nodes in such a way that there will be less interference and higher connectivity. This also may require the algorithm to ensure that there are multiple paths between the nodes in the network in order to increase the reliability and robustness of the network.

A node in a multi-channel multi-interface ad hoc network should set up a common channel between it and its neighbour to be able to communicate with each other. However the limitation of available radio interfaces as well as the number of channels adds more restrictions on the process of channel assignment. Each node should minimize the number of shared channels while trying to increase the connectivity. Figure 2.5 shows an example of the problem of channel assignment. Figure 2.5a shows the network and node connections if a single channel and single interface is used. The network is fully connected; however the interference is too high as all the nodes share the same media and channel. It should be noted that even if four channels were available, in order for the nodes to communicate at a certain time they all have to use the same channel or else the connectivity will decrease as in Figure 2.5b. to increase the connectivity and decrease interference Figure 2.5 c and d show an example of using two interface cards and at the same time there are four available channels to use. In Figure 2.5c, where the network is optimized for maximum connectivity, the problem of interference arises. As there are two interface cards in each node, both node b and d are using one of their interface cards to connect to two of their neighbours. Figure 2.5d shows the network where there is no interference and all the nodes can send and receive simultaneously, however the network is not fully connected. Obviously, to solve the above network problem and have maximum connectivity and minimum interference each node should be equipped with three interface cards as well as the number of available channels should be five. It is clear that both the limitation in number of interface cards and available channels adds more complexity to channel assignment problems. Channel assignment can give different topology to the network depending on how communication channels are assigned [28].

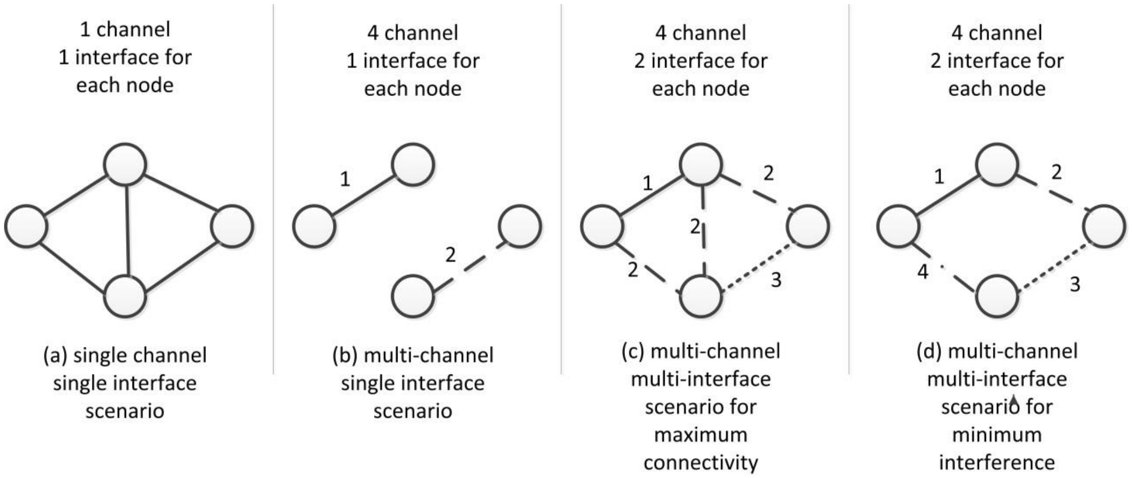


Figure 2.5 Trade-off between connectivity and interference (figure adapted from [28])

There are three approaches to solving the problem of channel assignment, fixed channel assignment method, dynamic channel assignment method, and hybrid channel assignment method. In the fixed channel assignment approach, a specific channel number is assigned to the interface card and remains fixed for the entire network lifetime or for a long time period with respect to its network life time. An advantage of this method is its simplicity, however it has many drawbacks. The limitation of the number of interface cards which is usually less than the available number of channels leads to inadequate utilization of available channels especially if a common channel assignment approach has been used. On the other hand, varying channel assignment may lead to network partitioning [29].

Unlike fixed channel assignment, dynamic channel assignment changes the assigned channel number of an interface card frequently in order to cope with network requirements.

However, frequent change of channel number introduces a problem as two neighbours cannot guarantee that they have a common channel in order to communicate and exchange control information. As neighbours need to share a common channel in order to communicate and discover each other, many approaches have been proposed to solve the problem. One approach is to force all nodes to return to a certain channel periodically to exchange information with each other. Slotted Seeded Channel Hopping (SSCH), is another method where each node changes its channel number based on pseudo-random sequences [30] periodically. This will enable nodes to meet at certain channels in order to exchange control information. Although dynamic assignment gives the nodes the opportunity to change channel and thus use all available channels, the problem of synchronization arises which is difficult to achieve in mobile ad hoc networks.

Hybrid channel assignment tries to overcome the problems introduced in static and dynamic channel assignments. Hybrid channel assignment uses fixed channels on some interface cards as well as dynamic ones on others. Most often the fixed channels in this strategy are used for exchanging control information.

2.11 Network security objectives

The total objective of any network is to deliver information to users. Security is an essential part of any network in order to protect network resources and prevents internal or external misbehaviour. The operation of any network can be easily disrupted if the security mechanism has not been added into the network. The main objectives of network security could be summarized as authenticity, confidentiality, integrity, availability, access control, and non-repudiation [5, 28, 31].

- **Authentication:** is an important objective to verify the identity of the node and ensure that it is the right party. When two parties need to communicate with each other, they should be able to ensure that the other user is who they claim to be (genuine).
- **Confidentiality**: is concerned with the content of data message and protect it from unauthorized disclosure. Any two end users should guarantee that they are they only one that can read the content of a message and the content of the message remains secret to other parities in the network. An exception is the authorized parties where they can understand the content of the message.

- **Integrity**: as the data messages move in the network, the user must be sure that the content of the message has not been modified by unauthorized parties. It is a technique to make sure that the received message has not been tampered with.
- **Availability**: this measure is concerned with the probability of an entity to provide services when needed.
- **Access control**: guarantees that only authorized parties have the right to access resources and services that they have the right to access.
- **Non-repudiation**: concerned with the fact that any party which sends a message cannot deny the sending as well as the fact that any receiver cannot deny the reception of a message.

2.12 Threats in ad hoc networks

The success of any network strongly depends on whether the network is secure or not. In general, characteristics and constrains of ad hoc networks are main causes of the threats. The wireless medium poses a great risk to ad hoc network security. Moreover, the fact that each node in ad hoc network works like a router adds extra risk to the network. The infrastructure-less of the network as well as the distributed and non-centralized structure of ad hoc network makes it Vulnerable to various kinds of attacks.

Attacks in ad hoc networks can be divided into two categories, passive and active attacks. Attacks can target any layer in protocol stack. Examples of attacks that target physical layer are vulnerable to jamming, interceptions, and eavesdropping attacks. Other attacks like repudiation and data corruption may target application layer. Other attacks like denial of service attack, and man in the middle attacks may target multiple layers of protocol stack.

2.12.1 Passive and active attacks

Passive attacks are the kind of attacks that do not disturb network operation. The attacker’s goal is to obtain information from the network without any disruption. Passive attacks are divided into two groups; eavesdropping and traffic analysis. Eavesdropping is done by tapping the communication line. Because the transmission media in mobile ad hoc network is wireless, it is easier to tap ad hoc networks [32]. However, as the transmission range of ad hoc network is shorter than other kinds of wireless network, this makes ad hoc networks less vulnerable to eavesdropping attacks. An attacker needs to get within the node transmission range to be able to tap. On the other hand, traffic analysis tries to extract information by analysing the traffic pattern in the network. For example, when the clustering method is used in ad hoc networks, the cluster head could be detected according to the traffic pattern in the network. Traffic analysis is useful in organizing and targeting the attack on the network where the location of important nodes could be detected [33].

Active attack is the other type of attack where the attacker affects the operation of the network. This kind of attack may degrade the network performance. Active attacks could be detected as they attempt to alter, inject, delete or destroy the data being exchanged in the network. Examples of these kinds of attacks are masquerade, replay and message modification denial of service attacks.

2.13 Routing attacks in ad hoc networks

Attacks that target network layers in ad hoc networks mainly try to disrupt the operation of the network. Attacks with the objective of disrupting the availability of services in the network are known as denial of service attacks. These attacks either try to disrupt routing procedure or consume resources. Examples of these attacks are [34]:

- Hello flood attack: routing protocols usually use hello messages to maintain vicinity information where nodes can detect their neighbours. In this kind of attack, the attacker gets advantage from this and cons other nodes to make them believe that the attacker is their neighbour. The attacker increases the transmission power of its hello messages which in turn increases the range of hello messages. Distant nodes may send packets to this node which is not in their transmission range. This in turn will degrade network performance.
- Wormhole attack: The attackers of this kind consist of at least two nodes. A node in this attack listens to the packets around it and sends it to the other node. The second node then replays the packets. Attacking nodes use special links or tunnelling to exchange information.. These links are usually fast channels that connect the malicious nodes. When the neighbours of the second node receive the replayed packet, they start to think that they are one hop away from those nodes that are neighbours of the first attacker node. In fact if the wormhole transfers all packets and has high speed channel, it will enhance network performance. The only reason for this kind of attack will be to

collect as much as possible of the information passing through the attackers. However, if only part of the packets have been forwarded, just control packets or a selective control and data packets, this will disrupt the routing protocol.

- Detour attack: the attacker tries to convince other nodes to send packets through routes that are not optimal by showing them as optimal routes. This may poison routing tables and make some parts of the network unreachable.
- Silent route changes: the attacker forwards the packet to an unintended node causing the packet to move in a different route than the intended one.
- Routing table overflow: the attacker tries to create fake routing table entries which lead to overflow in the routing table. This will prevent new legitimate entries from being created in the routing table.
- Sink hole attacks: in this kind of attack, the attacker tries to convince other nodes that it has the best route to the destination. This make other nodes think that the attacking node has best route will prefer sending messages through the attacking node. The main reason for this kind of attack is collecting information about the network. Moreover, it also leads to jamming the network.
- Black hole attacks: the goal of this attack is to drop all packets that have been forwarded to the attacker.
- Gray hole attack: the attacker selectively drops packets that have been forwarded to it. This kind of attack is harder to detect than black hole attack
- Packet replication: the attacker tries to consume network resources by sending replicates stale packets.
- Sybil attack: where the attacker presents multiple identities. This will corrupt routing table, consume bandwidth as the node tries to send to non-existing nodes, and may result in route table poisoning.
- Rushing attack: Rather than modifying control packets to become attractive nodes, this kind of attack modifies the timing of forwarding route request messages. This will increase the probability that routes which include the attacker will be selected before the better routes.

2.14 Misbehaving

The distributed nature of MANET means it mainly depends on the cooperation of its entities in order to deliver information successfully and efficiently around the network. Attacks on MANETs are divided into external and internal attacks. External attacks are carried out by external nodes that do not belong to the network and are usually known as intruder nodes. Malicious misbehaviour by external node is interested in attacking and damaging the network. Internal attacks are carried out by nodes belonging to the network itself. These attacks are more difficult to detect as they are carried out by nodes that have been authenticated and considered legitimate members of the network, but at some point have become compromised. Internal attacks could be carried out by selfish nodes. Although selfish nodes are part of the network, they refuse to cooperate with other nodes in order not to consume their resources. Apart from above where nodes deliberately misbehave, other misbehaving may occur due to mistakes. A node may mistakenly deliver information to unauthorized nodes [35].

The damage caused by selfish behaviour is less than the damage caused by malicious misbehaviour. A selfish node tries to maintain its resources, and refuses to cooperate with other nodes in the network. However a selfish node does not attend to disturb network operation. Moreover, it is usually easy to convince selfish nodes to cooperate by offering motivations like services in favour of cooperation.

Malicious nodes' main objective is to devastate network operation. It does not look to get any services from the network. The only solution to malicious attacks is to detect and isolate the malicious nodes.

2.15 Trust and reputation management in MANETs

Over the last few years, mobile devices have received tremendous attention. The advance in communication technology and the widespread use of these mobile devices have introduced new problems. The security, safety and privacy of the users have become at risk. Information security has become a major research area to overcoming this problem. As these mobile devices become more compact, mobile and affordable, this will decrease their tamper-resistant property. As MANETs consist mainly of a group of mobile devices where each device acts as a router and can make decisions individually, the probability of attacks increases. Mobile nodes in MANET have limited resources and this will increase the opportunity of selfish behaviour in order for a node to reserve their resources. Moreover, these mobile devices could be seized and physically captured by an adversary and this person could behave maliciously to disrupt the network. In general, if a device is captured and permeated, the adversary will have enough information about the system as was available to node itself. If standard cryptographic is being used in the network, then the adversary has a device that is authenticated to the network and he can start an attack on the system and cause maximum damage to the network. Misbehaviour attacks, especially an internal attack, are difficult to be detected by standard cryptographic system [35]. Moreover, most of these techniques that are based on cryptographic techniques in order to separate non legitimate nodes from legitimate nodes are suffering from drawbacks. In general, these approaches introduce a heavy traffic load on the network which consumes network resources. This is considered as the main drawback of using cryptographic techniques for security in ad hoc network. More difficulty comes from the distributed nature of ad hoc networks which requires distributed authentication and key management services [36, 37].

These kinds of attacks could be dealt with by using reputation and trust based systems. Reputation is the degree of confidence of one entity towards another entity. This degree could be positive, negative or neutral value. Reputation directly depends on the interaction with the entity or could be indirectly influenced by other opinions regarding the same entity. Reputation is cumulative and represents the behavioural history of an entity. Trust, on the other hand, is built upon the reputation of the entity. According to social science, trust could be defined as the expectation of an entity (trustor) on how another entity (trustee) will act in future depending on the degree of reputation of trustee according to the trustor’s perspective [38]. However as long as the action of the trustee is in the future, this adds uncertainty to the result of the action especially as the trustor has no control over the trustee’s behaviour. Trust is a very complicated measure in real human life. A person may intentionally believe in another person in order to finish his job. An example is a person may trust Amazon more than eBay and this trust affects his decisions when buying from these sites. However, this depends on his experience or maybe he just find the item at Amazon and just want to buy it. The degree of trust in Amazon and eBay is a consequence of that person’s experience and may be affected by other opinions. That doesn’t mean another person will also have the same opinion. However, if the second person knows the first one, his opinion may be affected by the first person’s opinion. The effect may be positive or negative depending on how much trust the second person has in the first person. Finally, even if the second person concludes to the same trust idea, he may buy from eBay and intentionally put his trust in eBay for some beneficial reason like price for example.

Trust in human lives is used widely and it is usually a very complicated process in real life. Many parameters in life and personal experience effect a person's decisions and changes according to how trust is calculated. Many approaches have been done to apply and simulate trust. For example in an economic transection, each party calculates trust based on costs and benefits and trust is given if any action of the other party will always be beneficial. Trust models also have been implemented to detect node misbehaviour in MANETs. Trust is an effective way to discover the intentions of selfish and malicious nodes. Various trust approaches have been implemented in MANET detect node misbehaviour. These misbehaviours are classified as non-cooperative behaviours. Reputation and trust systems in MANET are usually based on observing neighbour nodes. Using this technique, a node can detect many kinds of misbehaviours. It should be easy for any monitoring system to detect if a packet has been modified. Reputation and trust systems should also be able to detect black hole attacks. Reputation and trust performs better if the attack is not a cooperative group attack. Although the trust system offers a better solution to detect malicious behaviour than cryptographic based system, it has some drawbacks. Reputation and trust systems require time to build reputation tables and the trust algorithm may itself become a target for the attacks.

2.16 Reputation and trust system model

The process of building a reputation and trust system is mainly made up of five parts, initialization, information gathering, information sharing, information modelling, and decision making [35].

2.16.1 Initialization

The initialization process takes place when a new system is built or when a new node joins the network. When the reputation and trust system starts, each node should assign a value of trust and reputation to other nodes in its tables. Three options are available:-

- All trusted: A system may be built on the basis that every node in the system is trusted until they show themselves not to be. In this case a positive value is assigned to each node in the trust table. When a new node joins the network it will be trusted until it shows the opposite.

- All suspicious: this approach starts by setting all nodes to be distrusted. In which case, a new node that joins the network is also considered distrusted until it shows the opposite.
- Unknown: all nodes’ trust values are set to a neutral value. That means the exact trust value is unknown and node trust value will be built based on future interaction.

2.16.2 Information gathering

The first step in any MANET reputation and trust system is information collection. Each node in MANET can communicate and monitor its neighbour nodes that lie within its transmission range. As neighbour nodes are the nodes that are in direct access and considered as having higher importance to a node, the node collect information about these nodes by observing and monitoring their behaviours. The information gathered may be as a result of direct interaction with other neighbours. This information also may be collected by monitoring the interaction between neighbour nodes.

2.16.3 Information sharing

There are two types of information in reputation and trust systems, first-hand information and second-hand information, although some systems only use first-hand information. First-hand information is the kind of information gathered by the node itself, it is reliable and the nodes have full confidence about this information. On the other hand, second-hand information is other nodes’ information that has been shared with the node. This information is not reliable as this information is vulnerable to false report attacks. Information sharing speeds up the process of building reputation tables of nodes. Over time, nodes in the network will share the same point of view. However, sharing information could be attacked, especially in group attack methods. In group attacks some nodes may start strategic attacks by cheating other nodes to believe that other attacker nodes are highly reputed and sabotaging the reputation of trustworthy nodes.

Some approaches to solving this problem rely on sharing only positive or negative information. However both of these approaches use half of the information in the system which delays the building up process of the reputation table. Moreover, they both suffer from the same problem as above as the attacker can increase the reputation of other attack nodes in positive information sharing methods, and decrease the reputation of trustworthy nodes in negative information sharing methods. Another method is to share all of the information;

however the updating process depends on giving weights to each sender of information based on the first hand reputation information.

An important point raised here is when, how and what to share the information in the network. There are two main methods to diffuse the information in the network; proactive diffusion and reactive diffusion. In proactive diffusion, reputation information is diffused periodically every predefined interval of time regardless of any reputation change in the system. On the other hand, in reactive reputation diffusion, whenever the amount of information change is greater than predetermined value, the information will be published. To decrease the traffic consumed by diffusing reputation messages, some methods piggyback the reputation information on control packets. Moreover, the diffusion may be global to all nodes in the network or a specific group of nodes like first or second neighbour nodes. Finally, the information shared could be the first-hand reputation information only or overall reputation information.

2.16.4 Information modelling

Information modelling deals with how to store reputation information and how to deal with the first-hand and the second hand information. A system may store only one table representing the reputation of nodes, while another may choose to have a table for reputation and another for trust. As second hand reputation information is not reliable, a weighting method may be used to accept it based on how much the sender is trusted. Finally, because reputation information is accumulated over time, there should be some aging process and some forgetting process should be used depending on whether old information is more important than new or not.

2.16.5 Decision making

The final part of reputation and trust systems is decision making. After gathering information and modelling first and second hand information, the system should decide whether a node is trustworthy or not. This is a binary decision which makes the node forward packet to trusted nodes and stop sending packets to malicious nodes. As long as this is a binary decision, the final value of trust is compared to predetermined value to decide where to set the node as trusted or not.

2.17 Trust property

Many characteristics are related to trust which define how trust is built in MANET [37]:

1. Trust is not reciprocal: that means if a node trusted another node, the other node need not trust the first node based on the first node's trust relationship.
2. Trust is not transitive: for any three nodes, if the first one trusted the second and the second trusted the third, the first one may or may not trust the third one. There is no rule that forces the first one to trust the third.
3. Trust is reflexive: every node trust in its own information
4. Trust is a measure of uncertainty: whenever trust is needed, there is a place for uncertainty and there is a probability for a node to be trusted or not.
5. Trust is time dependent: trust depends on reputation which is gathered over a period of time. Whenever there is a need for trust systems there should be enough time to build the reputation information.

2.18 Summary

This chapter starts with brief explanation of mobile ad hoc networks and explains their properties. The chapter also gives an overview on routing and security protocols in mobile ad hoc networks. A detailed study on different routing approaches is given, showing different categories of MANETs routing protocols, and focusing on the joint problem of routing and channel assignment which will be addressed in chapter four. The chapter also focuses on cognitive routing protocol as it is the basis of smart data packet protocol which will be introduced in chapter five. Moreover, the chapter addresses routing security issues and different types of attack that are related to MANETs focusing on trust and reputation system in order to add security to routing protocols.

References

[1] D. Sivakumar, B. Suseela and R. Varadharajan, "'A survey of routing algorithms for MANET," *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on,* 2012, pp. 625-640.

[2] Lei Chen and W.B. Heinzelman, "'A Survey of Routing Protocols that Support QoS in Mobile Ad Hoc Networks," *Network, IEEE*, vol. 21, no. 6, 2007, pp. 30-38.

[3] E. Alotaibi and B. Mukherjee, "'A survey on routing algorithms for wireless Ad-Hoc and mesh networks," *Computer Networks*, vol. 56, no. 2, 2012, pp. 940-965.

[4] A. Boukerche, B. Turgut, N. Aydin, M.Z. Ahmad, L. Bölöni and D. Turgut, "'Routing protocols in ad hoc networks: A survey," *Computer Networks*, vol. 55, no. 13, 2011, pp. 3032-3080.

[5] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu and Lixia Zhang, "'Security in mobile ad hoc networks: challenges and solutions," *Wireless Communications, IEEE*, vol. 11, no. 1, 2004, pp. 38-47.

[6] D.B. Johnson, D.A. Maltz and J. Broch, "'DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," *Addison-Wesley*, 2001, pp. 139-172.

[7] C.E. Perkins and E.M. Royer, "'Ad-hoc on-demand distance vector routing," *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, 1999, pp. 90-100.

[8] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum and L. Viennot, "'Optimized link state routing protocol for ad hoc networks," *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, 2001, pp. 62-68.

[9] C.E. Perkins, P. Bhagwat, C.E. Perkins and P. Bhagwat, "'Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers;" *SIGCOMM Comput.Commun.Rev.*, vol. 24, no. 4, 1994, pp. 234-244.

[10] P. Samar, M.R. Pearlman and Z.J. Haas, "'Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks," *IEEE/ACM Trans.Netw.*, vol. 12, no. 4, 2004, pp. 595-608.

[11] F. Cadger, K. Curran, J. Santos and S. Moffett, "'A Survey of Geographical Routing in Wireless Ad-Hoc Networks," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 2, 2013, pp. 621-653.

[12] Sun-Ho Lee, Eunjeong Choi and Dong-Ho Cho, "'Timer-based broadcasting for power-aware routing in power-controlled wireless ad hoc networks," *Communications Letters, IEEE*, vol. 9, no. 3, 2005, pp. 222-224.

[13] C.E. Jones, K.M. Sivalingam, P. Agrawal and J.C. Chen, "'A Survey of Energy Efficient Network Protocols for Wireless Networks," *Wirel.Netw.*, vol. 7, no. 4, 2001, pp. 343-358.

[14] Jiageng Li, D. Cordes and Jingyuan Zhang, "'Power-aware routing protocols in ad hoc wireless networks," *Wireless Communications, IEEE*, vol. 12, no. 6, 2005, pp. 69-81.

[15] K. Kaur and I.K. Aulakh, "'Power aware metrics and routing techniques in MANETs," *Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on*, 2011, pp. 369-373.

[16] S. Singh, M. Woo and C.S. Raghavendra, "'Power-aware routing in mobile ad hoc networks," *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, 1998, pp. 181-190.

[17] E. Gelenbe, Zhiguang Xu and E. Seref, "'Cognitive packet networks," *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on*, 1999, pp. 47-54.

[18] E. Gelenbe, R. Lent and Zhiguang Xu, "'Networks with cognitive packets," *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2000. Proceedings. 8th International Symposium on*, 2000, pp. 3-10.

[19] R. Lent, "'Smart packet-based selection of reliable paths in ad hoc networks," *Design of Reliable Communication Networks, 2005. (DRCN 2005). Proceedings.5th International Workshop on*, 2005, pp. 5 pp.

[20] E. Gelenbe and R. Lent, "'Power-aware ad hoc cognitive packet networks," *Ad Hoc Networks*, vol. 2, no. 3, 2004, pp. 205-216.

[21] Hon Sun Chiu, K.L. Yeung and King-Shan Lui, "'J-CAR: An efficient joint channel assignment and routing protocol for IEEE 802.11-based multi-channel multi-interface mobile Ad Hoc networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 4, 2009, pp. 1706-1715.

[22] P.S. Kiran, "'A Survey on Mobility Support by Mac Protocols Using Directional Antennas for Wireless Ad Hoc Networks," *Ad Hoc and Ubiquitous Computing, 2006. ISAUHC '06. International Symposium on*, 2006, pp. 148-153.

[23] Wu Zemin and Qiu Zhenglun, "'A Survey on Directional Antenna Networking," *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, 2011, pp. 1-4.

[24] J. Mietzner, R. Schober, L. Lampe, W.H. Gerstacker and P.A. Hoeher, "'Multiple-antenna techniques for wireless communications - a comprehensive literature survey," *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 2, 2009, pp. 87-105.

[25] S. Schwarz, J.C. Ikuno, M. Simko, M. Taranetz, Qi Wang and M. Rupp, "'Pushing the Limits of LTE: A Survey on Research Enhancing the Standard," *Access, IEEE*, vol. 1, 2013, pp. 51-62.

[26] Jiun-Ying Wu, Wen-Rong Wu and Nan-Chiun Lien, "'Low-complexity MIMO detection using a list projection technique," *Signal Processing and Communication Systems, 2008. ICSPCS 2008. 2nd International Conference on*, 2008, pp. 1-5.

[27] C. Toham and F. Jan, "'Multi-interfaces and Multi-channels Multi-hop Ad hoc Networks: Overview and Challenges," *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, 2006, pp. 696-701.

[28] Y. Xue, Y. Cui, and K. Nahrstedt, "'Channel Assignment Strategies for Wireless Mesh Networks,"in Wireless Mesh Networks, Architectures and Protocols. Anonymous , Springer Science+Business Media, LLC, 2008, pp.113-142.

[29] R. Soua and P. Minet, "'A survey on multichannel assignment protocols in Wireless Sensor Networks," *Wireless Days (WD), 2011 IFIP*, 2011, pp. 1-3.

[30] Long Le, "'Practical Multi-Channel MAC for Ad Hoc Networks," *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, 2010, pp. 1-9.

[31] C.R. Erdal Çayırcı, "'Security in Wireless Ad Hoc and Sensor Networks,"John Wiley & Sons Ltd., 2009.

[32] A. Mishra, K. Nadkarni and A. Patcha, "'Intrusion detection in wireless ad hoc networks," *Wireless Communications, IEEE*, vol. 11, no. 1, 2004, pp. 48-60.

[33] Yang Qin and Dijiang Huang, "'Least Squares Disclosure Attack in Mobile Ad Hoc Networks," *Communications (ICC), 2011 IEEE International Conference on*, 2011, pp. 1-5.

[34] A.K. Abdelaziz, M. Nafaa and G. Salim, "'Survey of Routing Attacks and Countermeasures in Mobile Ad Hoc Networks," *Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on*, 2013, pp. 693-698.

[35] Azzedine Boukerche, "' Algorithms and Protocols for Wireless and Mobile Ad Hoc Networks ,"Wiley-IEEE Press, 2008.

[36] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato and A. Jamalipour, "'A survey of routing attacks in mobile ad hoc networks," *Wireless Communications, IEEE*, vol. 14, no. 5, 2007, pp. 85-91.

[37] Han Yu, Zhiqi Shen, Chunyan Miao, C. Leung and D. Niyato, "'A Survey of Trust and Reputation Management Systems in Wireless Communications," *Proceedings of the IEEE*, vol. 98, no. 10, 2010, pp. 1755-1772.

[38] Wikipedia, **Trust (social sciences)**, Accessed: July 2013, [Online] Available: http://en.wikipedia.org/wiki/Trust_(social_sciences).

Chapter Three

Swarm Intelligence and Routing

3.1 Introduction

Nature conceals many mysteries. In the past, some of these behaviours like ants foraging and flight flocks were considered as magical secrets of nature. Others were considered as fact like the falling of raindrops and rivers which always run to the sea. These and other phenomena inspired some people to study and understand their secrets. The unravelling of many of these mysteries and secrets led to the foundation of new artificial intelligence science known as Swarm Intelligence (SI). Swarm intelligence is inspired by the social behaviour of insects and other animals and is widely used to solve optimization problems. Swarm intelligence is typically made up of a population of simple individuals interacting locally with one another and work in cooperative distributed fashion. This social behaviour could be seen in many insects organizations in which they act as super organisms. Their success does not lie in the ability or the strength of an individual but in the social behaviour of such organization. These behaviours have been mimicked into optimization algorithms.

Swarm intelligence is a good replacement for traditional search algorithms as traditional algorithms are facing big challenges and become inefficient in solving non-deterministic polynomial-time hard (NP-hard) optimization problems. As the computational complexity of a problem increases, more time and computation power are required to solve such a problem. Routing in mobile ad hoc networks is considered NP-hard problems. Due to the limitation power and time of routers, other methods are used to solve such problems. Swarm intelligence is one of these methods that has been adapted to solve routing problems.

This chapter presents some methods of swarm intelligence and their applications in routing protocols.

3.2 Self-organizing systems

A Self-organizing system is defined as a system with a set of dynamic mechanisms that changes its states to reflect some structure or organization without any external guidance, these changes could be as a response to some operating conditions or environmental conditions change [1]. The actual works on self-organizing systems returning back to the previous century although self-organization discussions go right back to the ancient Greek ages [2].

Self-organizing algorithms have many drawbacks. In general, it is not guaranteed that the global optimum solution could be found using self-organizing algorithms. Instability could appear in the algorithm and especially if opposing actions appeared among system units.

Many artificial self-organizing systems have been implemented based on the self-organizing mechanism inspired from nature. Self-organizing systems in nature can be classified into three types; Social self-organizing systems, physical self-organizing systems and biological self-organizing systems.

Ant, termite and bee colonies are examples of social self-organizing systems in nature. Other examples are schools of fish and flocks of birds. Social behaviour could be seen in many organizations; starting from small bacteria cells, up to human behaviour. The interaction among the individuals in a system serves a global idea of survival and continuity of such a system. Individuals in these systems may be unaware of the ultimate goal; however the interaction and the collaborative work among these individuals serve the ultimate goal.

Originally, the appearance of macroscopic patterns out of some chemical and physical processes led to first development of self-organizing theories. Physical self-organizing systems share a common characteristic. When they reach a threshold point, the state of systems immediately change. An example could be seen in mixture of chemical which can lead to organized oscillation [1]. Another example of non-living system is the way rain drops join together to make rivers and the find best path to the sea.

Examples of biological self-organizing systems are the immune system of mammalians, and the process of evaluation in nature through natural selection.

3.3 Real ants in nature

One of the well-known swarm intelligence algorithms is the Ant Colony Optimization. This is a heuristic stochastic search algorithm based on population search of optimal resolution. The inspiring source of ACO is the food foraging behaviour of some real ant species. In their search for food, ants first try to explore the surrounding of their nest randomly. When a food is found, the ant carries a sample of it back to the nest. At the same time it deposits a chemical pheromone trail on the returning path. At the nest other ants will check the sample and may join the path to the food through the use of the pheromone track. As more ant travel from the food source back to the nest they lay more pheromones on the track. The more intense the pheromone, the more attractive the path is for other ants and the path will attract more ants. The main principle behind these interactions is called stigmergy, or communication through the environment [3].

The behaviour of real ants food foraging is adaptable and can lead to the shortest path between the nest and food source. Whenever a path becomes expired due to some external environmental changes like obstacles, ants are able to find another shortest path to the food source.

3.4 Ant colony optimization

ACO is an artificial ant algorithm that mimics the behaviour of the real ants and is widely used to solve optimization problems. ACO makes use of groups of independent artificial ants who are able to communicate with each other through an indirect pheromone-based communication. Ants work cooperatively and share their solutions with each other using pheromone intensity on each link. This work usually leads to good solutions to problems.

Italian scientists Dorigo M. et. al. firstly presented the heuristic algorithm known as Ant-System (AS) in 1991 [4, 5]. The Travelling Salesman Problem (TSP) is used as an example application. TSP is a problem of finding the shortest closed loop tour while visiting all cities. Each should be visited exactly once. TSP has been widely used for testing optimization problems. It is considered as a NP-Hard problem [6].

The problem defines N number of cities. The distance between city i and city j is $d_{ij.}$. The problem could be symmetric ($d_{ij}=d_{ji}$) or asymmetric (d_{ij} not equals d_{ji}). Each edge or link between two cities (i and j) is associated with pheromone intensity value τ_{ij}. The algorithm starts with m ants. Each ant is randomly placed in a chosen city. Starting from there, each ant moves in N steps to complete a tour. In order to choose the next city, the ant uses a random proportional transition rule to select a city between all non-visited cities. The algorithm is repeated for $t_{\max}$ iterations. The random proportional transition rule is given by

$$p_{ij}^{k} = \begin{cases} \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}} & if \quad j \in N_i^k \\ 0 & otherwise \end{cases} \tag{3.1}$$

where p_{ij}^{k} is the probability of ant k selecting city j from city i. N_i^k represents the set of feasible non visited cities by ant k from city i. η_{ij} is the inverse of the distance and called visibility , is equal to 1/ d_{ij}. α and β are the relative weight controlling parameters of pheromone intensity and visibility respectively. When $\alpha = 0$, the probability of selection is based on the visibility values, i.e. heuristic desirability. The selections of the closest cities are more preferable. On the contrary, when $\beta = 0$, the selection is based on pheromone intensities.

After compiling a tour, each ant updates the pheromone level of the path it has visited. The amount that is added to each edge it has visited is according to the following

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q/L^{k}(t) & if \; edge(i,j) \in T^{k}(t) \\ 0 & if \; edge(i,j) \notin T^{k}(t) \end{cases} \tag{3.2}$$

where $\Delta\tau_{ij}^{k}(t)$ is the amount of pheromone update added to edge *i,j* by *k* ant at time *t*. $T^{k}(t)$ is the tour taken by ant *k* at time *t*. Q is the amplification constant. $L^{k}(t)$ is the length of the tour that is taken by *k* ant at time *t*.

At the same time, each pheromone intensity decays after each tour. The new pheromone intensity will be

$$\tau_{ij}(t) = (1-\rho) * \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}\,(t) \tag{3.3}$$

where ρ is the decay coefficient and its value is between zero and one, and m is the number of ants. The algorithm is repeated until satisfaction quality is reached or when a maximum number of iterations are reached.

Many variations to the ant system algorithm had been proposed to improve the algorithm. MAX-MIN Ant System (MMAS) introduced by [7, 8] is based on the same principles of AS with the variation in the method of pheromone updating procedure. Only the best solution is chosen from current iteration or throughout all previous iterations, which is used to update pheromone intensities. Pheromone updating only occurs for the best chosen selected path. The values of pheromone intensities are bound between two values which are called τ_{max} and τ_{min}. The MMAS updates pheromones according to the following equation

$$\tau_{ij}(t) = \max\left\{\tau_{min}\ , \min\left\{\tau_{max}\ ,\ (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}^{best}(t)\right\}\right\} \qquad (3.4)$$

where $\Delta\tau_{ij}^{best}(t)$ is the amount of pheromone updates belonging to the best solution through i, j edge.

$$\Delta\tau_{ij}^{best}(t) = \begin{cases} Q/L^{best}(t) & if\ edge(i,j) \in T^{best}(t) \\ 0 & if\ edge(i,j) \notin T^{best}(t) \end{cases} \qquad (3.5)$$

where Q is the amplification constant. $L^{best}(t)$ is the minimum length of the tour chosen from all tours that passed through edge i, j at time *t*. $T^{best}(t)$ is the best tour chosen from all tours at time *t*.

Dorigo and Gambardella in 1997 introduce the Ant Colony System (ACS) [9, 10]. ACS differs from AS in both the transition and pheromone updating methods. The selection of the next city depends on the uniform probability function q over the [0, 1]. The ant moves to the next best *j* city for probability of q below of equal to q_o.

$$j = \arg max_{h \in N_i^k} \left\{\tau_{ih}(t), [\eta_{ih}]^{\beta}\right\} \qquad if\ q \leq\ q_o \qquad (3.6)$$

For all other values of q that are greater than q_o, ACS uses the same method of AS to select the next city.

$$p_{ij}^{k} = \begin{cases} \frac{[\tau_{ij}][\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k}[\tau_{ij}][\eta_{il}]^{\beta}} & if\ \ j \in N_i^k \\ 0 & otherwise \end{cases} \qquad if\ q >\ q_o \qquad (3.7)$$

Like the MMAS, ACS only updates the best bath. At the same time, a local pheromone updating procedure is used whenever an ant moves from one city to another.

$$\tau_{ij}(t) = (1-\rho) * \tau_{ij}(t) + \rho * \tau_o \quad (3.8)$$

where τ_o is the pheromone's initial value which is set to 1/(L_{nn}. *N).* L_{nn} is the length of a tour produced by the nearest neighbour algorithm.

3.5 Ant colony and routing problem

Mobile ad hoc network (MANET) routing has motivated many researchers in the past two decades. A mobile ad-hoc network is based on self-organizing networks where each node participates in the routing protocol. Since the topology of MANET network is dynamic and frequently changing, it composes great challenges to the routing algorithms. A number of routing methods have been proposed to address these problems [11-15].

In general, routing algorithm is an optimization process that tries to maximize network performance while minimizing costs. Swarm intelligence has shown to be a good optimization algorithm and has been applied to the field of routing optimization. In [16] the author describes the fundamental value of swarm intelligence and how the cooperative and collaborative work of simple agents could be used in solving routing problems. Ant Colony Optimization (ACO) is the most popular algorithm among other Swarm algorithms. Ant Based Control algorithm (ABC) [17] is considered the first SI routing algorithm for telecommunication networks. ABC algorithm addresses load balancing problem through using ant agents. The routing algorithm is proposed to work on circuit switched networks. The ants move in one direction from sources to other nodes. As ants move they deposit pheromones which will eventually guide data packets.

ANTNET [18] introduced to work on packet switch networks. ANTNET uses forwards and backwards ants. ANTNET uses distance vector for data routing, while it uses source routing for control packets. Thus it will introduce high overheads especially in large networks. Ant-AODV [19] is a hybrid routing algorithm that combines the idea of AODV and ant-based algorithm. It uses a small number of ants that keep moving around the network in a random manner. Each visited node will be aware of previous n visited nodes by the ants and update its routing table accordingly. Ant-AODV reduces the end to end delay but it increases the overhead of route discovery and maintenance. Ant Colony Based Routing Algorithm (ARA) [20] is using distance vector routing and supports multipath. It is a reactive routing algorithm. The algorithm tries to limit the overhead caused by ants but it losses the proactive feature of ants algorithms. Termite [21] routing protocol is an enhanced version of ABC and is related to

ANTNET. In Termite each ant is allowed to carry a fixed predetermined amount of pheromones (can carry one pebble). It simulates how termites build hills in natures.

3.6 River Formation Dynamics

RFD is another type of swarm intelligence which simulates different natural behaviour. The main inspiration behind RFD is how the water forms rivers in nature [22]. In nature, when rain falls on the mountains it joins together forming rivers and moves towards lower lands which eventually fall into the sea. As these water drops move, they change the environment by eroding the surface by an amount proportional to the decreasing slope of the surface and deposits carried sediments on a flatter surface. The speed and direction of moving water drops depend on the gradient of the surface. In this way, the altitude of the surface is changed and water drops creates rivers. Progressively, new water drop will reinforce the best shortest path which will be considered as an optimal path between the place where it is raining and the sea.

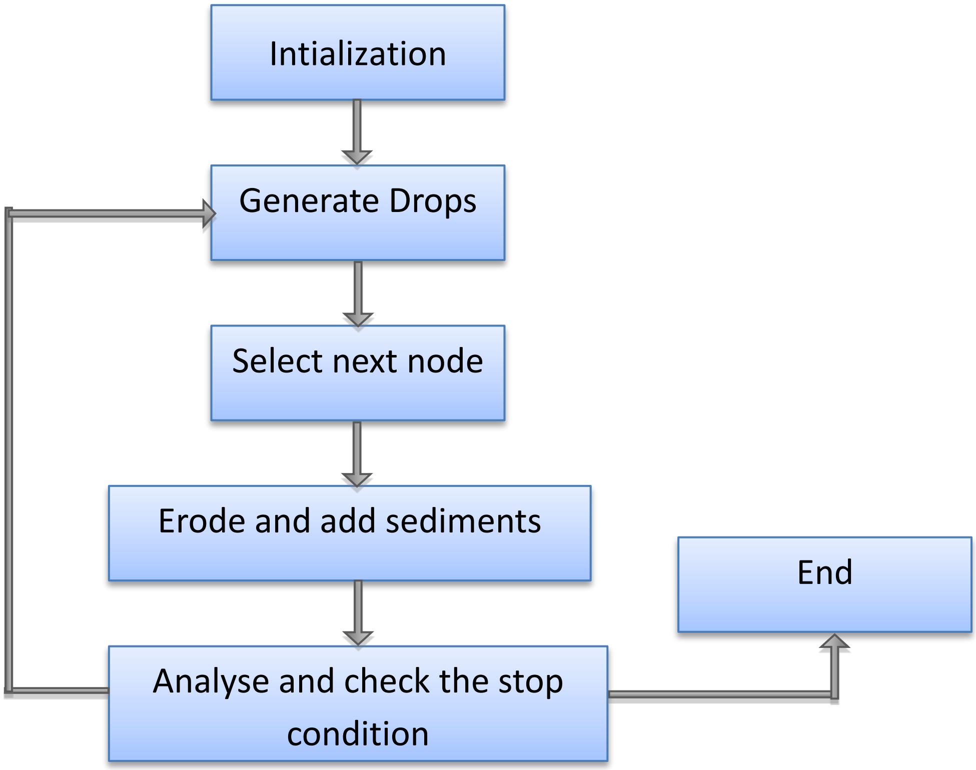


Figure 3.1 The RFD algorithm

Figure 3.1 shows the flowchart of the RFD algorithm. The RFD algorithm process starts with initializing the nodes to predetermined positive values, which reflect a flat surface at the beginning. The only exception is the goal point or destination which will have the altitude value of zero. Destination is considered as the sea where the drops should end. Drops are generated at the source (sources). At the beginning as all the nodes have the same altitude value, the drops will spread around the flat environment. When some drops find the destination, they fall into it. As they fall and because there is a difference in altitude between these nodes and the destination, they will be eroded. This erosion will create a down slope and throughout many training cycles the slope will be propagated backward to the source. Drops move according to the following probability random selection [22, 23].

$$P_k(i,j) = \begin{cases} \frac{\text{decreasingGradient(i,j)}}{\sum_{l \in V_k(i)} \text{decreasingGradient(i,l)}} & \text{if } j \in V_k(i) \\ 0 & \text{if } j \notin V_k(i) \end{cases} \quad (3.9)$$

where $P_k(i,j)$ is the probability of drop k at node i to select node j. V_k is a set of neighbors nodes that can be visited by the drop from node k. decreasingGradient(i, j) represents the negative gradient between nodes i and j, which is defined as follows:

$$\text{decreasingGradient}(i,j) = \frac{\text{altitude(i)} - \text{altitude(j)}}{\text{distance(i,j)}} \quad (3.10)$$

where altitude(x) is the altitude of the node x and distance(i, j) is the length of the edge connecting node i and node j. At the beginning of the algorithm, all nodes have the same altitude, and the sum of the decreasing gradient is also zero. RFD protocol suggests giving a special treatment to flat gradients, where the probability that a drop moves through an edge with zero gradient is set to some (non-null) value. This enables drops to spread around a flat environment, which is mandatory, in particular, at the beginning of the algorithm.

When a drop moves off a node to lower altitude, the node that the drop moved from will be eroded. The amount of erosion is proportional to the difference between the altitudes of two nodes. One must keep in mind that the altitude of the final destination is always zero and it is not eroded.

$$\text{erosion(j)} = \propto \left(\text{altitude(i)} - \text{altitude(j)}\right) \quad (3.11)$$

where α is a positive constant number.

Another process which follows the erosion is sediments deposit. This process is divided into two kinds. First, the algorithm periodically adds sediment to all nodes. This is done by slightly and uniformly increasing the altitudes of all nodes in the network. The main idea behind this is to avoid the situation when all altitudes become close to zero. This could occur after a long number of iterations which make the gradients close to zero or even negligible value and will lead to losing all formed paths. Second, drops add sediments as they move through the network. When drops move in the network it carries sediments. This sediment is the results of erosion. The amount of sediments that a drop carries throughout its path from source to destination is accumulative. This is equal to the sum of sediment carried from erosion in each node minus the amount of sediments that had been deposed at each node. The amount of sediments that will be deposited at each node is proportional to the amount that the drop carries according to the following equation.

$$\mathrm{sediment} = \beta * \mathrm{carried\_sediment} \qquad (3.12)$$

β is positive constant number.

Finally, the path from source to destination is analyzed and the stop condition is checked. If the quality is not good the procedure of drop sending will be repeated until satisfaction quality is reached or when a maximum number of iteration is reached.

RFD has been applied in many combinatorial optimization problems such as the asymmetric traveling salesman problem [22, 24], Optimal Quality-Investment Tree problem [25], minimum spanning Tree Problem [23], and others [26-28]. The RFD algorithm is a competitor to ACO algorithm and has shown to perform better than ACO in many applications [25, 27]. Finally the idea behind RFD drops can be adapted to propose intelligent data packets that contribute and collaborate with drops in the learning process.

3.7 Intelligent water drop

Intelligent Water Drop (IWD)is another swarm algorithm that has been proposed by Hamed Shah_Hosseini in 2007 [29]. Like RFD, IWD is also inspired by the way water drops create rivers. It is based on actions and reactions between water drops and its environment which leads to the creation of rivers paths that take the shortest path to the sea. One important feature that IWD is mimicking is the velocity of the water drop. The amount of erosion is relative to the speed of a drop, which has been referred to as soil transferring in [30]. Water drop velocity and the amount of soil carried by a drop are two important properties that IWD is based on. Velocity of water drop changes depending on the gradient of its path. The more velocity the drop has the more it erodes the surface. At the same time, the amount of sediment added is proportional to the time required for a drop to move from one location to another. As with RFD, IWD prefers to take paths with higher gradients. As with other swarm intelligent algorithms, the first application of IWD algorithm was Travel Salesman Problem.

The process of IWD algorithm starts with setting up the number of drops and parameters of the algorithm. IWD assigns initial soil to links between cites and gives initial velocity for the drops. These drops are distributed among the cities. Each IWD keeps a list of its visited cities [29, 31].

The selection of the next city in IWD algorithm is according to the following random probability function

$$p_{ij}^{IWD}(t) = \frac{f(soil_{ij}(t))}{\sum_{k \notin vc(IWD)} f(soil_{ik}(t))} \tag{3.13}$$

where

$$f\left(soil_{ij}(t)\right) = \frac{1}{\varepsilon_s + g(soil_{ij}(t))} \tag{3.14}$$

and

$$g\left(soil_{ij}(t)\right) = \begin{cases} soil_{ij}(t) & if\ \min_{l \notin vs(IWD)} soil_{il}(t) \geq 0 \\ soil_{ij}(t) - \min_{l \notin vs(IWD)}\left(soil_{il}(t)\right) & otherwise \end{cases} \tag{3.15}$$

$p_{ij}^{IWD}(t)$ is the probability of IWD selecting city j from city I at time t. $soil_{ij}(t)$ is the amount of soil at link i to j. vc(IWD) is a vector of previous visited cities by the IWD. ε_s is a small positive number to prevent dividing over zero.

After selecting the next city, the IWD updates its speed according to the following

$$vel^{IWD}(t) = vel^{IWD}(t) + \frac{a_v}{b_v + c_v * soil_{ij}(t)} \tag{3.16}$$

where $\mathrm{vel}^{\mathrm{IWD}}(\mathrm{t})$ is the velocity of the IWD. $\mathrm{a_v}$, $\mathrm{b_v}$, and $\mathrm{c_v}$ are velocity tuning constants.

As the IWD moves through a link it also erodes that link. The amount of erosion is calculated from the following equation

$$\vartriangle soil_{ij}(t) = \frac{a_s}{b_s + c_s * time_{ij}(vel^{IWD})} \qquad (3.17)$$

where

$$time_{ij}(vel^{IWD}) = \frac{\|pos_i - pos_j\|}{\max(\epsilon_v, vel^{IWD})} \qquad (3.18)$$

$\vartriangle \mathrm{soil_{ij}}(\mathrm{t})$ represents the amount of erosion in link i,j at time t. $\mathrm{a_s}$, $\mathrm{b_s}$, and $\mathrm{c_s}$ are erosion constants. $\mathrm{time_{ij}}\left(\mathrm{vel}^{\mathrm{IWD}}\right)$ is the time needed for an IWD to move from city i at $\mathrm{pos_i}$ to city j at $\mathrm{pos_j}$. ϵ_v is a threshold value used to compensate negative and zero speeds to this value.

This remaining soil at link i to j is

$$soil_{ij}(t) = (1 - \rho) * soil_{ij}(t) - \Delta soil_{ij}(t) \qquad (3.19)$$

The amount of carried sediment (referenced soil in the original paper) is

$$soil^{IWD} = soil^{IWD} + \Delta soil_{ij}(t) \qquad (3.20)$$

$soil^{IWD}$ represents soil carried by IWD.

After each iteration the length of the tours are computed, and the soil of the best tour since the beginning of the algorithm is updated [2]

$$soil_{ij}^{best} = (1 - \rho) soil_{ij}^{best} - \rho * \frac{2 * soil_{best}^{IWD}}{N_c(N_c - 1)} \; for\ each\ link\ in\ the\ best\ path$$

$$(3.21)$$

[2] The original paper shows an addition in equation 3.21, while all other papers that reference the original paper show subtraction rather than addition. The subtraction is more convincing as the best tour is enforced to become better and more attractive for other IWDs.

$\mathrm{soil}_{ij}^{best}$ is the soil at link i,j belonging to the best path. $\mathrm{soil}_{best}^{IWD}$ is the soil carried by IWD for the best path. Nc is the number of cities.

The algorithm is repeated until a satisfactory result is reached or up to a specific number of iterations.

IWD algorithm has been used to solve many applications like TSP, the n-queen puzzle, and Automatic Multilevel Thresholding (ATM) [29].

The main features that differentiate the IWD and the RFD algorithm are the IWD mimics the speed of water drop, RFD mimics the gradient. Moreover RFD has erosion and sedimentation processes while IWD has erosion only. Most importantly, RFD uses local feedback technique where there is no need for global feedback; however IWD is using global feedback. Finally, IWD has many parameters which need tuning and more explanations.

3.8 Particle Swarm Optimization

Kennedy and Eberhart introduced Particle Swarm Optimization (PSO) in 1995 [32,33]. PSO become popular due to its speed and relative ease of implementation. It is inspired by animal social behaviour like a flock of birds or a school of fish when they travel and try to find sources of food. The algorithm mimics how a bird in a flock will be guided by the rest of the birds in the flock. At the same time the rest of the birds will also be influenced by that specific bird. This is a distributed swarm system where every entity gets influenced by the experience of other members in the system. This interaction develops search behaviour towards the objective goal.

The PSO algorithm starts by initializing a population of random solutions. Each member of this population is called a particle. Each particle is defined as multidimensional points. Therefore, in a D dimensional search space, a particle *i* is a D-dimensional vector. Each particle defines its position $\overrightarrow{\mathrm{x_i}} = (\mathrm{x_{i1}}, \mathrm{x_{i2}}, \dots., \mathrm{x_{iD}})$ and its velocity $\overrightarrow{\mathrm{v_i}} = (\mathrm{v_{i1}}, \mathrm{v_{i2}}, \dots., \mathrm{v_{iD}})$. A fitness function is used to quantize the quality of the solution represented by each particle. The best particle position is stored as parameter named global best particle $\overrightarrow{\mathrm{gp_i}} = (\mathrm{gp_{i1}}, \mathrm{gp_{i2}}, \dots., \mathrm{gp_{iD}})$. A local best fitness $\overrightarrow{\mathrm{lp_i}} = (\mathrm{lp_{i1}}, \mathrm{lp_{i2}}, \dots., \mathrm{lp_{iD}})$ is found among the local neighbours of each particle. A set of updating equations is used to move these particles throughout the search space. The velocity and position in every dimension D of each particle *i* in the swarm is updated at each time step according to following equations

$$\mathrm{v}_{\mathrm{ij}}^{\mathrm{t+1}} = \mathrm{v}_{\mathrm{ij}}^{\mathrm{t}} + \mathrm{c} * \mathrm{r}_{1\mathrm{j}} * \left(\mathrm{lp}_{\mathrm{ij}}^{\mathrm{t}} - \mathrm{x}_{\mathrm{ij}}^{\mathrm{t}}\right) + \mathrm{c} * \mathrm{r}_{2\mathrm{j}} * \left(\mathrm{gp}_{\mathrm{ij}}^{\mathrm{t}} - \mathrm{x}_{\mathrm{ij}}^{\mathrm{t}}\right) \quad (3.22)$$

where $\mathrm{v}_{\mathrm{ij}}^{\mathrm{t+1}}$ is the velocity of particle *i* at time *t+1* for every *j* dimension. *c* is a positive constant, r_{1j} and r_{2j} take independent uniform random values in the range [0, 1] and uniquely generated at every update for each individual dimension *j* = 1 to D.

$$\mathrm{x}_{\mathrm{ij}}^{\mathrm{t+1}} = \mathrm{x}_{\mathrm{ij}}^{\mathrm{t}} + \mathrm{v}_{\mathrm{ij}}^{\mathrm{t}} \quad (3.23)$$

where $\mathrm{x}_{\mathrm{ij}}^{\mathrm{t+1}}$ is the position of particle *i* at time *t+1* for every *j* dimension.

Each particles in PSO movement is controlled by attractive forces of both global best and local best position as shown in equation 3.22 .

Particle velocities had to be clamped at a maximum value v_{max}

$$\mathrm{v}_{\mathrm{ij}} < |\mathrm{v}_{\mathrm{max}}| \quad (3.24)$$

where v_{ij} is the velocity of particle *i* for every dimension *j*.

Like the RFD algorithm, the concept of particle swarm algorithm is simple. However, as PSO tracks the best global position in the system which require a method to access all nodes to select the best global position. This usually requires a global feedback if implemented in ad hoc network which will introduce extra traffic in the network. On the other hand, the RFD algorithm only requires local data exchange which makes it more suitable algorithm for ad hoc networks.

3.9 Studying the RFD algorithm

This section shows some properties of the RFD algorithm. The convergence of the method and the effect of the algorithm parameters on the learning process of the algorithm are explained.

The first characteristic of the RFD algorithm is the slow convergence rate of the algorithm especially at the beginning of the learning process. The main reason for this is the way that the RFD algorithm returns feedback. Most swarm algorithms, which are using feedback, are using feedback agents. These backward agents move from destinations to sources and they update the entire bath in their way back to the sources. Some other swarm approaches use global variable or global update equation. An example of global update equation is in the IWD where equation 3.21 represents a global updating. This equation updates the best path which could be a collection of any nodes in the problem. In the RFD there is no global feedback. The feedback effect is generated locally by the erosion and sedimentation processes. When a node sends a drop to the next node, it checks the altitude of the next node and then it erodes or adds sediment to itself. This acts like a local feedback where the change in the nodes altitude is depending on the altitude of the next node and of course the node itself's altitude. The less the altitude of the next node which leads to more altitude difference, the higher the erosion. With each drop part of this difference will propagate back as erosion in previous nodes.

To show the slow convergence of the RFD algorithm, a network consisting of a grid of 5 by 10 nodes is used. Drops are sent from node S (node number 1) to node D as shown in Figure 3.2. The first scenario is implemented to show how drops create a river and to show the process of local feedback. To show the number of drops needed to create a path, drops are forced to move along the shortest path in order to show the delay in path creation. A path is created whenever there is a decreasing slope from the source to the destination. Figure 3.3 shows the number of drops required to create a path. It shows that at least 8 drops are needed to create a path. This is equal to the number of nodes between the source and the destination. To create a waterfall between the source and destination the nodes in between should be eroded. The destination altitude is zero, so the first node that will be eroded is the penultimate node. The erosion should at least propagate back to the second node, which in this case will be less than the initial value of the first node. As a result, for ten nodes, eight nodes should be eroded. This will require eight drops of water.

In the second scenario, the effects of erosion factor and sedimentation factor are studied. Two tests have been carried out. Figure 3.3a shows the effect of variable $\propto$ (erosion factor) on the nodes altitude curvature (waterfall curvature) for the shortest path between the source and the destination. The value of β is set to 0.5 in the first test. It is clear that low value will result in slower learning at nodes close to the source as the slope of the nodes near the source is very small. Moreover, a surface that is almost flat near the source increases the probability of drops moving in a direction opposite to the direction of the destination. However, setting the erosion factor to high value may lead the algorithm to non-optimal solutions.

In the second test, the effect of sedimentation factor is studied. Figure 3.3.b shows the effect of β on the curve created by the RFD algorithm for the same network and same number of drops. The value of $\propto$ has been set to 0.5. It could be seen that the curve become concave for low values and as the value of β increases it converted to convex curve. When the surface near the destination becomes flatter, it gives the opportunity to many nodes to deliver the drop to the destination.

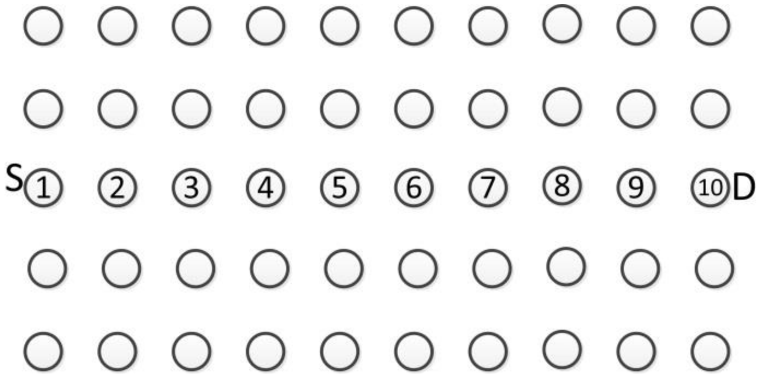


Figure 3.2 Network topology

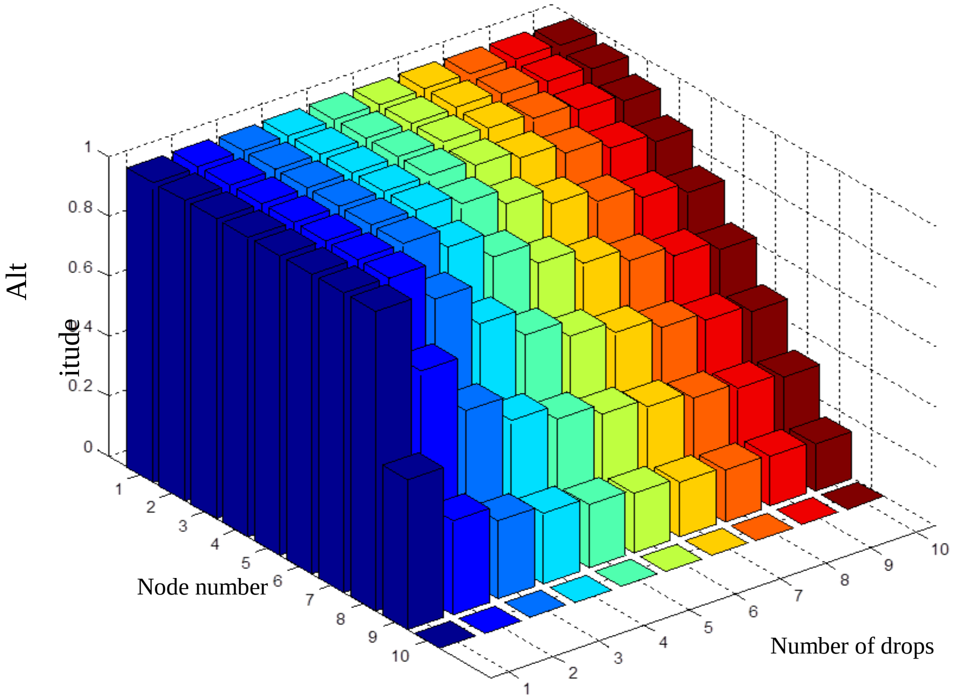
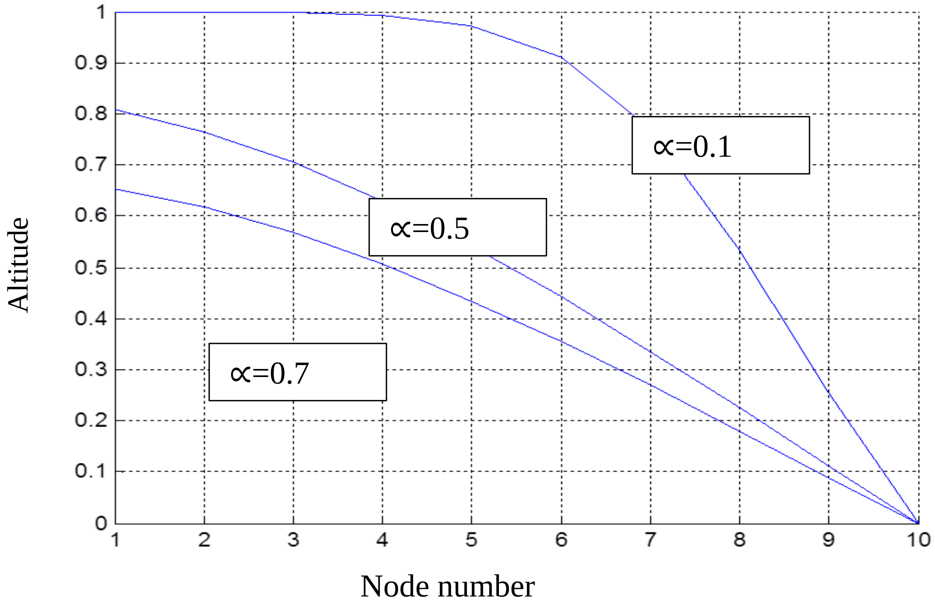
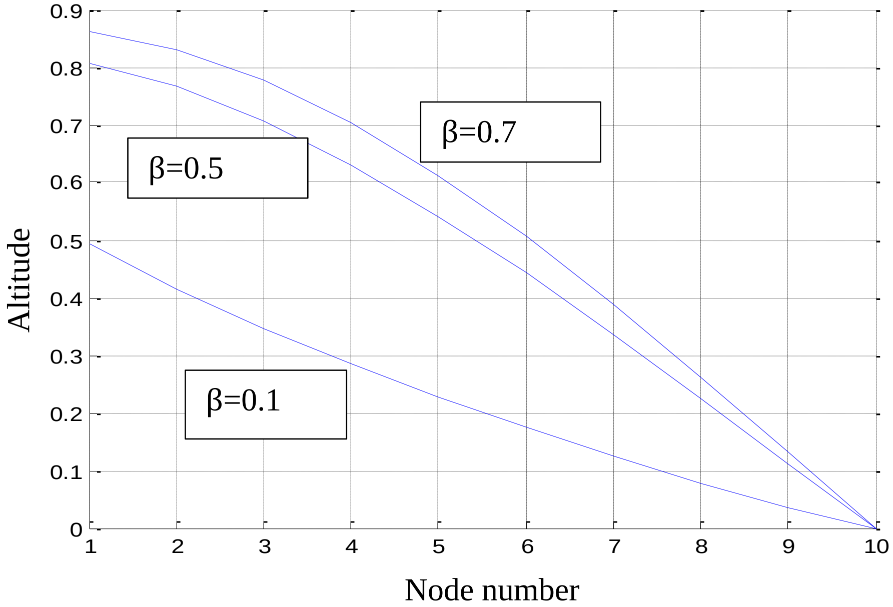


Figure 3.3 Effect of number of drops on nodes altitude



(a)



(b)

Figure 3.4 Effect of ∝ and β on the curvature of altitude surface.

3.10 Summary

Different swarm technologies have been reviewed in this chapter and we mainly focused on ACO and RFD algorithms as they will be used along this thesis. First the concepts of self-organizing system are explained. ACO is a swarm algorithm which is inspired from the natural food foraging behaviour of real ants. ACO is a self-organizing system where all entities in the system are working in cooperative way. There is central control over the system in ACO system. River Formation Dynamics is another type of swarm intelligence. The main concept behind River Formation Dynamics is how raindrops create rivers which finds shortest path to the sea. Other swarm intelligence techniques like IWD and PSO is also reviewed. Finally, the effects of various parameters in RFD algorithm are examined.

References

[1] G. Di Marzo Serugendo, M. Gleizes and A. Karageorgos, "'Self-organising Software From Natural to Artificial Adaptation,". G. Rozenberg. , Ed.london new york: Springer Heidelberg Dordrecht, 2011, pp.8-15.

[2] Scholarpedia, Self-organization, Accessed: 27/10/2013, [Online] Available: http://www.scholarpedia.org/article/Self-organization.

[3] Wikipedia, Stigmergy, Accessed: September 2013, [Online] Available: http://en.wikipedia.org/wiki/Stigmergy.

[4] D. Maniezzo, M. Dorigo, V. Maniezzo and A. Colorni, "'Ant System: An Autocatalytic Optimizing Process,", 1991.

[5] M. Dorigo, V. Maniezzo and A. Colorni, "'Ant system: optimization by a colony of cooperating agents," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 26, no. 1, 1996, pp. 29-41.

[6] E. Bonabeau, M. Dorigo and G. Theraulaz, "'Swarm intelligence: from natural to artificial systems,"Oxford University Press, Inc, 1999.

[7] T. Stützle and H.H. Hoos, "'MAX–MIN Ant System," *Future Generation Computer Systems*, vol. 16, no. 8, 2000, pp. 889-914.

[8] T. Stutzle and H. Hoos, "'MAX-MIN Ant System and local search for the traveling salesman problem," *Evolutionary Computation, 1997., IEEE International Conference on*, 1997, pp. 309-314.

[9] T.S. M. Dorigo, "'Ant Colony Optimization,"MIT Press, Cambridge, MA, 2004.

[10] H. Shah-Hosseini, "'An approach to continuous optimization by the Intelligent Water Drops algorithm," *Procedia - Social and Behavioral Sciences*, vol. 32, no. 0, 2012, pp. 224-229.

[11] S. Marwaha, J. Indulska and M. Portmann, "'Biologically Inspired Ant-Based Routing in Mobile Ad hoc Networks (MANET): A Survey," *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC '09. Symposia and Workshops on*, 2009, pp. 12-15.

[12] G.S. Sharvani, N.K. Cauvery and T.M. Rangaswamy, "'Different Types of Swarm Intelligence Algorithm for Routing," *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on*, 2009, pp. 604-609.

[13] B. Kalaavathi, S. Madhavi, S. Vijayaragavan and K. Duraiswamy, "'Review of ant based routing protocols for MANET," *Computing, Communication and Networking, 2008. ICCCn 2008. International Conference on*, 2008, pp. 1-9.

[14] H. Shokrani and S. Jabbehdari, "'A Survey of Ant-Based Routing Algorithms for Mobile Ad-hoc Networks," *2009 International Conference on Signal Processing Systems*, 2009, pp. 323-329.

[15] S.D. Shirkande and R.A. Vatti, "'ACO Based Routing Algorithms for Ad-hoc Network (WSN, MANETs): A Survey," *Communication Systems and Network Technologies (CSNT), 2013 International Conference on*, 2013, pp. 230-235.

[16] I. Kassabalidis, M.A. El-Sharkawi, R.J. Marks II, P. Arabshahi and A.A. Gray, "'Swarm intelligence for routing in communication networks," *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 6, 2001, pp. 3613-3617 vol.6.

[17] R. Schoonderwoerd, J.L. Bruten, O.E. Holland and L.J.M. Rothkrantz, "'Ant-based load balancing in telecommunications networks," *Adapt.Behav.*, vol. 5, no. 2, 1996, pp. 169-207.

[18] G. Di Caro and M. Dorigo, "'Mobile agents for adaptive routing," *System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on*, vol. 7, 1998, pp. 74-83.

[19] S. Marwaha, Chen Khong Tham and D. Srinivasan, "'Mobile agents based routing protocol for mobile ad hoc networks," *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 1, 2002, pp. 163-167.

[20] M. Gunes, U. Sorges and I. Bouazizi, "'ARA-the ant-colony based routing algorithm for MANETs," *Parallel Processing Workshops, 2002. Proceedings. International Conference on*, 2002, pp. 79-85.

[21] M. Roth and S. Wicker, "'Termite: ad-hoc networking with stigmergy," *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, vol. 5, 2003, pp. 2937-2941.

[22] P. Rabanal, I. Rodríguez and F. Rubio, "'Using River Formation Dynamics to Design Heuristic Algorithms," *Unconventional Computation Lecture Notes in Computer Science*, vol. 4618, 2007, pp. 163-177.

[23] P. Rabanal, I. Rodríguez and F. Rubio, "'Applying River Formation Dynamics to the Steiner Tree Problem," *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*, 2010, pp. 704-711.

[24] Hifza Afaq , Sanjay Saini, "'On the Solutions to the Travelling Salesman Problem usingNature Inspired Computing Techniques," *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 4, July 2011, pp. 326 -334.

[25] P. Rabanal, I. Rodríguez and F. Rubio, "'Applying RFD to Construct Optimal Quality-Investment Trees," *J.UCS Journal of Universal Computer Science*, vol. 16, no. 14, May 2010, pp. 1882-1901.

[26] P. Rabanal, I. Rodriguez and F. Rubio, "'A Formal Approach to Heuristically Test Restorable Systems,"in Theoretical Aspects of Computing - ICTAC 2009, vol. 5684. Martin Leucker and Carroll Morgan. , Springer Berlin Heidelberg, 2009, pp.292-306.

[27] P. Rabanal, I. Rodriguez and F. Rubio, "'Finding Minimum Spanning/Distances Trees by Using River Formation Dynamics,"in Ant Colony Optimization and Swarm Intelligence, vol. 5217. Marco Dorigo, et al. , Springer Berlin Heidelberg, 2008, pp.60-71.

[28] P. Rabanal and I. Rodriguez, "'Testing Restorable Systems by Using RFD,"in Bio-Inspired Systems: Computational and Ambient Intelligence, vol. 5517. Joan Cabestany, et al. , Springer Berlin Heidelberg, 2009, pp.351-358.

[29] H. Shah-Hosseini, "'Problem solving by intelligent water drops," *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 3226-3231.

[30] S. Noferesti and H. Shah-hosseini, "'Article: A Hybrid Algorithm for Solving Steiner Tree Problem," *International Journal of Computer Applications*, vol. 41, no. 5, 2012, pp. 14-20.

[31] H. Shah-Hosseini, "'The intelligent water drops algorithm; a nature-inspired swarm based optimization algorithm," *Int.J.Bio-Inspired Comput.*, vol. 1, no. 1/2, 2009, pp. 71-79.

[32] Kennedy, J. and Eberhart, R. C. "'Particle swarm optimization, " *Proc. IEEE Int'l Conf. on Neural Networks, Piscataway, NJ: IEEE Service Center*, vol. 4, 1995, pp1942–1948.

[33] Eberhart, R. C. and Kennedy, J., "'A new optimizer using particle swarm theory, " *Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan*, 1995, pp. 39-43,

Chapter Four

Multi-Channel Multi-Interface Routing Algorithm

4.1 Introduction

Most 802.11 protocols use one channel, although it supports many non-overlapped channels. The 802.11b/g for example supports 3 non over-lapped channels and the 802.11a supports 12 non overlapped channels. Assigning different channels to different nodes can increase the quality of services in ad hoc networks as this provides less interference between the nodes [1]. One advantage of using multi-channel multi-interface is that a node can receive and transmit data simultaneously. Another advantage is it reduces the interference with other nodes in its range.

This chapter proposes a new geographical ant routing protocol for mobile ad hoc network with limited number of interfaces that are utilized dynamically to improve network performance. This protocol is inspired by ant colony algorithm which reacts dynamically and cognitively toward networks environments changes. Channel selection is dynamically allocated as a reaction to link breaking or in case a channel has been used by licensed users. At the same time, this protocol uses local geographical information by the traveling ants and is combined with mapping algorithm to represent a temporary global source to destination geographical information. The limited number of interfaces is posing more challenge to multi-channel routing protocol. The proposed method tries to overcome the channel switching problem occurs due to the limited numbers of radio interfaces. The problem of route fragility is addressed. The protocol focuses on cognitive channel selection in ad hoc networks with limited numbers of interfaces using geographical information in order to reduce interference and increase the throughput and decrease the amount of channel switching. Simulation results show that the proposed protocol can improve network throughput.

4.2 Related work

Routing and channel assignment have been a point of interest in the last several years [2, 3]. Many approaches have been introduced to solve the joint optimization problem of channel assignment and routing [4, 5]. One of these methods is to use swarm technology to solve this problem [6-8].

In [9] the authors developed a QoS-aware routing mechanism for wireless ad-hoc networks. The protocol uses static channel assignment method. The protocol tries to load balance messages around the network. Each node shares its available bandwidth in each channel with other nodes in the network. The protocol is based on OLSR protocol and is using its control messages to distribute bandwidth information around the network. The source node tries to find the best path to the destination using the available information at the node. Each intermediate node will then try to select the best channel in stochastic manner. The transmission quality is measured using the Signal-to-Interference and Noise Ratio (SINR).

In [10] a multipath protocol proposed where the protocol sets up different local backup path for each link. The protocol locally optimizes the selection of backup path which must have the highest lifetime among other available backup paths.

In [11] a link life time based on energy which measures the received signal strength, and calculates the distance between the node according to this measurement. The protocol assumes all nodes transmitting with the same power. The algorithm is combined with DSR protocol to build a link prediction protocol. In [12] the protocol joint with Particle Swarm Algorithm. Both of these protocols are based on summing the value of received power and try to maximize it. However, selecting a route with higher sum of received energy will not result in best path.

In [13] a location base system is used to predict next node position. The protocol tries to maintain connectivity by using controlled ad hoc network. A particle swarm algorithm was designed to solve the optimization problem and to direct the motion of the mobile node agents.

Joint channel assignment and routing scheme used in [4] to build a receiver conflict avoidance interference model. The approach sets one interface card for transmission and the others for receiving. In order to compensate the channel switching delay, each transmitter interface remains selected for predetermined slot time between 10ms to 100ms [14, 15].

Bowen LI *et. al* [16] introduced ASAR: Ant-based Spectrum Aware Routing for Cognitive Radio Networks. This is an ACO based routing protocol where all nodes have at least one control interface. ASAR detects if a channel has been used by licensed or primary users and switches channel to another free channel. Forward ants are used to sense the availability of channels in the route, while backward ants are used to update route parameters. In their work, they assume that the number of interface cards is equal to the number of available channels. The protocol assigns a fixed interface to each channel. The protocol only forwards a packet if there is an available free channel, otherwise the packet will be discarded. If the number of interfaces is low, the system will be forced to use part of the overall available channels. The number of used channels will be equal to the number of interface cards. This in turn will degrade the performance of the network. However, the protocol does not have the problem of channel switching as they use fixed channel for each interface. The main drawback of the protocol is it cannot use all available channels if the number of interfaces is less than the available channels.

Geographical routing assumes that each node has at least the geographical information of its neighbours and geographic information of the destination [17]. This geographical information could be location, speed and direction. The node finds the best forwarding node in its neighbour and delivers the packet to that node. Many approaches have been introduced to select best forward node. An example is greedy forwarding where the best forwarding node is the one that is closer to the destination. This way the algorithm tries to minimize the distance to the destination each time when the packet is forwarded. However, to prevent loops, the packet will be dropped if there is no node closer to the destination than the node itself that is processing the packet. Face routing is another approach where the virtual line between the source and the destination is known and the packets are routed on the interior of the faces of the communication graph without crossing the virtual line between the source and destination. Geographical routing does not guarantee that best path will be found, moreover loops can occur. If a local minimum is reached, geographical routing protocols usually try to find a detouring strategy to leave this local minimum [18]. Combining geographical information with Ant colony optimization could solve some problems in geographical routing [19, 20]. More information about the route path and its availability could be extracted from geographical information and play a major role for the ants to select their paths to destination node.

Shahab Kamali et al [21] presented an ant Position Based Ant Colony Routing algorithm (POSANT). In their work they introduce zone concept with ant colony algorithm. They divide the neighbours nodes into three zones and the ant stochastically select the next node which is based on the values of pheromone trails. The protocol decreases the route setup time. Basic position biased ant protocol has some drawbacks as it cannot cope with node mobility. The idea of clustering has been added to POSANT by Maumita Bandyopadhyay et al [22] to address its limitation. Both of these protocols were single channel single interface algorithms.

4.3 Proposed routing protocol

The main objective of the proposed protocol is to minimize the number of interfaces, decreasing the amount of channel switching by estimating route fragility, and increasing the data rate. The protocol uses dynamic channel assignment where three interface cards have been used. One interface card is fixed and used for control packets and the two other cards are dynamically assigned to various channels. Minimizing the number of interface cards will reduce the cost and size of the mobile node as it uses less hardware. Ant colony estimation is used as an optimization method where ants are searching for the best route and collecting geographical information. The geographical information is mapped into another domain in order to select a stable route as well as selecting the data link with higher data rates. The proposed protocol uses source routing to set up a route. The protocol sets up these routes for a predetermined amount of time. These routes will remain constant and will not change during these periods, even if the ant table changed, until rerouting time. Depending on the information gathered by the moving ants the route's quality is estimated as well as route's life time. In order to minimize the probability of route failure the protocol tries to establish new route prior to estimated link breakage in the route. This will also minimize the delay caused by channel switching during route setup as the route will be ready to use almost directly if the new one has no intersection with the old route. The following sections will discuss the details of the proposed routing protocol.

4.3.1 Network structure

The network consists of N nodes, each node occupied with M radio interface cards for data exchange and one radio interface for control messages exchange. There are C orthogonal channels that are available to use in the network, these channels could be licensed or not. Each node is assumed to be equipped with a GPS device and can get its own X and Y coordinate and its speed S and direction. The available bandwidth in each channel is defined in a vector Ch. The available bandwidth is calculated as the maximum bandwidth divided by the number of nodes that are using this particular bandwidth.

4.3.2 Geographical information usage

In multi-channel multi-interface networks, route setup includes both channel allocation process and route finding. In order to establish a route, each node should set up a communication channel with its next neighbouring node that is within its communication range. The routing process and route set up in such a network can lead to significant downgrade of network performance. One way to enhance network throughput is to select a stable route and decrease the number of route breakage using geographical information. Figure 4.1 shows two neighbour nodes that are moving in different directions and speeds. Assuming both nodes are transmitting at the same power and thus having the same range of transmission. Therefore these two nodes have a valid link between them for a certain amount of time. To predict the amount of time that the link will be valid (we called it ‘time to live’), the relative speed and direction should be first calculated from their speeds and directions. The relative speed and direction of the second node towards the first node is shown in the figure. Then according to their positions this amount of time can be calculated. The algorithm of calculating time to live is shown below

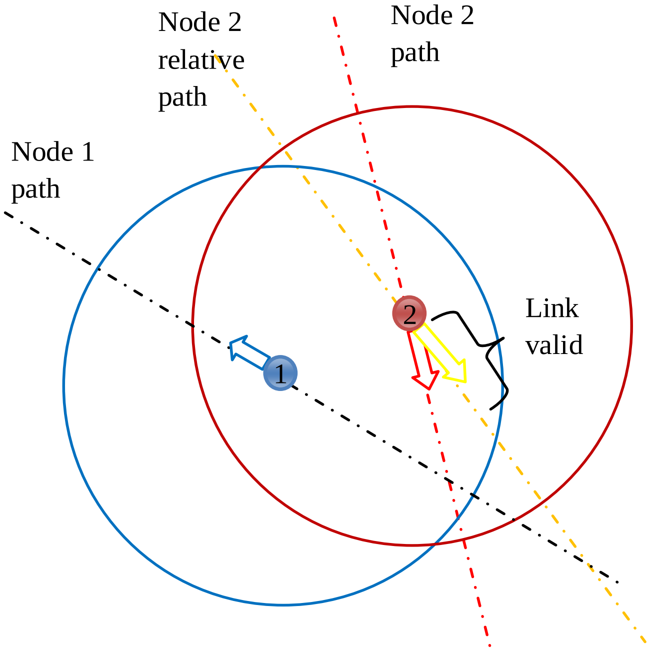


Figure 4.1 The duration of a link between two nodes to remain valid

```
Algorithm: Calculating time to live `
Constant: RADIO_RANGE
Input: Node (A) {$P_A$ : position, $V_A$ : speed vector}, Node (B) {$P_B$ : position, $V_B$ : speed vector}
Output: Time_to_live
Begin
Calculate (B) relative speed with respect to (A) $V_{B/A}= V_B - V_a$ .
Calculate the relative position ($P_{B/A}$) between nodes (A) and (B) from their Position information.
Calculate the intersection of vector $V_{B/A}$ from position $P_{B/A}$ with the circle whose centre is node (A) and radios RADIO_RANGE.
Calculate the time needed (time_to_live) to travel from $P_{B/A}$ to intersection point with velocity $V_{B/A.}$
End
```

The main problem in multi-channel multi-interface network and that has a big impact on the network performance is the number of channel switching that occurs in the network due to the limited number of interfaces. Whenever a node changed its transmitting channel, a chain of channel switching may take place all over the route. The same is true when a new route is being set up. When an interface switches from one channel to another channel, it takes a considerable amount of time. Although newer interfaces are very fast, this amount of delay should not be neglected [23, 24].

The process of selecting the best link is fuzzy in nature. There are many input parameters that affect the quality. A set of rules have been defined in order to select the best link. Both time to live and distance as input parameters to select next node.

- *Time to live value:* Practically, when two nodes are close enough and their link life time will be high, especially if they move in the same direction. Any two nodes that are close to each other could be assured that they are going to be in range for a longer period of time, than the nodes that are far from each other with the same velocity and direction. Selecting routes that are expected to endure longer are preferred and will add significant performance improvement to the network [25]. The first constraint on how to select a link is to have high time to live value in order to select more stable routes.

- *Number of nodes in the route:* Increasing the number of nodes in a route from a source node to a destination node may lead to many problems. First, if many close to each other nodes are in a selected route, this will add more end to end delay, as the packet will be buffered in each intermediate node in its way to the destination. Moreover, the time required to set up the route will increase as each node will require channel switching (this problem is not presented in networks where there is no limit on the number of interfaces). Second, using too many nodes will soon lead to a situation where all channels are used within the transmission range of a node. This leads to a situation where the node has to use a channel that may overlap with another node's channel and will lose the benefits of the multi-channel network.
- *Data rate:* As the distance is one of the primary factors that determines data rate in wireless networks, nodes that are very far away from each other are not guaranteed to support the link for long durations of time. Moreover they have low bandwidth [26]. The proposed protocol is set to be simulated in open space environment, so the main factor that affects the data rate is distance. Other factors like noise and obstacles effects are neglected and the study of their effect is postponed to future works.

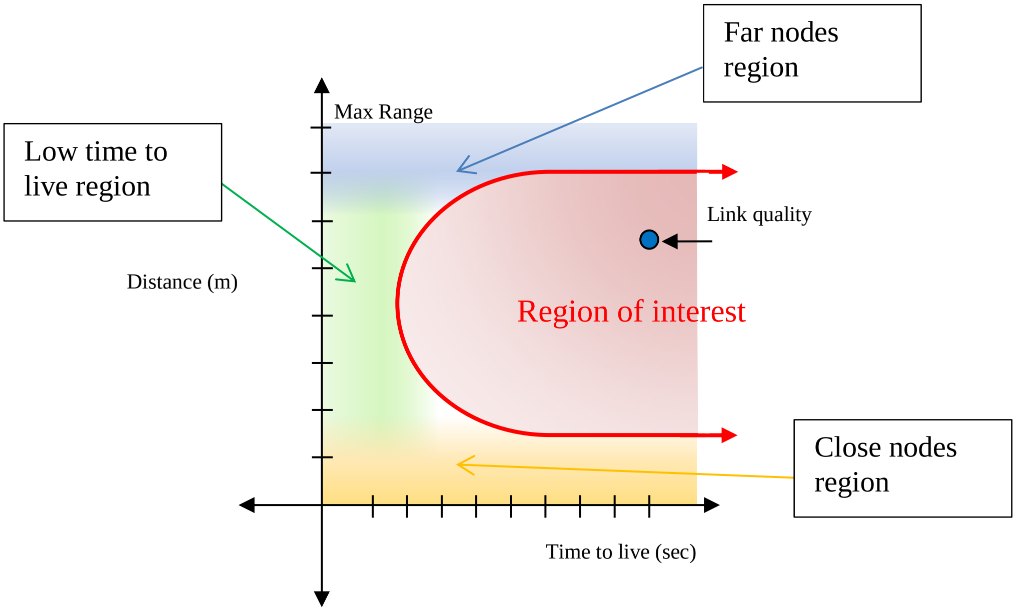


Figure 4.2 Distance vs. time to live graph

Figure 4.2 shows the four different regions created according to the above rules. The three regions low time to live, far nodes, and close nodes are reflecting the above rules, while the region of interest represents the remaining region where the link quality is good. The exact boundaries of these regions are fuzzy and the red line represents an approximation of the interested just for clarification. The exact boundary limits of distance and time to live are fuzzy and uncertain. Practically the upper boundary, lower distance boundary and minimum time to live could be set according to network state and usually related to the speed of the nodes. In order not to have a crisp value of the boundaries a mapping algorithm has been defined which gives the probability of goodness of the values. Moreover, as the value of time to live is unbounded, the mapping algorithm should be bounded.

$$x = 1 - \frac{1.5}{e^t} \tag{4.1}$$

$$y = \begin{bmatrix} 2d - 0.5 & 0 < d < 0.25R \\ (d - 0.25)/0.5 & 0.25R < d \leq 0.75R \\ 1 - (d - 0.75)/0.05 & 0.75R < \ d \ \leq 0.8R \\ (0.8 - d) * 2.5 & 0.8 < d \end{bmatrix} \tag{4.2}$$

where *t* is time to live, d is distance between the nodes and *R* is the maximum range of transmission. Equation 4.1 maps the time to live to x domain where this equation has a bounded output for all values of time to live. This exponential function is used to reflect more intrust to the difference in low values of time to live as these differences are more important than the small differences for different links with high time to live values. For example if the time to live of two links are 0.5s and 1.5s, it is important to reflect the difference between them in the mapped domain because selecting the link with 1.5 will result in better stable route. On the other hand, for high time to live values like 1000s and 1001s, the mapping function will not produce the same difference (the difference will be less) as both links are good.

Equation 4.2 maps the distance according to the rules explained previously. Where close nodes are mapped to bad region and mid-range nodes are mapped to good region and far nodes are mapped to bad region. When the distance between nodes is less than 25% of maximum transmission range, the number of nodes that are sharing the same media will be five or more which is not desirable as explained in our rules. Nodes with distance up to 75% of maximum transmission range are better to select. When the distance between nodes are about 75% of maximum transmission range, the probability of three node to share the same transmission media is less. Finally, when the range is higher than 80% of maximum transmission range, the node are far from each other and are not preferred to be selected.

The mapped linked quality is shown in Figure 4.3 where the upper right corner represents good link quality, while the lower left corner represents bad link quality. The link quality is calculated as the distance from the good quality corner and is normalized to the total diagonal length of the square area (from upper right good corner to the lower left bad corner). The more the quality is closer to the upper right corner the better the link. The link quality is used to select next node in ant routing algorithm (will be explained in section 4.3.6). as the link quality is calculated as the distance on the diagonal of the square and its value is normalized, the size of the square has no effect on the quality of the link. Moreover, the position of the axis has no effect on the quality of the link. However, the position of the axis has effect on route quality. The selection of y axis position is based on calculation of time to live value. If two nodes with good quality change their direction of movement to opposite direction and with maximum speed they will need about 0.625s to be out of range of each other. The value is set to 0.4 to ensure that the nodes will be in communication range if they suddenly change their dirction of movement. The position of x axis is selected so that it represents the boundary ranges of the previous explained rules.

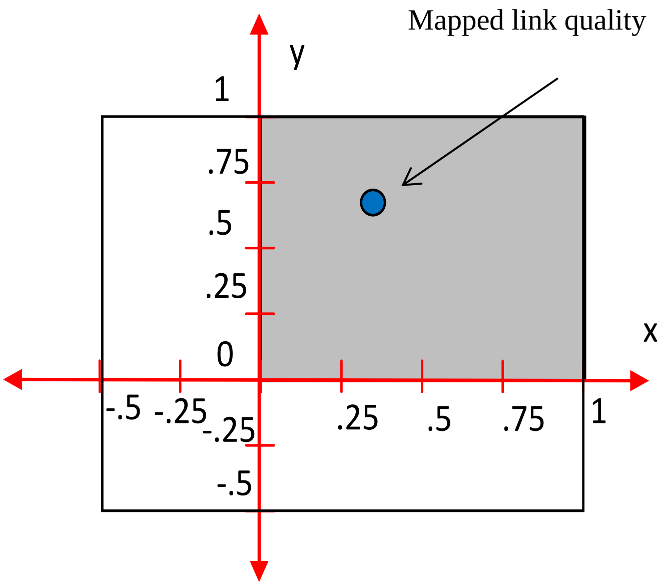


Figure 4.3 Mapped domain

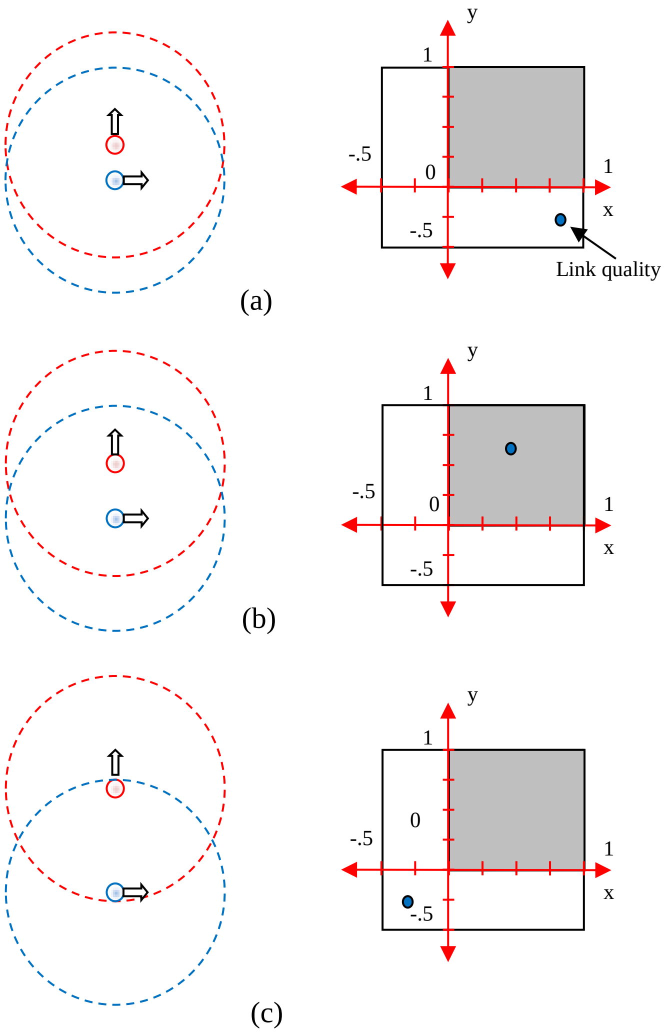


Figure 4.4 The relationship between distance and mapped link quality. a- Short distance and high data rate, b- Medium distance and medium data rate, c- Long distance and low data rate

Figure 4.4 shows the relationship between distance and mapped link quality. Figure 4.4a shows two nodes that are very close to each other. As explained previously, using close node could lead to excessive use of available channels which could lead to over lapping in channels. The link quality is shown to be in the fourth quarter of the graph. The first quarter is for good quality and the closer the link's quality to the upper right corner, the better the quality. The closer to the bottom left corner, the worse the quality is. The midrange nodes will have better probability and will be in the first quarter of the graph as shown in Figure 4.4b. boundary node will have bad link quality as shown in Figure 4.4c.

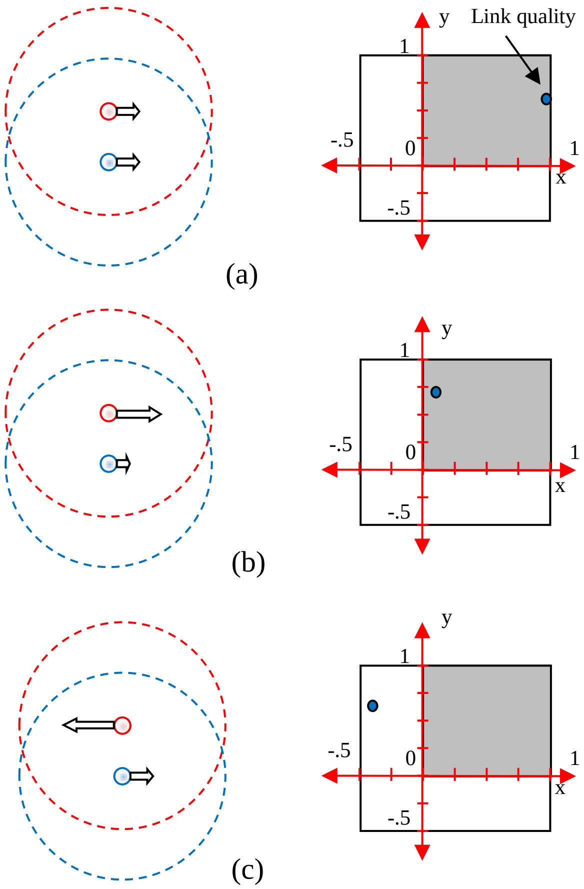


Figure 4.5 Examples of the relationship between speed, direction and mapped link quality. a- same speed same direction, b- different speed same direction c- different speed opposite direction

Figure 4.5 shows examples of the effect of both speed and direction on the mapped link quality. The nodes are in acceptable range. It can be seen from Figure 4.5a if two nodes have the same speed and same direction, they will have better link quality as they are predicted to move together and the link will remain stable. Figure 4.5b shows that if the two nodes have various speeds the link quality moves away from the upper right corner. When two nodes are moving in opposite direction the time to live value decreases and the link quality moves towards the negative side of the x axis as shown in Figure 4.5c.

4.3.3 Ant tables

In order to route packets, the proposed routing protocol uses probability ant table at each node in the network, where the routing table is maintained. The table includes the probability of selecting next neighbour (pheromone intensity) (τ), neighbour time to live (T), neighbour position, speed, direction (G), and available channels bandwidth (Ch). Where D represents destination node, Ni represents neighbour node, k is the number of neighbour nodes and n is the number of destination nodes in the table.

Table 4.1 Ant routing table

Neighbour node	Ni_1	Ni_2	……..	Ni_k
time to live	T_1	T_2	……..	T_k
Geographical info.	G_1	G_2	……..	G_k
Bandwidth destination node	Ch_1	Ch_2	……..	Ch_K
D_1	τ_{11}	τ_{12}	……..	τ_{1k}
D_2	τ_{21}	τ_{22}	……..	τ_{2k}
……..	……..	……..	……..	……..
D_n	τ_{n1}	τ_{n2}	……..	τ_{nk}

Hello message is used to discover neighbour nodes and exchange geographical information. The rate of sending hello message is proportional to node speed to ensure the delivery of geographical information.

4.3.4 Route Quality

Efficient routing algorithm should respond adaptively toward network states changes. Ant colony algorithm with reinforcement learning is used as an adaptive learning algorithm. Routing quality is used in reinforcement learning and is computed at the destination node.

While a forward ant travels from source to destination, each node adds a copy of its time to live value and distance to the forward ant packet. At the destination, a temporary graph of overall route quality will be available as shown in Figure 4.6. This graph reflects the overall source to destination geographical information. When the parameters on the graph are close to each other, that mean the nodes are moving at homogenous speeds and spaced by equal distances as shown in Figure 4.6a. Moreover, if the parameters are in the first quarter, the nodes will support the route for a longer time. Otherwise, if any of these parameters were in other quarters, the route has a bottleneck at that link. It can be noted that the centroid reflects the route quality and when it is in the first quarter it will be better. Figure 4.6b shows a bad route quality, the centroid is very close to the y axis. Another important parameter is the contour length. Figure 4.6a and Figure 4.6c almost have the same centroid but in Figure 4.6c the nodes are diverge. The contour length in Figure 4.6c is longer than the one in Figure 4.6a. The shorter the contour length is, the better the route quality is.

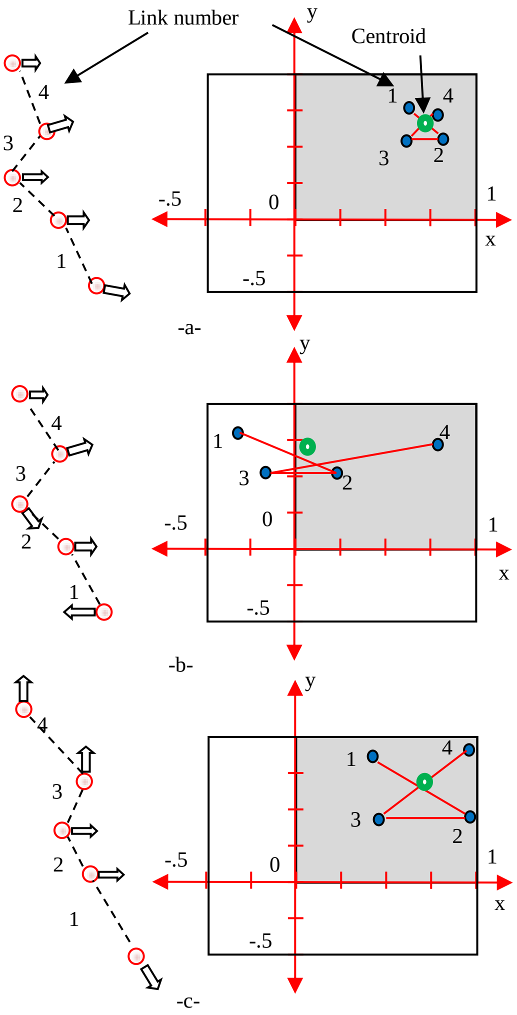


Figure 4.6 Examples of route quality (the green circles represent centroids and the red lines between the link's quality are contours).

The main idea of finding routing quality is to check if the links quality within a route are close to the good link quality (upper right corner). Moreover, to check if these links quality are close to each other which represent homogenise movement as explained in previous example. Route quality is computed according to the following functions, the magnitude of centroid (*cen*) of links' quality, the contour length of links (*con*) and the difference between the centroid of absolute links' quality and their real value (Δ*cen*). The magnitude of centroid (*cen*) used to check if the centre of all links' qualities are good and how far they are from the origin. The best *cen* will be when the route quality centre is close to the upper right corner. However, this function does not indicate in which quarter the *cen* is located. Another function is used which calculates the difference between the centroid of absolute links' quality and their real value (Δcen). This function reflects if all the links' quality are in the first quarter or if some of them are not, it will give us an indication of their negative values (how far they are from the axis). Finally, length of the contour that connect the links quality is calculated which reflects if the nodes are moving in the same directions and speeds as explained in previous example. One important parameter that could be extracted from the graph is the estimation of the time that the route will be available. This equals to the smallest node's time to live in the graph. Because ants may take time to reach their destination where route quality is calculated, the y axis is set so that the smallest time to live which considered as bad time to live is 0.4 seconds. As explained previously and from our results the end to end delay in its worse cases is close to 0.4 seconds. The route quality is directly proportional to the time that the route will be available, centroid and (1-centroid difference) and inversely proportional to (1+contour length).

$$cen = \sqrt{\left(\frac{\sum_{i=1}^{N} x_i}{N}\right)^2 + \left(\frac{\sum_{i=1}^{N} y_i}{N}\right)^2} \quad (4.3)$$

$$\Delta cen = \sqrt{\left(\frac{\sum_{i=1}^{N} x_i}{N} - \frac{\sum_{i=1}^{N} |x_i|}{N}\right)^2 + \left(\frac{\sum_{i=1}^{N} y_i}{N} - \frac{\sum_{i=1}^{N} |y_i|}{N}\right)^2} \quad (4.4)$$

$$con = \sum_{i=1}^{N-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \quad (4.5)$$

$$quality = gain * mintime * cen * \frac{1-\Delta cen}{1+con} \quad (4.6)$$

where *xi* , *yi* are mapped coordinate of link quality *i*, *mintime* is the estimated time for the route to be available, *N* node visited by the forward ant and gain is a positive number used to increase the adaptation gain of the feedback ant usually between 1 and 100.

The higher *cen* as better the route quality is. The nodes will have a long time to live and good distance. Small *con* means all nodes of that route have the same link quality and moving in homogeneous way.

When an extra node is added to the route it will add extra weight to *con* which will decrease the route quality (unless it has the same quality as its preceded node).

4.3.5 Channel selection

In order to reduce interference and increase throughput, the channel allocation algorithm in a node should be aware of channels being used by neighbour nodes. When the forward ant moves from source to destination, it checks for available channels. First it removes the two previous channels of the preceding nodes from the list of available channels. Then it checks if one of the previous visited nodes is one of its neighbours in order to detect if the route comes close to a previous node. If so, the node selected channels will be removed from available channels. Finally, if there remains a free channel the node will select it. Otherwise, the node selects the largest available bandwidth channel from shared channels. The channel availability parameter is calculated as follows:

$$\epsilon = \frac{\sum available\ channels * bandwith}{maxbandwidth * \max number\ of\ channels} \qquad (4.7)$$

A copy of the selected channel will be added to the forwarding ant. If the node uses a shared channel then a flag bit on the ant will be set up. For example, if a node has two interface cards and both of them are being used, then it has to use a shared channel. Otherwise, it will assign available channels to them.

Upon the previous flag bit, the destination node will decrease the value of quality in equation 4.6 in order to decrease the reinforcement leaning for this route. This will allow other better combination of channels and routes to rise up.

It should be noted here that in each node there are local flags for each interface card. Each interface is bonded to a flag which is set if the interface sends or receives data packets. Otherwise this flag is cleared after a predetermined period of time if no data activity is detected. When all of the node interfaces are in use then another flag alongside the selected channel number on the forward ant is set. This flag denotes that this selected channel is shared. A node can detect if its prior node is forced to use a shared channel. If two consecutive nodes are forced to use shared channels and have no common channel between them, the forward ant will die.

4.3.6 The Ant colony algorithm

Route discovery procedure starts when a source node starts a new data session towards a destination node. If the destination address is not found in the ant route table, the node starts broadcasting forward ants to find the destination. When a broadcasted forward ant reaches a destination a channel set up backward ant is sent which will set up the channels on the selected route. Although this could not be the best path, it will shorten the route set up time. Then bursts of forward ants are sent. These ants are moving according to random probability selection. The probability of a forward ant at node *i* selecting next node *k* is

$$p_i(k) = \frac{(\delta_{ik})^{\alpha}(\tau_{ik})^{\beta}(\epsilon_{ik})^{\gamma}}{\sum_{h\in Ni}(\delta_{ih})^{\alpha}(\tau_{ih})^{\beta}(\epsilon_{ih})^{\gamma}} \tag{4.8}$$

where α, β and γ parameters are used to control the relative importance of the pheromone intensity τ versus the channel quality ϵ and link quality δ. For forward ants, α and γ are greater than β to give the ant the ability to search. In general, for data routing and for route setup request ant, the value β is greater. *Ni* represents neighbour nodes. The link quality δ between the node and its neighbour could be calculated from mapped link quality, shown in Figure 4.3, from the following equation

$$\delta = \left(2.121 - \sqrt[2]{(1-x)^2 + (1-y)^2}\right)/2.121 \tag{4.9}$$

Equation 4.9 is calculating the goodness of the link quality by measuring the distance from the lower left corner, the worst link quality. this values is normalize to be between zero and one. The best value of link quality will occur when it is in the upper right corner of Figure 4.3. This value is always positive as the maximum distance between any two corners is 2.121.

The route life time which is extracted from minimum link quality in the route is calculated and sent to the source with the backward ant as well. The route quality is calculated as described earlier and used to update the intermediate pheromone table according to the following equation

$$\tau_{ij} = \begin{cases} (1-\varphi)\tau_{ij} + quality * \varphi & \text{if ant moves from i to j} \\ (1-\varphi)\tau_{ij} & \text{otherwise} \end{cases} \tag{4.10}$$

where τ_{ij} is the pheromone intensity from node *i* to node *j*. φ is small value between 0 and 1 which represents the evaporation factor.

Then the pheromone intensity values τ to each destination are calibrated to ensure their sums are equal to one as the sum of the probabilities to each destination should be equal to one.

In ant based routing protocol, data routed locally depending on pheromone intensities in ant routing tables. The ant routing table is frequently updated while the ants are moving and trying to find the best route. Accordingly, if data is routed directly upon the intensity values this may lead to a vast amount of channel switching in the network. New selected route needs new channel assignment which will add delays to the network. The probability of link breakage increases as the mobility increases. For long routes that have many nodes in between, this probability increases. To solve this problem the route table is separated from the ant table and to set up the route for a pre-predicted amount of time. This amount of time could be estimated from the overall route quality at the destination. When the destination node sets up a route, it computes the amount of time that the route will be available. Then it sends back the next reroute time which should be at least less than the amount of time that the route will be available. This value is sent back with the backward ants that setup the route. These ants are normal backward ants but with a flag bit that is being set up to indicate it will set up a route.

When the time comes for reroute, the most promising route learned by the ants should be selected. At reroute time, the source will send a route request ant. This is the same as the forward ant but it has two differences. Firstly, it depends more on pheromones intensities which reflect the learned path of the ants. For this the value of β is greater than α and γ in equation 4.6. Secondly, it has a flag that indicates it is a route request ant so that the node route it differently and the destination node sends back the route setup when the ant reaches the destination.

If the pre-predicted amount of time finished and the destination did not receive the route request ant, it will try to set up a route to prevent route break. This is done by checking a table of the last few ants that arrived at the destination. If it found a route with no overlapped channel it will set up the route.

In spite of the above precautions, a route break could not be prevented. Whenever a link is broken, the node sends back a link break to the source. Meanwhile, it communicates with the next best forwarding node from its ant table to establish a communication channel between them. They setup a common channel for data packet transfer and then it sends the data packet to it. The same procedure is repeated in the next node until the data reaches the destination node. When the source node receives a route break it starts the route setup procedure.

4.4 Implementation and results

4.4.1 Simulation environments

The proposed protocol is implemented using OMNeT 4.1 and INETMANET[27]. The number of interfaces is set to three, one for control packet exchange and two for data exchange. 11 channels have been used for data and one channel for control packets exchange as use used 802.11a. Each channel has a bandwidth of 54Mbps. The source and the destination are randomly selected. The first channel is used for control messages. Simulation area is 600m *600m. The transmission range is 150 m. UDP traffic application is used with packet size equals to 512 bytes with transmission interval of 0.0004 second. Random waypoint mobility is used with various speeds. Pause time is set randomly to be between 0.01 and 8 seconds.

Gain in equation 4.6 is set 80. The evaporation factor φ in equation 4.10 is set to 0.1 . For the forwarding ant α, β and γ are set to 1 (equation 4.8) while for route request ant β *is* set to *1* and α, γ are set to 0.3. Simulation time is set to 80 seconds and repeated twenty times.

In order to measure the improvement in throughput, the ASAR model proposed by Bowen LI et al [16] is being used for comparison as it is the closest protocol to the proposed protocol. Both protocols are using the same number of interfaces on both networks and the same rate of ants with interval equal to 0.005 seconds. This rate of forward ant generation is less than the rate proposed by Bowen LI in his paper. This is preferred in order to decrease the ant packets in the network for both protocols especially as high data rates are being used. The average ant generation rate is one forward ant packet per 12.5 data packets. Although a dedicated channel for control messages is being used and ant messages are relatively small in size compared to the UDP packets, using the rate proposed in Bowen LI paper (one forward ant per three data packets) may exhaust the network. All other parameters are set for both networks as mentioned above.

The following end to end network characteristic has been studied.

1-Throughput: is the measure of the total number of successful delivered data bits over simulations time for a specific node, averaged over the number of source-destination pairs.

2-End to end delay: is the measure of average delay of data packets. This is the time from sending the packet from the application layer at the source node to the time that the packet arrives to the application layer at the destination node, averaged over the number of source-destination pairs.

4.4.2 Results

Figure 4.7 shows the throughput for ASAR and the proposed ant protocol for different nodes speed. The network consists of 32 nodes.

The results show that the protocol offers better throughput then ASAR. The proposed protocol has the ability to utilize all the available channels, this will reduce the interference and allow the nodes to transmit and receive at the same time. While in ASAR, as channels are bound to interfaces, it cannot get the advantage of other channels. At the same time the interference in ASAR protocol is greater. As there are only two interfaces, there is always interference between the selected channels whatever the sequence of selecting them was.

Another factor that increases the proposed protocol throughput is the method of selecting the next forwarding nodes. The selection is depending on geographical information, thus links with higher data rates are more likely to be selected. While in ASAR it optimizes the route for optimal spectrum available and not the actual data rate being used. i.e. there could be a high bandwidth available, which is usually equal for all channels. ASAR do not predict if the nodes are going far from each other and they have to communicate at a low data rate. It could be noted when node speed increases the proposed protocol throughput decreases for two reasons. Firstly, at higher speeds the probability of link breakage is higher. Secondly, the route life time is shorter and the protocol needs to set up routes frequently which adds more delay to the network and decreases throughput. While in ASAR the throughput also decreases but the ratio of decreasing is less. This is because there is no channel switching in ASAR and the node forwards the data to the next best node when link breakage occurs.

Figure 4.8 shows the throughput for a network of 64 nodes. When the density of nodes increases the opportunity of interference increases, thus the efficiency of ASAR decreases more than the proposed protocol especially at 10m/s speed.

One of the problems of increasing the number of nodes is it will increase the search space for the ants; however it added one advantage to the proposed protocol. The number of neighbour to any node may increase and the possibility of a node with better link quality increase. Selection criteria in equation 4.8 will have more nodes to select from and will select nodes that have better link quality.

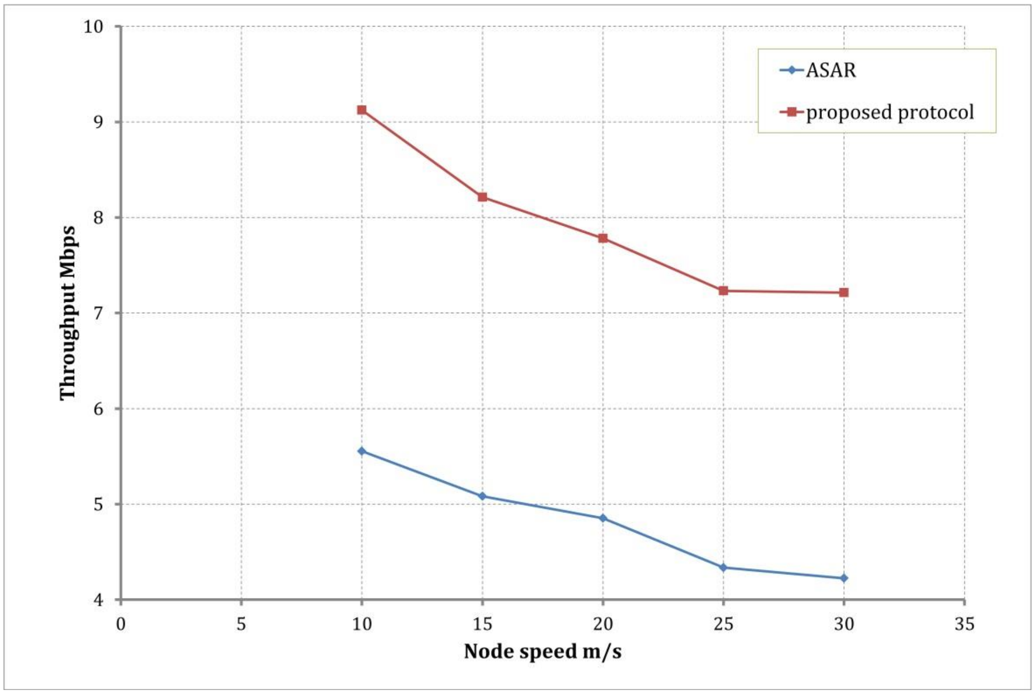


Figure 4.7 Throughputs under different speed for 32 nodes network

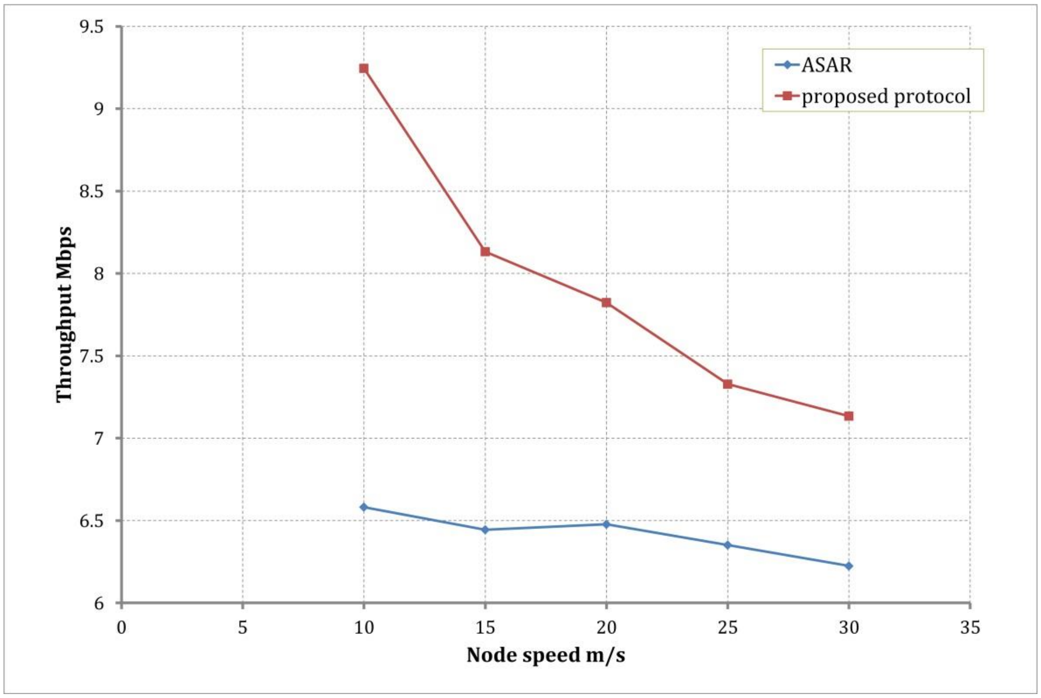


Figure 4.8 Throughput under different speeds for 64 nodes network

The better throughput of the proposed protocol comes with a cost. The end to end delay of the proposed protocol increased. Figure 4.9 and Figure 4.10 show the end to end delay for both protocols for 32 and 64 nodes network. The proposed protocol has more end to end delay as a result of channel switching which results in more queuing.

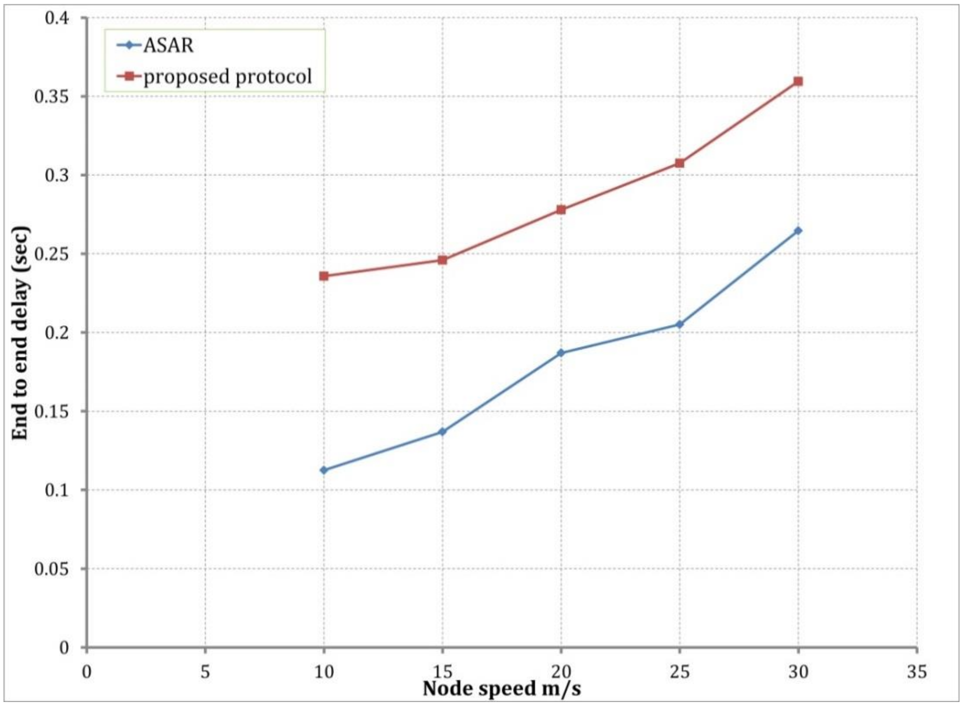


Figure 4.9 End to end delay under different speeds for 32 nodes network

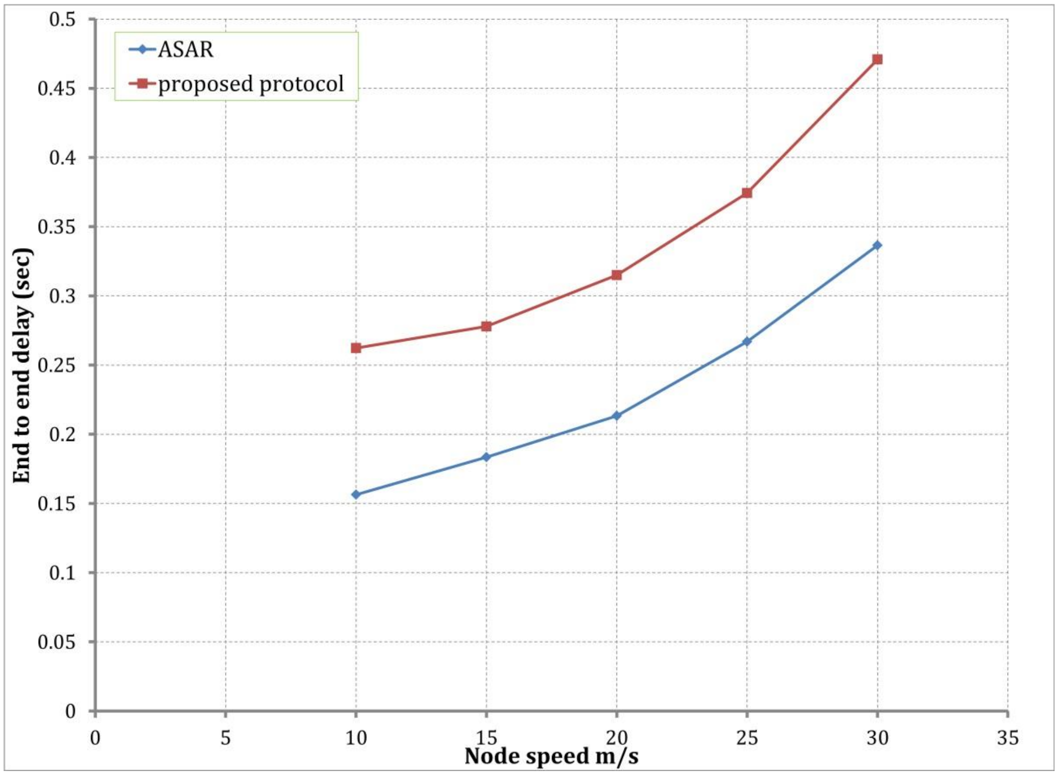


Figure 4.10 End to end delay under different speeds for 64 nodes network

4.4.3 Varying the routing coefficient

In this experiment, we vary the routing coefficient used by route setup ants in equation 4.8. The α parameter controls the amount of exploration the ants are allowed to do when they are setting up a route towards their destination based on link quality. When α is low the ants are concentrated on the paths with the best pheromone values. The ants will explore paths that have been indicated to be good by previous ants based on pheromone intensity. On the other hand, when α is high, the ants can follow paths with higher link quality more than the good paths that had been learned based on pheromone intensity. This way the selected routes will not reflect the learned information. It is clear from Figure 4.11 that the throughput will decrease when the value of α is increased. The results for different node speeds are shown with different curves. Moreover, the end to end delay will increase when α increases as the selected path is not the best path as shown in Figure 4.12.

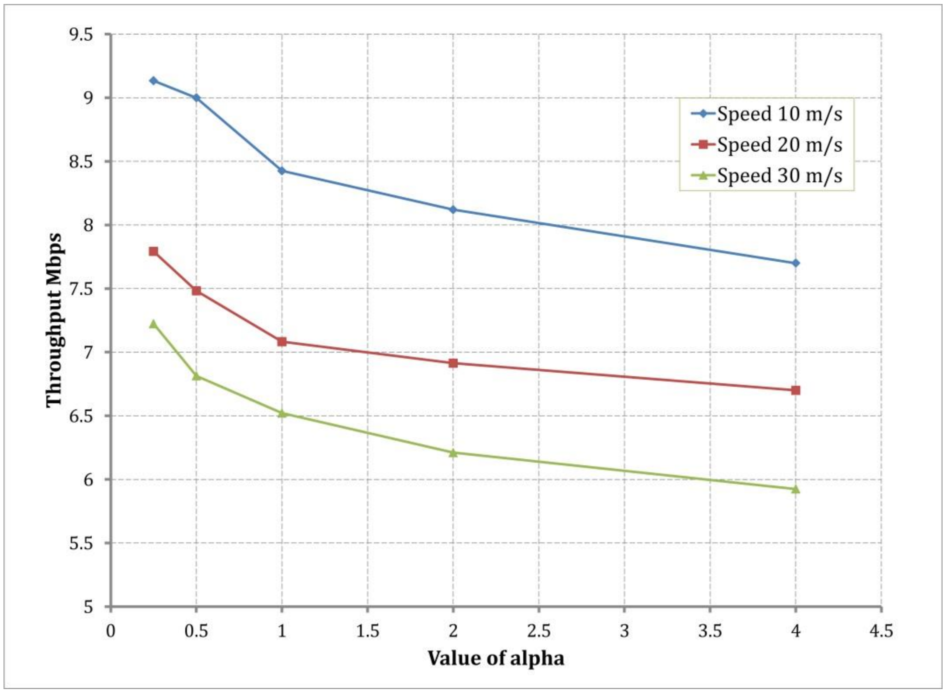


Figure 4.11 Network throughput for various values of alpha

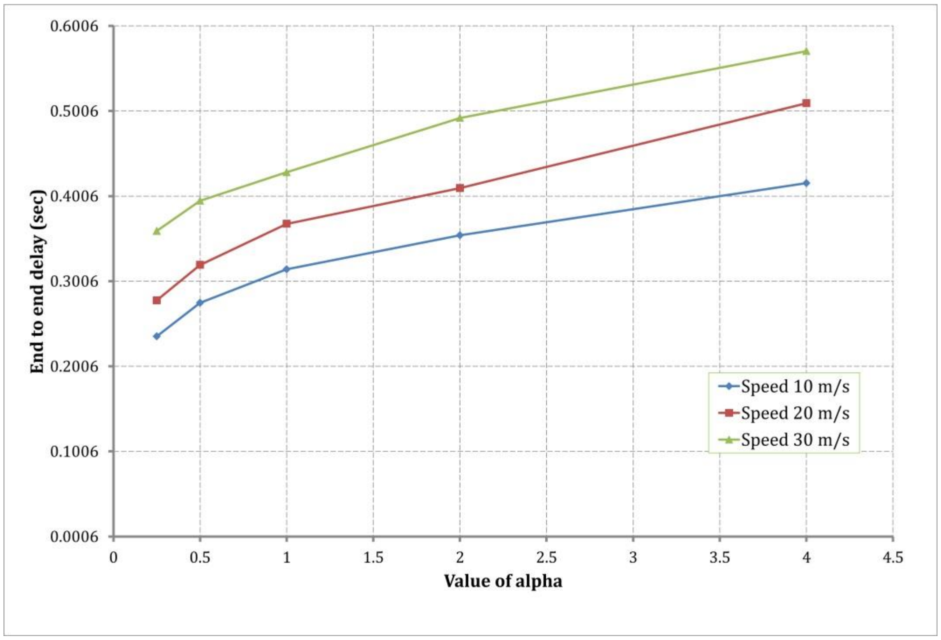


Figure 4.12 Network end to end delay for various values of alpha

Figure 4.13 shows the effect of increasing β on route selection. β controls the stochastic selection of route set up ants based on pheromone intensity. It defines how strong the preference of selecting a route based on learned information. When the value is low the preference is low and when it is high the preference is high. The figure shows that it is better to route data packet based on pheromone intensities (high value of β). Figure 4.14 shows the end to end delay of the network. It is clear that when the data packets are routed based on pheromone intensities, the delay is less.

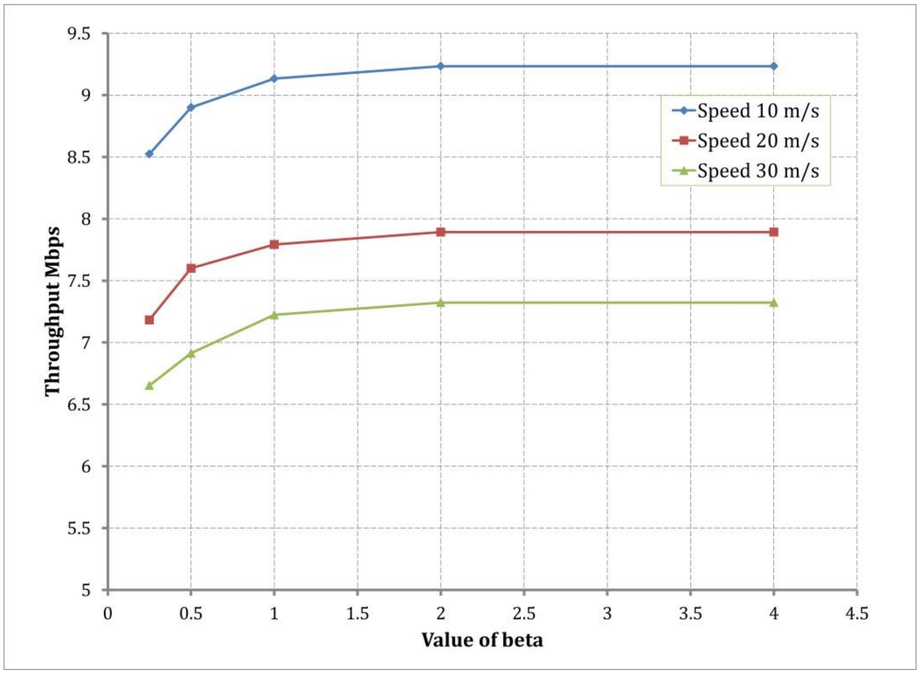


Figure 4.13 Network throughput for various values of beta

Figure 4.15 show the effect of selecting different values of γ on the performance of the network. γ controls the probability of selecting a path based on available channel bandwidth. When the value of γ is high, the probability of selecting a path based on the available channel bandwidth is high. The figure shows that the effect of increasing the value of γ has less effect than the other previous two variables. It shows that increasing γ will result in decreasing the throughput. This is due to the fact that the number of available channel in the network is high (10 channel) and the algorithm firstly selects from unused channels where all the bandwidth is available for the link. Figure 4.16

shows the end to end delay for various values of γ. It also shows that increasing γ will slightly increase the end to end delay.

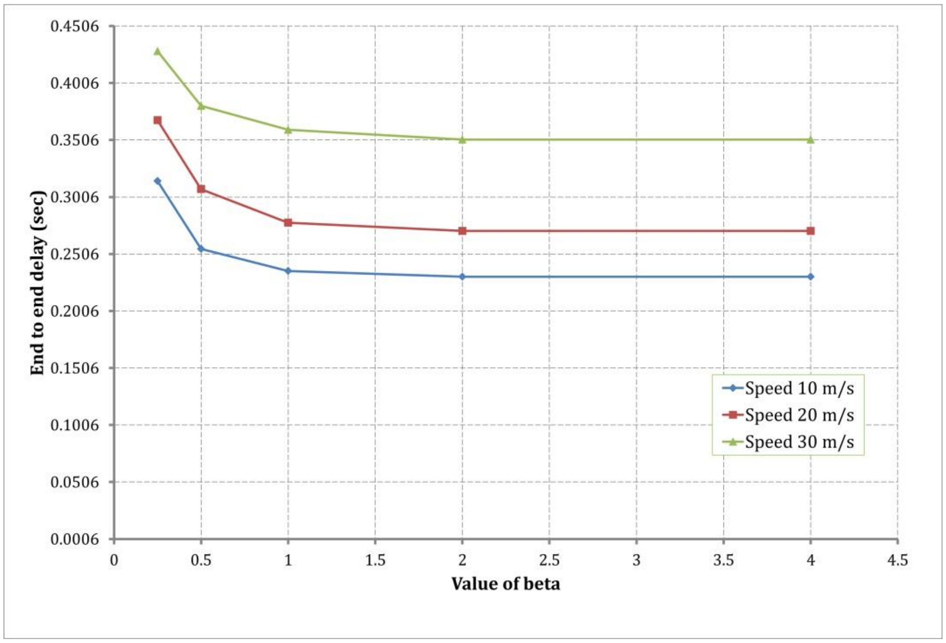


Figure 4.14 End to end delay for various values of beta

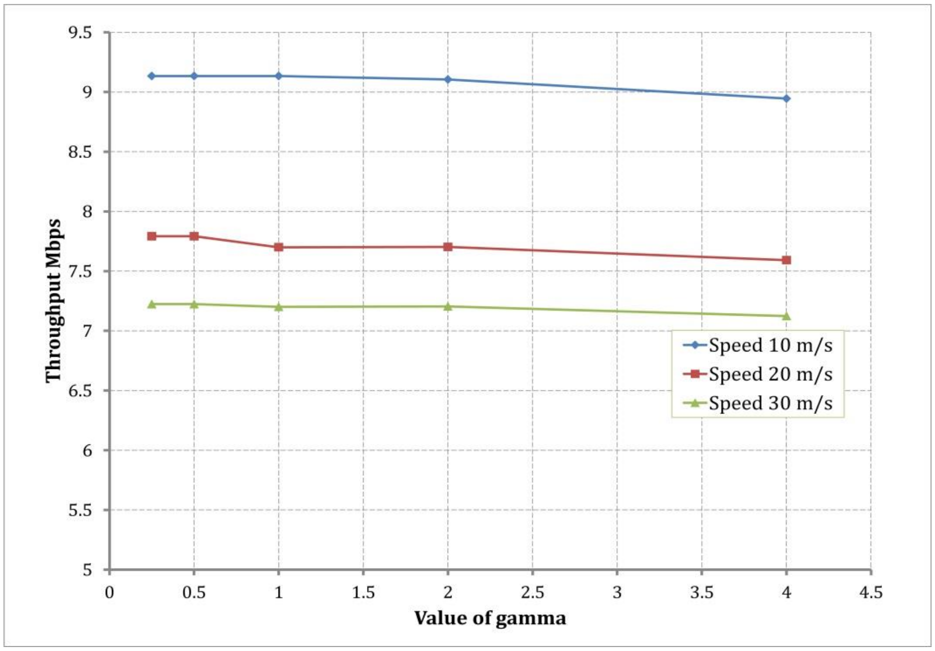


Figure 4.15 Network throughput for various valuses of gamma

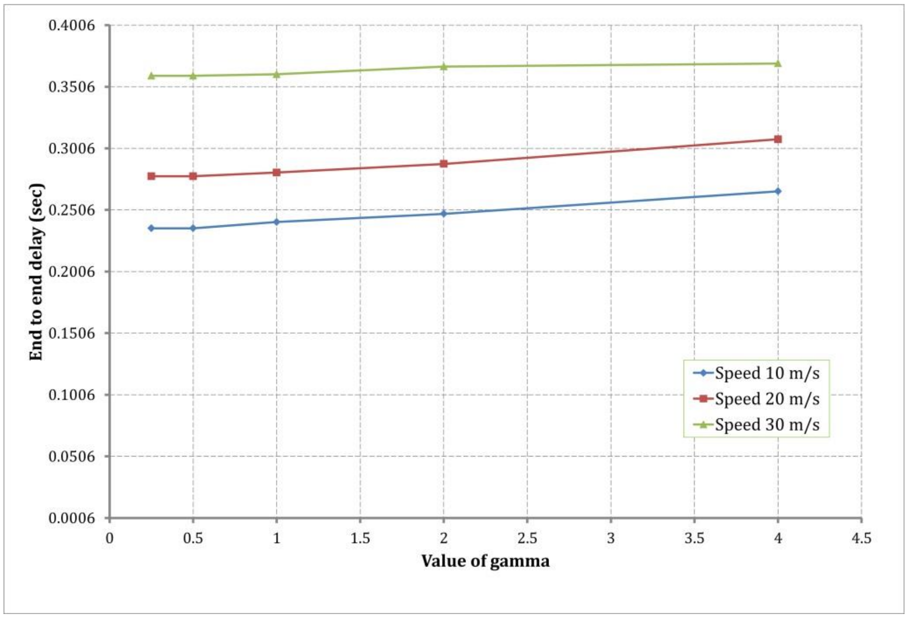


Figure 4.16network end to end delay for various values of gamma

4.5 Summary

This chapter introduced a routing method for multi-channel multi-interface mobile ad hoc network. The proposed protocol tries to find a stable route with good data rate between source and destination. The protocol is built to work with nodes that have three interface cards. The channels are dynamically assigned instead of using an interface per channel. In order to overcome the delay caused by channel switching, the protocol tries to decrease the number of channel switching by finding a stable route that lasts for a long time and with less probability of breaking. The protocol also tries to predict the length of time in which the route is valid and tries to establish a new route prior to the route break. Moreover, the protocol is inspired by ant colony algorithms which will react to route breaks in ant colony method. This protocol tries to find a link that has less interference as well as assigning the best sequence of available channel to the route in order to get benefit from the multi-channel multi-interface configuration. The results show that, using the same number of interfaces, the proposed protocol can increase the throughput over the ASAR protocol. Whenever the numbers of interfaces are less than the numbers of channels, it is better if the protocol can dynamically assign channels to the interfaces rather than using fixed channel per interface configuration. Using fixed configuration like in ASAR protocol leads to two problems. First, the protocol cannot use all available channels. Second, the interference between nodes increases and the nodes that interfere with each other should wait until the shared media becomes free in order to use it. This will decrease the throughput as it does in ASAR protocol.

References:

[1] Jipeng Zhou, Zhengjun Lu, Jianzhu Lu and Shuqiang Huang, "'A Channel Assignment Scheme for Location Service in Multi-Channel Mobile Ad Hoc Networks," *Ubiquitous Information Technologies and Applications (CUTE), 2010 Proceedings of the 5th International Conference on*, 2010, pp. 1-6.

[2] R. Soua and P. Minet, "'A survey on multichannel assignment protocols in Wireless Sensor Networks," *Wireless Days (WD), 2011 IFIP*, 2011, pp. 1-3.

[3] C. Toham and F. Jan, "'Multi-interfaces and Multi-channels Multi-hop Ad hoc Networks: Overview and Challenges," *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, 2006, pp. 696-701.

[4] Hon Sun Chiu, K.L. Yeung and King-Shan Lui, "'J-CAR: An efficient joint channel assignment and routing protocol for IEEE 802.11-based multi-channel multi-interface mobile Ad Hoc networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 4, 2009, pp. 1706-1715.

[5] F. Tang, L. Barolli and J. Li, "'A Joint Design for Distributed Stable Routing and Channel Assignment Over Multi-Hop and Multi-Flow Mobile Ad Hoc Cognitive Networks," *Industrial Informatics, IEEE Transactions on*, vol. PP, no. 99, 2012, pp. 1-1.

[6] Xiaofang Zhuang, Hongju Cheng, Naixue Xiong and L.T. Yang, "'Channel Assignment in Multi-Radio Wireless Networks Based on PSO Algorithm," *Future Information Technology (FutureTech), 2010 5th International Conference on*, 2010, pp. 1-6.

[7] Qian He and Ping Zhang, "'Dynamic Channel Assignment Using Ant Colony Optimization for Cognitive Radio Networks," *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, 2012, pp. 1-5.

[8] F. Bokhari, "'Channel assignment and routing in multiradio wireless mesh networks using smart ants," *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, 2011, pp. 403-404.

[9] S. Kajioka, N. Wakamiya, H. Satoh, K. Monden, M. Hayashi, S. Matsui and M. Murata, "'Implementation and evaluation of multichannel multi-interface routing mechanism with QoS-consideration for ad-hoc networks," *EURASIP J.Wirel.Commun.Netw.*, vol. 2010, 2010, pp. 17:1-17:12.

[10] Wenjing Yang, Xinyu Yang and Shunsen Yang, "'A Stable Backup Routing Protocol Based on Link Lifetime in Mobile Ad Hoc Networks," *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2009. UBICOMM '09. Third International Conference on*, 2009, pp. 202-207.

[11] C. Priyadharshini and K. ThamaraiRubini, "'Predicting route lifetime for maximizing network lifetime in MANET," *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*, 2012, pp. 792-797.

[12] C. Priyadharshini and K.T. Rubini, "'Integration of route lifetime prediction algorithm and particle swarm optimization algorithm for selecting reliable route in MANET," *Computing, Communication and Applications (ICCCA), 2012 International Conference on*, 2012, pp. 1-6.

[13] O. Dengiz, A. Konak and A.E. Smith, "'Connectivity management in mobile ad hoc networks using particle swarm optimization," *Ad Hoc Networks*, vol. 9, no. 7, 2011, pp. 1312-1326.

[14] P. Bahl, R. Chandra and J. Dunagan, "'SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks,", 2004, pp. 216-230.

[15] J. So and N.H. Vaidya, "'Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, 2004, pp. 222-233.

[16] Bowen Li, Dabai Li, Qi-hui Wu and Haiyuan Li, "'ASAR: Ant-based spectrum aware routing for cognitive radio networks," *Wireless Communications & Signal Processing, 2009. WCSP 2009. International Conference on*, 2009, pp. 1-5.

[17] Hong Huang, "'Adaptive geographical routing in wireless ad-hoc networks," *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 4, 2004, pp. 2749-2753 Vol. 4.

[18] P. Hsiao and H.T. Kung, "'Gravity routing in ad hoc networks: integrating geographical and topology-based routing," *Parallel Architectures, Algorithms and Networks, 2004. Proceedings. 7th International Symposium on*, 2004, pp. 397-403.

[19] B.R. Sujatha, V.P. Harigovindan, M.N.A. Namboodiri and M.V. Sathyanarayana, "'Performance analysis of PBANT (PBANT: Position based ANT Colony Routing algorithm for MANETs)," *Networks, 2008. ICON 2008. 16th IEEE International Conference on*, 2008, pp. 1-6.

[20] M. Kudelski and A. Pacut, "'Geographical Cells: Location-Aware Adaptive Routing Scheme for Ad-Hoc Networks," *EUROCON, 2007. The International Conference on "Computer as a Tool"*, 2007, pp. 649-656.

[21] S. Kamali and J. Opatrny, "'POSANT: A Position Based Ant Colony Routing Algorithm for Mobile Ad-hoc Networks," *Wireless and Mobile Communications, 2007. ICWMC '07. Third International Conference on*, 2007, pp. 21-21.

[22] M. Bandyopadhyay and P. Bhaumik, "'Zone based ant colony routing in mobile ad-hoc network," *Communication Systems and Networks (COMSNETS), 2010 Second International Conference on*, 2010, pp. 1-10.

[23] Pingshi He and Ziping Xu, "'Channel assignment and routing in multi-channel, multi-interface wireless mesh networks," *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, vol. 3, 2010, pp. V3-69-V3-74.

[24] K.V.S. Prashanth and A. Trivedi, "'Distributed channel assignment and routing in Mobile Ad Hoc Networks," *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, vol. 9, 2010, pp. 245-248.

[25] A. Trivino Cabrera, A. Garcia Mozos and E. Casilari, "'On the Suitability of Using the Residual Lifetime as a Routing Metric in MANETs," *Telecommunications: The Infrastructure for the 21st Century (WTC), 2010*, 2010, pp. 1-6.

[26] Cao Trong Hieu and Choong-seon Hong, "'A new routing protocol with high throughput route metric for Multi-Rate Mobile Ad-hoc Networks," *Local and Metropolitan Area Networks (LANMAN), 2010 17th IEEE Workshop on*, 2010, pp. 1-5.

[27] OMNET++ community, OMNeT++ Network Simulation Framework, Accessed: July 2013, [Online] Available: http://www.omnetpp.org/.

Chapter Five

Smart Data Packet Ad Hoc Routing Protocol

5.1 Introduction

The absence of infrastructure in Mobile ad hoc network (MANET) and the dynamic and continuously changing network topology compose real challenges to routing algorithms [1]. Routing in such a network is considered as an optimization process of locating the optimal paths between sources and destinations. A number of algorithms have been proposed to address routing in ad hoc networks [2-6].

Designing a protocol with two or more QoS constraints is known to be a NP-complete problem [7]. As problem size increases, the computational complexity of a problem increases, and more time and computation power are required to solve such a problem. Accordingly, finding shortest path as well as minimizing delay and avoid congested nodes is an NP-complete problem. Swarm intelligence has been adapted to solve routing problems due to their efficiency and distributed characteristic. However, to solve complex problem using swarm algorithms, the number of iterations required is proportional to problem complexity.

The main challenge facing a routing protocol is to cope with the dynamic environment of mobile ad hoc networks. As the network become more dynamic, the topology will change faster, and more swarm agents are needed to cope and quickly adapt with this changes in the network. In highly dynamic environment, more agents are required to detect link qualities in order to find best route. Increasing the rate of agent, in order to quickly adapt to the changes, will consume more resources. Moreover, each agent will require a feedback agent to adapt learned parameters which will increase resource consumption.

This chapter introduces a smart data packet routing protocol (SMART) based on swarm technology for mobile ad hoc networks. A new infrastructure will be presented where data packets are smart enough to guide themselves through best available route in the network. This approach uses distributed swarm learning approach which will minimize convergence time by using smart data packet which in turn will decrease the number of control packets in the network as well as it provides continues learning with the transmission of data packets and fast reaction to changes in the network environment. The learning information is distributed throughout the nodes of the network. This information can be used and updated by successive packets in order to maintain and find better routes. This protocol is a hybrid Ant Colony Optimization (ACO) and River formation dynamics (RFD) swarm algorithms protocol. While the main idea is based on the RFD algorithm, ACO algorithm is used to address the bottleneck of slow convergence rate of the RFD algorithm in the starting process of learning procedure [8]. ACO is used to set up multiple routes to destination at the initialization, while RFD is mainly used as a base algorithm for the routing protocol. The RFD algorithm offers many advantages toward implementing this approach. The main two reasons of using RFD are the small amount of information that required to be added to the packets (12 bytes in our approach) and the main idea of the RFD algorithm which is based on one kind of agent called drop that moves from source to destination only. This will eliminate the need of feedback packets to update the network and offers a suitable solution to change data packet into smart packets.

The proposed protocol introduces the use of data packet in the learning process of swarm algorithm. Rather than increasing the number of agents, data packets are used to act like drop agents. These packets are called “smart data packets” because they behave in smart way by avoiding congested nodes and they incorporate in the learning process. Moreover, they are called smart to distinguish them from ordinary packets (packets that are not acting like drops). As these data packet moves in the network, they adapt altitude tables, and the network learn about its environment. Using smart data packet will accelerate the sensing and reaction toward network parameters change.

ACO has been widely used in solving routing problems. In general, in ant based routing protocol, a node generates forward ant packets to find a destination. Ant packets move around the network in a random walk to find the destination. This movement is based on some stochastic probability function. When the destination is found, a backward ant is sent from the destination toward the source. As the backward ants move back to the source, they update the pheromone intensity on the links between the nodes. The path with higher pheromone intensity will attract more ants and data packets. After a period of time, the optimal route, depending on the optimization parameters, will be become more attractive and will be chosen by the data packets. More information about ant colony optimization can be found in chapter three.

River formation dynamics is a subset of swarm intelligence. It reflects how raindrops on highlands join together to form rivers [9]. These rivers tend to take shortest path to the sea. The RFD algorithm has been explained in chapter three. Implementing the RFD algorithm in ad hoc routing protocols provides many advantages. First of all, as there is no backward agent in the RFD algorithm, this will decrease the total number of control packets in the network. Another advantage is the simplicity of the algorithm; especially it relates altitudes to nodes rather than links. As the number of nodes is usually less than the number of links in a network. This minimizes the resource usage. More advantages come from the fact that the implemented RFD based protocols are using promiscuous communication mode, so the learning process is not local but all neighbour will be affected by the learning process. As drops are moving in the network, they update the altitude of the nodes. The drop carries recent node altitude which will be detected by neighbouring nodes. All neighbour nodes will update the corresponding node altitude in their tables. In other words, one change in a node's altitude which is announced by one drop packet is corresponding to changes of all links between that node and all their neighbours. A farther advantage is since RFD uses just drops, which element the need for backward agents and RFD drops adapts network parameters while they are moving from source(s) to destination(s), this will offer the opportunity to use data packets and make them act like drops. This allows data packets to guide themselves and contribute in the learning process. Other protocols usually require backward agents to adapt routing parameters. With other protocol, assigning a backward agent for each data packet will exhaust the network. Finally, the amount of control information appended to data packets is small, which make it easy to integrate the information into the data packets.

5.2 Related works

In [10] the authors analyse the performance of different types of routing protocols used in Wireless Networked Robotics (WNR). Different scenarios have been proposed to identify the features that affect the performance of traditional ad hoc routing protocols. The authors study four types of routing protocols Optimized Link State Routing protocol [11], Destination-Sequenced Distance Vector routing protocol [12], Ad hoc On Demand Distance Vector routing protocol [13], and Dynamic Source Routing protocol[14]. The study shows both node density and traffic have the major impact on the performance of routing protocols in WNR. Finally, the study shows that in average the AODV protocol performance could be considered better than OLSR, DSR, and DSDV.

In [15] HODVM routing protocol for Spatial Wireless Ad Hoc (SWAH) networks is proposed. Spatial Wireless ad hoc network consists of both mobile and static nodes. Two different protocols have been adapted to work with backboned network and non-backboned network. Static routing is used for static nodes, while AODV routing protocol is used for dynamic nodes. Moreover a node behaviour distinguishing algorithm is used to select multiple routes.

Adaptive Neuro-Fuzzy Inference System (ANFIS) has been used in [16] to select a single destination (server) from a group members belonging to anycast group in mobile ad hoc networks. The protocol uses three kinds of agents in order to cope with the QoS required. These agents are: static anycast manager agent, static optimization agent, and mobile anycast route creation agent. Moreover the protocol tries to select stable routes. Gradient Based Routing protocol (GBR) is a special kind of protocol developed for wireless sensor networks [17-21]. Many researches have been done on designing, enhancing, and analysing the performance of the GBR protocol. The main idea behind GBR is coming from the infrastructure of wireless sensor networks. Data streams in sensor network are directed towards sink node, which acts as a destination for all nodes in the network. As long as the destination is known, this destination node, the sink, floods the network with a cost metric toward itself. The sink cost metric is zero as it is the destination required. Each intermediate node that receives the message adds its cost to the message and rebroadcast it. Some conditions are added to check the cost with the cost that the neighbour nodes had to flood the message. As a result of flooding and accumulation cost values each time, a gradient is created toward the sink node. GBR protocol is suitable for static sensor network; however it produces many problems in dynamic networks. For dynamic moving network, the sink should broadcast cost messages more frequently in order for the nodes to update their routes and cost function toward the sink. Applying GBR in ad hoc network will produce more problems as each node in ad hoc network could be a source or a destination. As there is no infrastructure in ad hoc networks and there is no defined destination, a source node should first find its destination usually by flooding the network with broadcast message, than the destination will also broadcast a cost message and flood the network in order to build a gradient function. This will consume the network resources and time. The problem will be become more critical when the nodes are mobile as the rate of flooding should increase. A similar approach has been proposed in [22]. Temporally-

Ordered Routing Algorithm (TORA) [23] is an enhanced algorithm based on [22]. TORA builds a height metric to establish Directed Acyclic Graph (DAG). TORA is loop free algorithm which consists of three important phases, route creation, route maintenance, and route erase. TORA respond to link failure using localized control messages. Node in TORA should be synchronized and timing is important factor in the network where height metric depends on logical time of failure. Both route creation and route erasing build on broadcasting. Although the above protocols show the idea of gradient and heights, they introduce many problems. The problem of broadcasting, coping with dynamic mobile structure, synchronization, and the amount and types of control messages needed to exchange information. Most importantly, in GBR, gradient metrics are not adaptable and changed by flooding the network, and in TORA the height metrics are changed only to reflect links failure. No optimization technique is used and it is not guaranteed that the protocol will find best route. Moreover, the protocols use extensive number of control packets (forward and backward packets).

Swarm intelligence is well known optimization algorithm which inspired from the social behaviour of insects and other animals and used to solve optimization problem. Ant Colony Optimization is one of swarm technology algorithm that has been used to solving routing problems. AntHocNet [24] is a multipath hybrid routing protocol that uses source routing principles combined with ACO. Ants in this protocol compare both the travel time and hop count with previous visited ants and only broadcast if it is better. Certain concentrate have been added to decide if ants are going to be broadcasted or unicasted in the network. AntHocNet protocol features automatic load balancing as backward ants take into account the delay in each hops. Hello messages have been used to discover neighbour nodes and defuse pheromones. Link failure is treated using local repair technique. In case of route failure, the node even forward the packet to next best available neighbour node or it will try to locally repair the route by broadcasting route repair ant if no more links is available in its route table. On both cases the node informs its neighbour about link failure. This local repair technique usually will not lead to optimal solution rather than it only finds another path to the destination.

Cognitive Packet Network (CPN) introduced the use of intelligent packets where the capabilities for routing and control have been moved towards the packets themselves. In CPN, Random Neural Networks (RNNs) has been used in order to make routing decision. CPN has been used as routing protocol for ad hoc network in [25, 26]. More information is described in chapter two. To enhance the performance of the protocol, in [27] the authors present a multiple path approach for CPN in order to perform load balance among all network nodes. The algorithm carried out in two steps. The first step collect the information about available multipath in the network then a Hopfield Neural Network algorithm is used to refine and balance the distribution of packet around the network. In spite of replacing the RNN with Hopfield Neural Network the protocol still suffers from its main drawback which is the size of the packet. Moreover, each smart packet in CPN requires a backward acknowledgement packet. The use of random neural network or genetic algorithm adds more cost regarding processing power. Although information is shared using mailbox, the overall process is not distributed.

The proposed protocol is based on the RFD algorithm. It shares some common feature with the above presented protocols. It is a hybrid routing protocol. It uses reactive route setup whenever a route is not available. At the same time it uses hello messages to defuse topology information. Like AntHocNet and other swarm protocols, SMART protocol uses agents to search for best path. However, SMART protocol uses data packet to act like drop agents. Unlike other swarm protocols, there is no need for backward agents. Like CPN, routing decisions are influenced by the information carried by data packets. SMART protocol distributes the learned information around the network instead of carrying it within the packets. Only part of the information (altitudes) which reflect the change of network state will be carried by the packets. Moreover, SMART protocol uses distributed learning which is more efficient than local learning algorithm proposed by CPN.

5.3 Smart data packet ad hoc routing protocol

The proposed protocol is a hybrid routing protocol. Hello message has been used to discover neighbour nodes as well as to propagate information around the network and build a proactive routing table. At the same time, when a node requires a connection to any other node, it starts a reactive route procedure through broadcasting ant-drop messages around the network. Smart data protocol is implemented in network layer and uses promiscuous communication mode to monitor neighbours’ packets in order to update its routing tables. Each node will contain an altitude table where the altitudes of other nodes are stored. Another table is used to store the node itself altitudes towards other nodes. Based on the RFD algorithm the selection of next node, regarding a specific final destination, is proportional to altitude differences between the node and its neighbours. The lower the altitude of the next node, the higher the probability to be selected. Drop packets are routed stochastically over different paths using altitude tables. Smart data packets are routed using restricted stochastic algorithm where they can guide themselves and contribute in the learning process. Non smart data packets are routed using minimum altitude as minimum represents the best discovered path. Drops are always sent from source node to destination throw-out the whole data session in order to maintain and monitor the route as well as to search for better route.

Nodes can communicate and share information directly with neighbours and indirectly with other nodes that are out of its transmission range using the diffusion method which is implemented by hello messages.

The route table is represented as altitude table as shown in Table 5.1,where Ni represents neighbour node i. Ti represents time delay of node i. and ALT_{ij} is the altitude to destination j through node i (node i altitude to destination j)

Table 5.1 Altitude table

		Destinations				
Neighbours	Time delay	D_1	D_2	D_3		D_k
N_1	T_1	Alt_{11}	Alt_{12}	Alt_{13}		Alt_{1k}
N_2	T_2	Alt_{21}	Alt_{22}	Alt_{23}		Alt_{2k}
N_3	T_3	Alt_{31}	Alt_{32}	Alt_{33}		Alt_{3k}
N_j	T_j	Alt_{j1}	Alt_{j2}	Alt_{j3}		Alt_{jk}

Table 5.2 My_altitudes table

destination	My altitude
D_1	Alt_1
D_2	Alt_2
D_{k-1}	Alt_{k-1}
D_k	Alt_k

Another important table is my_altitudes table shown in Table 5.2. This table contains the node itself altitudes to other destinations.

5.3.1 Route setup

Route discovery starts by checking altitude table for the destination. If the source node has no information about the route to the destination, it starts broadcasting ant-drop messages. When an intermediate node receives the ant-drop, it will either unicast or broadcast it. If the intermediate node do not has a route to the destination it will broadcast it, otherwise it will unicast it. In unicast mode, instead of using link pheromones intensity to route packets, node altitudes are used. Probability selecting of next node is based on equation 5.1. The decreasing gradient is calculated according to equation 5.2, as following

$$\mathrm{P_n(j,k)_d} = \begin{cases} \dfrac{\mathrm{decreasingGradient_d(j,k)}}{\sum_{\mathrm{l \in V(j)}} \mathrm{decreasingGradient_d(j,l)}} & if\ k \in\ V_n(j) \\ 0 & if\ k \notin\ V_n(j) \end{cases} \qquad (5.1)$$

$$decreasingGradient_d(j,k) = \frac{altitude_d(j) - altitude_d(k)}{T(k)} \qquad (5.2)$$

where $P_n(j,k)_d$ is the probability of drop *n* at node *j* with destination *d* to select next node *k*. *$V_n(j)$* is set of neighbour node of node *j* for drop n. *T(k)* is the average time that node k needs to send a packet (well be explained in the next subsection), *altitude$_d$(j)* is altitude of the node *j* toward destination *d*.

Due to broadcasting, the number of ant-drops will increase and build a multipath from the source to the destination. Ant-drop packet contains a field called travel time. When a node selects the next node it adds the corresponding next node expected time delay from it altitude table to this field. This accumulated value will reflect the expected time to travel from the source node to that node. If a node receives another copy of the same ant-drop packet, having the same sequence number, it will forward it only if the number of hops or the expected travel time is less than the previous forwarded ant-drop packet. This selective approach will reduce the overall number of ant-drops in the network which decreases the overhead by removing unpromising ant-drop packets from the network. Another condition on forwarding a duplicate copy of the same ant-drop messages is when the first hop that taken by ant-drop is different from the previous ones. This will allow us to build sufficiently multiple disjoint paths which will add more protection against link failure [24, 28].

When an ant-drop reaches the destination, it will be converted to a backward ant-drop. This backward ant-drop moves backward from destination to source depending on the recorded route in the ant-drop received. This backward ant-drop message will update the altitudes table (my_altitudes) which reflects nodes altitude to that destination. Intermediate node will calculate updating factor as following

$$dt_j = \frac{1}{\frac{T_{route}^{j} + t_{hop} * h_j}{2}} \tag{5.3}$$

where T_{route}^{j} is the total accumulated route time by backward ant-drop while traveling back from destination to node j. t_{hop} is a parameter used to represent time required to send a packet in unloaded condition which is set to 3 msec [24]. h_j is number of hops from destination to node j. equation 5.3 is a cost function which averages two cost functions, the end to end delay and hop count.

The proposed protocol is mainly based on the RFD algorithm, so the idea of ant colony optimization is adapted to modify altitudes (nodes altitudes) rather than links weights (pheromones intensity). The backward ant-drop will adopt altitude as following

$$altitude_d(j) = \nu * altitude_d(j) - (1 - \nu) * dt_j \qquad \nu \epsilon [0,1] \tag{5.4}$$

where *altitude*$_d$*(j)* is altitude of the node *j* (receiver node) toward destination node *d*. *v* is the adaptation factor. Choosing small values for *v* will lead to rapid change of altitudes and forget its learning information while larger values lead to more smother changes. Value of 0.9 is chosen.

Each node saves a copy of its altitude before changing and updating by equation 5.4. Each time a copy of backward ant-drop passes through a node, equation 5.4 is computed for the original saved value. Only the best value is conducted.

When the source node receives backward ant-drop packet, it starts sending data packets.

5.3.2 Smart data packets and drops routing

In order to find better paths and maintain the route, drop packets are sent throughout data session. Data packets are being used to act like drops and called smart packets. As long as data packet travels from source to destination it is better to use these data packets to reinforce the learning process. This makes data packets detect congested nodes and update altitudes. Therefor successive data packet will select different route as the altitudes change. Smart data packets act like drops; they search and react to network condition as well as they contribute in the learning process of the network. The learning information is stored in the altitude tables throughout the network. Although drops and smart packets act the same way, smart data packet are routed in in restricted way. This minimizes the latency as well as decreases the probability of data packet being sent far away from the destination. At the same time drops will continue discovering other parts of the network.

The rate of drops is set to one drop per 0.5 second. The drops are propagated according to the gradient probability function in equations 5.1 and 5.2. Using time delay parameter will reduce the probability of selecting congested nodes. The more congested the node, the more it is not preferred to be selected. The higher value *T(k)* is representing higher distance according to the RFD algorithm and the node is not preferable to be next forwarding node.

Each drop packet contains a field that represents the recent altitude of sender node. Drops change node altitudes by the process of eroding and adding sediments. The node new altitude to the final destination of drop is attached to the drop packet. All neighbour nodes will update their tables according to the new altitude carried by the drop packet. This acts like a distributed learning procedure where one change affects a group of nodes. Each node also monitors its sent packets. Unlike ant based algorithms where an ant updates specific link pheromone intensity that is moving along, one drop alters a node altitude and all neighbours are updating their tables according to this change.

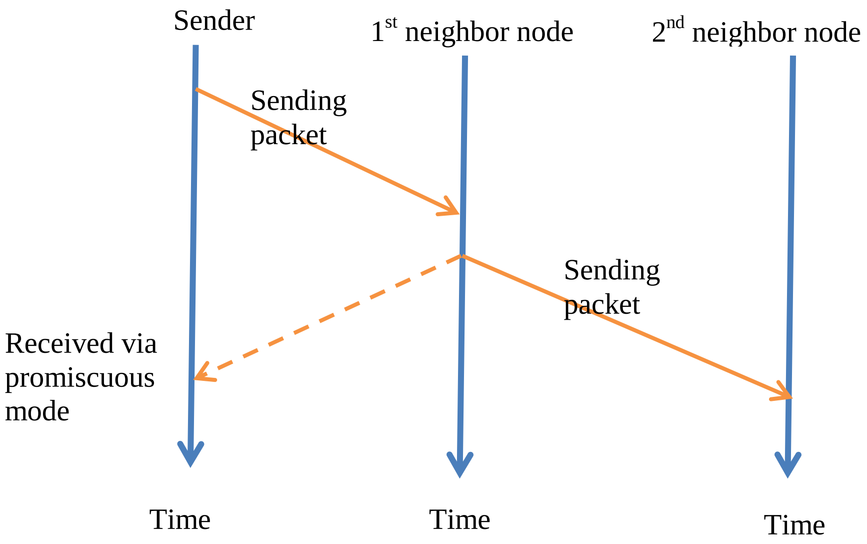


Figure 5.1 Time between sending a packet and receiving a copy of it

When a node starts sending a packet, it computes how much time is needed for the next node to send it again as shown in Figure 5.1. The time delay is the amount of time between sending the packet and receiving a copy when transmitted by next node. Whenever this time is available, upon successful reception of the transmitted packet using promiscuous mode, the node updates the average time delay value. It keeps tracking of this by computing the running average of the time needed by its neighbour to forward its packets. This value is stored in altitude table.

$$T_k(t) = \gamma T_k(t-1) + (1-\gamma)C_k(t), \quad \gamma \in [0,1] \qquad (5.5)$$

where $C_k(t)$ is new delay in node k at time t, T_k is time delay for node k. T initially is set to be equal to the time required by an unloaded node to send a packet (set to 3ms) [24]. γ is a parameter regulating how quickly the formula adapts to new information (set to 0.7). Using this type of distance will help in selecting uncongested node. When a node becomes more congested, its time delay will increase. This makes it undesirable and nodes will forward packets to other neighbour. This reflects how water drops behave in nature. When a group of rivers pour in the same valley, and the amount of outgoing water from the valley is less than the amount of incoming water, a water lake is created. As the water level is increased, the water on some edges will start to leak out to the other sides of the mountains around the lake. Water drops that fall on places close to these edges will follow to other sides with the leaking water rather than to the valley.

When drops move in the network they erode the altitudes of the nodes. The amount of erosion is proportional to altitude difference between sender and receiver nodes. The selected forward node altitude is taken from the altitude table and used to calculate the gradient. The erosion is calculated using

$$erosion_d(j) = \propto (altitude_d(j) - altitude_d(k)) \qquad (5.6)$$

$$altitude_d(j) = altitude_d(j) - erosion_d(j) \qquad (5.7)$$

where *$erosion_d(j)$* is the amount of erosion at the node *j* toward destination *d*, *$altitude_d(j)$* is the altitude of node itself, k is the next selected node. α is a positive constant number between 0 and 1 which reflects erosion factor (set to 0.7). High value of α will lead higher erosion and missing of optimal solution. While very low value may lead the algorithm to fall in local minimum. Due to nodes mobility and dynamic infrastructure of mobile ad hoc network, low values for α are not preferred.

At the same time drops add sediment to node altitude. The amount of sediment is proportional to the amount of sediment carried by the drop as well as inversely proportional to the altitude difference of the node and next node altitudes. When the altitude difference is low which represent flat surface, the drop will deposit more sediment.

$$sediment_d(j) = (\beta + \varepsilon * (1 - (altitude_d(j) - altitude_d(k)))) * carried\_sediment, \qquad \beta + \varepsilon \in [0,1] \qquad (5.8)$$

$$altitude_d(j) = altitude_d(j) + sediment_d(j) \tag{5.9}$$

β and ε are constants controlling the amount of sediments deposit (set to 0.1 and 0.1 respectively). Carried sediment reflects the path characteristic. Paths with higher slopes will result in more carried sediment. Setting β and ε to high value will lead to quickly depositing sediments. This will lead to a convex topographical structure where the altitudes around the source are less curvature. While lower values will lead to concave structure and the flat part will close to the destination. Lower value will make drops search around the destination for better delivery nodes. Flat surface around the source will increase the probability of sending data packet to directions opposite to the destination direction, which in turn will move them far away from the destination.

The sediment carried by the drop to next node is calculated as the sediments carried by the drop after subtracting the amount of sediment that has been deposit by the process of sedimentation and adding the amount of sediment that eroded from the node.

$$carried_{sediment} = carried_{sediment} + erosion_d(j)\ -\ sediment_d(j) \tag{5.10}$$

Another type of sediment adding occur periodically every fixed amount of time

$$altitude_{d\in w}(j) = altitude_{d\in w}(j) + \theta, \qquad \theta \in [0,1] \tag{5.11}$$

where w is a set of all known destinations. θ is the amount of sediment to be add(set to 0.01). θ acts as forgetting factor. High value will quickly erase learned information, especially for recent learned paths. Low value is chosen because high value may corrupt recent learned information, beside that there are other methods that will also help in adapting the information like the punishment procedure if a link broken. The time period between regular additions is set to 1 second.

To implement smart data protocol, extra fields should be added to data packets in order to act like drops. It is significant to keep the added information to data packets as small as possible and not to overload the data with many extra bytes. Drop packet themselves are small in size. Three parameters have been added to data packet, altitude, carried sediment and the sender address each of them are four bytes in size.

Smart data packets should have the opportunity to discover new routes. This means they should move like drops. In order to limit them from exploring long paths and keep them close to the best known path, nodes that have minimum altitude and less congested are having minimum distance and considered as best shortest path, data packets are routed using greedy and restricted method. Data packets use altitude table and move according to the following random probability selection function below.

$$P_n(j,k)_d = \begin{cases} \frac{decreasingGradient_d(j,k)^{\partial}}{\sum_{l \in V_n(j)} decreasingGradient_d(j,l)^{\partial}} & if\ k \in\ V_n(j) \\ 0 & if\ k \notin\ V_n(j) \end{cases} \tag{5.12}$$

where ∂ is a constant number greater than one, and is set to 4 to achieve the greedy movement. *$V_n(j)$* is set of neighbour node of node j.

Usually the number of data packets is much higher than drop packets. This high number of data packets could ruin the learning process especially as these packets are moving in a greedy way. This reflects flooding in nature. To limit the erosion that is occurring due to this high number of data packet, the percentage of erosion and sedimentation by data packets is reduced 10% of drop packet.

5.3.3 Hello messages and information diffusion

Hello messages are used to propagate information around the network. Essentially, hello message is used to declare node presence; moreover it is used to carry information about node neighbours. Hello message is extended to carry K elements from its altitude table (K=10 is used) [24]. Using this idea the altitudes of the nodes will be distributed around the network.

The receiving node will update the altitude toward this node (my_altitudes) and updates its altitude to those destinations that are carried by the hello messages by a factor proportional to the difference between node altitude and received altitude.

$$altitude_d = altitude_d - \mu(altitude_d - hello_{altitude_d}) \ , \quad \mu \in [0,1] \tag{16}$$

$hello_{altitude_d}$ is the altitude of destination d in hello message, μ is the adaptation constant (set to 0.1).

Although distributing information by hello messages help to form paths to destinations but it has some drawbacks. The reliability of this information is not high. First, this information does not address the congestion in the nodes. Second, since hello messages are sent every hello interval period this information may be out of date.

5.3.4 Link failure

The protocol detects route failure in two ways. The first one is by detecting the missing of hello message from a neighbour for a time period more than *allowed hello loss*. *Allowed hello loss* period is set to be twice as *hello interval* which is inherited from AODV protocol. The second way of detecting link failure is through missing acknowledgment after sending data or drop packet.

If a route failure occurred, the node deactivates the route in route altitude table. Then it updates its destination attitude table. If the loss of a neighbour affects the table then it will broadcast a notification to its neighbours.

Deactivating helps in situations where the neighbour node is temporarily unreachable. Instead of removing the node from the table and lose all information gathered through time about it, just deactivating it will keep the information. If the disconnection is temporary then when the node returns back there is some information left. For long time periods, the regular sediment addition will erase the information of deactivated node’s altitudes.

Whenever the reason of link failure is due to the failed transmission of data packet, than the node will try to send the packet to the next best neighbour. At the same time if the altitude of this neighbour is higher than the altitude of the node itself, it will change its altitude to the same amount of the next best neighbour. In a worse case, if there is no node to deliver the packet to, the node will return the packet to its sender. In this case the altitude of the node to this destination becomes one. The previous node and all neighbours will update their tables with the new altitude value of that node and will route further new packets according to new altitudes. When a node becomes a dead end, there is an opportunity that this node and its neighbours are constructing a local valley. Local valleys are small cavities on the altitude surface. These local valleys prevent drops from reaching their final destinations, global minimum. Local valleys should be filled quickly as they work as attracting zones for packets which in turn increases the number of lost packets. The above procedure will prevent wasting many packets on local valleys. Without this procedure, the number of packet required to fill a valley is at least equal to the number of nodes between the dead end node and a node with valid link to the destination. To prevent wasting this amount of packets, this procedure will damp it with the same packet that is returned. At the same time, the packet is not lost and the learning process is going on. However, the number of punished nodes is set to two nodes in our implementation. This will minimize the probability of losing sensitive information in case of temporarily link breakage. This prevents the loss of learned information as returning back packet to many nodes will erase altitudes toward the destination and set them all to one.

5.3.5 Smart data packets routing example

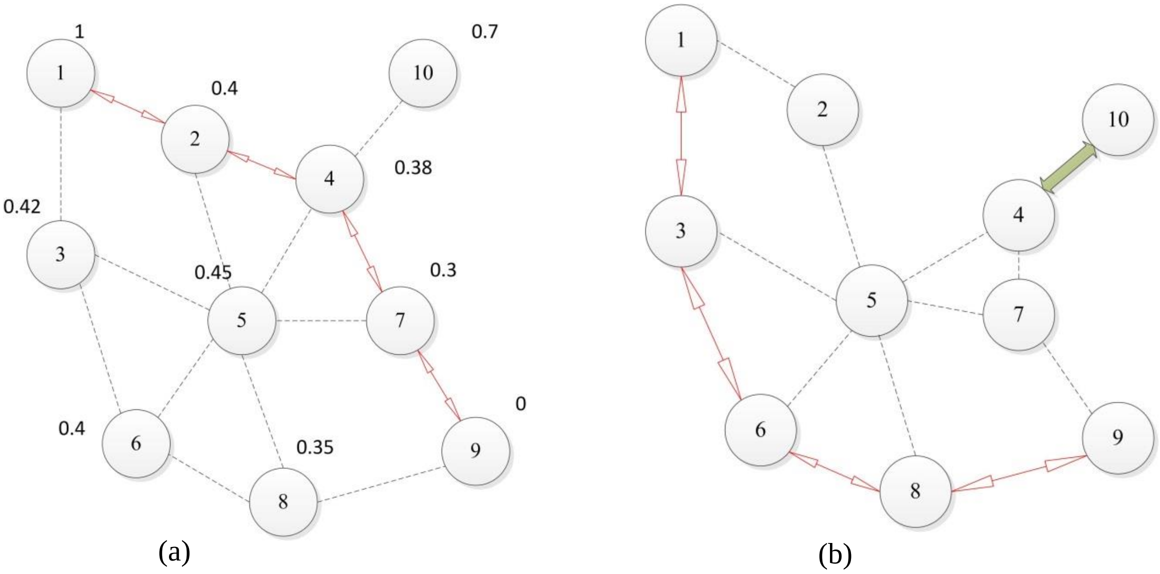
The section explains how smart data packets are routed in the network. In Figure 5.2.a, node 1 sends packets to node 9. The numbers over each node represent nodes altitude toward node 9. Red arrows represent best route. To explain different approaches of how the routing problem is solved and the advantages of the proposed protocol, all protocols are assumed to start by selecting the route 1-2-4-7-9. After selecting the route, node 4 has been involved in a communication session with node 10. Then node 4 moves far from node 2 and the link between them is broken. Node 4 becomes congested and the route is not available as shown in Figure 5.2.b.

Starting with AODV protocol, when node 4 becomes congested, AODV has no mechanism to detect congestion and it waits until route break occurs. After all, a new route set up procedure is required.

If an ACO based protocol like AntHocNet is used; first let us assume that the rate of ant generation is equal to one ant per second. When route break occur and as AntHocNet is multipath protocol, node 2 forwards the packet to next available node (node 5) and the route become 1-2-5-7-9. Both nodes 5 and 7 are in the transmission of node 4 and will be affected by it, however the protocol does entirely depends on ant agents to detect network status. The protocol will wait for at least one second, if not more, until an ant pass through the route and change the pheromone intensities of the links in that route (by the backward ant).

In the proposed protocol, as node four becomes congested, its distance become higher (equations 5.2 and 5.5). The time needed to send the packet from node 2 to node 4 will increase. Node 1 will update the time of node 2 according to equation 8.

Accordingly, in node 1, the probability of selecting node 2 or 3 changes (probability of selecting node 2 decreased depending on the delay). When the link breaks, node two will forward the data to node 5 at same time it changes its altitude to be equal to node 5 (0.45). Node 1 will get a copy of the sent packet and update the altitude of node 2 in its altitude table to 0.45. Now the altitude of node 3 is less than node 2, therefor node 3 will have higher probability to be selected as next node and the route will change to 1-3-5-7-9. The data packets behave in smart way as they always try to avoid congested area in the network. An important point here is all the above occurred during data packets sending. There is no need to wait for one second like ACO based protocol to send an agent and detect the change in the network parameters. The altitudes have been changed with data packets. No need for feedback agents. Moreover, the changes not only affect the node but it will propagate backward and affect previous nodes' altitudes. It should be noted that the process here is not only message forwarding, it is a distributed learning and optimization technique which continuously learn and react according to network status to find best route. Another advantage of using smart packets is it will minimize convergence time. For example, if the number of ants required in order to find a solution (best route) was 100 ants, with a rate of one ant per second, this will take 100 second. As for SMART protocol with the same rate of agents (drops) and with data rate of five packets per second, the time will be shorter. Each second there will be one drop and five smart data packets that act like drops, which means there will be six drops per second. The convergence time will be about 16.66 second.



(a) (b)

Figure 5.2 An example of smart data packet routing.

5.4 Implementation and simulation results

The performance of the proposed protocol was evaluated by comparing it with AntHocNet and standard AODV protocols (with local repair) [9]. AODV is chosen because it is a well-known and almost considered as reference protocol in this research area. AntHocNet protocol is a swarm based algorithm and is chosen because it outperforms many ant routing protocols in many aspects [29, 30].

Simulation results are generated using OMNet++ as simulation software. A model for AntHocNet protocol was implemented based on [24]. As for the AODV protocol, the INETMANET add-on package of the OMNet++ is used. An important point worth mentioning here is that all the setting of protocols are tried to be match as possible. The rate of ant's generation is equal to the rate of drop's generation. The rate of hello messages for all protocol is equal as well. As we used the standard AODV with no modification as a comparison routing protocol, we kept with its setting and any other setting required for proposed protocol was inherited form AODV protocol.

5.4.1 Simulation environments

Three scenarios have been implemented to test our protocol. Previous studies show that ad hoc network can produce best performance if the number of neighbours is between six to eight [31]. However, the node density has been chosen to be close to 6.25 in order to have good connectivity. With this node density, there is a good probability for a node to have multipath to its destination. At the same time, as the environment become more aggressive, nodes speed increases, the probability of link failure increases, which provide a good environment to test our protocol.

In the first scenario, 32 nodes have been randomly placed in a 600 *600 m^2 environment. Simulation time was set to 200 seconds. The simulations are repeated for twenty times with different seeds.

The medium access control protocol is the IEEE 802.11 DCF. Packet size is 512 bytes. Five mobile nodes selected randomly to act as sources and five other nodes acts as receivers. Each node generates a packet every 0.2 second. The network remains silent in the first second. The nodes start sending data at third second and keep sending until the end of simulation, which gives one second for some hello packet to be generated before starting data session. Data traffic is generated using constant bit rate (CBR) UDP traffic sources. Two mobility models are used, random waypoint (RWP) model and Gauss Markov (GM) model [32], to test the performance of the protocols.

In the second scenario, the number of nodes increased to 64 and to keep node density almost equal to the first test, the area is increased to 850*850 m^2. This scenario is used to test the scalability of the protocol, keeping the same node density and increasing network size. Simulation time was set to 400 seconds. The simulations are repeated for twenty times with different seeds. The number of sources is increased to ten and the number of destination nodes is increased to ten as well. Only random waypoint (RWP) model has been used.

In order to test the performance of the protocol under different node densities, a third scenario is introduced. This scenario is similar to the first one, except that we fixed the speed of the nodes to 10 m/s. The number of nodes varied from 20 up to 100 nodes. Random waypoint (RWP) model is used in this scenario. Moreover, to analyse the impact of the traffic load on the performance of the protocol, we use this scenario to test the protocols under different traffic loads. The number of nodes is set to 32, the speed is set to 10 m/s, the load is increased by increasing the number of packets sent per second (packets rate).

Other common parameters for all scenarios are set as following. Each node has a radio propagation range of 150m and channel capacity of 54 Mb/s. Hello time intervals is set to 0.5 second [33, 34]. Pause time is set randomly between 0.1 and 1 second.

The following end to end network characteristic has been studied [24, 35].

1-Throughput: is the measure of the total number of successful delivered data bits over simulations time for a specific node, averaged over the number of source-destination pairs.

2-End to end delay: is the measure of average delay of data packets. This is the time from sending the packet from the application layer at the source node to the time that the packet arrives to the application layer at the destination node, averaged over the number of source-destination pairs.

3-Jitter: is the variation of packet delay which is averaged over the number of source-destination pairs, averaged over the number of source-destination pairs.

4-Routing overhead is the total number of control packets sent divided by the number of data packets delivered successfully.

5- Number of route requests: is the total number of route request generated (route setup procedure) by specific node averaged over the number of source-destination pairs.

5.4.2 Results

Figure 5.3 and Figure 5.5 show the throughputs of SMART routing protocol compared with AODV and AntHocNet protocols under different nodes speed. It is clearly seen that SMART protocol performs better than both AODV and AntHocNet. Figure 5.3 shows that a significant increase in throughput can be achieved by using SMART protocol. For 10 m/s the throughput increased up to 17% over AODV and 13% over AntHocNet under RWP mobility model. When nodes become more mobile, the altitude table, which reflects the topology, will need more updates, therefor the throughput decreases. Comparing the results of the network under RWP mobility with GM mobility, we can see the protocol preforms better under RWP mobility. There are two main reasons for that. Firstly, in RWP mobility, the nodes tend to move toward the centre of the simulation area and move away from the simulation area boundary. This will leads to fluctuation in node density and as a result, the path lengths become shorter. The nodes are better distributed under GM mobility model. Secondly, nodes movement in GM mobility is correlated. In RWP, node can make a sudden change in its direction independent on its previous direction. These two reasons led to better results under RWP mobility model.

Figure 5.4 shows the throughputs for 64 node network under RWP mobility. The throughput increased about 30% over AODV in low speed and about 13% over AntHocNet. We can notes that SMART protocol preforms better than the other protocols because it uses data packet in learning process. Comparing Figure 5.3 and Figure 5.4, the effect of network size and number of sources is obvious and the throughput for all protocol decreases.

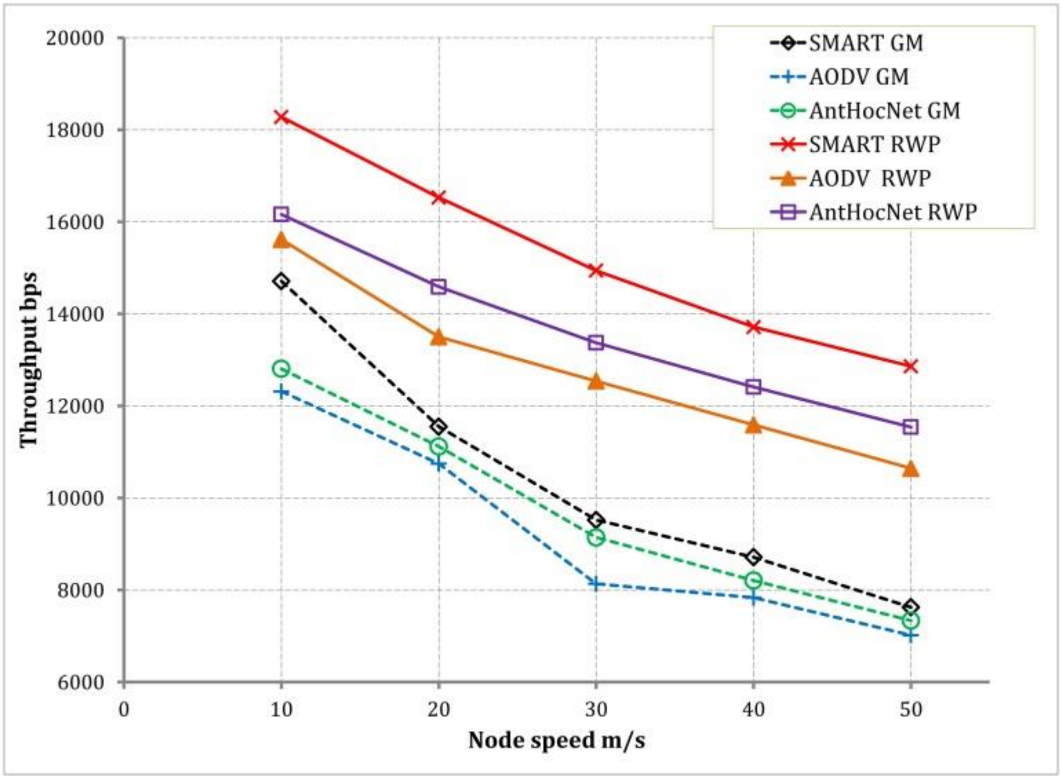


Figure 5.3 Average throughput for 32 nodes network under various speed values.

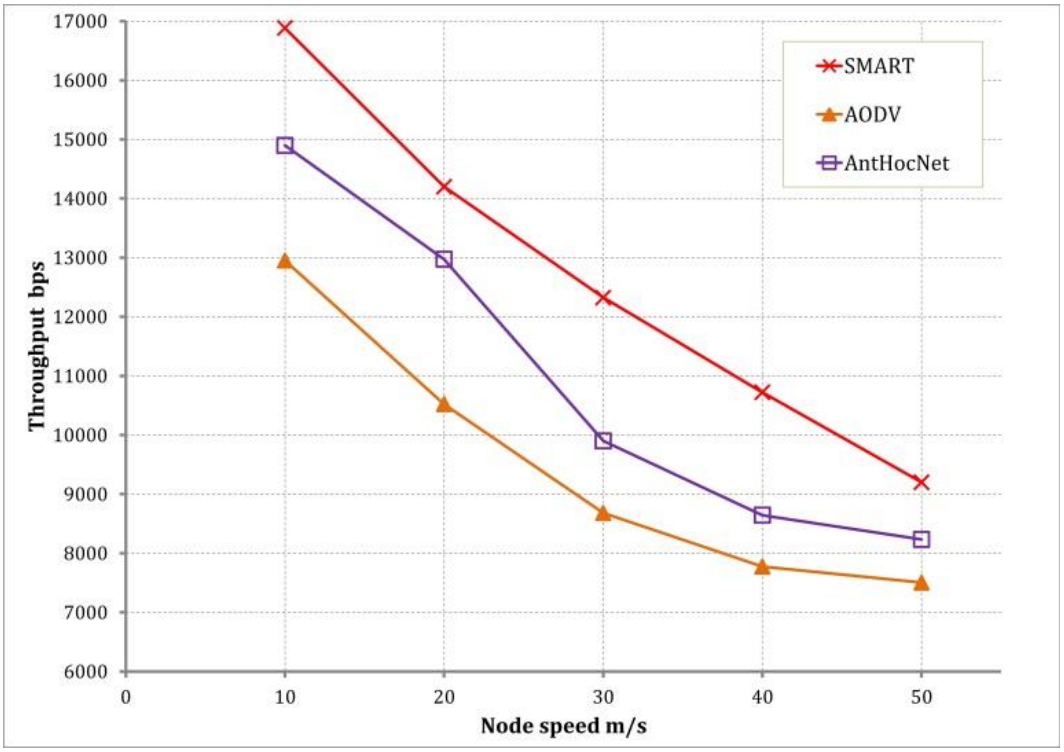


Figure 5.4 Average throughput for 64 nodes network under various speed values.

The end to end delay for the first scenario is shown in Figure 5.5 and for the second scenario is shown in Figure 5.6. Both figures show more than 96% enhancement in end to end delay. SMART data protocol produces better results than the others. The main factors that contribute in network latency and jitter are congestion, queuing and route changing. In AODV most of the delay occurs due to route setup and broadcasting of control packets and route maintenance as well as queuing of data packets. At the same time, the criteria of selecting next forwarding node and finding the optimal path from source to destination also contributes in the delay. SMART protocol minimizes the number of broadcasting, and tries to select a non-congested node. If a node had a route failure, it forwards the packet to other best node. At the same time when a packet is moving around the network, the process of learning is still carried on and nodes update their altitude tables continually.

AntHocNet forward data depending only on regular pheromone (there are two type of pheromones tables in AntHocNet, regular and virtual) which mostly trained by backward ants. This table may not have all possible paths to destinations as this require huge number of ants and training cycles, so in many situation route break occurs. The method of packet forwarding and continues real time learning of SMART protocol leads to better performance. Smart data packets allow the network to learn more rapidly and the packets can find other routes. Another important factor, if a better route is found, the AntHocNet needs many ants to enhance the weight of the route to be selected. SMART protocol is based on RFD and adopts faster.

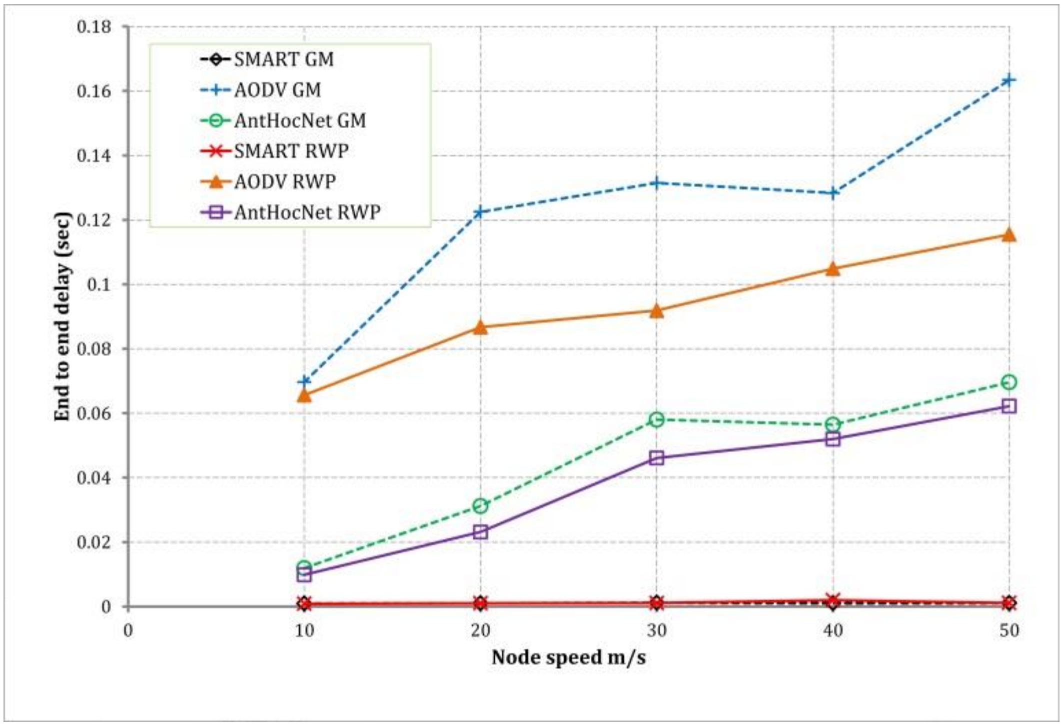


Figure 5.5 Average end to end delay for 32 node network under various speed values.

Comparing Figure 5.5 with Figure 5.6, the effect of network size and number of sources is obvious. As the network become larger and number of source nodes increases, the probability of link breakage increases and the repair time will be longer as well as the network becomes more congested. As the network size increases, the search space for finding better routes also increases.

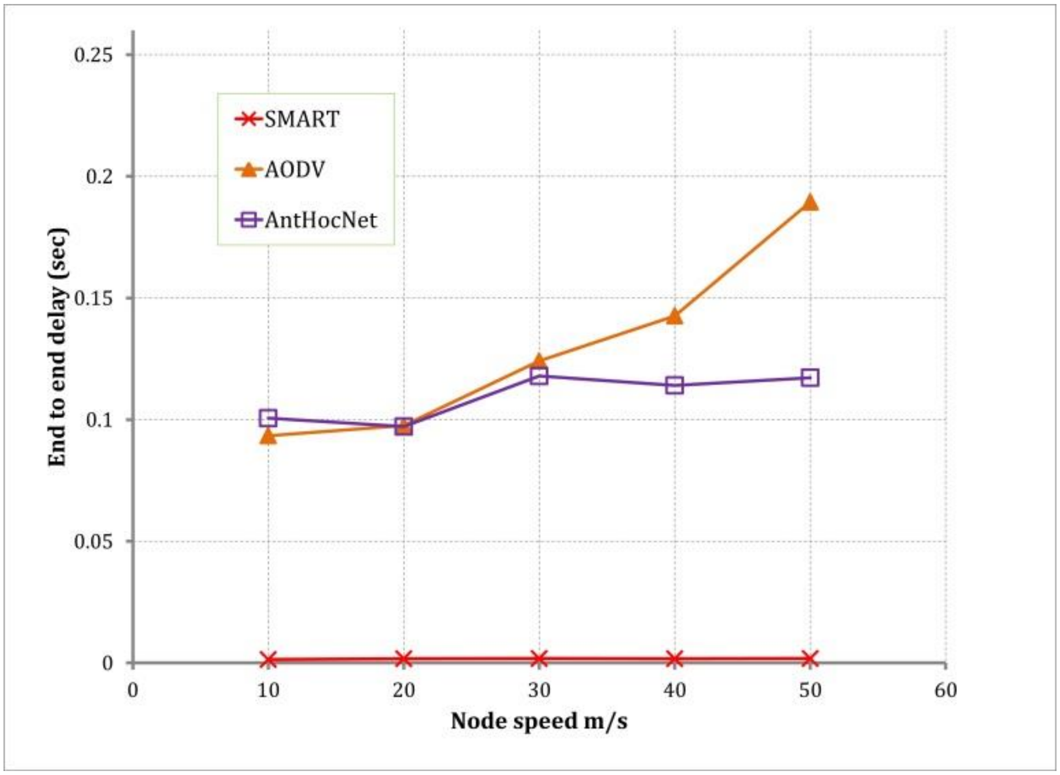


Figure 5.6 Average end to end delay for 64 node network under various speed values.

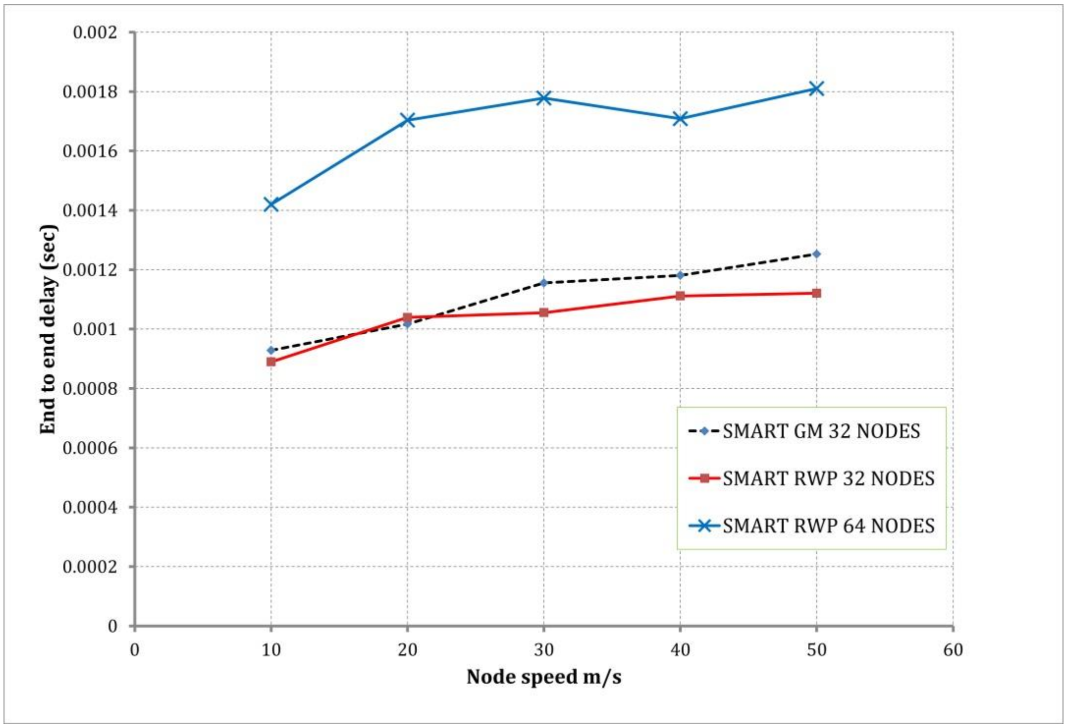


Figure 5.7 SMART protocol end to end delay under various speed values.

The end to end delay of the SMART protocol is very low. Figure 5.7 shows the effect of speed on the end to end delay for the first and second scenarios. For 32 node network, the average end to end delay under both RWP and GM mobility is between 0.89 up to 1.3 millisecond and increasing slowly as node speed increases. The delay is higher in 64 node network and starts from 1.4 millisecond up to 1.8 millisecond.

The time needed (T_p) to send a packet in the network, using two way handshaking (DATA- ACK), could be calculated as below [36-38]

$$T_p = T_{DIFS} + T_{SIFS} + T_{BO} + T_{DATA} + T_{ACK} \qquad (17)$$

where T_{DIFS} is the Distributed InterFrame Space time (34 µs), T_{SIFS} is Short InterFrame Space (SISF) time (9 µs), T_{BO} is backoff interval time (min 67.5 µs) , T_{DATA} is MAC Protocol Data Unit (MAC PDU) plus Physical Layer Convergence Protocol(PLCP) header plus PLCP preamble (for 512 byte it will be 106.6 µs), and T_{ACK} is acknowledgement time (24 µs). For 32 node network, in average, the end to end delay time is equal to five successful data transmission times.

Figure 5.8 and Figure 5.9 show the jitter for the first and second scenarios respectively. Again SMART protocol overcome both protocols and has less jitter. Variation in network structure due to nodes mobility causes link breakage. Both AODV and AntHocNet have route recovery mechanism which consists of queuing and rebroadcasting. SMART protocol is based on instantaneous data packet rerouting to other better route in case of route failure. As the data packets moves from source to destination, all nodes within the route will continue their learning process by the data packet. Moreover, all nodes that are neighbours to the route will also learn as the process of learning in SMART protocol is distributed. An extra point to address here, data packets are routing themselves through multipath to the destination through nodes around best discovered path. This will load balance the network, as well as it creates a valley like structure which its lowest end is at the destination. When a node in-between becomes unreachable, the data packets should easily find another path to the destination. Figure 5.10 shows SMART protocol’s jitter.

Both AntHocNet and SMART protocols are hybrid routing protocols and they send control messages throughout the entire data session in order to maintain the route between the source and the destination. This could produce more overhead in the network and costing the network to use more resources.

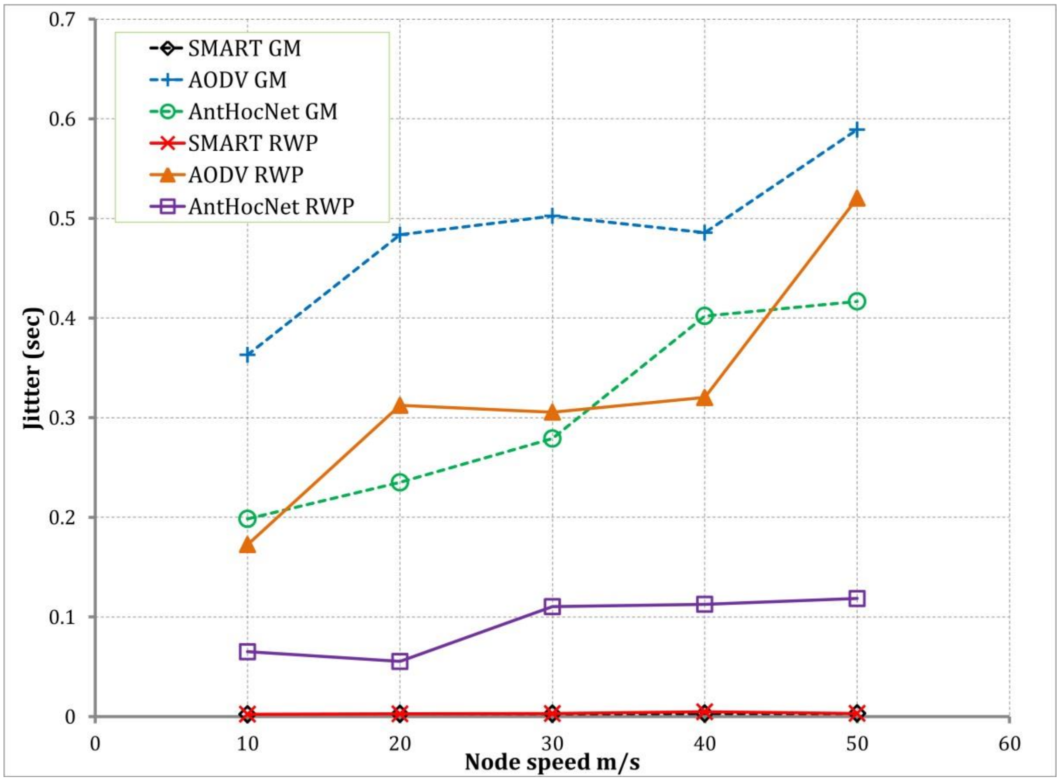


Figure 5.8 Average jitter for 32 node network under various speed values.

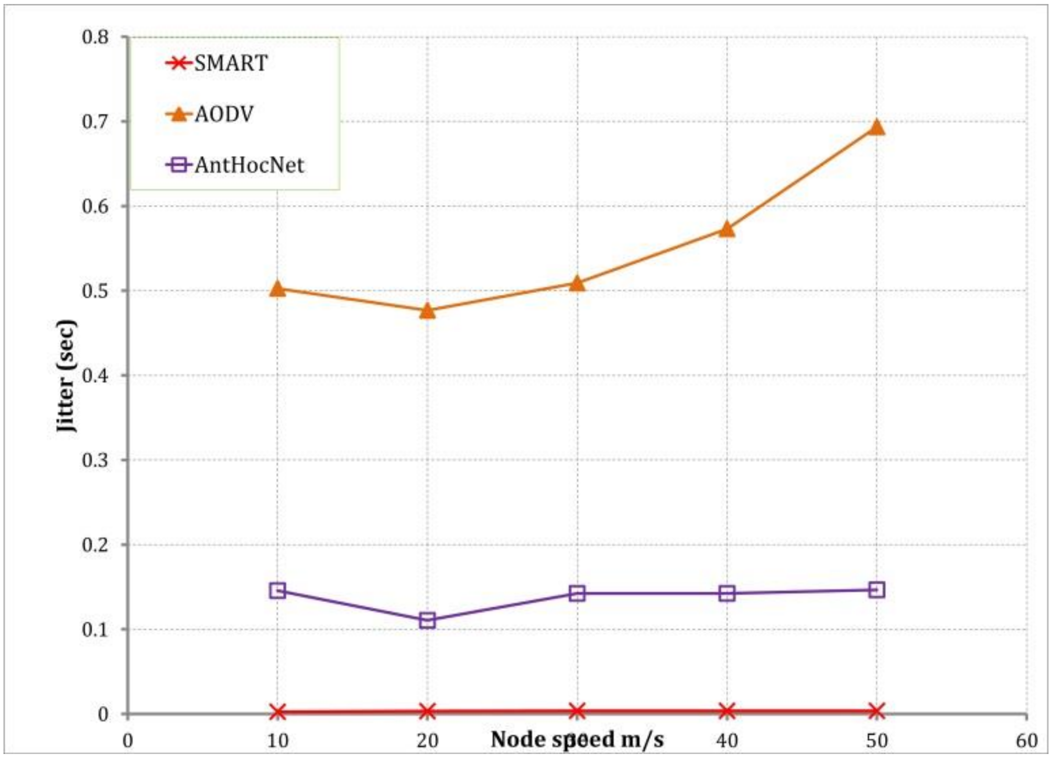


Figure 5.9 Average jitter for 64 node network under various speed values.

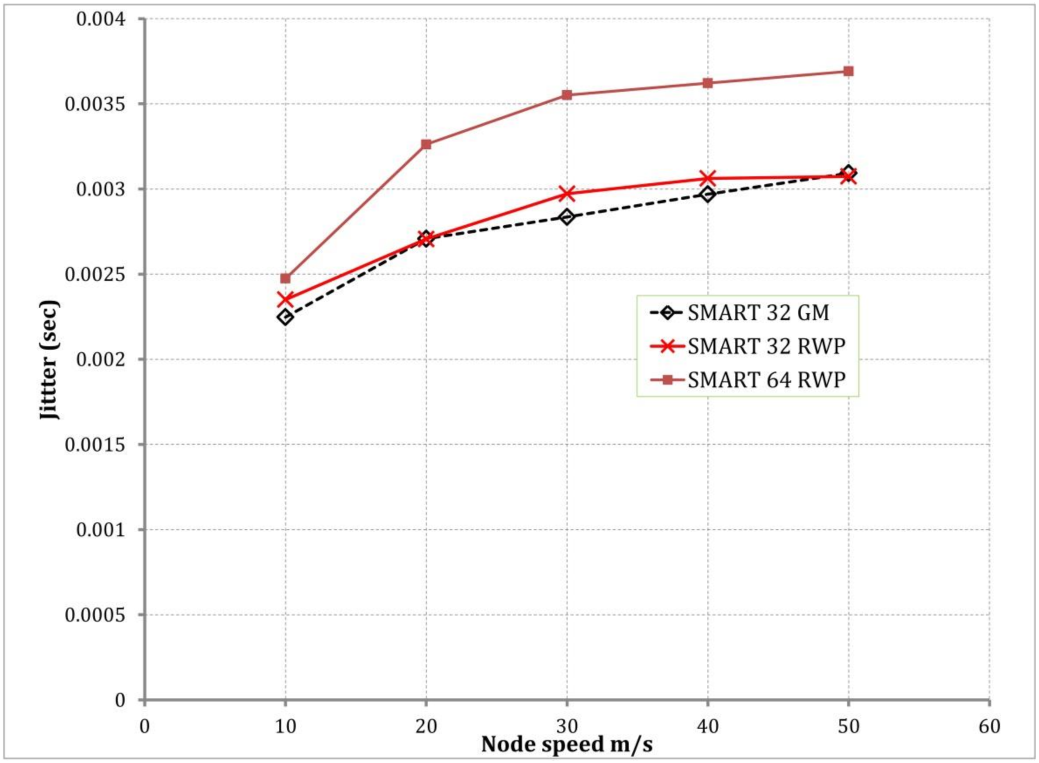


Figure 5.10 SMART protocol jitter under various speed values.

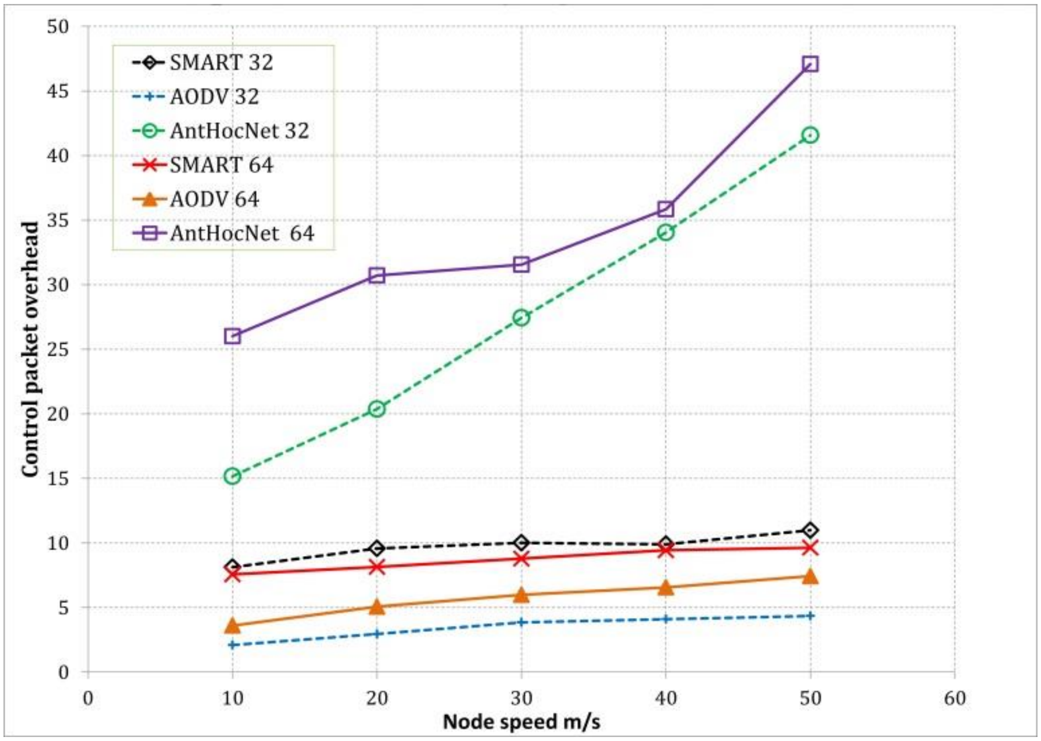


Figure 5.11 Control packet overhead under various speed values.

Figure 5.11 shows the control packet overhead of the three protocols for both the first and second scenarios under RWP mobility. SMART protocol generates more control overhead than AODV protocol, however the control overhead is less than AntHocNet. The absence of backward agent and the use of SMART data packets led to less control overhead than AntHocNet.

The third scenario is used to study the effect of node density on the protocol performance. Figure 5.12 shows the effect of node density on network throughput. When the number of nodes is less than 40 nodes in the network, the connectivity of the network is degraded and the throughput decreased. When the number increases over 60 nodes, the network become more congested and the throughout decreases again. From the figure it is clear that the proposed protocol overcome both other protocol throughout all number of nodes.

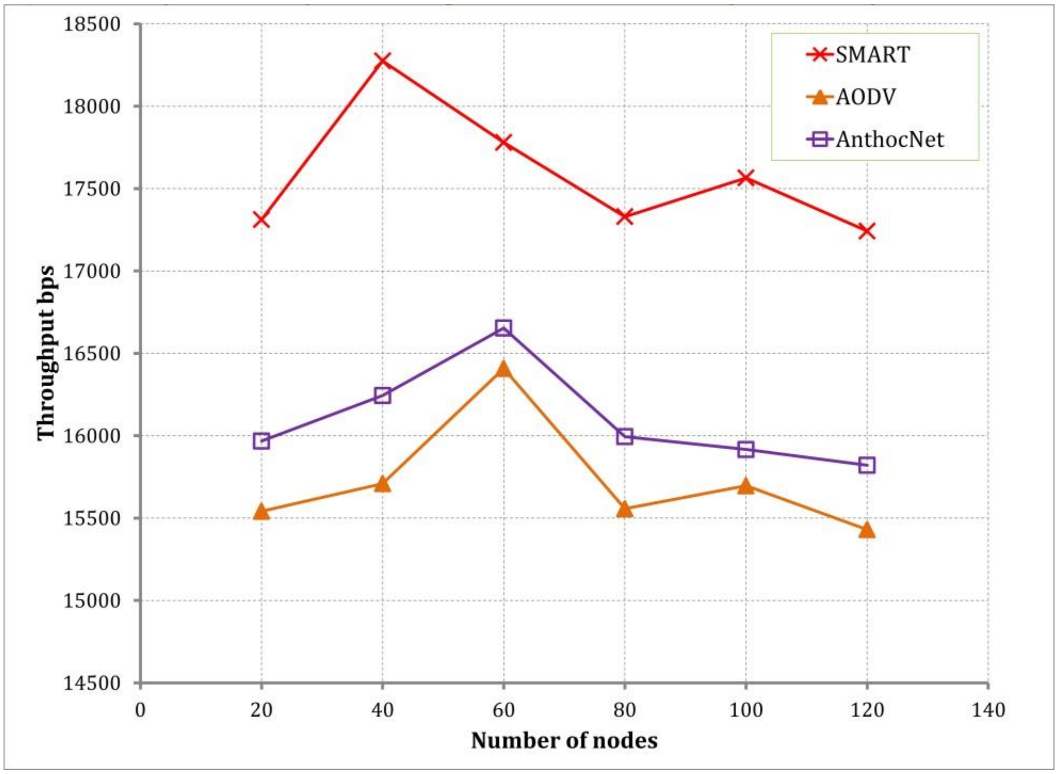


Figure 5.12 Network throughput under various numbers of nodes

Figure 5.13 and Figure 5.14 shows the end to end delay for various numbers of nodes. When there are few nodes in the network, the number of route failure increases. For each route failure, the AODV route recovery requires more time which causes more delay because of broadcasting. AntHocNet has faster recovery procedure [7]; however it also uses broadcasting algorithm whenever a route break occurs. SMART protocol usually do not buffer data packet, unless if the destination is unreachable. As the network become denser, the number of nodes involved in a route may increase in both AODV and AntHocNet. The cost of broadcasting increases as well as the probability of collision increases. SMART protocol is more efficient in searching for better routes, and it minimizes the number of broadcasting in the network. It can be seen that the end to end delay at low node density is high. This occurs because at low node density the probability of the destination being unreachable is high. The delay occurs usually as a consequence of queuing and broadcasting. Putting in mind, in our protocol queuing occurs only at the original source when the destination is unreachable and all the nodes around the original source do not have a route to the destination. In other words all the nodes around the source have an altitude equal to one. This situation is rarely occurs unless the source node is isolated alone or with only few nodes. The reason that reduces this situation is hello message and distributed learning. Whenever a drop is moving through the network it erodes its path, and node surrounding the path will be also eroded by distributed learning. Hello message is also eroding the altitudes of neighbour nodes. However, two methods are participating in increasing the altitudes, the sediment addition, and the punishment process. The ratio of sediments addition is always low. The punishment in our approach is limited to two nodes to decrease network traffic and prevent the loss of learned information. Limiting the punishment procedure to two nodes, described in section 5.3.4, will require many packets to set the altitude of all the nodes around the source to one. Moreover, as explained in section 5.3.3 that hello messages erode altitude of neighbour nodes, if a node with an altitude less than one broadcasts its hello messages to those neighbour that have been punished, it will result in decreasing their altitudes. This will also decrease the probability of bringing the altitudes of all nodes around the source to one. Another factor also contribute in decreasing this probability is when a node joins these group, and it has an altitude lower than one to the desired destination, it will decrease the altitude of its neighbour in the group when it send hello messages or when a data activity occur at that node as all the neighbours will listen to its activity (promiscuous mode updating). Smart data packets as well as drops may be forwarded to that node. If this node joins another group to the original source group and become a link between them, smart data packets and drops will try to find a route to destination through this new group, and in case if there is a valid path to the destination from that node, they will follow that route. Even if the topology of these nodes and links changed and become expire, these smart packets will adapt and search for a link to the destination. After all, this is why we can see that the end to end delay in our protocol is always low, as data packets are always being sent and searching and creating new routes. The probability of queuing as well as broadcasting is low and data packets may be deleted in case if they reach a dead end or when maximum number of hops is reached. Throughout all our simulation, we observed that the maximum number of broadcasting for any source (route set up) in each simulation was always less or equal to three. This indicates that the probability of queuing is very low.

Table 5-3 shows the average number of route request generated by source nodes in AODV protocol compared to SMART protocol. AODV has a local route repair procedure and the average number of route repair at each node in the network is also shown in the table. The results are for scenario one and for random way point mobility. The effectiveness of the RFD algorithm can be seen as drops and smart packets are always moving and searching for a route rather than depending on a recovery mechanism as in AODV or AntHocNet. Whenever a link becomes expire, drops as well as smart packets will follow to new offered paths to search for destination. Similarly, when a flow of water drop is closed, water drops will follow new paths until they reach the sea. In their way to the sea, they will continue their erosion and sedimentation process, i.e. the learning process is never stopping.

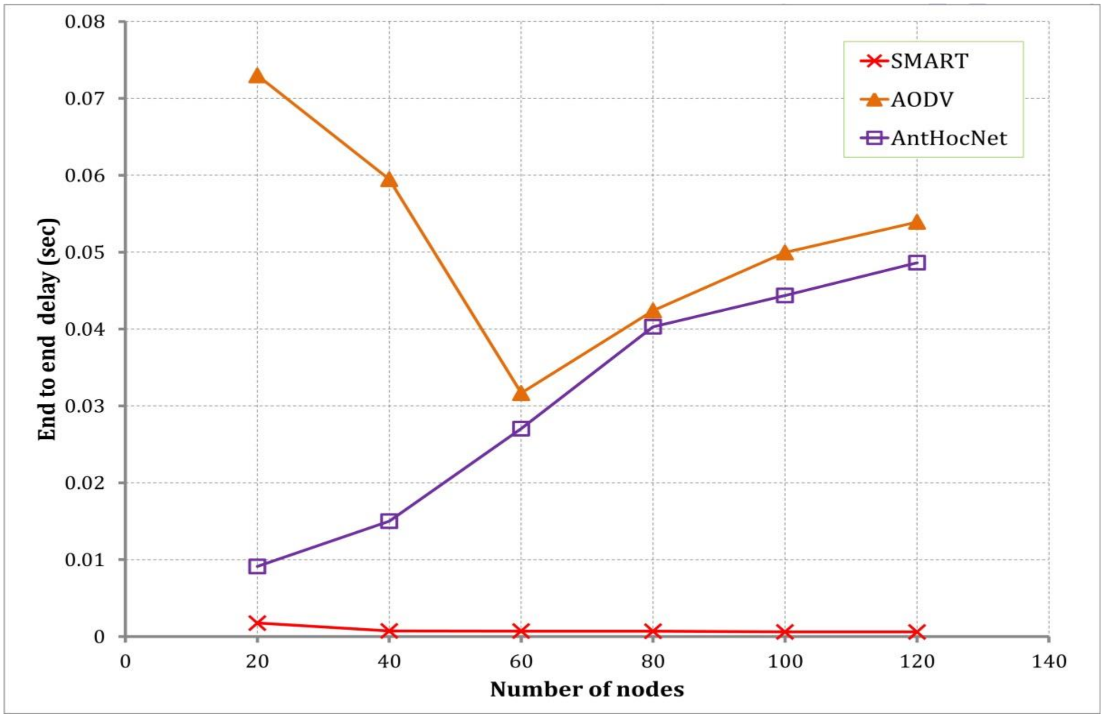


Figure 5.13 End to end delay under various numbers of nodes

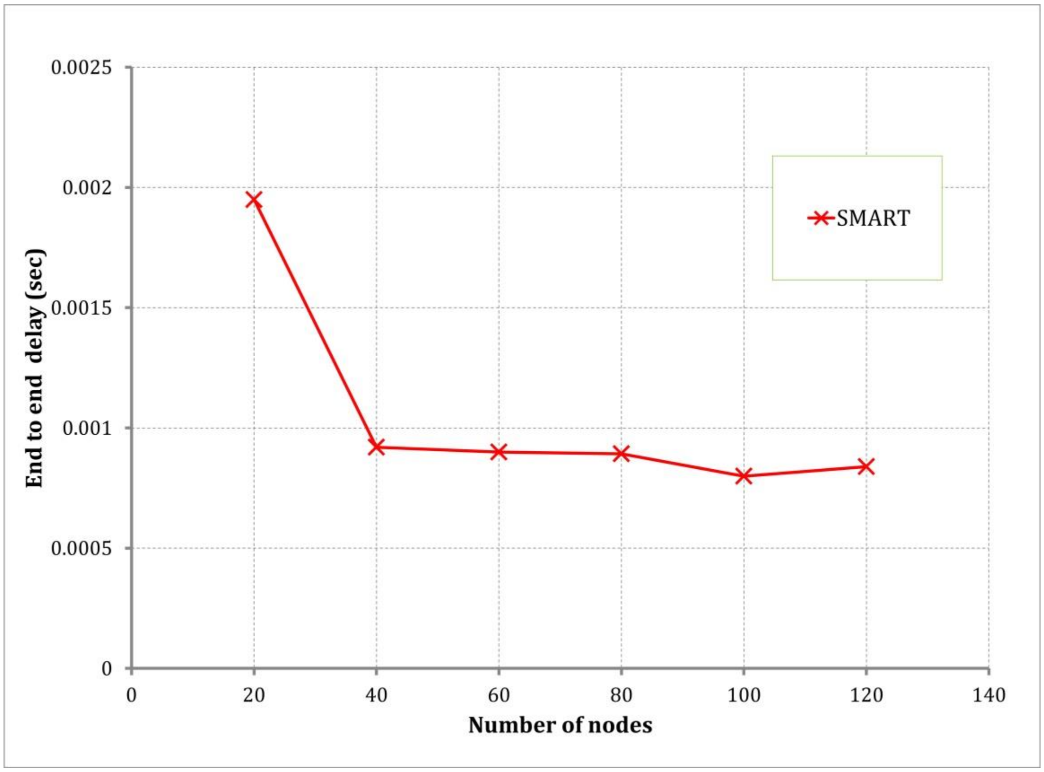


Figure 5.14 SMART protocol end to end delay under various numbers of nodes

Table 5-3 Average number of route request for the first scenario under RWP mobility

	Speed (m/s)				
	10	20	30	40	50
Number of Route discovery for AODV protocol	19.95	26.45	32.24	35.81	39.28
Local route repair for AODV protocol	2.18	2.62	2.48	2.59	2.70
Number of Route discovery for SMART Protocol	1.22	1.24	1.34	1.38	1.46
Number of Route discovery for AntHocNet	12.54	16.38	22.02	26.66	28.56

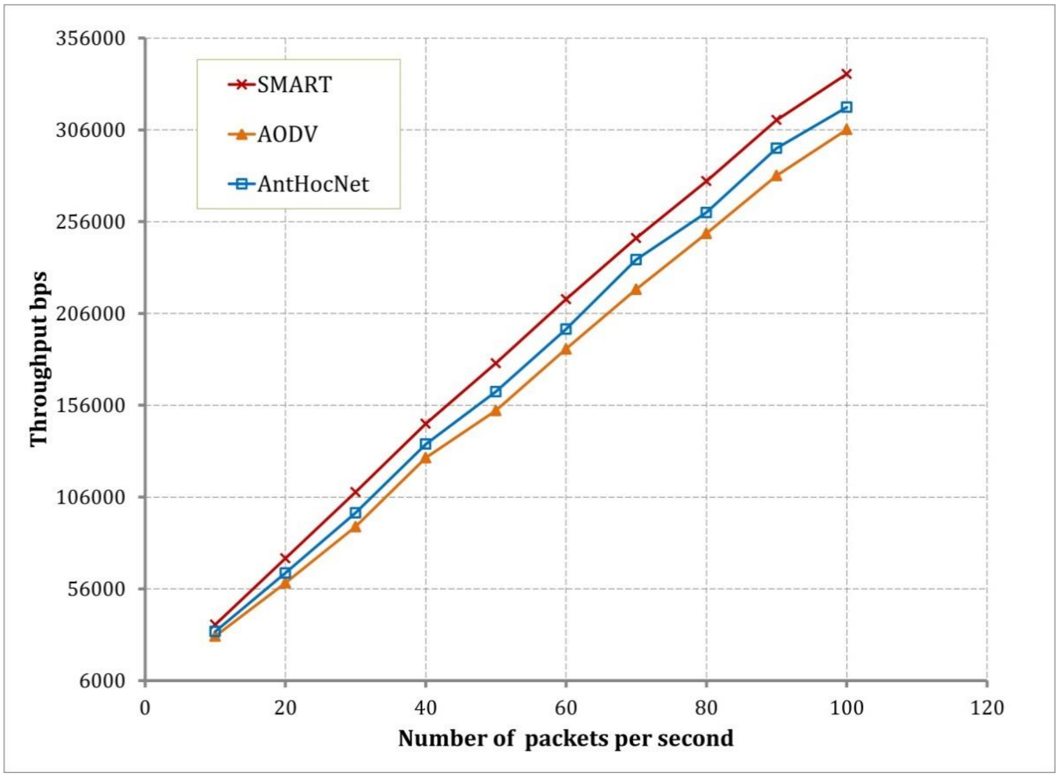


Figure 5.15 Network throughput under various traffic loads.

Figure 5.15 shows the effect of increasing traffic load on the proposed protocol compared to AODV and AntHocNet protocols. The number of packet per second varied from 10 packets to 100 packets per second. It is clear that the throughput of SMART protocol is better than others especially in high traffic load at 100 packets per second. Figure 5.16 shows the end to end delay under variable packet rate. It can be emphasized from this figure that the network is becoming congested when the rate of packets are more than 40 packets per second. The delay of SMART protocol at all rates is less than the others. The negative slope of the AODV end to end delay at the beginning is due to the rate of encountering route failure for low data rate is high or the probability of finding difficulties to build valid route to the destination [10].

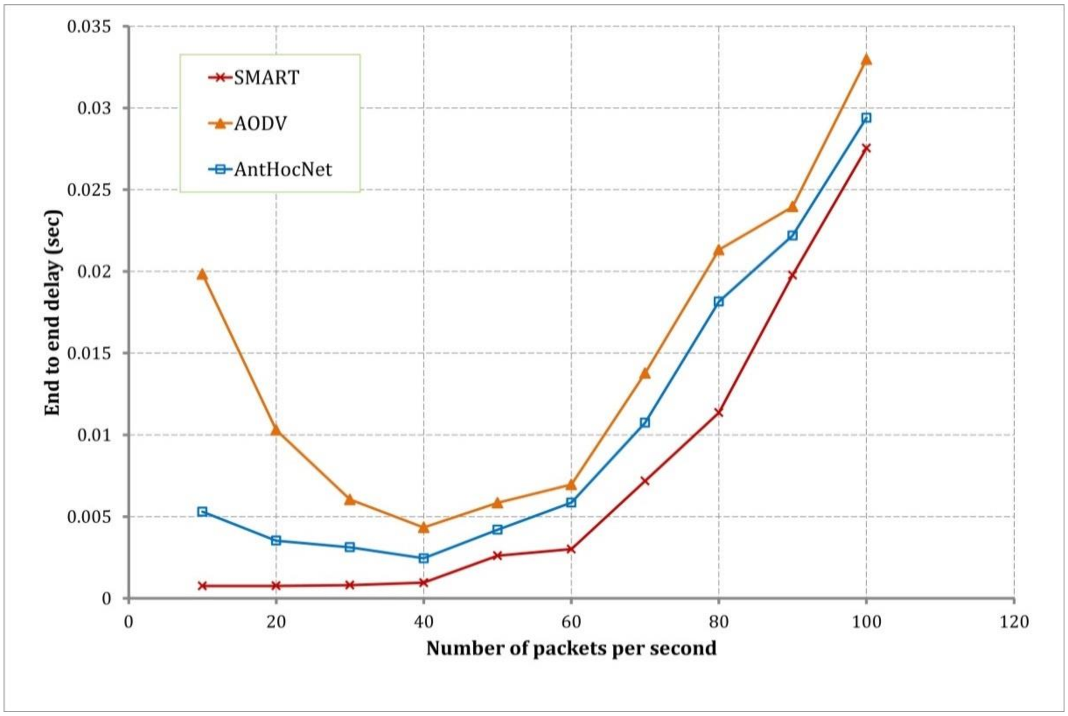


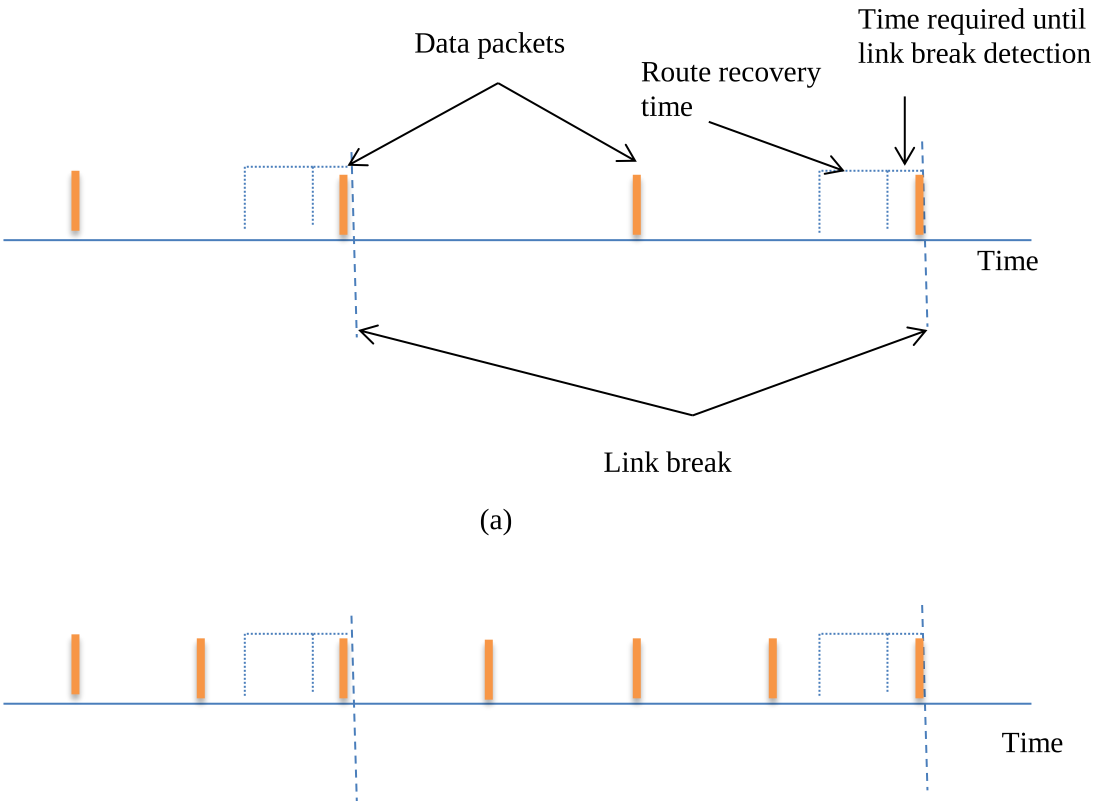
Figure 5.16 Network end to end delay under various traffic loads.

The negative slope of AODV protocol at low data rates requires more explanation. Table 5-4 shows the average number of packets buffered at MAC layer per second. It is clear that the main cause of delay at high data rate is because of buffering at MAC layer. The design of MAC protocol and the detection of link failure at MAC layer caused more packets to be buffered and added delay to the network. At low data rate, the source of delay is due to network layer protocol. The number of packets buffered at MAC layer is very low.

Figure 5.17 shows an example of the reason that at low data rate the delay is high in AODV protocol and decreasing as the data rate increases until certain data rate. In Figure 5.17a, the ratio of packets that suffering from delay to overall number of packets passing through a node is ½ (1 packet delayed per 2 packets). When the data rate increases, as in Figure 5.17b the ratio will decreases and as example become ¼ so the average delay decreases as data rate increases until certain data rate. It should be noted that when a source node is dealing with route recovery the new packets will be buffered and network layer until a route will be found.

Table 5-4 Average number of buffered packets per node

data rate (p/s)	10	20	30	40	50	60	70	80	90	100
AODV	0.105469	0.196387	0.394531	14.28906	39.97314	246.9009	886.0752	1209.461	1344.716	1791.581
SMART	0.098633	0.188477	0.323242	14.35547	25.60547	170.8007	533.3301	743.4785	1274.656	1622.425



(a)

(b)

Figure 5.17 Example showing slow data rate

The SMART protocol rarely has route recovery; however the delay of link breakage detection occurs as it is related to MAC and link layers protocols. The link break detection depends on the retry limit constant set at MAC layer which defines how many times the node will try to send the packet before reporting link breakage. The delay in AODV will be higher as route recovery is based on broadcasting while the SMART protocol will forward the packet to next available node.

At a certain data rate, the traffic in the network will suffer more delay as network become congested and the number of packet that are buffered in both network and MAC layer increases resulting in more end to end delays. As this delay at high data rate is inherited from MAC layer protocol, we preferred to use low data rate to show the delay caused by network protocol.

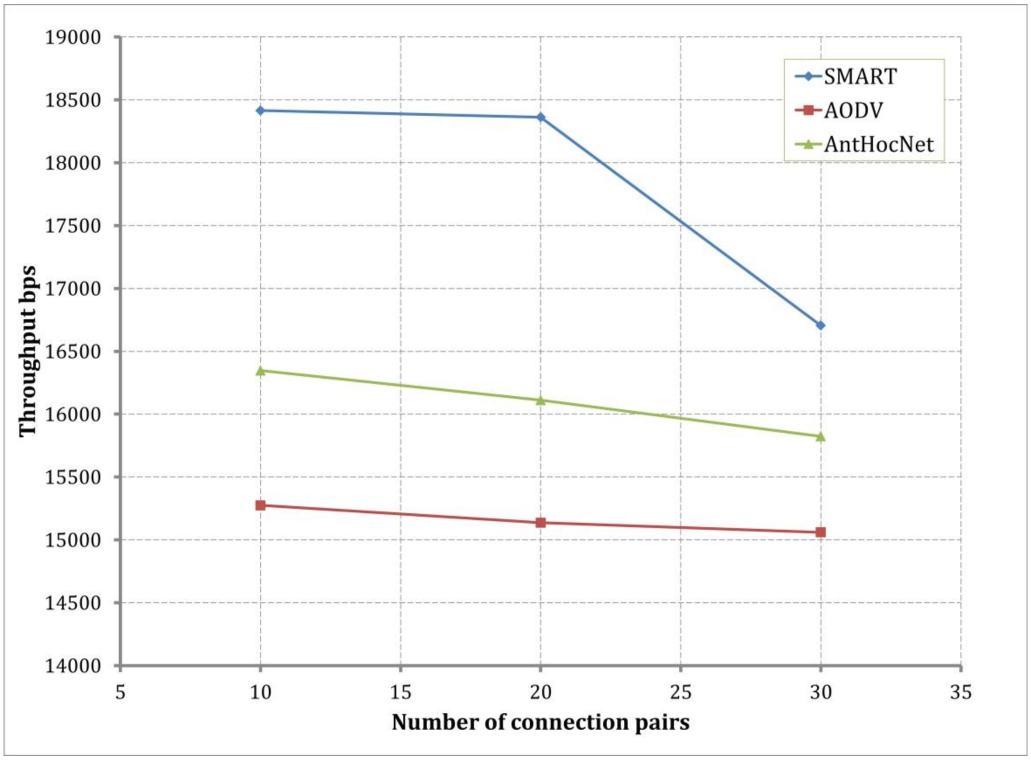


Figure 5.18 Throughput for various number of connection pairs

The effect of different number of connection pairs is shown in Figure 5.18 and Figure 5.19. The number of sources and destinations pairs varied from 10 to 30, where for 10 the number of source nodes is set to 10 and other 10 nodes are set as destinations. For 20 connection pairs, 20 nodes are set as sources and other 20 nodes are set as destinations, some nodes act a sources as well as they act as destinations for other nodes. For 30 connection pairs and as the network consists of 32 nodes, the nodes act as sources and send data to other nodes as well as they act as destinations for other source nodes. In Figure 5.18 the throughput of the network is clearly affected by number of connection pairs. In general, all protocols throughputs decreases as the number of pairs increases due to increase of data traffic which will introduce congestion to the network. The performance of proposed protocol decreases in higher rate when almost all nodes in the network are acting as sources and destinations at the same time. This is due to difficulty in load balancing the network as all nodes are transmitting. The end to end delay also increases as number of connection pairs increases as the network becomes more congested as shown inFigure 5.19. However, the end to end delay of AODV decreases slightly as increasing number of connection pairs will increase the traffic in the network leading to better detecting link breakage but still the protocol introduces high end to end delay.

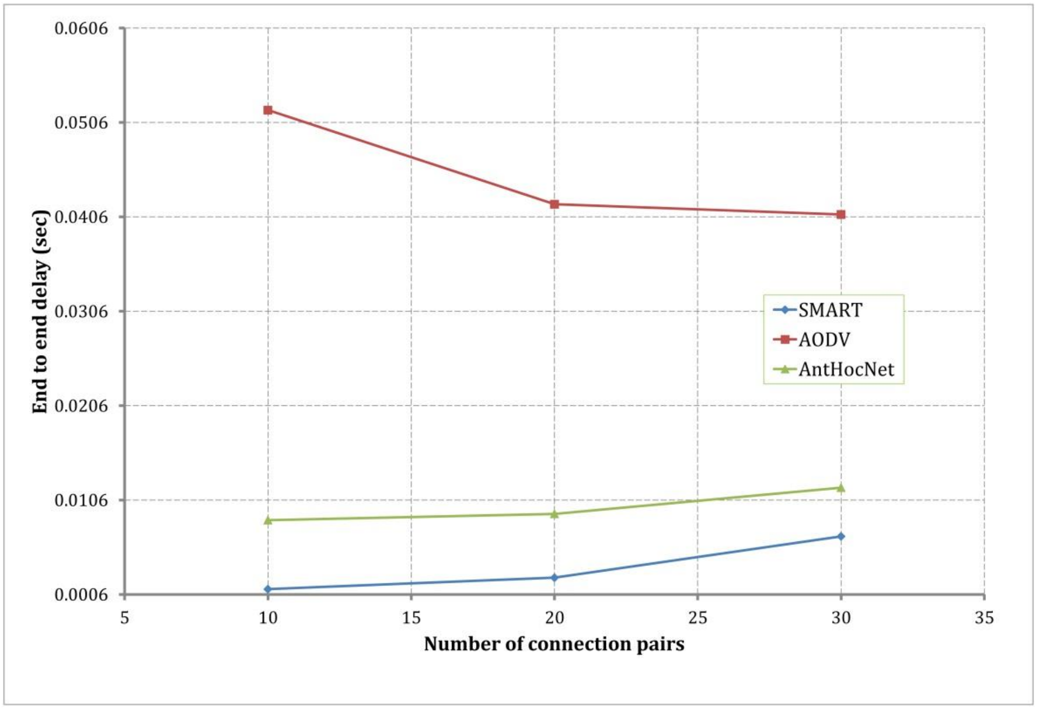


Figure 5.19 End to end delay for various number of connection pairs

Finally the effect of large network size on the protocol performance is tested. Three different sizes have been chosen 600*600 m^2, 1000*1000m^2, and 1400*1400m^2. The node density of these networks is set be close 6.25 as possible so the number of nodes chosen are 32, 89, and 173 nodes. Due to the limitation capabilities of OMNet++ and especially as the protocol is using a lot of vector variables, the largest size of the network is limited to 1400*1400 and 173 nodes to cope with available memory of OMNet++ simulator. The maximum speed of the nodes is 10 m/sec . Increasing the size of the network area will increase the average path length between nodes which will introduce more end to end delay to the network. Moreover, the search space will increase and adds more complexity to the network. The effect of size on throughput is shown in Figure 5.20, the throughputs of all protocol decrease as the size is increased. However, SMART protocol overcomes other protocols and its throughput is always higher than the others.

Considering the effect of network size on the delay, Figure 5.21 shows the results. It could be seen for large network size the end to end delay of AODV increased rapidly as the length of the route increases and the broadcasting will take longer time. Figure 5.22 shows the end to end delay of smart protocol for more clarification.

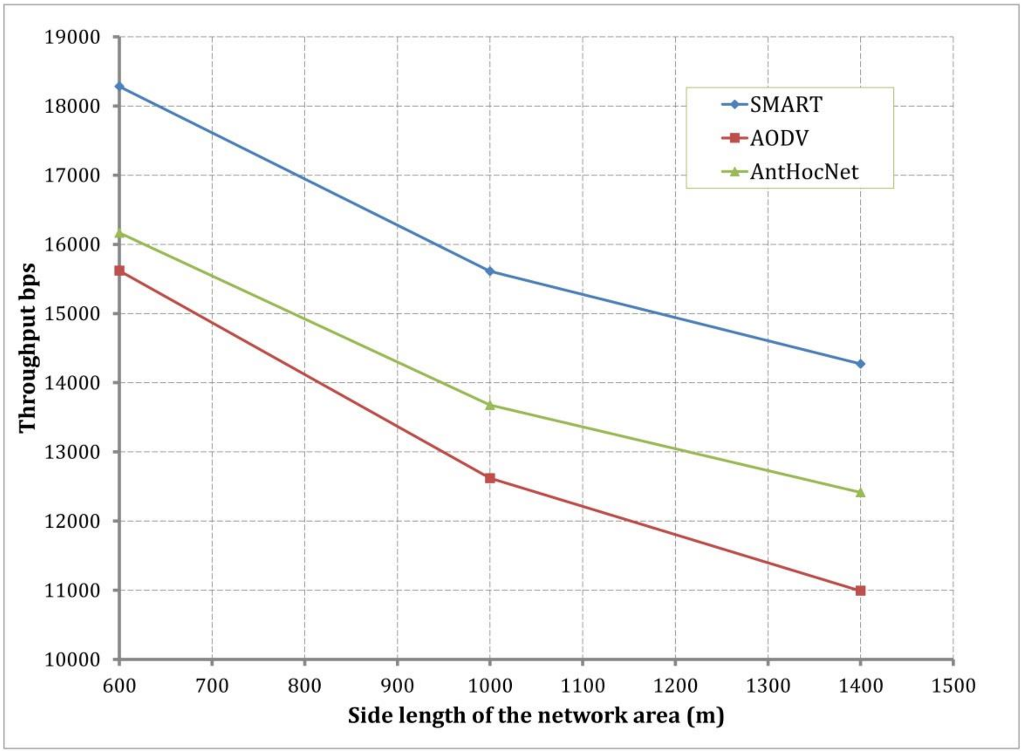


Figure 5.20 Protocols throughput for various network sizes

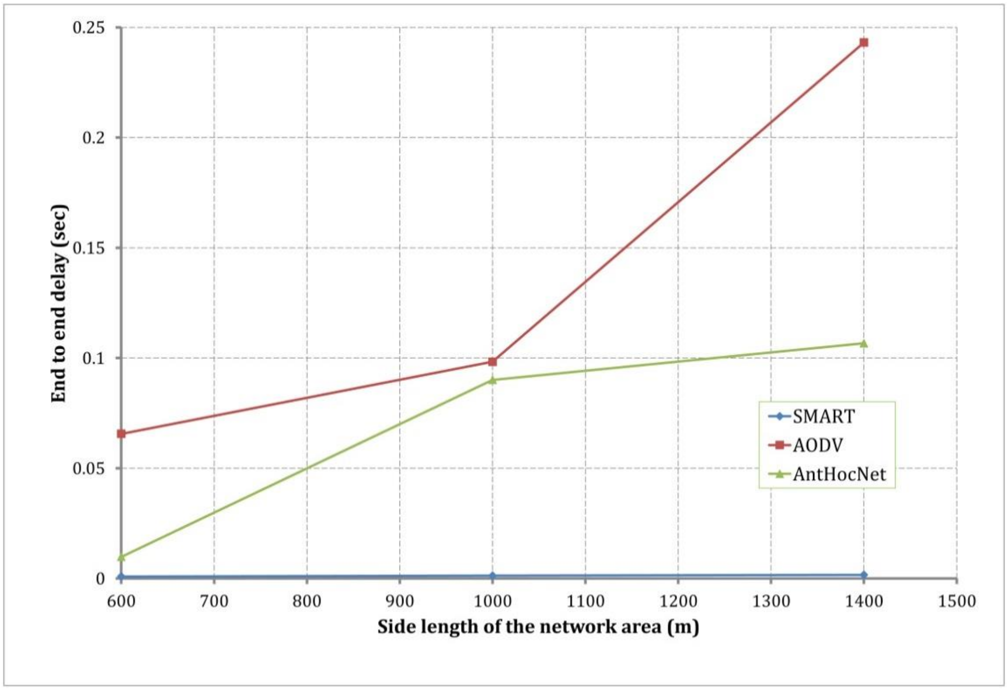


Figure 5.21 Protocols end to end delayfor various network sizes

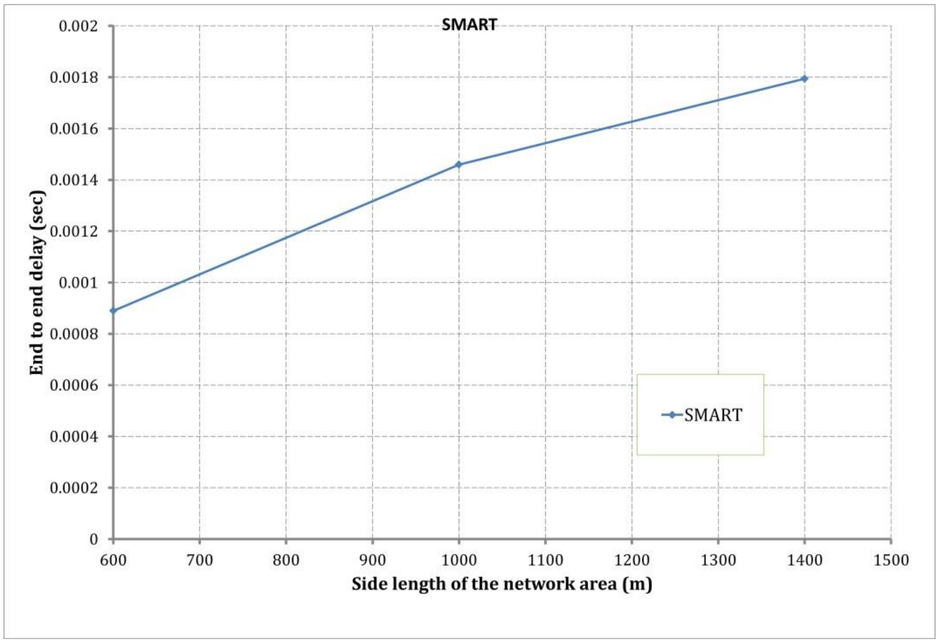


Figure 5.22 SMART protocol end to end delay for various network sizes

5.4.3 Effects of varying the parameters of the algorithm

In the following tests, some of the most important parameters of the algorithm are changed to show the effects of selecting these parameters on the protocol performance. The first scenario has been chosen with minor changes to cope with the test requirements. The speed is set to 10, 30 and 50 m/sec and the test only applied to the SMART protocol as this section intended to test the protocol with different parameters values.

5.4.3.1 Varying the ratio of erosion

The erosion ratio α defined in equation 5.6 controls the ratio of erosion related to altitude difference between two nodes. More details have been given in chapter three section 3.8. Figure 5.23 shows the throughput of the protocol for various values of α for various speeds. For low speeds, low values of speed, low α values produce more throughputs. The nodes are slowly moving and the topology is changing slowly so drops have more time to explore the network and find best path. Giving α small value will smoothly change altitude surface and can lead to better find routes. On the opposite side, when the speed is high, topology changes quickly and drops will not have that time to search for best paths. Higher values will adapt the altitude surface faster. Looking to Figure 5.24, the end to end delay of slow speed is always less than higher speeds. However, the important factor here is the slope of the end to end delay line as it relate the delay to the value of α . It is clear that higher values of α will not give good end to end delay as the altitudes rapidly hanged which will not give good search results. Overall, the changes in the end to end delay are not big.

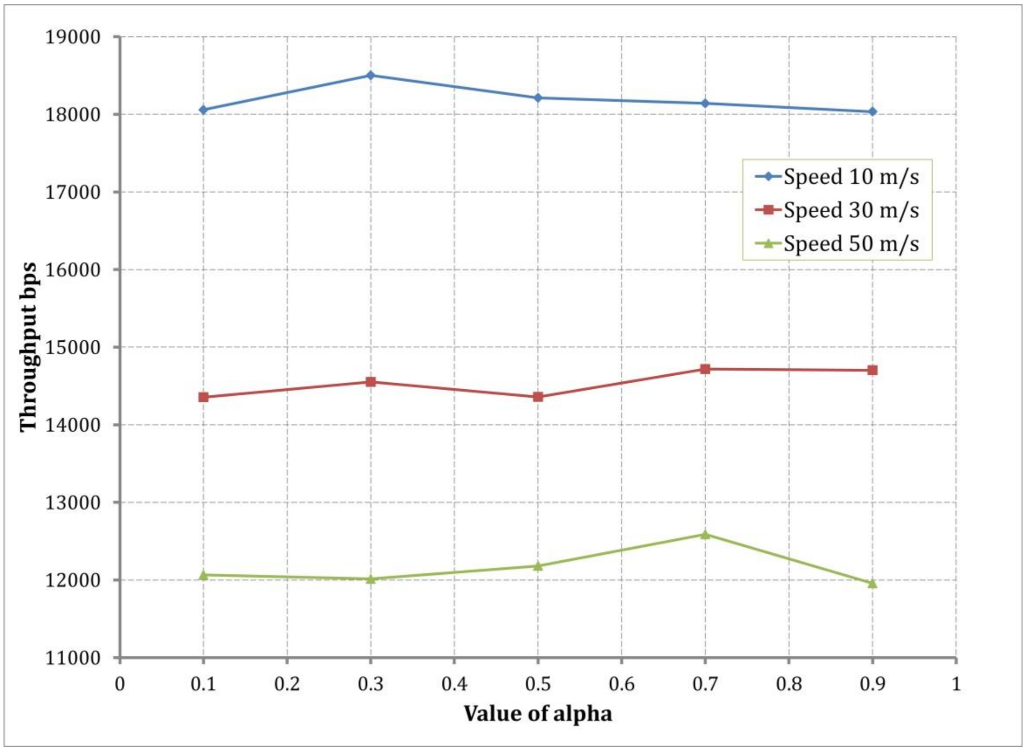


Figure 5.23 Throughput for various values of erosion ratio α

5.4.3.2 Varying the sedimentation ratio

The sedimentation ratio is controlled by variables β and ε in equation 5.8. In the following tests, the values of β and ε are set to equal amount. It is clear from Figure 5.25, the less the values for sedimentation ratio will result in better throughputs for different speed values. As explained in chapter three and in previous discussion higher values will result in a convex topographical structure where the altitudes around the source are less curvature which may lead packet to directions opposite to the direction of the destination.

The delay in Figure 5.26 shows that medium values is best as high value will lead the packet to drop most of carried sediments and the packet will not carry the path information very well.

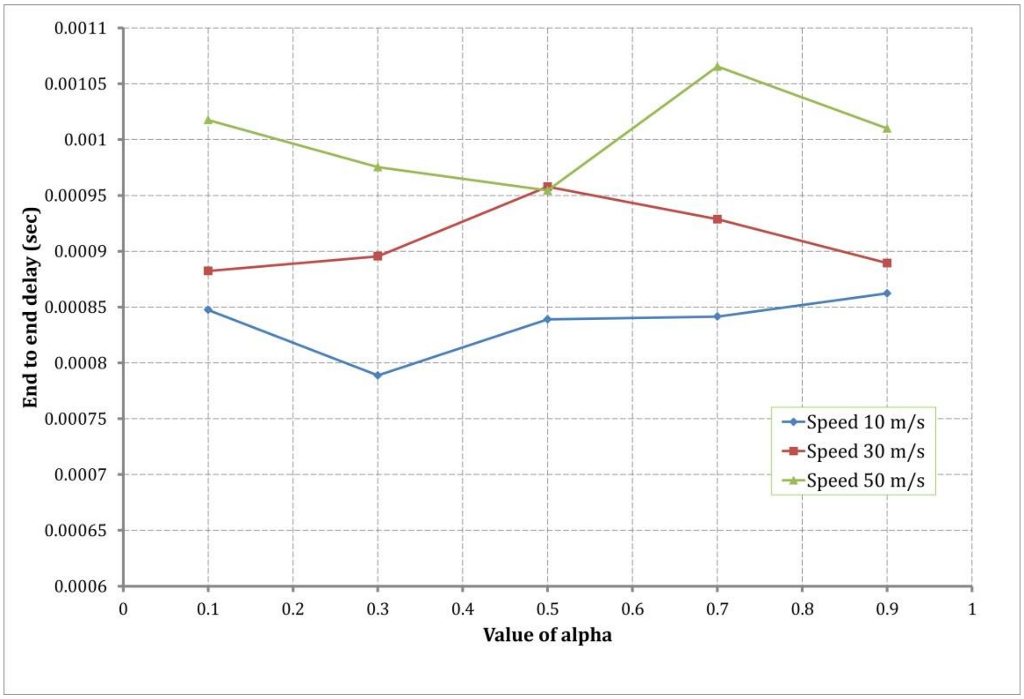


Figure 5.24 End to end delay for various values of erosion ratio α

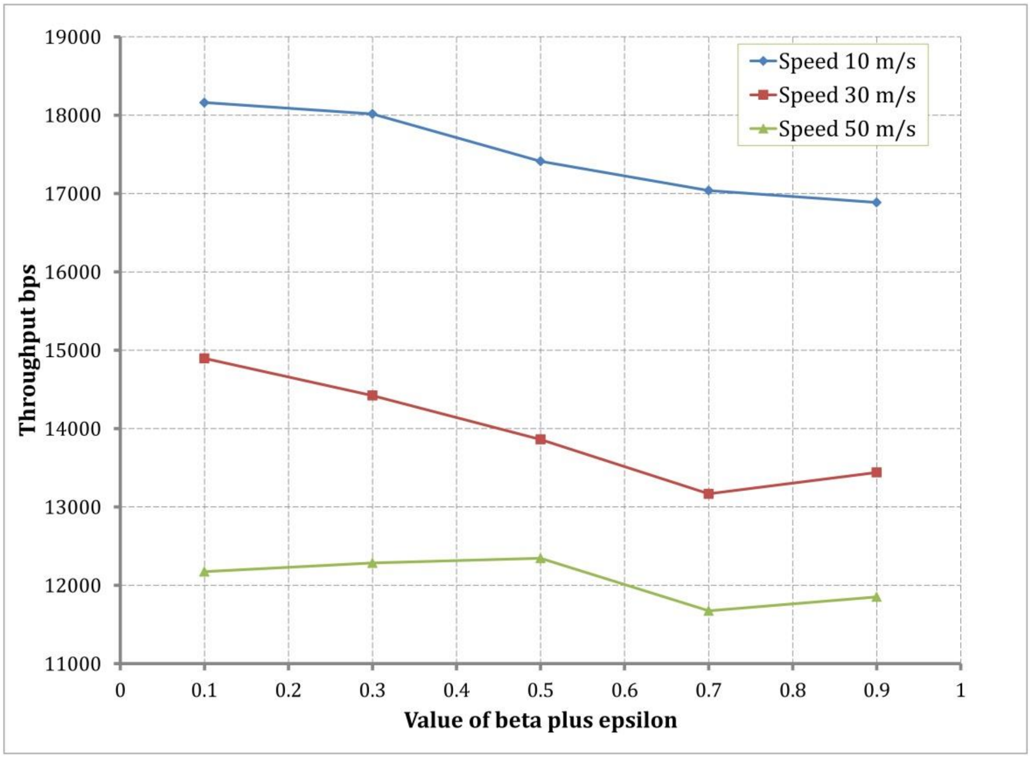


Figure 5.25 Throughput for various values of sedimentation values

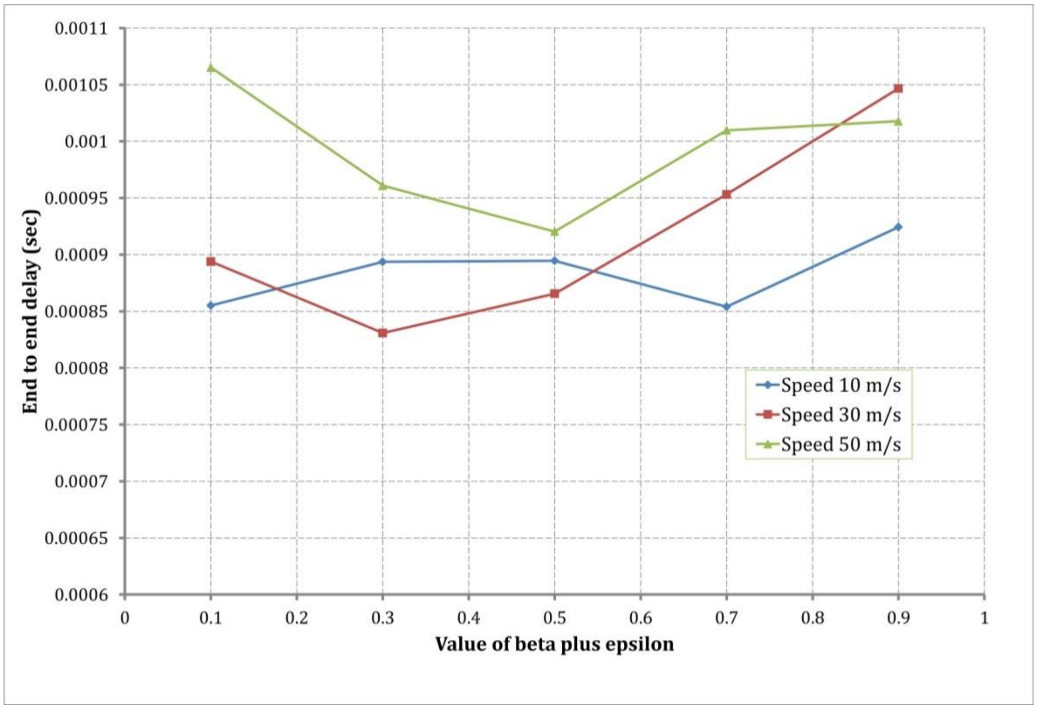


Figure 5.26 End to end delay for various values of sedimentation values

5.4.3.3 Varying forgetting factor value

Sedimentation in the proposed protocol is divided in two processes, on the move and periodical sedimentation. On the move is occurred when a drop is moving toward its destination, discussed in previous section. The periodical one is occurred on regular time bases. This represents the forgetting factor of the protocol denoted θ in equation 5.11. Different values from 0.01 to 0.9 have been chosen to test the effect forgetting factor on the performance of the protocol. Figure 5.27 shows the effect of different values of θ on the throughput of the protocol. Increasing the value of θ will lead to quickly forgetting the learned information and will degrade the performance of the protocol as shown in the figure. For large values of θ the throughput decreases. It is clear also from Figure 5.28 that high values will lead to higher end to end delays. Small values will maintain the learned information especially as there are different sources for sedimentation like the previous sedimentation process and the punishment process.

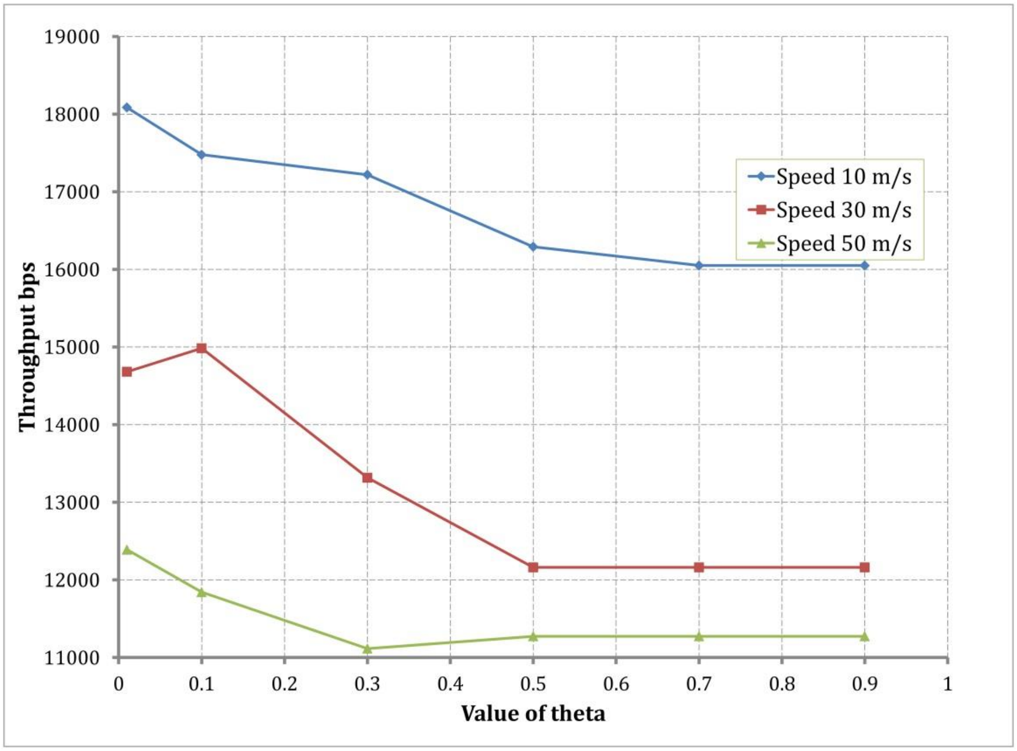


Figure 5.27 Effect of varying theta on throughput

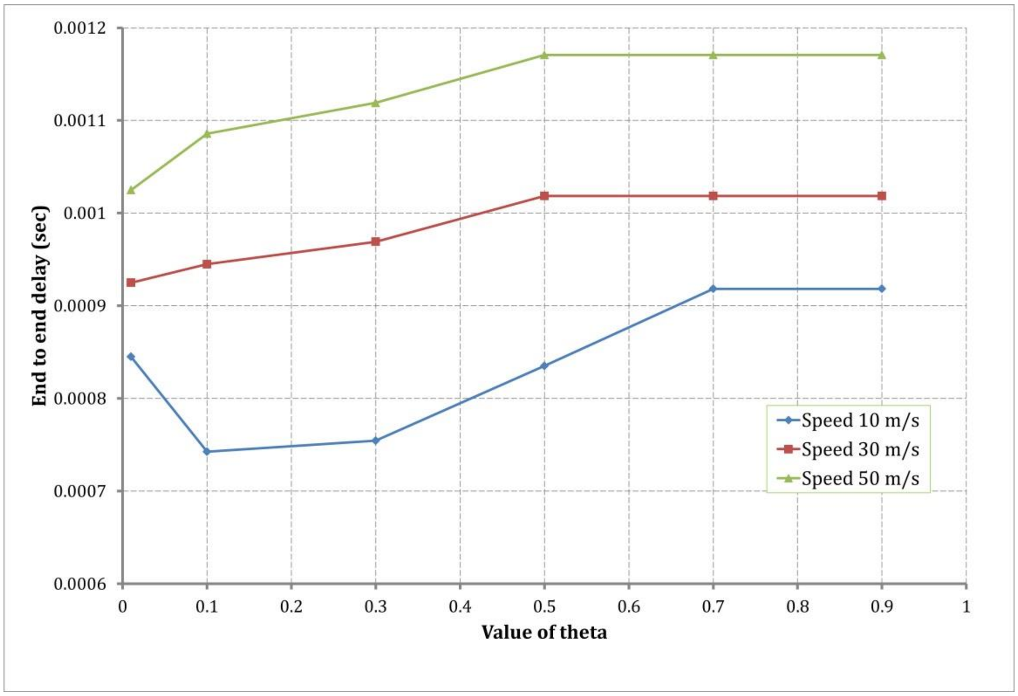


Figure 5.28 Effect of varying theta on end to end delay

5.4.3.4 Varying drop send interval

In this section, the effect of varying the time between the launching of successive drops is tested. Drops are responsible for exploring the network and searching for new paths to a destination. Drops can move far away from best known path. To test the effect of drop rate on network performance, the time interval between drops changed from 0.1 to 2 second. The reflected to starting from 10 drop packets per second to one drop every two second. The throughput in Figure 5.29 shows similar patterns for all speeds. The throughput decreases with the increase of time interval between the drops. The reason behind that is less drops in the network will result in less searching and the protocol cannot find better routes. Moreover data packets are moving in restricted way, they cannot search as drops and they will deepen the found path to the destination. An important factor that differentiates the three patterns is the range of change in throughput. The change in throughput for low speed is higher than the range of change for high speed. Two reasons are participating in these results. First, the throughput itself is high at low speed so the ratio of change will be higher. Second, the movement of nodes at high speeds gives the data packet the ability to search and test paths through other nodes that come closer to the best fount route. The delay in Figure 5.30 shows higher values as the drop rate is high (low interval) this is due to higher number of packets in the network. The delay increases slowly after 1 second interval, this is due to the fact that the selected path is not the best path.

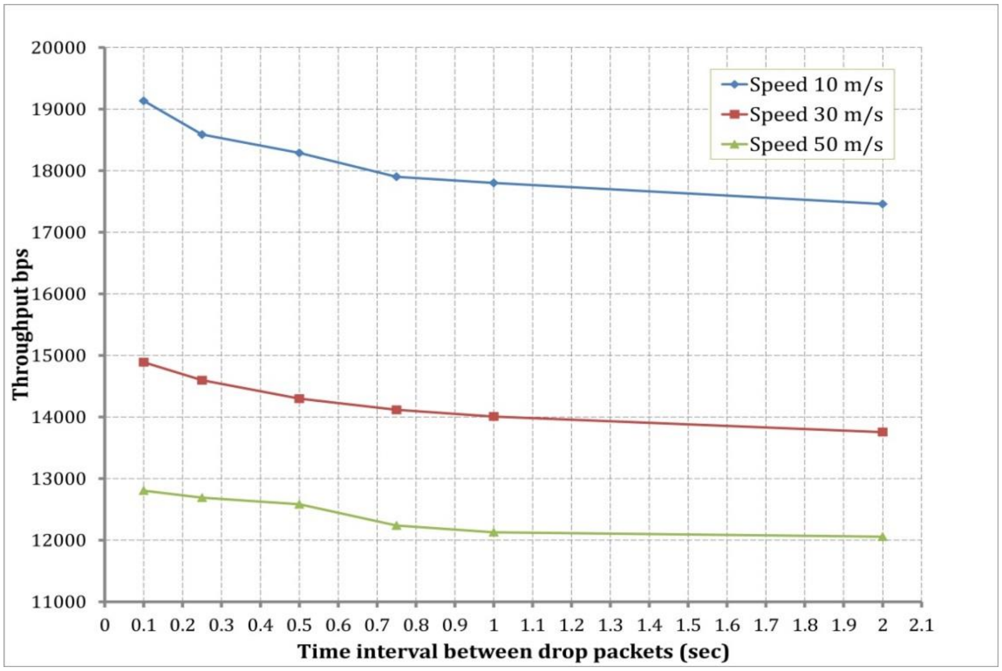


Figure 5.29 Throughput for various drop time intervals

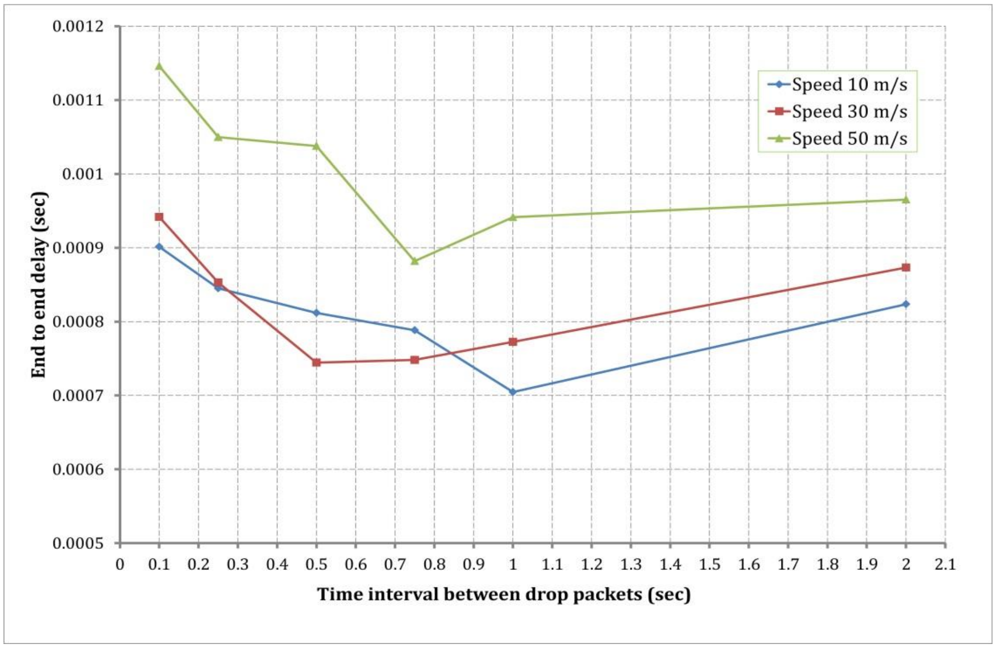


Figure 5.30 End to end delay for various drop time intervals

5.5 Summary

This chapter proposed smart data routing protocol for mobile ad hoc networks based on the RFD algorithm. RFD is a swarm algorithm inspired by the way rain drops make rivers.

The learning in the RFD algorithm is feed forward and eliminates the need for backward packets. This reduces the number of control packets in the network and offers a good opportunity to change and implement smart data packets.

Data packet in the proposed protocol could be ordinary or smart packet. The proposed protocol is flexible and can work on both smart and ordinary data packet. Smart packets carry extra fields in order to contribute in learning process which as result affects the movement of the packets in the network. These extra fields are appended to the end of IP packet header, which adds more compatibility and flexibility to the protocol in order to handle ordinary data packets.

The results show that smart data protocol performs better than AODV and AntHocNet. In average, the throughput is increased and both end to end delay and jitter decreased.

References

[1] P. Lalbakhsh, B. Zaeri, A. Lalbakhsh and M.N. Fesharaki, "'AntNet with Reward-Penalty Reinforcement Learning," *Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on*, 2010, pp. 17-21.

[2] A. Boukerche, B. Turgut, N. Aydin, M.Z. Ahmad, L. Bölöni and D. Turgut, "'Routing protocols in ad hoc networks: A survey," *Computer Networks*, vol. 55, no. 13, 2011, pp. 3032-3080.

[3] Kwang Mong Sim and Weng Hong Sun, "'Ant colony optimization for routing and load-balancing: survey and new directions," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 33, no. 5, 2003, pp. 560-572.

[4] S. Marwaha, J. Indulska and M. Portmann, "'Biologically Inspired Ant-Based Routing in Mobile Ad hoc Networks (MANET): A Survey," *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC '09. Symposia and Workshops on*, 2009, pp. 12-15.

[5] G.S. Sharvani, N.K. Cauvery and T.M. Rangaswamy, "'Different Types of Swarm Intelligence Algorithm for Routing," *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on*, 2009, pp. 604-609.

[6] B. Kalaavathi, S. Madhavi, S. Vijayaragavan and K. Duraiswamy, "'Review of ant based routing protocols for MANET," *Computing, Communication and Networking, 2008. ICCCn 2008. International Conference on*, 2008, pp. 1-9.

[7] Z. Wang and J. Crowcroft, "'Quality-of-service routing for supporting multimedia applications," *Selected Areas in Communications, IEEE Journal on*, vol. 14, no. 7, 1996, pp. 1228-1234.

[8] P. Rabanal and I. Rodriguez, "'Hybridizing River Formation Dynamics and Ant Colony Optimization,"in Advances in Artificial Life. Darwin Meets von Neumann, vol. 5778. George Kampis, et al. , Springer Berlin Heidelberg, 2011, pp.424-431.

[9] P. Rabanal, I. Rodríguez and F. Rubio, "'Using River Formation Dynamics to Design Heuristic Algorithms," *Unconventional Computation Lecture Notes in Computer Science*, vol. 4618, 2007, pp. 163-177.

[10] B. Blanco, F. Liberal and I. Taboada, "'Suitability of ad hoc routing in WNR: Performance evaluation and case studies," *Ad Hoc Networks*, vol. 11, no. 3, 2013, pp. 1165-1177.

[11] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum and L. Viennot, "'Optimized link state routing protocol for ad hoc networks," *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, 2001, pp. 62-68.

[12] C.E. Perkins, P. Bhagwat, C.E. Perkins and P. Bhagwat, "'Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers;" *SIGCOMM Comput.Commun.Rev.*, vol. 24, no. 4, 1994, pp. 234-244.

[13] C.E. Perkins and E.M. Royer, "'Ad-hoc on-demand distance vector routing," *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, 1999, pp. 90-100.

[14] D.B. Johnson, D.A. Maltz and J. Broch, "'DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," *Addison-Wesley*, 2001, pp. 139-172.

[15] L. Guo, L. Zhang, Y. Peng, J. Wu, X. Zhang, W. Hou and J. Zhao, "'Multi-path routing in Spatial Wireless Ad Hoc networks," *Computers & Electrical Engineering*, vol. 38, no. 3, 2012, pp. 473-491.

[16] V.R. Budyal and S.S. Manvi, "'ANFIS and agent based bandwidth and delay aware anycast routing in mobile ad hoc networks," *Journal of Network and Computer Applications*,. http://dx.doi.org/10.1016/j.jnca.2013.06.003.

[17] Hongseok Yoo, Moonjoo Shim, Dongkyun Kim and Kyu Hyung Kim, "'GLOBAL: A Gradient-based routing protocol for load-balancing in large-scale wireless sensor networks with multiple sinks," *Computers and Communications (ISCC), 2010 IEEE Symposium on*, 2010, pp. 556-562.

[18] D. Guo, Y. He and Y. Liu, "'On the Feasibility of Gradient-Based Data-Centric Routing Using Bloom Filters," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 1, 2014, pp. 180-190.

[19] O. Erdene-Ochir, M. Minier, F. Valois and A. Kountouris, "'Toward Resilient Routing in Wireless Sensor Networks: Gradient-Based Routing in Focus," *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, 2010, pp. 478-483.

[20] Y. Koizumi, H. Kanai, H. Ohsaki and M. Imase, "'On the gradient formation incorporating node movement for gradient-based routing in Delay Tolerant Mobile Sensor Networks," *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on*, 2012, pp. 1456-1461.

[21] L. Xia, X. Chen and X. Guan, "'A New Gradient-Based Routing Protocol in Wireless Sensor Networks," *Springer Berlin Heidelberg*, vol. 3605, 2005, pp. 318-325.

[22] M.S. Corson and A. Ephremides, "'A distributed routing algorithm for mobile wireless networks," *Wireless Networks*, vol. 1, no. 1, 1995, pp. 61-81.

[23] V.D. Park and M.S. Corson, "'A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 1997, pp. 1405-1413.

[24] G.D. Caro, F. Ducatelle and L.M. Gambardella, "'AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications*, vol. 16, SEP 2005, pp. 443-455.

[25] E. Gelenbe and R. Lent, "'Power-aware ad hoc cognitive packet networks," *Ad Hoc Networks*, vol. 2, no. 3, 2004, pp. 205-216.

[26] R. Lent, "'Smart packet-based selection of reliable paths in ad hoc networks," *Design of Reliable Communication Networks, 2005. (DRCN 2005). Proceedings.5th International Workshop on*, 2005, pp. 5 pp.

[27] L. CHEN, H. JI, Y. LI and X. LI, "'Multi-path routing based on load-balance for cognitive packet networks," *The Journal of China Universities of Posts and Telecommunications*, vol. 18, no. 5, 2011, pp. 71-75.

[28] M.K. Marina and S.R. Das, "'On-demand multipath distance vector routing in ad hoc networks," *Network Protocols, 2001. Ninth International Conference on,* 2001, pp. 14-23.

[29] B.B. Vasundhara Uchhula, "'Article:Comparison of different Ant Colony Based Routing Algorithms," *IJCA Special Issue on MANETs*, no. 2, 2010, pp. 97-101.

[30] J.S. E, "'Performance Comparison of ACO Algorithms for MANETs," *International Journal of Advanced Research in Computer Engineering & Technology(IJARCET)*, vol. 2, no. 1, 2013, pp. 027-032.

[31] E.M. Royer, P.M. Melliar-Smith and L.E. Moser, "'An analysis of the optimum node density for ad hoc mobile networks," Communications, 2001. ICC 2001. IEEE International Conference on, vol. 3, 2001, pp. 857-861.

[32] C. Bettstetter, "'Smooth is better than sharp: a random mobility model for simulation of wireless networks," *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems,* 2001, pp. 19-27.

[33] S.R. Azzuhri, M. Portmann and Wee Lum Tan, "'Evaluating the performance impact of protocol parameters on ad-hoc network routing protocols," *Telecommunication Networks and Applications Conference (ATNAC), 2012 Australasian*, 2012, pp. 1-6.

[34] C. Gomez, M. Catalan, X. Mantecon, J. Paradells and A. Calveras, "'Evaluating performance of real ad-hoc networks using AODV with hello message mechanism for maintaining local connectivity," *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, vol. 2, 2005, pp. 1327-1331 Vol. 2.

[35] S. Rajagopalan and C. Shen, "'ANSI: a swarm intelligence-based unicast routing protocol for hybrid ad hoc networks," *Journal of Systems Architecture: the EUROMICRO Journal*, vol. 52, no. 8, 2006, pp. 485-504.

[36] Jangeun Jun, P. Peddabachagari and M. Sichitiu, "'Theoretical maximum throughput of IEEE 802.11 and its applications," *Network Computing and Applications, 2003. NCA 2003. Second IEEE International Symposium on*, 2003, pp. 249-256.

[37] Shao-Cheng Wang, Y.-. Chen, Tsern-Huei Lee and A. Helmy, "'Performance evaluations for hybrid IEEE 802.11b and 802.11g wireless networks," *Performance, Computing, and*

Communications Conference, 2005. IPCCC 2005. 24th IEEE International, 2005, pp. 111-118.

[38] P. Raptis, V. Vitsas and K. Paparrizos, "'Packet Delay Metrics for IEEE 802.11 Distributed Coordination Function," *Mob.Netw.Appl.*, vol. 14, no. 6, 2009, pp. 772-781.

Chapter Six

RFDManet: A Swarm Based Routing Protocol Using Intelligent Data Packets

6.1 Introduction

This chapter proposes RFDManet routing protocol for mobile ad hoc network. RFDManet uses the River Formation Dynamic swarm algorithm as a routing protocol. Swarm intelligence has been widely used to solve routing problems [1]. RFD is a subset of swarm intelligence and closely related to ACO [2]. RFD mimics how raindrops try to take the shortest path when they move from highlands to sea. RFD algorithm has been explained in chapter three. RFDManet protocol introduces the approach of using intelligent data packets which directly participate in the learning process. Data packets in RFDManet protocol are intelligent and can route themselves as well as contribute in the learning process. Eventually due to this learning process, the data packets will guide themselves through the best available route. Altitude tables replace routing tables in RFDManet protocol. These tables are updated by drop packets[3], data packets as well as hello packets. The updating process by hello packets could lead to an expired route while it is a crucial process in order to defuse altitude information. RFDManet protocol tries to increase the reliability of data delivery by separating altitude tables into two groups of tables. Information that is propagated by hello message only affects one group of these tables, which are not used for data packets routing.

RFDManet is a hybrid approaches which tries to balance the load throughout the network. This protocol uses two routing tables called reactive and proactive tables. It separates the data packets routing table from control packets the routing table to add more reliability to data routing. At the same time, using two routing tables will address some issues of node mobility. The results show that RFDManet protocol has better performance than AODV and AntHocNet protocols.

[3] Throughout this chapter, drop packet is used to refer to a packet that acts like water drop.

The first and most important advantage of using the RFD algorithm is converting dumb data packets into intelligent packets where they can guide themselves based on required quality of services. Intelligent data packet can learn as they move in the network. They can decide autonomously and instantaneously and select new paths rather than waiting for control packets and route setup procedures.

In general, most of ACO protocols are source routing protocol where they suffer from both bandwidth and hop-count drawback. RFDManet is a distance vector routing protocol and doesn't have such problems. Moreover, RFD uses one kind of messages called drops which move in one direction, from source to destination, while ACO usually uses two types, forward and backward ants. Other advantages of RFD algorithm are explained in chapter five.

6.2 RFDManet Routing Protocol

RFDManet is a hybrid distance vector routing algorithm based on RFD algorithm. It uses two altitude tables, proactive and reactive one, where altitudes of other nodes are stored. It essentially uses hello message to build a proactive route table. On the other hand, drops and data packets are primarily used to update reactive tables. RFDManet protocol is implemented in network layer and uses promiscuous communication mode to monitor neighbours' packets in order to update its routing tables and record other nodes altitude. Data packets are routed using restricted stochastic algorithm. Maintenance drops are sent throughout data session.

6.2.1 RFDManet tables and general structure

Altitudes tables in RFDManet are considered as routing tables. Unlike original the RFD algorithm these tables are updated asynchronously and there is no global access to this information. Nodes can communicate and share information directly with neighbours and indirectly with other nodes that are out of their transmission range. The real distance between nodes is unknown as geographical information is not used in this approach in order to make the protocols more generic and simple.

The route table is represented as altitude table as shown in Table 6-1. This table contains neighbour altitudes to other destinations. Both reactive and proactive altitude tables have the same structure but they are updated differently. Ni in the table represents neighbour node *i*. T_i represents time delay of node *i*. ALT_{ij} is the altitude to destination *j* through node *i* (node *i* altitude to destination *j*). The altitudes in this table are mostly updated when a node hears a message from its neighbors and/or by hello messages.

Table 6-1 Altitude Table

		Destinations				
Neighbours	**Time delay**	**D_1**	**D_2**	**D_3**		**D_k**
N_1	T_1	Alt_{11}	Alt_{12}	Alt_{13}		Alt_{1k}
N_2	T_2	Alt_{21}	Alt_{22}	Alt_{23}		Alt_{2k}
N_3	T_3	Alt_{31}	Alt_{32}	Alt_{33}		Alt_{3k}
N_j	T_j	Alt_{j1}	Alt_{j2}	Alt_{j3}		Alt_{jk}

The node also contains other tables representing its altitudes toward other nodes. These tables are called reactive my_altitude and proactive my_altitude tables. Proactive my_altitude table is updated by drops and hello messages while Reactive my_altitude table is updated by drops and hello messages.

Table 6-2 My_altitudes Table

Destination	**My altitude**
D_1	Alt_1
D_2	Alt_2
D_{k-1}	Alt_{k-1}
D_k	Alt_k

6.2.2 Route discovery

At the beginning of each data session, RFDManet starts by checking its altitude table. If no route to the destination is found, it starts broadcasting drop messages. Broadcast drops update both reactive and proactive tables.

When an intermediate node receives a drop broadcast message, it rebroadcast it if the number of hops is less than previous copies of the message. If there is no copy, it will rebroadcast it. This selective rebroadcasting will reduce the overall number of broadcasting in the network.

To record the path to the original source node the broadcast message carries the altitude of the broadcasting node to the original source node. For the original source its altitude to itself will be zero. The receiver of the broadcast message will update its altitude to the source according to the following

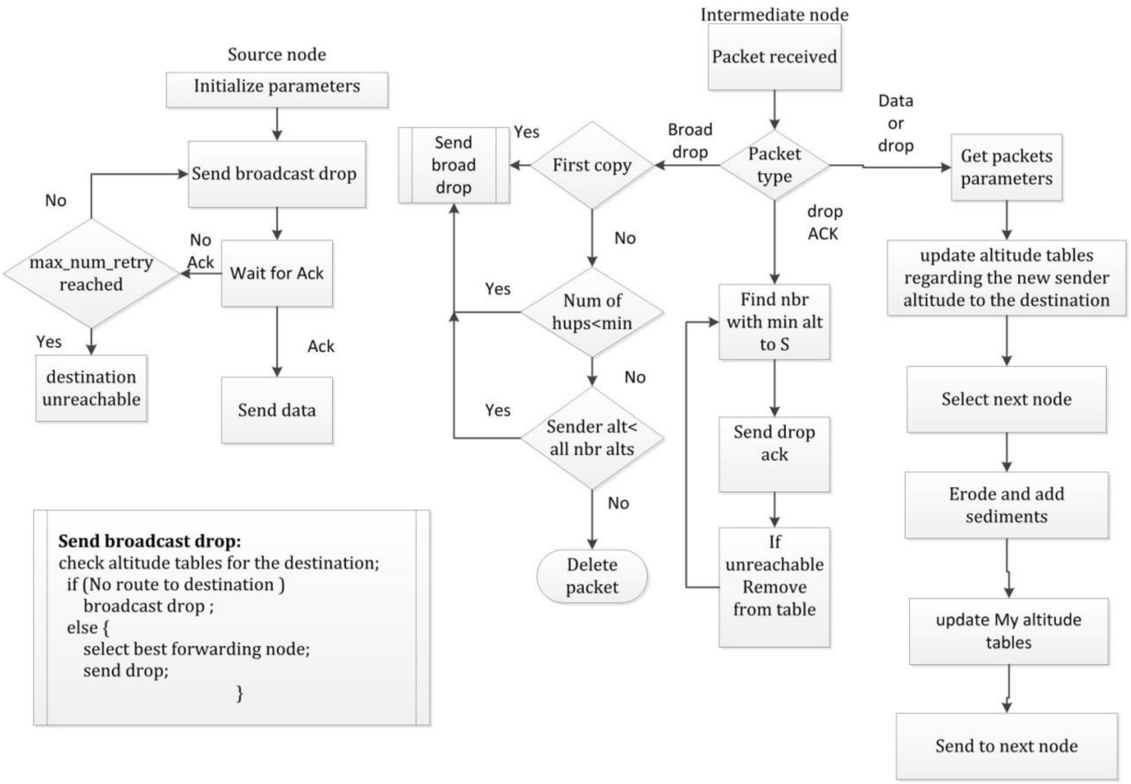
$$altitude_s(j) = altitude_s(j) - \mu_s(altitude_s(j) - Broad_{altitude_s}), \ \ \mu_s \in [0,1] \qquad (6.1)$$

where *altitude$_s$(j)* denote the altitude of receiving node *j* (my_altitude) to the original source *s*. *Broad$_{altitude\ s}$* is the altitude of broadcasting node to the original source *s* in the broadcast message, μ_s is the adaptation constant (value of 0.1 is used). Neighbour altitude is recorded in the *altitude table*. Low value for μ_s is more preferable to prevent overriding old learning information.

It should be noted that when a node receives a message with an altitude higher than its altitude to the source through a specific neighbour, this denotes an expire route. In this case the node will assume it has no route to this source and its altitude will be one and then it will update its altitude according to the above equation. An extra condition for rebroadcasting is when the node receives a neighbour altitude less than its altitudes of all other neighbours to the source. This will assure that best route to the source will be propagated to the destination. Figure 6.1 shows a flowchart for more conception.

When a broadcasted drop reaches the destination, it will send back an acknowledge packet to the source. This packet will crack the surface in a gradient way creating a downslope from the source to destination using the following equation.

$$altitude_d(j) = min_{l \epsilon V}\big(altitude_d(l)\big) - grd * (min_{l \epsilon V_i}\big(altitude_d(l)\big) - altitude_d(i)) \qquad (6.2)$$



```
Send broadcast drop:
check altitude tables for the destination;
 if (No route to destination )
    broadcast drop ;
 else {
    select best forwarding node;
    send drop;
                    }
```

Figure 6.1 RFDManet flowchart

where *altitude$_d$(j)* is altitude of the node *j* (receiver node) toward destination node *d*. *altitude$_d$(i)* is altitude of sender node. *V* is a set of neighbour nodes not including the sender *i*. V_i is a set of neighbour including the sender node *i*. *grd* controls the gradient of the crack should be set between 0 and 1 (set to 0.1). Low value of *grd* will prevent giving high biasing to routes discovered by broadcasting.

The reason behind replying with acknowledgment is in RFDManet algorithm all nodes are set on initialization to the same altitude, i.e. a flat surface at the beginning of data session. Sending data in such a flat surface will spread the packets everywhere through the network and may not reach the destination. The solution is to set up a small crack on the surface between the source node and the destination node by the acknowledgment packet. This crack will guide the data packet at the beginning of the route setup phase and minimize route setup time.

The path that acknowledgment packet takes in its return to the source is depending on attitudes to the source created throughout the broadcast process and previous information in the nodes if available. The broadcast change nodes altitude levels to a funnel structure, where the source is located at the bottom end of that funnel. Intermediate nodes will forward the acknowledgment packet to a neighbour with minimum altitude regarding the source. If due to mobility the next node is not available and a link break occurred, the node will forward the acknowledgment packet to the second best neighbour with minimum altitude toward the source.

After the first broadcasting, the source node will wait for a time period equal to *network traversal time*. If no acknowledgment received throughout this period, the source node may send another broadcast message until the *maximum number of retries* is reached. When the source node receives the acknowledgment packet, it starts sending data packets.

6.2.3 Route maintenance

RFDManet uses drop messages to maintain and find better routes throughout data session. The rate of these drops could be proportional to data rate or constant (constant rate=0.5 sec is used). The drops are propagated according to the gradient probability function

$$P_m(j,k) = \begin{cases} \frac{decreasingGradient(j,k)}{\sum_{l \in V_m(j)} decreasingGradient(j,l)} & if\ k \in\ V_m(j) \\ 0 & if\ k \notin\ V_m(j) \end{cases} \quad (6.3)$$

where $\mathrm{P_m}(\mathrm{j,k})$ is the probability of drop m at node j to select node k. Vm is a set of neighbour nodes that can be visited by the drop based on proactive table. *decreasingGradient(j,k)* represents the negative gradient between nodes j and k, which is calculated as follows:

$$decreasingGradient_d(j,k) = \frac{altitude_d(j) - altitude_d(k)}{T(k)} \quad (6.4)$$

where *T(k)* is the average time that node *k* needs to send a packet, *altitude$_d$(j)* is altitude of neighbour node *j* toward destination *d*. As stated before, the exact distance between nodes is unknown. Time delay parameter has been used where it detects the congested node and used to represent the distance. The more congested the node, the more it is not preferred to be as a forward node and though it had higher distance (*T(k)*).

Each node monitors its sent packets. When it starts sending a packet it computes how much time it needs for the forwarded node to send it again. It keeps tracking of this, in both the reactive and proactive altitudes table, by computing the running average of the time needed by its neighbour to forward its packets.

$$T_k(t) = \gamma T_k(t-1) + (1-\gamma)C_k(t), \quad \gamma \in [0,1] \tag{6.5}$$

where $C_k(t)$ is new delay in node k, T_k is time delay for node k. T initially is set to be equal to the time required by an unloaded node to send a packet. γ is a parameter regulating how quickly the formula adapts to new information (set to 0.7). Using this type of distance will help in selecting uncongested node.

In order to give drops the ability to explore more paths and search for better solutions, another method inspired from nature is used. When a drop of water falls from high altitude and hit a surface, it will bounce up from the surface (or some time an equal amount of water). The amount that the drop will bounce off is proportional with altitude difference. This is set to 5% of the difference. This gives the node the ability to climb little elevation and prevents local loops as a drop will never follow the same path twice, without any extra technique to prevent local loops and cycles. This method is better than the one introduced by [3, 4] as depending on probability may lead to some cycle loops as the author declared.

When drops move in the network they erode the altitudes of the nodes in both reactive and proactive tables. The amount of erosion is proportional to the gradient. After the next node has been selected according to above equations, the altitude of the selected forward node altitude from the table is used to calculate the gradient. The amount of eroding is calculated according to the following equation

$$erosion_d(j) = \propto (altitude_d(j) - altitude_d(k)) \tag{6.6}$$

$$altitude_d(j) = altitude_d(j) - erosion_d(j) \tag{6.7}$$

where $erosion_d(j)$ is the amount of erosion at the node j toward destination d, $altitude_d(j)$ is the altitude of node j, k is the next selected node. α is a positive constant number between 0 and 1 (set to 0.7) which controls the convergence of the algorithm. Due to nodes mobility and dynamic infrastructure of mobile ad hoc network, low values for α are not preferred.

At the same time drops add sediment to node altitude. The amount of sediment is proportional to the amount of sediment carried by the drop as well as the flatness of the surface between the node and the next receiving node.

$$sediment_d(j) = (\beta + \varepsilon * (1 - (altitude_d(j) - \ altitude_d(k)))) * carried\_sediment, \ \ \beta + \varepsilon \in [0,1] \qquad (6.8)$$

$$altitude_d(j) = altitude_d(j) + sediment_d(j) \qquad (6.9)$$

altitude$_d$(j) is the altitude of node *j* where the drop is being processed. k is the next selected node. β and ε are constants controlling the amount of sediments deposit (set to 0.1 and 0.1 respectively). Low values will give the drops more searching opportunity around the destination.

The sediment carried by the drop to next node is

$$carried_{sediment} = carried_{sediment} + erosion_d(j) - sediment_d(j) \qquad (6.10)$$

Another type of sediment adding occur regularly every fixed amount of time

$$altitude_{d\in w}(j) = altitude_{d\in w}(j) + \theta, \qquad \theta \in [0,1] \qquad (6.11)$$

where *w* is a set of all known destinations. θ is the amount of sediment to be add(set to 0.01). θ acts as forgetting factor. The time period between regular additions is set to 1 second.

Finally, the destination altitude is always zero.

6.2.4 Hello message

In RFDManet Hello message is used to discover neighbour nodes as well as to propagate information around the network. Hello message is extended to carry K elements from its my_altitude table (k=10 is used). Using this idea, the altitudes of the nodes will be distributed around the network. Hello messages only update the proactive altitude tables. It should be noted that carried altitudes in hello messages can be selected randomly from proactive or reactive altitudes tables using uniform random generator.

When a neighbor node receives hello message it will contain a list of destinations and their altitudes. The node will update the altitude toward those destinations over that neighbour in altitude table with the new ones. At the same time node altitude to that destination (my_altitude) will be changed in a factor proportional to the difference between node altitude and received altitude.

$$altitude_d(j) = \ altitude_d(j) - \ \mu(altitude_d(j) - hello_{altitude_d}), \qquad \mu \in [0,1] \qquad (6.12)$$

$hello_{altitude\ d}$ is the altitude of destination d in hello message, μ is the adaptation constant (set to 0.1).

The reliability of information carried by hello message is not high for many reasons. First, this information does not address the congestion in the nodes. Second, since hello messages are sent every hello interval period consequently this information may be out of date. Using two tables to separate proactive tables, information that hello message contribute in it, and reactive tables will add more reliability to routes that guide data packets.

6.2.5 Data routing

In RFDManet, the learning process is using drops like agents (drop and data packets). Drops move from sources to destinations. There is no need for backward messages or agents like backward ants in ACO based routing protocols. This gives the opportunity to use data packets as extra agents that act like drops and contribute in the learning process. These data packets are acting like intelligent packets as they will cooperate directly in the learning process. There is no need for any type of other packets to work with these data packet in order to perform the learning process. As a result of this learning process, data packets will guide themselves through better paths.

Some other fields should be added to the data packets in order to make the data packets more intelligent. It is significant to keep the added information as small as possible and not to overload the data with many extra bytes. Three parameters have been added to the data packet. The node altitude, carried sediment and the sender address are added. Each of them is four bytes in size. Figure 6.2 shows an example of UDP packet format for intelligent data packet.

Although all data packets in this proposed protocol were set to cooperate in the learning process, the protocol can handle both normal data packets and intelligent data packets. The protocol is flexible and can handle both types of data packets. The effective ratio of intelligent data packets to normal packets will be addressed in future works.

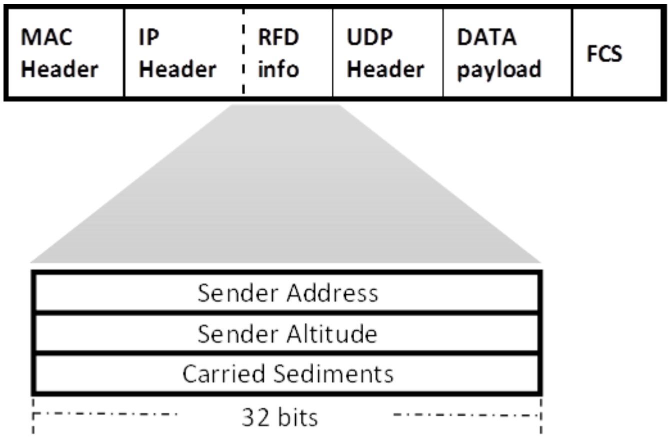


Figure 6.2 Packet format for intelligent UDP data packet where the extra RFD information is appended to the IP header

Like SMART protocol (explained in chapter five), data packets are routed using greedy and restricted method. Data packets use reactive altitude table and move according to the following random probability function.

$$P_n(j,k)_d = \begin{cases} \frac{decreasingGradient_d(j,k)^{\partial}}{\sum_{l \in V_n(j)} decreasingGradient_d(j,l)^{\partial}} & if\ k \in\ V_n(j) \\ 0 & if\ k \notin\ V_n(j) \end{cases} \qquad (6.13)$$

where ∂ is a constant number greater than one, and is set to 4 to achieve the greedy movement. *$V_n(j)$* is set of neighbour node of node *j*.

The ratio of data packets to drop packets is high, which could ruin learned information. This reflects flooding in nature. When it rains repeatedly there is increasing tendency for the water to follow along one path way, and this progressively deepens of the flows will artificially increase the significant of some categories over others. This effect has great influence on the route when the nodes are moving slowly. The data packets will move almost through same restricted area from the source to destination. While at high speed, the topology is changing faster and the likelihood of eroding same area is decreased and the learning process is enhanced. To limit the erosion that is occurring due to this high number of data packet, the percentage of erosion and sediment deposit by data packets is reduced.

6.2.6 Route failure

As with SMART routing protocol (more details in chapter five), there are two ways that RFDManet detects link failures. The first one is by detecting the loss of hello message from a neighbour. The second way of detecting link failure is through missing acknowledgment.

When a link failure is detected the node takes number of actions. First it deactivates the route in route altitude table. The second action after detecting link failure is updating node destination attitude table. If the loss of a neighbour affects the table then it will broadcast a notification to its neighbour.

When a link failure occurs in data packet transmission, the node will try to send the packet to the next best neighbour. The same procedure of SMART protocol is used. The only exception is the reactive table are used in forwarding data packets in RFDManet protocol.

6.3 Implementation and Simulation Results

In this section, we discuss the performance of RFDManet protocol, the performance was evaluated by comparing the simulation results of the proposed algorithm with those of AntHocNet and the standard AODV protocols.

All of the simulation results are generated using OMNet++ as simulation software. A model for AntHocNet protocol was implemented based on [5]. As for the AODV protocol, the INETMANET extension package of the OMNet++ is used.

6.3.1 Simulation environments

The node density in our experiment is set to about 6.25 in order have a good connectivity [6] , as well as providing a good probability for a node to have multiple paths to its destination. The first scenario consists of 32 nodes distributed randomly in 600 *600 m^2 area. Simulation time was set to 200 seconds and repeated for twenty times with different seeds. Five mobile nodes selected randomly to act as sources and five other nodes acts as receivers. Each node had a radio propagation range of 150m and channel capacity of 54 Mb/s. Hello time intervals is set to 0.5 second. Pause time is set randomly between 0.1 and 1 second.

In the second scenario, the number of nodes increased to 64 and to keep node density almost equal to the first test. The area is increased to 850*850 m^2. Simulation time was set to 400 seconds. The simulations were repeated for twenty times with different seeds. The number of sources is increased to ten and the number of destination nodes is increased to ten as well.

The medium access control protocol was the IEEE 802.11 DCF. Packet size was 512 bytes. Each node generates a packet for every 0.2 second. The network remains silent in the first second. The nodes start sending data at third second and keep sending until the end of simulation, which gives one second for some hello packet to be generated. Data traffic was generated using constant bit rate (CBR) UDP traffic sources. Random waypoint (RWP) and Gauss Markov (GM) mobility model have been used [7] .

All results are averaged over the number of source-destination pairs [5], and all graphs are showing 95% confidence intervals of the measured values.

The following end to end network characteristic has been studied and chosen based on [5, 8], and we show them using graphs showing 95% confidence intervals of the measured values.

1- Throughput: is the measure of the total number of successful delivered data bits over simulations time for a specific node, averaged over the number of source-destination pairs.

2- End to end delay: is the measure of average delay of data packets. This is the time from sending the packet from the application layer at the source node to the time that the packet arrives to the application layer at the destination node, averaged over the number of source-destination pairs.

3- Jitter: is the variation of packet delay which is averaged over the number of source-destination pairs, averaged over the number of source-destination pairs.

4- Routing overhead is the total number of control packets sent divided by the number of data packets delivered successfully.

6.3.2 Results

Four factors have been tested to compare the performance of the proposed protocol with other two protocols. The throughputs of the proposed protocol compared with AODV and AntHocNet is shown Figure 6.3 Figure 6.4. It is clearly obvious that RFDManet protocol outperforms both AODV and AntHocNet. Figure 6.3 shows that RFDManet throughput is increased about 10.8% over AODV protocol at low speeds up to 19.9% at high speeds under RWP mobility. As stated earlier, in low speeds data packet enforce the same path which affect the route selection. RFDManet throughput is about 7% higher than AntHocNet when the speed is 10 m/s. This ratio is increases to 10.6 at 50m/s. The throughput under GM mobility is less than RWP mobility mostly because in RWP mobility the nodes tend to move toward the centre of the simulation area, which will lead to fluctuation in node density.

Figure 6.4 shows the throughput for 64 node network under RWP mobility. RFDManet throughput is more than 22% compared with AODV. The ratio of throughput increase over AntHocNet varies from minimum 3% at 20m/s, and maximum 20% increase at 40m/s. Comparing with Figure 6.3,where both networks have equal node densities but different sizes, we can observe the effect of network size on the performance of the protocols. As the simulation area become larger, the lengths of the path lengths are growing, where many nodes involved in the routes between the sources and destinations. The probability of link failure and disconnection are in greater long routes. The repairing processes in shorter routes are faster. We can see that all the protocols have been affected by the network size. The throughput in this scenario is less than the previous one.

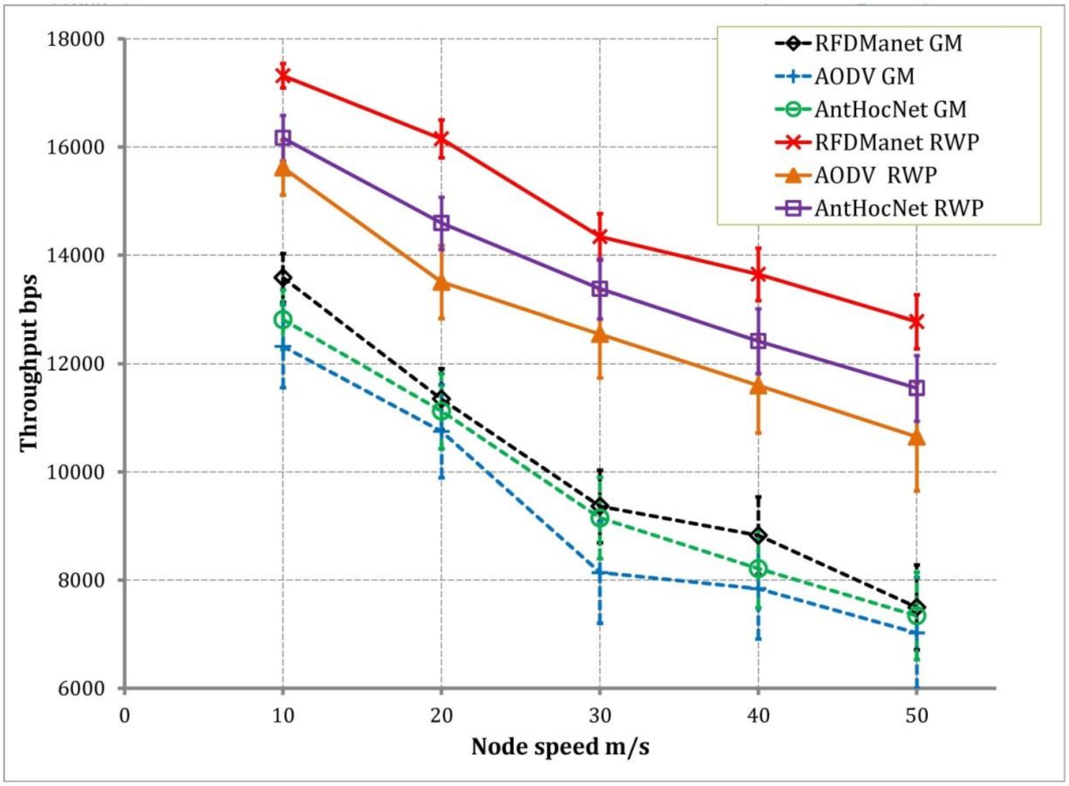


Figure 6.3 Average throughput for 32 nodes network under various speed values.

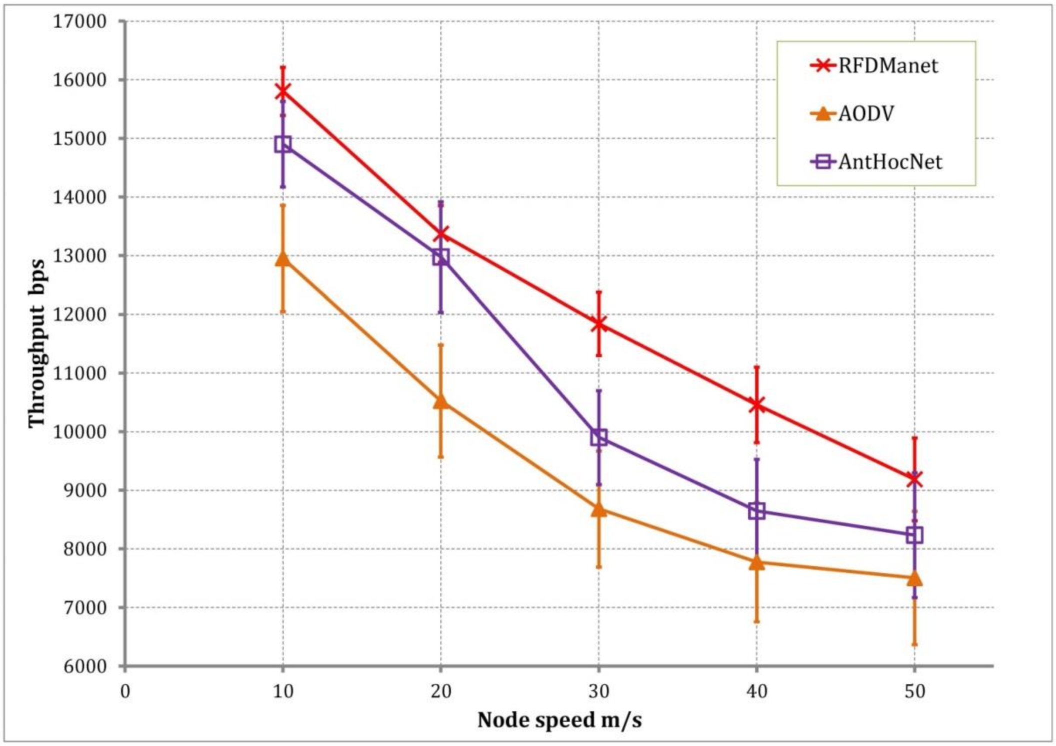


Figure 6.4 Average throughput for 64 nodes network under various speed values

Figure 6.5 shows that RFDManet protocol has better end to end delay than the other protocols. RFDManet end to end delay is about 94.24% less than AODV in low speeds up to 98.9 % at high speed. Most of the delays in AODV are due to route setup and broadcasting of control packets, route maintenance and the data packets queuing. RFDManet protocol minimizes the number of broadcasting and there is no backward traffic from destination node to source node. In the event of route failure, data packets will move until they fall into the destination node. If the destination is unreachable data packet either will be returned to source node to be buffered or lost. Keeping in mind, as the packet is moving around the network, the process of learning is still carried out and the nodes update their altitudes tables continually.

RFDManet end to end performance is about 61.4% better than AntHocNet in low speeds. In general RFDManet has better convergence to solution than AntHocNet. The contribution of data packets in the learning process adds significant advantages to RFDManet over AntHocNet. Other factor like quick adaptation to network characteristic also contributes in these results.

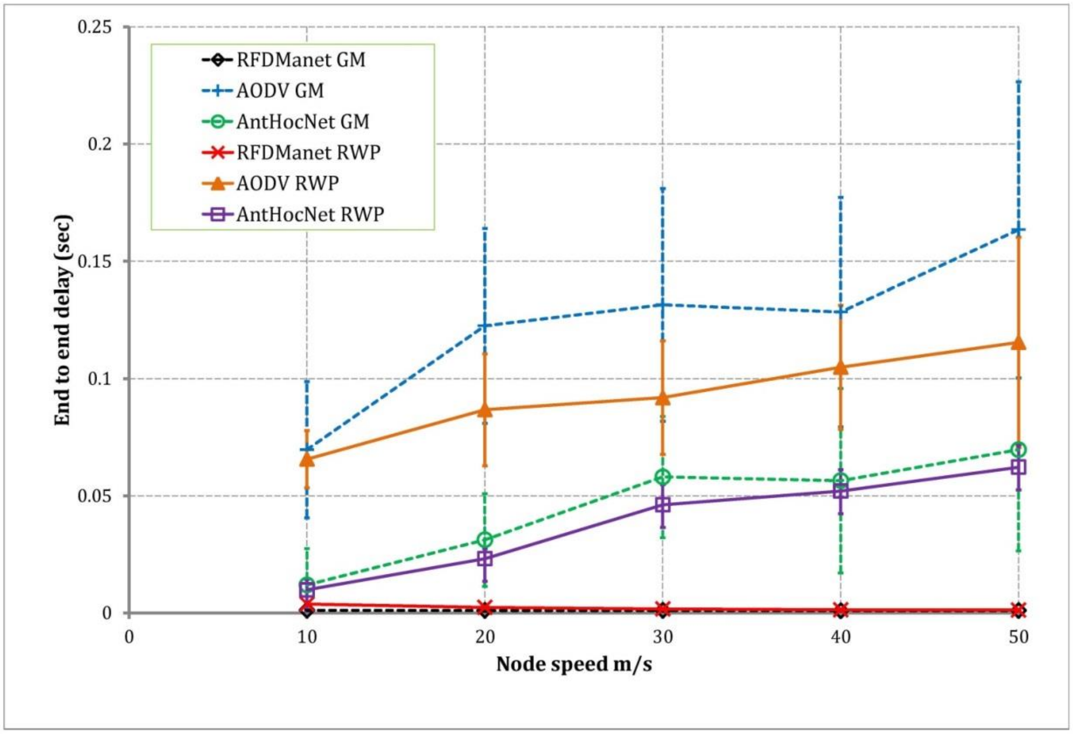


Figure 6.5 Average end to end delay for 32 nodes network under various speed values.

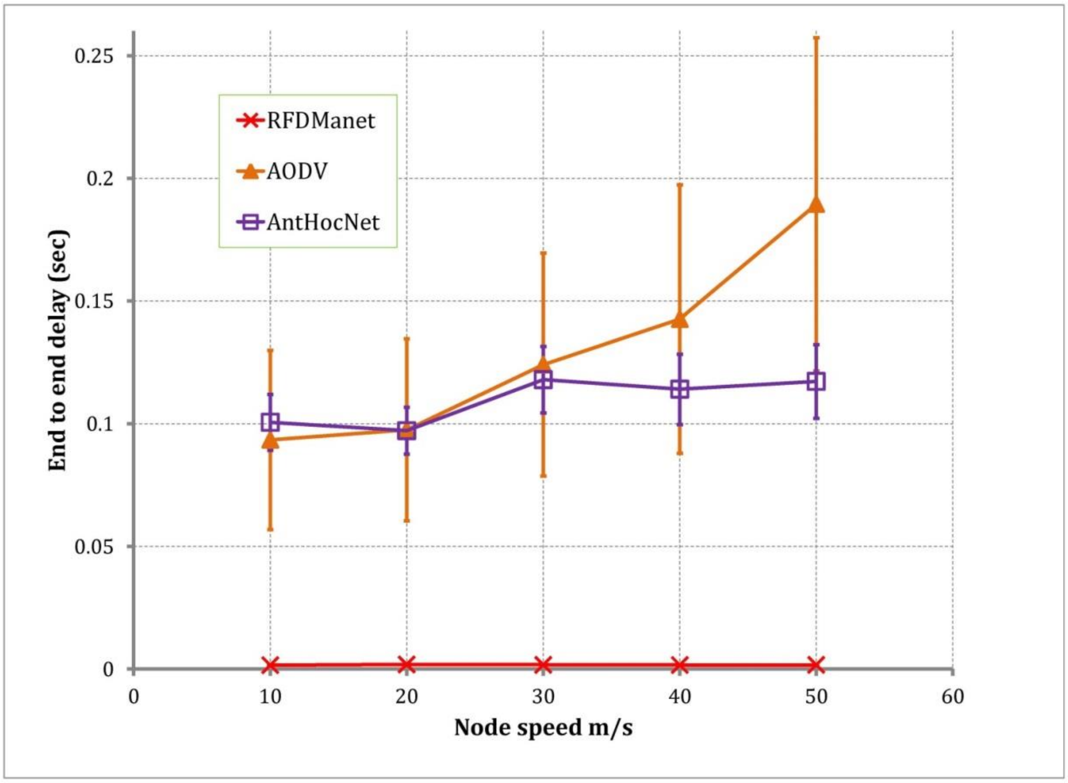


Figure 6.6 Average end to end delay for 64 nodes network under various speed values.

AntHocNet protocol also uses local repairing technique although it uses some restriction on broadcasting but it also adds delay to network. Figure 6.6 shows the end to end delay for 64 nodes again RFDManet protocol performs better.

Figure 6.7 and Figure 6.8 show the jitter for 32 and 64 nodes. The RFDManet protocols compared with AODV and AntHocNet. Again RFDManet protocol outperforms both protocols and has less jitter. The proposed protocol reduces the queuing in the network. At the same time, the protocol is congestion aware and even data packet can sense the congested nodes in the network. These congested nodes are not preferable by proposed protocol.

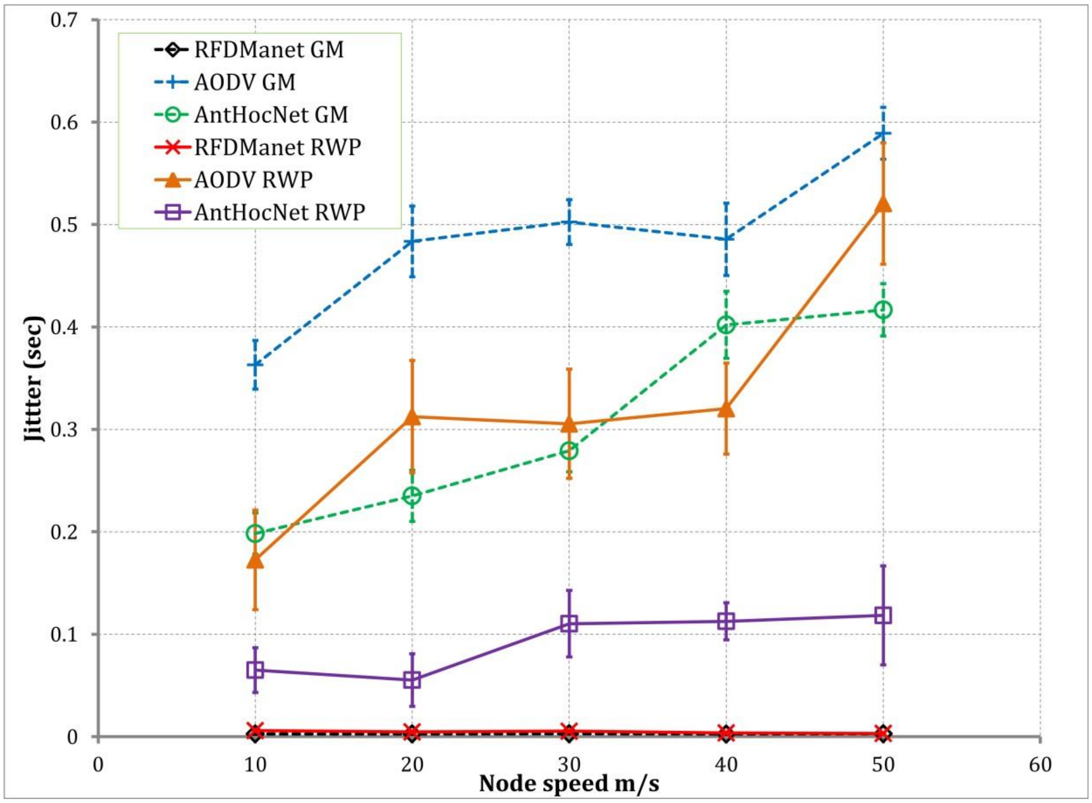


Figure 6.7 Average jitter for 32 nodes network under various speed values.

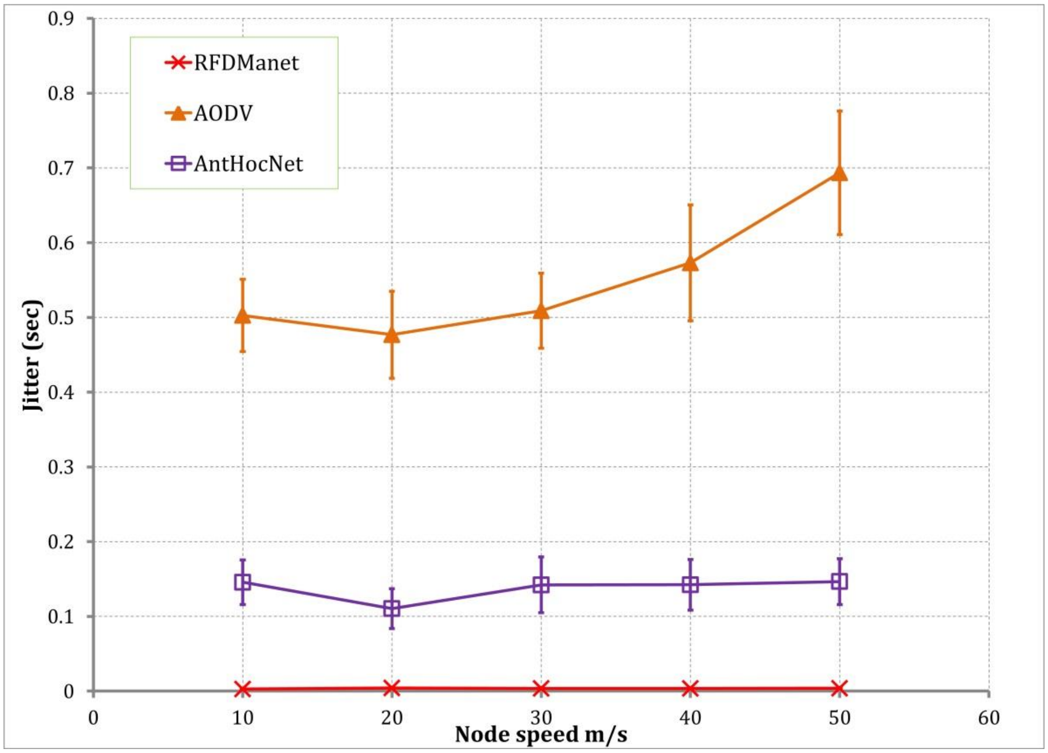


Figure 6.8 Average jitter for 64 nodes network under various speed values.

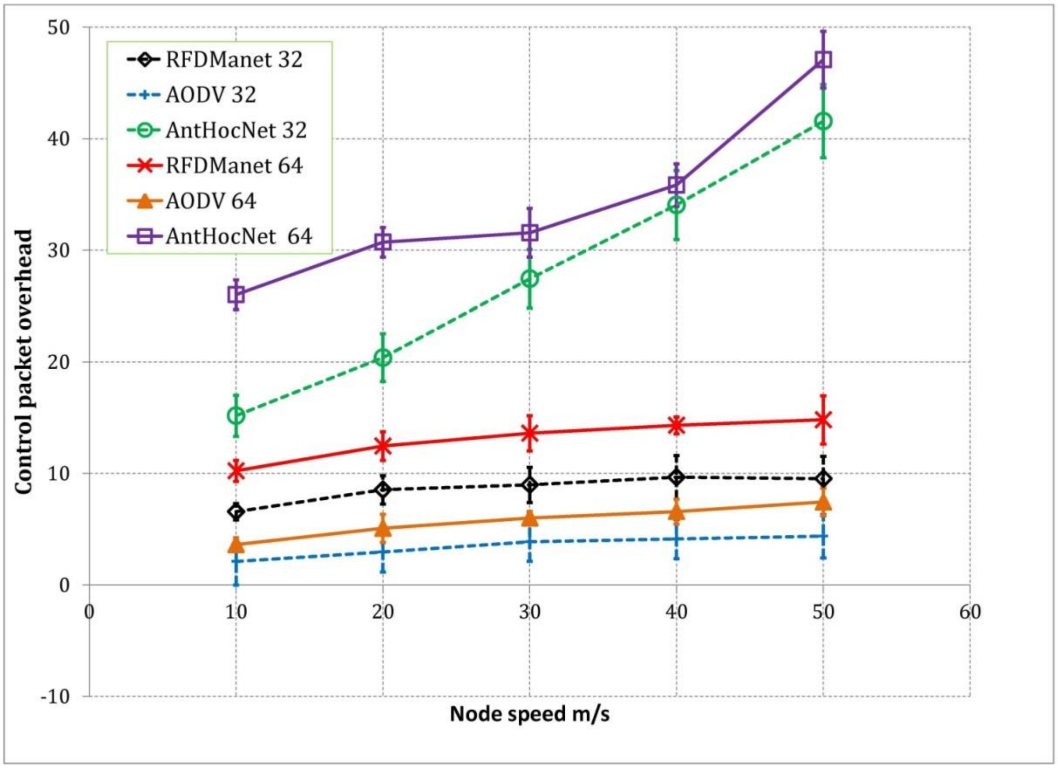


Figure 6.9 Control packet overhead under various speed values

Figure 6.9 shows the control packet overhead of the three protocols under RWP mobility. The better performance of RFDManet costs the network to generate more control overhead than AODV protocol. However, the control overhead is less than AntHocNet. The absence of backward agent and the use of intelligent data packet led to less control overhead than AntHocNet.

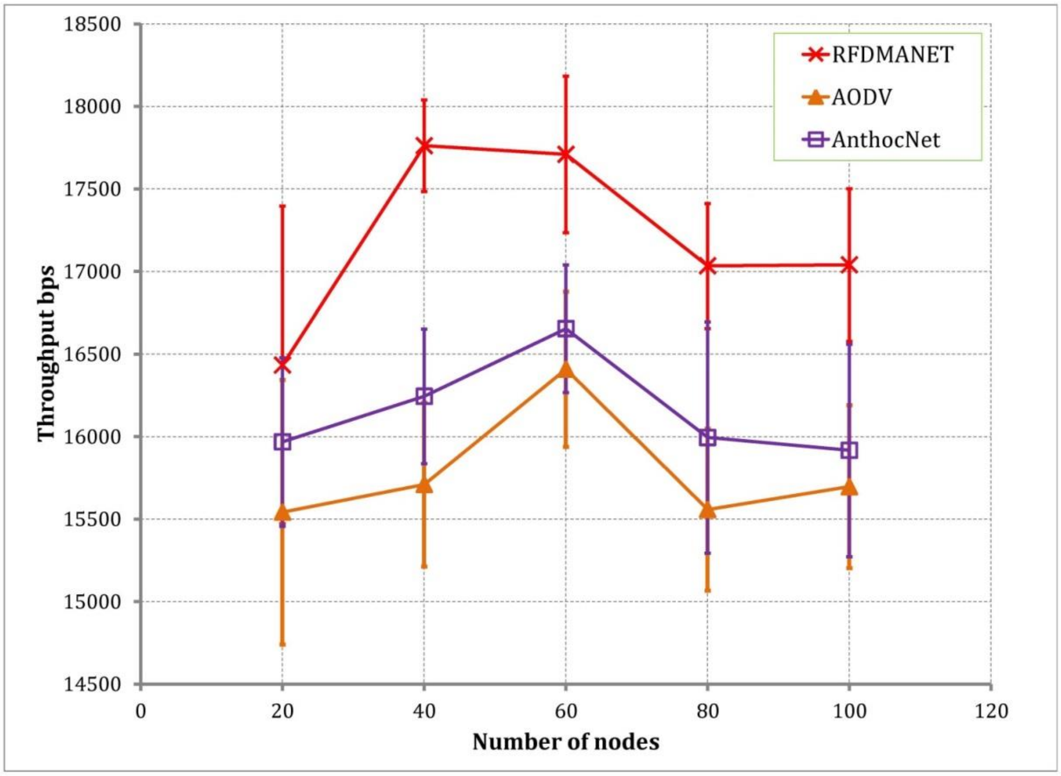


Figure 6.10 Network throughput under various numbers of nodes.

Figure 6.10shows the effect of node density on network throughput under RWP mobility, where nodes speed is set to 10m/s and number of nodes has been varied from 20 to 100 nodes. The best value for throughput is between 40 to 60 nodes. Less connectivity and more route breaks occur for low number of nodes. While the network become more congested and route length increases if nodes number increased. From the figure it is clear that the proposed protocol overcome both other protocol throughout all number of nodes.

Figure 6.11and Figure 6.12 shows the end to end delay for various numbers of nodes. Congestion, queuing and route change are the main factors that contribute in network latency and jitter. The proposed protocol reduces the queuing in the network. At the same time, the

protocol is congestion aware and data packets can sense the congested nodes in the network. For each route failure, the AODV route recovery requires more time which causes more delay because of broadcasting and queuing.

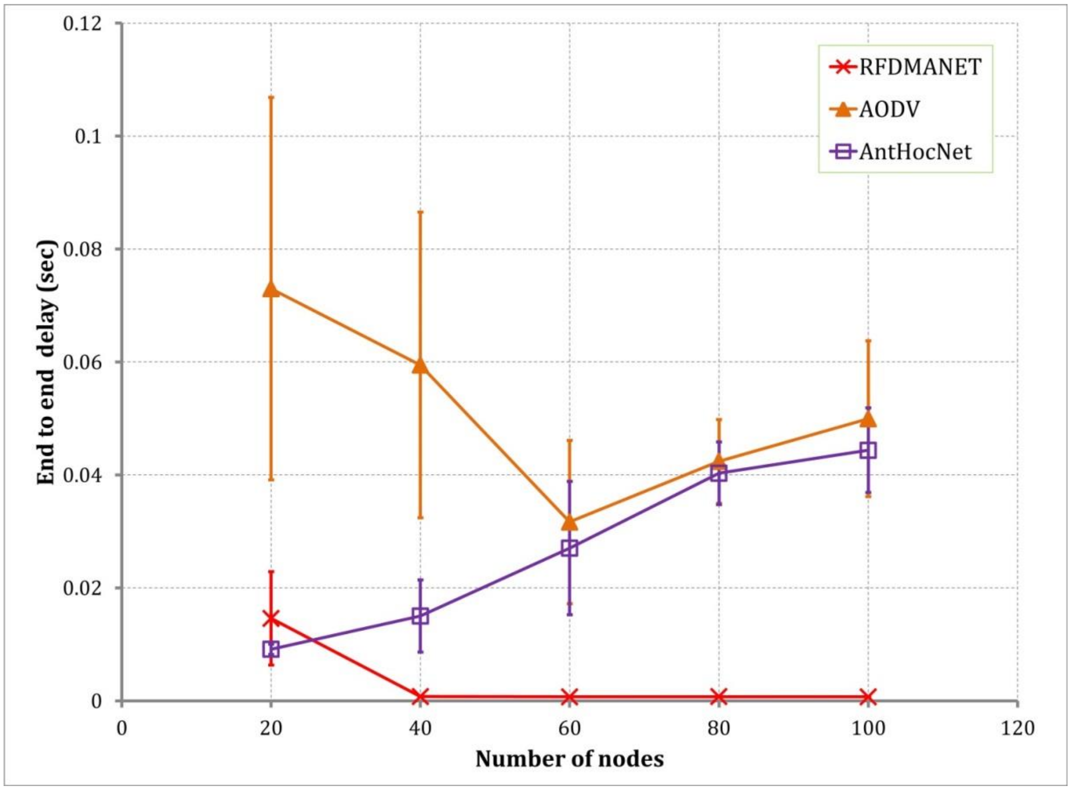


Figure 6.11 Network end to end delay under various numbers of nodes.

RFDManet protocol usually do not buffer data packet, unless if the destination is unreachable. As the number of nodes increases in the network, the cost of broadcasting increases as well as the probability of collision increases. RFDManet protocol is more efficient in searching for better routes, and it minimizes the number of broadcasting in the network. It can be seen that the end to end delay at low node density is high. This occurs because at low node density the probability of the destination being unreachable is high. The delay occurs usually as a consequence of queuing and broadcasting as the probability of the source or destination node being isolated from each other. Putting in mind, in our protocol queuing occurs only at the original source when the destination is unreachable and all the nodes around the original source do not have a route to the destination. In other words all the nodes around the source have an altitude equal to one. This situation is rarely occurs unless the source node is isolated alone or with only few nodes and usually at low node density. Moreover, as explained earlier, hello messages erode altitude of neighbour nodes, if a node with an altitude less than one broadcasts its hello messages to those neighbour that have been punished, it will result in decreasing their altitudes. The reason behind low end to end delay in our protocol is data packets are always being sent and search for new routes. The probability of queuing and broadcasting is low and data packets may be deleted in case if they reach a dead end or when maximum number of hubs is reached.

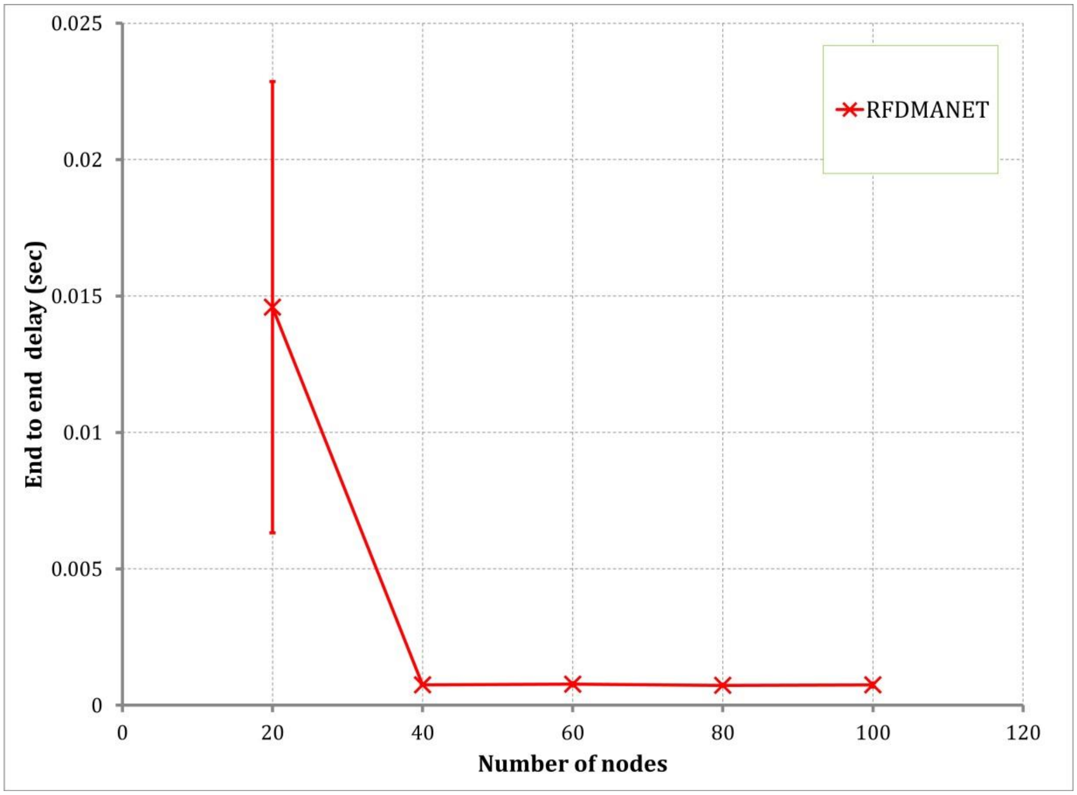


Figure 6.12 RFDManet end to end delay under various numbers of nodes.

6.4 Summary

In this chapter, RFDManet routing protocol for mobile ad hoc networks has been proposed. RFDManet is a swarm algorithm inspired by the way rain drops group together and create rivers.

RFDManet is a hybrid routing protocol, which uses reactive and proactive methods to update its routing tables. Moreover, it uses data packet in learning process so that the data packets can contribute in learning and guide themselves through best routes. Learning process in RFDManet does not require any types of backward agents as the learning is done with the drops and data packets. The protocol uses two kinds of tables, reactive and proactive. Both of these tables consist of two tables, altitude table and my_altitude table. The separation helps in directing data packets through reliable routes. The results show that RFDManet performs better than AODV and AntHocNet protocols.

References

[1] G.S. Sharvani, N.K. Cauvery and T.M. Rangaswamy, "'Different Types of Swarm Intelligence Algorithm for Routing," *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on*, 2009, pp. 604-609.

[2] P. Rabanal and I. Rodriguez, "'Hybridizing River Formation Dynamics and Ant Colony Optimization,"in Advances in Artificial Life. Darwin Meets von Neumann, vol. 5778. George Kampis, et al. , Springer Berlin Heidelberg, 2011, pp.424-431.

[3] P. Rabanal, I. Rodríguez and F. Rubio, "'Applying River Formation Dynamics to the Steiner Tree Problem," *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*, 2010, pp. 704-711.

[4] P. Rabanal, I. Rodriguez and F. Rubio, "'Finding Minimum Spanning/Distances Trees by Using River Formation Dynamics,"in Ant Colony Optimization and Swarm Intelligence, vol. 5217. Marco Dorigo, et al. , Springer Berlin Heidelberg, 2008, pp.60-71.

[5] G.D. Caro, F. Ducatelle and L.M. Gambardella, "'AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications*, vol. 16, SEP 2005, pp. 443-455.

[6] E.M. Royer, P.M. Melliar-Smith and L.E. Moser, "'An analysis of the optimum node density for ad hoc mobile networks," *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 3, 2001, pp. 857-861.

[7] C. Bettstetter, "'Smooth is better than sharp: a random mobility model for simulation of wireless networks," *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, 2001, pp. 19-27.

[8] S. Rajagopalan and C. Shen, "'ANSI: a swarm intelligence-based unicast routing protocol for hybrid ad hoc networks," *Journal of Systems Architecture: the EUROMICRO Journal*, vol. 52, no. 8, 2006, pp. 485-504.

Chapter Seven

Power and Congestion Aware Smart Data Packet Routing Protocol

7.1 Introduction

Mobile Ad hoc Network (MANET) consists of several mobile wireless nodes that communicate with each other through direct or indirect communication links. Nodes within these networks act like routers. The absence of infrastructure and the mobility of these nodes compose an enormous challenge to routing algorithms in such networks. Nodes in mobile ad hoc network are powered on batteries and therefor have limited energies. Reducing communication related power consumption is an important factor in designing routing algorithm. Routing protocol plays significant role in determining network performance. A routing protocol should offer acceptable QoS for applications and maintain the power in the network.

The previous proposed routing protocols are hybrid protocols. These protocols have higher packet overheads which usually results in consuming higher power. This chapter presents Power and Congestion Aware Swarm based routing protocol[4] for mobile ad hoc networks where data packets have the ability to cognitively route themselves through the network. This is a new approach to solve the problem of routing in cognitive packet ad hoc networks using River Formation Dynamics (RFD) optimization algorithm. The proposed protocol is based on the idea of giving cognitive capabilities to data packets in order to select their own route based on quality of service (QoS) requirements. Data packets can route themselves though routes that are less congested and throughout nodes with higher battery capacity in order to load balance the network.

[4] Throughout this chapter we refer to the protocol as smart protocol.

7.2 Related works

Mobile ad hoc network routing has motivated many researchers in the past two decades. A mobile ad-hoc network is based on self-organizing network where each node participates in the routing protocol. Since the topology of MANET network is dynamic and frequently changing, it composes numerous problems, and great challenges to the routing algorithms. A numbers of routing methods have been proposed to address these problems [1-3].

Routing algorithms in these networks can be classified into three main groups, proactive, reactive and hybrid routing protocols. Proactive routing protocol is a table driven routing protocol where all nodes are required to have complete knowledge of the network and therefor routing tables are periodically updated. Examples of proactive routing protocols are Optimized Link State Routing protocol (OLSR)[4] and Destination-Sequenced Distance Vector routing protocol (DSDV)[5]. Proactive routing protocols have low route setup latency. However, such routing protocols may not cope with the dynamic and rapid change of the network infrastructure of mobile ad hoc networks. Since nodes in ad hoc network are moving, links between nodes may become unavailable. Moreover, the quality of the links may change as these nodes are moving. Convergence to a new stable route after such a movement as well as maintaining route tables are costly in terms of resource usage and may lead to some delay in the network. On the other hand, reactive routing protocols set up routes on demands. Whenever a node requires a route to a specific destination, it starts route setup procedure. Route discovery process usually consists of broadcasting a route request message throughout the network. These types of protocols can be mainly divided into two categories, source routing protocols like Dynamic source routing (DSR)[6] and distance vector routing protocols like Ad hoc On Demand Distance Vector (AODV) [7]. Source routing protocols suffer from routing overhead especially in large networks. The frequent route break due to the mobile nature of mobile ad hoc networks requires the reinitiating of route discovery process, in both types, which adds more delay and queuing to the network. Hybrid routing protocols tries to address the problem in both reactive and proactive by combining features from both reactive and proactive protocols into a hybrid protocol. An example of hybrid routing protocol is zone routing [8]. The main drawback of hybrid routing protocol is the high resource usage.

In [9, 10] different power aware metrics are summarized which are mainly used to determine route quality in power aware routing protocols.

Different power aware routing protocols have been proposed to solve routing problems in ad hoc network [3, 11]. In [12] probabilistic analysis is used to show the effect of multi-user interference with and without distributed power control on network performance. The authors study the effect of interference, power control and different forwarding strategy on network lifetime. Two method of packet forwarding introduce by the authors, power controlled nearest forward (PCN) and network lifetime extending PCN (E-PCN). These methods try to increase the network lifetime.

In [13] a Power-aware Dual-Tree-based Multicast Routing Protocol (PDTMRP) for MANETs is proposed. The protocol tries to load balance the network in order to increase network lifetime. It divides the nodes into two groups and builds a multicast tree for each group in order to load balance the network.

Device-Energy-Load Aware Relaying (DELAR) protocol for heterogeneous mobile ad hoc networks is proposed in [14].The network is composed of two types of nodes. The first type of nodes is equipped with long range transmitter, while the second type of nodes is equipped with ordinary transmitter with normal ranges. A hybrid transmission technique is use to coordinate the transmission between these two types of nodes. The protocol deals with the unidirectional links, introduce as a result of different transmission power, by replying the acknowledgement frames through mini-routes, nodes with ordinary transmission ranges.

In [15]a combination of MAC and network layer algorithms is proposed to solve the problem of power consumption in ad hoc networks. The protocol tries to overcome the problem of rebroadcasting the RREQ message, where some efficient node can be neglected if they rebroadcast after inefficient nodes, by introducing some new field to the RREQ message and adding other decision to how to rebroadcast.

Power-Aware Link Quality estimation (PoLiQ) technique introduced in [16]to estimate the link quality in vehicular networks. The protocol selects reliable multi-hop forwarding nodes depending on link quality estimated through the beacon reception rates from neighbouring vehicles and the neighbours’ transmission settings, e.g. transmission power and data rate.

Swarm intelligence has shown to be a good optimization algorithm and has been applied to the field of routing optimization [17, 18]. ACO is a type of swarm intelligence and has been used to address ad hoc routing problems [19-21].

In Cognitive Packet Network [22], intelligent packets can route themselves in the network. Routing decision is made by Random Neural Networks. CPN contains at least three major types of packets: smart packets, dumb packets and acknowledgments. Packets contain extra fields for Cognitive Map and Executable Code. Each node equipped with a buffer named Mailbox which is used to exchange information between packets. Each router should process the content of the packet and a specific program interpreter is required to execute the codes. More information can be found in chapter two.

AHCPN [23] is an implementation of CPN in ad hoc networks where the algorithm tries to decrease power consumption. The protocol minimizes the packet content and tries to save power by minimizing the number of broadcasting in the network. Each smart packet recodes intermediate nodes energy in their move from the source to their destination. At the destination, the acknowledgment packet will carry the information of a route where nodes have higher battery energy. More information can be found in chapter two. CPN has been also used as routing protocol for ad hoc network in [24].

RFD is a swarm algorithm and has been applied in many combinatorial optimization problems such as the asymmetric traveling salesman problem [25, 26], Optimal Quality-Investment Tree problem [27], minimum spanning Tree Problem [28], and others [29-31]. The RFD algorithm mainly uses one kind of agents which is called drops. Drops moves only from sources to destinations. The RFD algorithm is a competitor to ACO algorithm and has shown to perform better than ACO in many applications [27, 30]. More information can be found in chapter three.

7.3 Network models

7.3.1 Introduction and routing tables

The proposed protocol uses different phases in order to setup route and maintain the routes in the network. First of all, the protocol uses on demand route setup, which is consists of flooding the network by broadcasting drop[5] messages. However, hello messages are being used to build a proactive route table. Drops are generated throughout all data sessions in order to maintain routes. With this perspective, the proposed protocol is considered a hybrid routing protocol. Moreover, data packets are considered as special kind of drops. Data packets can guide themselves through the network and update routing information according to available paths,

[5] Throughout this chapter, the word drop is used to refer to packet that acts like water drop.

using power and congestion metric. This kind of data packet is called smart packet. However the protocol can handle both smart and dumb packets. With this perspective, our protocol is a type of cognitive packet networks protocol. The proposed protocol is implemented in network layer and uses promiscuous communication mode to monitor neighbours' packets in order to update its routing tables. Each node contains an altitude table where the altitudes of other nodes are stored. Another table is used to store the node itself altitudes towards other nodes. Drop packets tend to select nodes with higher gradients.

Smart data packets are routed using restricted stochastic algorithm where they can guide themselves and contribute in the learning process. This will prevent the loss of data packet especially in large networks. To keep searching for other paths far from recent discovered path, drops are sent throughout data session. These drop packets are routed stochastically over different paths using altitude tables. Non smart data packets are routed using minimum altitude as minimum represents the best discovered path. Hello messages are responsible for information diffusion in the network.

The route table is represented as altitude table as shown in Table 7-1,where N_i represents neighbour node i. T_i represents the time delay of node i. P_i is the remaining battery power of neighbour i (normalized power). ALT_{ij} is the altitude to destination j through node i (node i altitude to destination j)

Table 7-1 Altitude table

			Destinations				
Neighbours	Node power	Time delay	D1	D2	D3		Dk
N_1	P_1	T_1	Alt$_{11}$	Alt$_{12}$	Alt$_{13}$		Alt$_{1k}$
N_2	P_2	T_2	Alt$_{21}$	Alt$_{22}$	Alt$_{23}$		Alt$_{2k}$
N_3	P_3	T_3	Alt$_{31}$	Alt$_{32}$	Alt$_{33}$		Alt$_{3k}$
N_j	P_j	T_j	Alt$_{j1}$	Alt$_{j2}$	Alt$_{j3}$		Alt$_{jk}$

Table 7-2 My_altitude table

Destination	My altitude
D_1	Alt_1
D_2	Alt_2
D_{k-1}	Alt_{k-1}
D_k	Alt_k

Another important table is my_altitudes shown in Table 7-2. This table contains the node itself altitudes to other destinations.

Power information usually decays slowly and gradually compared to other network parameters. To distribute power information through the network, both hello and drop messages are used. However, smart data packets do not carry power information. As usually the rate of data packets are much higher than control packets. It is better not to carry this information with data packets as it may add more cost to the network.

7.3.2 Route setup procedure

At network initialization the altitude of all nodes are set to one. This creates a flat surface. Sending data in such a flat surface will spread the packets everywhere through the network and may not reach the destination. Broadcast method is used to locate destination. The acknowledgement from the destination will serve creating a crack in this surface, which will temporally guide successive packets until better route will be found. When a source node starts a communication session, it checks its altitude table for the destination. When the altitude for the destination is one, no route for the destination, the source broadcast drop messages. Up on the receiving of drop message the destination replies with acknowledgment message.

Upon receiving broadcast message by an intermediate node, it will be either rebroadcasted or deleted. The packet will be rebroadcasted if it is the first copy arrived to the intermediate node. Broadcast message carries accumulated path power. This is used to check for paths with less power consumption. Another condition to rebroadcasting is if the number of hops or accumulated power is less than a previous received copy of the same broadcast. Due to this restricted broadcasting procedure drop packets can quickly spread around the network through different paths, and the number of broadcasting is reduced.

When the broadcast message propagated in the network, nodes which receive the broadcast will create a reference route to the original source. The broadcast message carries the altitude of the broadcasting node with respect to the original source node. For the original source its altitude to itself will be zero. The receiver of the broadcast message will update its altitude to the source according to the following

$$\mathrm{altitude_s(j) = \ altitude_s(j) - \mu_s(altitude_s(j) - \ Broad_{altitude_s}),} \quad \mu_s \in [0,1] \qquad (7.1)$$

where *altitude$_s$(j)* denotes the altitude of receiving node *j* (in my_altitude table) to the original source *s*. $Broad_{altitude_s}$ is the altitude of broadcasting node to the original source s in the broadcast message, μ_s is the adaptation constant (value of 0.1 is used. The value of adaptation constant controls the altitudes downslope toward the source node. If high value for adaptation constant is used, the recent broadcast information will override the previously learned information toward the source, but it will give better downslope at far distance node in large networks. On the other hand, low value will keep most of the previously learned information and will not add biasing to the network.

When a broadcasted drop reaches its destination, the destination node sends back an acknowledgment packet to the source. This packet will crack the surface in a gradient way that there is always a downslope from the source to destination according to the following equation.

$$\mathrm{altitude_d(j) = min_{l\epsilon V}\big(altitude_d(l)\big) - grd * (min_{l\epsilon V}\big(altitude_d(l)\big) - altitude_d(i))} \qquad (7.2)$$

where *altitude$_d$(j)* is altitude of the node *j* toward destination node *d*. *altitude$_d$(i)* is altitude of sender node *i*. *V* is a set of neighbour nodes not including the sender. grd controls the gradient of the crack should be set between 0 and 1 (set to 0.1).

The path that acknowledgment takes in its return to the source is depending on attitudes to the source created throughout the broadcast process and previous information in the nodes if available. The broadcast change nodes altitude levels toward the source to a funnel structure, where the source is located at the bottom end of that funnel. Intermediate nodes will forward the acknowledgment packet to a neighbour with minimum altitude regarding the source. If due to mobility the next node is not available and a link break occurred, the node will forward the acknowledgment packet to the second best neighbour with minimum altitude toward the source.

After the first broadcasting, the source node will wait for a time period equal to network traversal time. If no acknowledgment received throughout this period, the source node may send another broadcast message until the maximum number of retries is reached. This idea is inherited from AODV protocol. When the source node receives the acknowledgment packet, it starts sending data packets.

7.3.3 Drops and smart data packets routing

Smart data packets inherit their behaviour from drop control messages. This section is started by explaining how drops are routed in the network then the smart data packets are explained and how they are route themselves in the network.

In order to find better paths, swarm based algorithm uses maintenance agent packets throughout the data session. These agents are called drops. The rate of these drops can be proportional to data rate or constant (constant rate=0.5 sec is used). The drops are propagated according to the gradient probability function

$$\mathrm{P_n(j,k)_d} = \begin{cases} \dfrac{\mathrm{decreasingGradient_d(j,k)}}{\sum_{\mathrm{l} \in \mathrm{V}_n(\mathrm{j})} \mathrm{decreasingGradient_d(j,l)}} & \text{if } \mathrm{k} \in \mathrm{V_n(j)} \\ 0 & \text{if } \mathrm{k} \notin \mathrm{V_n(j)} \end{cases} \tag{7.3}$$

$$\mathrm{decreasingGradient_d(j,k)} =$$

$$\frac{\left(\mathrm{altitude_d(j)} - \mathrm{altitude_d(k)}\right) * \mathrm{norm\_power(k)}}{\mathrm{T(k)}} \tag{7.4}$$

where $P_n(j,k)_d$ is the probability of drop n at node j with destination d to select next node k. $T(k)$ is the average time that node k needs to send a packet. $altitude_d(j)$ is altitude of the node j toward destination d. $norm\_power(k)$ is normalized power of neighbor k equals to $\frac{\mathrm{P_k}}{\mathrm{p_{max}}}$. $V(j)$ is set of neighbour node of node j.

Time delay parameter is used where it detects the congested node and used to represent the distance. The more congested the node the more it is not preferred to be selected as a forward node and though the higher distance (T(k)) it had.

Each drop packet contains a field that represents the recent altitude of sender node. Drops change node altitudes by the process of eroding and adding sediments. The node new altitude to the final destination of drop is attached to the drop packet. All neighbour nodes will update their tables according to the new altitude carried by the drop packet. This is acts like a distributed learning procedure where one change affects a group of nodes. Each node also monitors its sent packets. Unlike ant based algorithms where an ant updates specific link pheromone intensity that is moving along, one drop alters a node altitude and all neighbours are updating their tables according to this change.

Each node monitors its sent packets. When it starts sending a packet it computes how much time is needed for the forwarded node to send it again. It keeps tracking of this by computing the running average of the time needed by its neighbour to forward its packets.

$$\mathrm{T_k(t)} = \gamma\mathrm{T_k(t-1)} + (1-\gamma)\mathrm{C_k(t)}, \quad \gamma \in [0,1] \tag{7.5}$$

where $C_k(t)$ is new delay in node k at time t, T_k is time delay for node k. T initially is set to be equal to the time required by an unloaded node to send a packet(set to 3 ms) [22]. γ is a parameter regulating how quickly the formula adapts to new information (set to 0.7). Using this type of distance will help in selecting uncongested node.

In order to give drops the ability to explore more paths and search for better solutions, another method inspired from nature is used. When a drop of water falls from high altitude and hit a surface, it will bounce up from the surface (or some time an equal amount of water). The amount that the drop will bounce off is proportional to altitude difference between the nodes. A ratio of 5% is used. Local loops are prevented as a drop will never follow the same path twice, without any extra technique to prevent local loops and cycles. This method is better than the one introduced by [28, 30] as depending on probability may lead to some loops as they declared, in which some method like source routing should be used, which is not desirable for our protocol.

As drops move in the network, they change the topographical structure of altitude surface. This change is proportional to remaining battery power as well as the gradient of the surface. Drops usually erode and add sediments to their path. Relating these changes to battery power will make the algorithm search for routes with lower power cost [10]. The altitude of node will be eroded more if it had more battery power. At the same time, when battery level is low, more sediment will be added to the node. This reduces the altitude of nodes that have more remaining battery level and make them more attractive for both drops and smart data packets, and gradually the best path will be through these nodes.

The amount of erosion is proportional to remaining battery power in the node as well as the altitude gradient between the node itself and the selected forwarding node.

$$\mathrm{erosion_d(j)} = \propto * \frac{\mathrm{P_j}}{\mathrm{p_{max}}} (\mathrm{altitude_d(j)} - \mathrm{altitude_d(k)}) \qquad (7.6)$$

$$\mathrm{altitude_d(j)} = \mathrm{altitude_d(j)} - \mathrm{erosion_d(j)} \qquad (7.7)$$

where *erosion$_d$(j)* is the amount of erosion of altitude toward destination *d* at the node *j*, *altitude$_d$(j)* is the altitude of node itself(node *j*), *k* is the next selected node. P_j is the remaining battery power of the node itself. P_{max} is the maximum battery power. α is a positive constant number between 0 and 1 which reflects erosion factor (set to 0.7). High value of α will lead higher erosion and missing of optimal solution. While very low value my lead the algorithm to sick in local minimum. Due to nodes mobility and dynamic infrastructure of mobile ad hoc network, low values for α are not preferred.

At the same time drops add sediment to node altitude. The amount of sediment is proportional to the amount of sediment carried by the drop, and inversely proportional to remaining battery power and the altitude difference.

$$\mathrm{sediment_d(j)} = \left(1 - \frac{\mathrm{P_j}}{\mathrm{p_{max}}}\right)(\beta + \varepsilon * (1 - (\mathrm{altitude_d(j)} - \mathrm{altitude_d(k)}))) * \mathrm{carried\_sediment}, \qquad \beta + \varepsilon \in [0,1] \qquad (7.8)$$

$$\mathrm{altitude_d(j) = altitude_d(j) + sediment_d(j)} \qquad (7.9)$$

β and ε are constants controlling the amount of sediments deposit (set to 0.1 and 0.1 respectively). Carried sediment reflects the path characteristic. Paths with higher slopes will result in more carried sediment. Setting β and ε will lead to quickly depositing sediments. Lower value will lead to flat surface around the destination making drops search around the destination for better delivery nodes. P_j is the remaining battery power of the node itself. P_{max} is the maximum battery power.

The sediment carried by the drop to next node is

$$\mathrm{carried_{sediment} = carried_{sediment} + erosion_d(j)\ - sediment_d(j)} \qquad (7.10)$$

Another type of sediment adding occur periodically every specific period of time

$$\mathrm{altitude_{d\in w}(j) = altitude_{d\in w}(j) + \theta,} \qquad \theta \in [0,1] \qquad (7.11)$$

where w is a set of all known destinations. θ is the amount of sediment to be add(set to 0.01). θ acts as forgetting factor. High value will quickly erase learned information, especially for recent learned paths. Low value is chosen because we don't want to corrupt recent learned information and there are other methods that will also help in adapting the information like the punishment procedure if the link become expires. The time period between regular additions is set to be linearly proportional to battery power.

$$\text{time period} \ = \max\left(\frac{\mathrm{P_o}}{\mathrm{P_{max}}}, 0.1\right) \qquad (7.12)$$

where P_0 is node itself power, the minimum time period is limited to 0.1 to prevent the altitudes of nodes with low battery power from increasing rapidly .

Smart data packets inherit their behaviours from drop packets. Both drops and smart packets are moving from source to destination. Both have the ability to search for batter routes and adopt routing information (altitude tables) while they are moving in the network. There are two main differences between drop and smart data packets. Data packets are routed in restricted way and the adaptation factor is reduced for data packets.

Smart data packets have extra fields that have been added to them in order to act like drop packets. The first parameter added is the altitude of the sender which is four bytes in size. The second parameter is carried sediments which is also four bytes in size. Finally we added source address to the packet in order to easily extract source address.

An important issue is the routing of smart packets is slightly different from drop packets. Data packets should not move in directions faraway from its destination. This will prevent the loss of data packet. At the same time drop packets are responsible of exploring other parts of the network. Smart data packets are routed in greedy way in directions close to current best discovered route to destination. Best discovered routes are the routes with higher gradients. Dumb packets are routed through these routes. Smart data packets are routed using equation 7 but with the following decreasing gradient

$$\mathrm{decreasingGradient}_{\mathrm{d}}(\mathrm{j},\mathrm{k}) = \frac{\left(\,\mathrm{altitude}_{\mathrm{d}}(\mathrm{j}) - \mathrm{altitude}_{\mathrm{d}}(\mathrm{k})\right)^{\partial} * \mathrm{norm\_power}(\mathrm{k})^{\delta}}{\mathrm{T}(\mathrm{k})} \qquad (7.13)$$

where ∂ and δ are constant numbers greater than one, and both set to 4 to achieve the greedy movement. Equal weights have been given to both gradient and power.

Usually the number of data packets is higher than drop packets. This high number of data packets can ruin the learning process especially as these packets are moving in a greedy way. If smart data packets allowed to erode with same ratio of drop packets than it will deepen the altitude surface around best discovered path more than other area. This will make the drops and data packets move towered these area and restrict the searching of other parts of the network as well as it will make it difficult to the new better paths to rise up. This reflects flooding in nature. To limit the erosion that is occurring due to this high number of data packet, the percentage of erosion and sediment deposit by smart data packets is reduced to ten per cent of the drop packets.

Finally, it should be noted that drops carry power information while smart data packet do not. Nodes update the neighbour power value in altitude table upon receiving a drop or hello packet.

7.3.4 Hello messages and information diffusion

Hello message is used to defuse proactive route information. The same procedure explained in previous chapters is used, except that a new field is added to the hello packet which is the sender normalized power.

7.3.5 Route failure

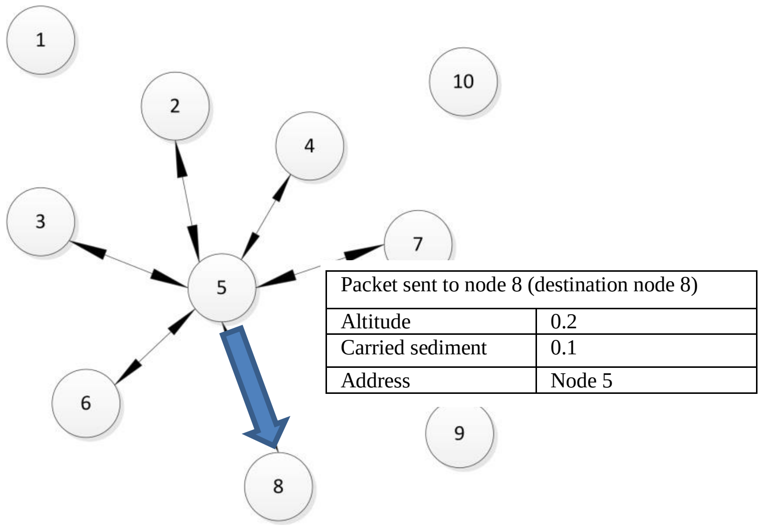
Route failure is handled the same way as explained in previous chapters except that the punishment procedure will not return back the packet and only one node will be punish to reduce the communication and power consumption.

7.3.6 Distributed updating

Both drops and smart data packet incorporate in the learning procedure. Smart data protocol includes two learning procedure. The first one is similar to other agent based protocols. As an agent moves through a node, it updates the node information. For example in ACO based protocol, the backward ant updates the node information as it pass through the node. Only the node that receives the backward ant will update its parameter and other neighbour nodes will not contribute or update their information in any way. As explained earlier, in similar way when a drop passes through a node it erodes its altitude and adds sediment to it. However, there is another type of updating procedure in smart data protocol where the information is spread to the neighbours of the node. When an agent (drop or smart packet) moves from a node to anther it also changes the altitudes of the surrounding nodes, as well as it gives them the new altitude of the sender.

Figure 7.1.a shows an example for a distributed updating in smart data protocol[6]. Node 5 sends a drop or smart data packet to node8. Prior to sending the packet, node 3 has no route to node 8 and its altitude to node 8 is 1 as shown in the table in Figure 7.1.b. at the time, node 4 has a route to node 8 and its altitude is 0.6. From altitude table of node 4, it is clear that it will forward packet with destination 8 to node 7.

[6] Only the altitudes related to destination node 8 are shown in the figure for simplicity.



Packet sent to node 8 (destination node 8)	
Altitude	0.2
Carried sediment	0.1
Address	Node 5

(a) Node 5 and its neighbours

NODE 3TABLES BEFORE SENDING A PACKET FROM NODE 5 TO NODE 8

Altitude table			
Neighbours	Power	Time	Dest.8 Alt.
Node 1	1	0.3	1.0
Node 2	1	0.3	1.0
Node 5	1	0.3	1.0
Node 6	1	0.3	1.0

My table	
Destination	Altitude
Node 8	1.0

NODE 4TABLES BEFORE SENDING A PACKET FROM NODE 5 TO NODE 8

Altitude table			
Neighbours	Power	Time	Dest.8 Alt.
Node 2	1	0.3	1.0
Node 5	1	0.3	1.0
Node 7	1	0.35	0.4
Node 10	1	0.3	1.0

My table	
Destination	Altitude.
Node 8	0.6

(b) Altitude tables before sending a packet.

NODE 3TABLES AFTER SENDING A PACKET FROM NODE 5 TO NODE 8

Altitude table			
Neighbours	Power	Time	Dest.8 Alt.
Node 1	1	0.3	1.0
Node 2	1	0.3	1.0
Node 5	1	0.3	0.2
Node 6	1	0.3	1.0

My table	
Destination	Altitude
Node 8	0.92

NODE 3 TABLES AFTER SENDING A PACKET FROM NODE 5 TO NODE 8

Altitude table			
Neighbours	Power	Time	Dest.8 Alt.
Node 2	1	0.3	1.0
Node 5	1	0.3	0.2
Node 7	1	0.35	0.4
Node 10	1	0.3	1.0

My table	
Destination	Altitude
Node 8	0.56

(c) Altitude tables after sending a packet

Figure 7.1 An example of distributed learning in RFD based protocol.

When node 5 sends a packet its neighbours may receive a copy of the packet as they use promiscuous communication mode. This will erode neighbours' altitude toward the destination of the packet, node 8 in this example. The amount of update is calculated using equation 18, replacing hello altitude by the altitude in the received packet. At the same time, the altitude of the neighbour node to the destination in altitude table is updated with the new value.

After sending the packet, node 3 altitude toward node 8 has been eroded and become 0,92 as shown in Figure 7.1.c. node 3 now has a route to destination 8. Looking to node 4 altitude table, it could be seen that now node 4 has probability to forward packets either to node 5 or node 7. If both node 5 and 7 have same power and delay, the probability of forwarding packets to node 5 is higher than node 7. This procedure not only enhances the spreading of information but it also erodes the edges of the route and creating a wider area (watershed) for routing packets. The advantage of using RFD algorithm is obvious where one packet changes all neighbours' altitude (sex nodes in this example). For ACO based routing it will require six backward ants in order to update the weights of the links between node 5 and its neighbours.

7.4 Implementation and simulation results

In order to evaluate the performance of our proposed protocol, the protocol implemented using OMNet++ as simulation software [32]. For the AODV protocol, the INETMANET add-on package of the OMNet++ is used. In a range of tests, smart data packet routing protocol performance has been compared with the AODV protocol (a reference algorithm in this research area).

7.4.1 Simulation environments

Three scenarios have been used to test the proposed protocol. The first scenario is used to test the performance of the protocol for various nodes speed. While in the second scenario the effect of node density is analysed. The third one is to see the scalability of the protocol.

The first scenario consists of 36 nodes randomly distributed in 600 * 600 m^2. Previous studies show that ad hoc network can produce best performance if the number of neighbours is between six to eight [33]. In our experiment, we have chosen node density close to 7 in order to have good connectivity. With this node density, there is a good probability for nodes to have multipath to their destinations. At the same time, as the environment become more aggressive, nodes speed increases, the probability of link failure increases, which provide a good environment to test our protocol. In order to test protocol performance against node mobility, the speed of nodes is varied from 10m/s up to 50m/s.

The second scenario is used to test the performance for variable nodes densities. The number of nodes varied from 20 to 100 nodes using the same area of the previous scenario. At the same time, nodes speed is set to 10 m/s.

The protocol is tested on larger network size with same node density in the third scenario. The size of the network is doubled to 850*850 m^2 and the number of nodes to 72 nodes. Moreover the number of sources is increased to 20 nodes. 20 nodes are set as destinations. The RWP mobility is used.

Two types of mobility frameworks have been used for both of the scenarios. Random waypoint (RWP) model and Gauss Markov (GM) model have been used [34].Other common parameters are set as following. Battery capacity is 100 mA.h, and their transmission and reception currents are 330mA and 280mA, respectively [35]. Supply voltage is 5V. The medium access control protocol was the IEEE 802.11 DCF. Data traffic was generated using constant bit rate (CBR) UDP traffic sources. Five mobile nodes selected randomly to act as sources and five other nodes acts as receivers. Each node generates a packet every 0.2 second. The network remains silent in the first second. The nodes start sending data at third second and keep sending until the end of simulation, which gives one second for some hello packet to be generated before starting data session. Each node had a radio propagation range of 150m and channel capacity of 54 Mb/s. Hello time intervals is set to 0.5 second. Pause time is set randomly between 0.1 and 1 second. Simulation time was set to 400 seconds. The simulations ware repeated for ten times with different seeds.

The following end to end network characteristic has been studied [36, 37].

1-Throughput: is the measure of the total number of successful delivered data bits over simulations time for a specific node, averaged over the number of source-destination pairs.

2-End to end delay: is the measure of average delay of data packets. This is the time from sending the packet from the application layer at the source node to the time that the packet arrives to the application layer at the destination node, averaged over the number of source-destination pairs.

3-Jitter: is the variation of packet delay which is averaged over the number of source-destination pairs, averaged over the number of source-destination pairs.

4-Routing overhead: is the total number of control packets sent divided by the number of data packets delivered successfully.

5-Power consumed per packet: is the total energy consumed in the network divided by the number of data packet delivered successfully [10].

7.4.2 Results

Starting with the first scenario, Figure 7.2 shows the throughput of proposed protocol compared to AODV protocol for different values of nodes speeds[7]. It clearly illustrates that smart protocol achieves higher throughput than AODV protocol under both RWP and GM mobility. As the speed of nodes increases, the probability of link break increases, this in turn, decreases the network throughput. We can also observe that both protocols throughputs become lower at high speed. Both protocols perform better under RWP mobility if compared to GM mobility. This behaviour can be returned to two main reasons. Firstly, in RWP mobility, the nodes tend to move toward the centre of the simulation area and move away from boundaries. This will leads to fluctuation in node density and as a result, the paths lengths become shorter and node density will be higher than the intended setting. The nodes are better distributed under GM mobility model. Secondly, nodes movement in GM mobility is correlated. In RWP, node can make a sudden change in its direction independent on its previous direction. These are the two main reasons which led to better results under RWP mobility model.

Figure 7.3 presents the average end to end delay in the network for AODV, and smart protocol. Smart protocol shows more than 96% enhancement in end to end delay. The main factors that contribute in network latency and jitter are congestion, queuing and route changing. In AODV most of the delay occurs due to route setup and broadcasting of control packets and route maintenance as well as queuing of data packets. Smart protocol minimizes the number of broadcasting, and tries to select a non-congested node. If a node had a route failure, it forwards the packet to other best node rather than rebroadcasting.

[7] Smart data packet protocol is referenced as SMART in all graphs

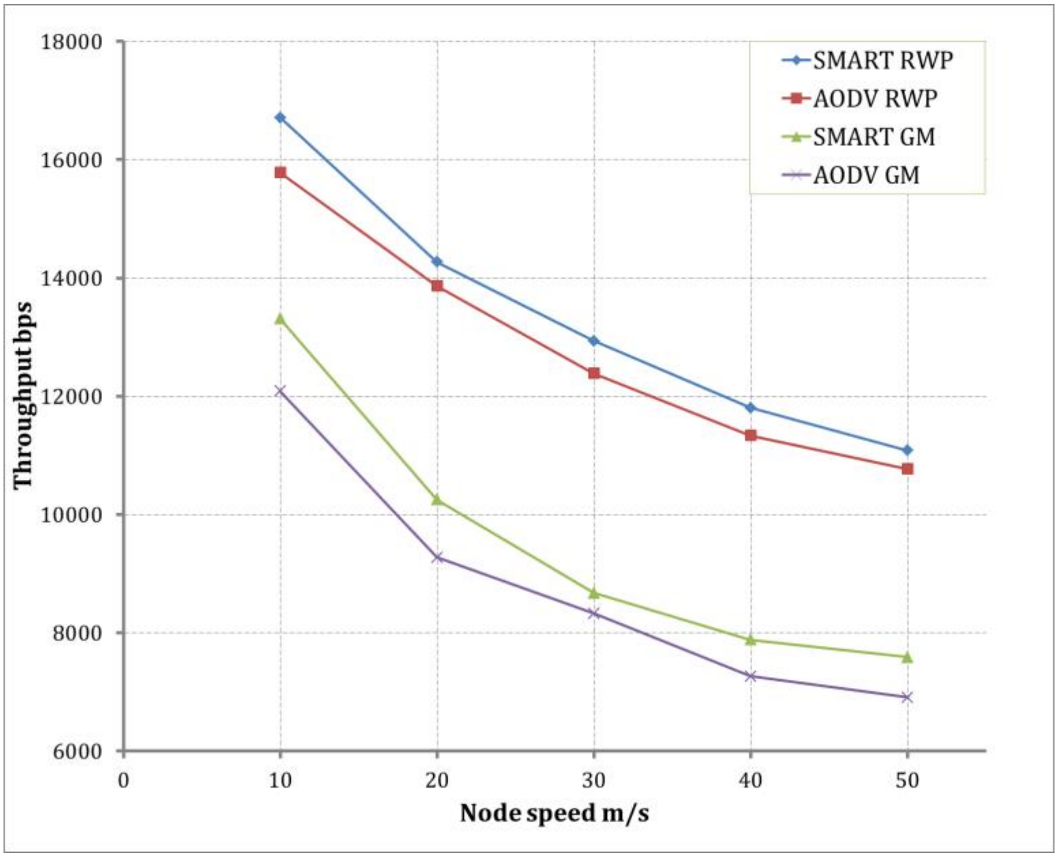


Figure 7.2 Average throughput under various speed values

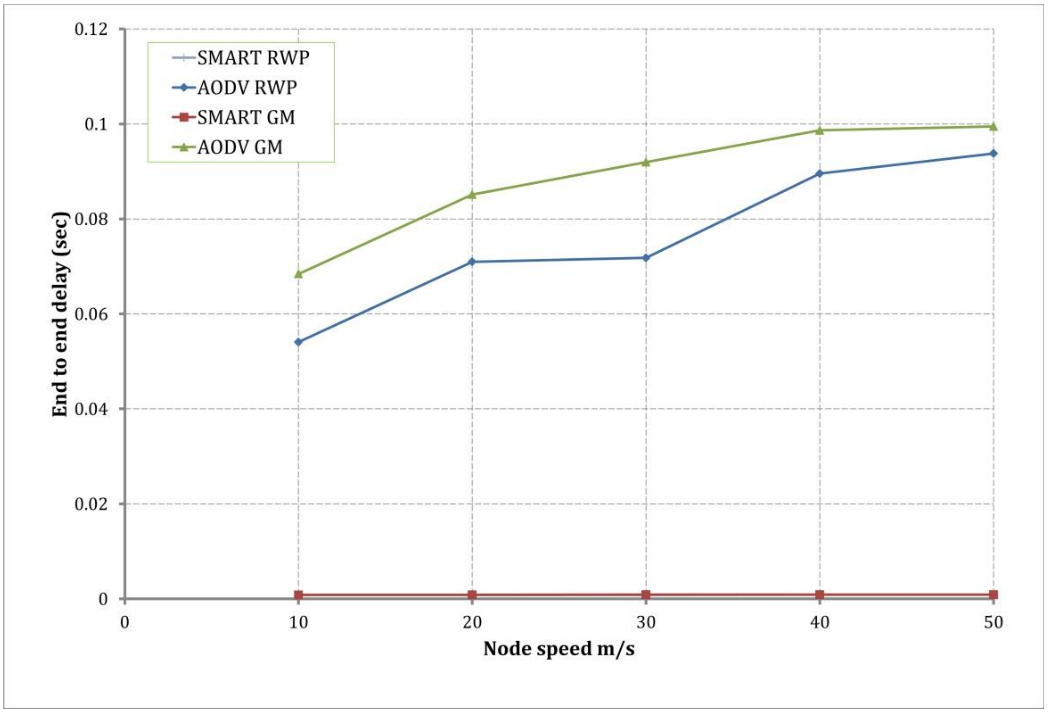


Figure 7.3Average end to end delay under various speed values

Figure 7.4 shows the end to end delay of smart data packet protocol for more conception. The method of packet forwarding and continues real time learning of smart protocol leads to better performance. Smart data packets allow the network to learn more rapidly and the packets can find other routes.

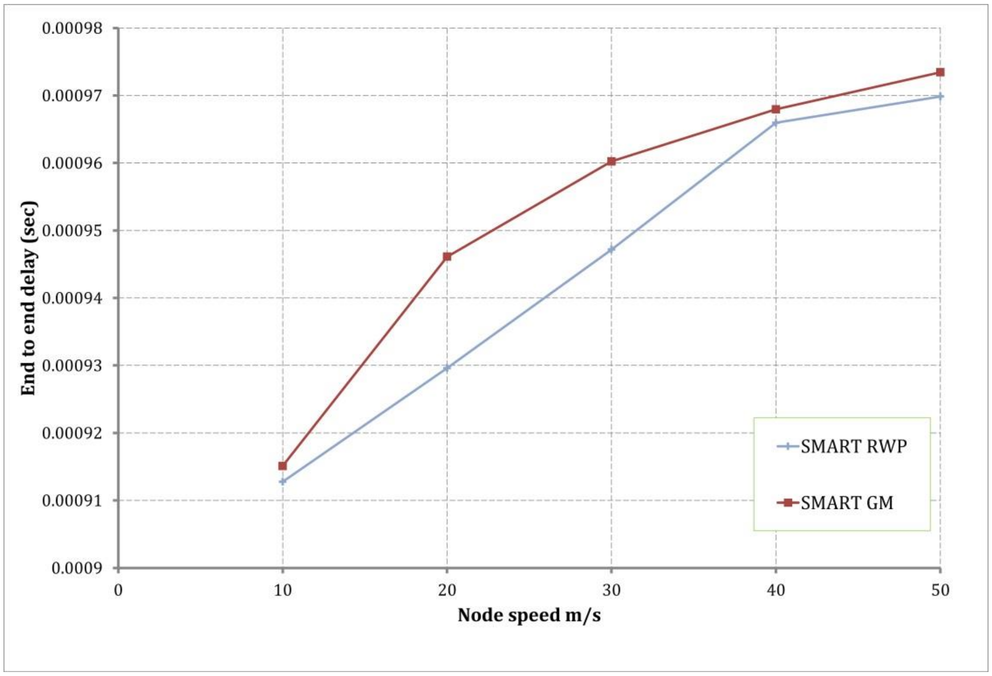


Figure 7.4 SMART protocol end to end delay under various speed values

Figure 7.5 shows the jitter for both protocols. Again smart protocol overcome AODV protocol and has less jitter. Variation in network structure due to nodes mobility causes link breakage. AODV and has route recovery mechanism which consists of queuing and rebroadcasting as well as local route repair. Smart protocol is based on instantaneous data packet rerouting to other better route in case of route failure. As the data packets moves from source to destination, all nodes within the route will continue their learning process by the data packet. Moreover, all nodes that are neighbours to the route will also learn as the process of learning in smart protocol is distributed. An extra point to address here, data packets are routing themselves through multipath to the destination through nodes around best discovered path. This will load balance the network, as well as it creates a valley like structure which its lowest end is at the destination. When a node in-between becomes unreachable, the data packets should easily find another path to its destination. For more clarification, the jitter for proposed protocol is shown in Figure 7.6.

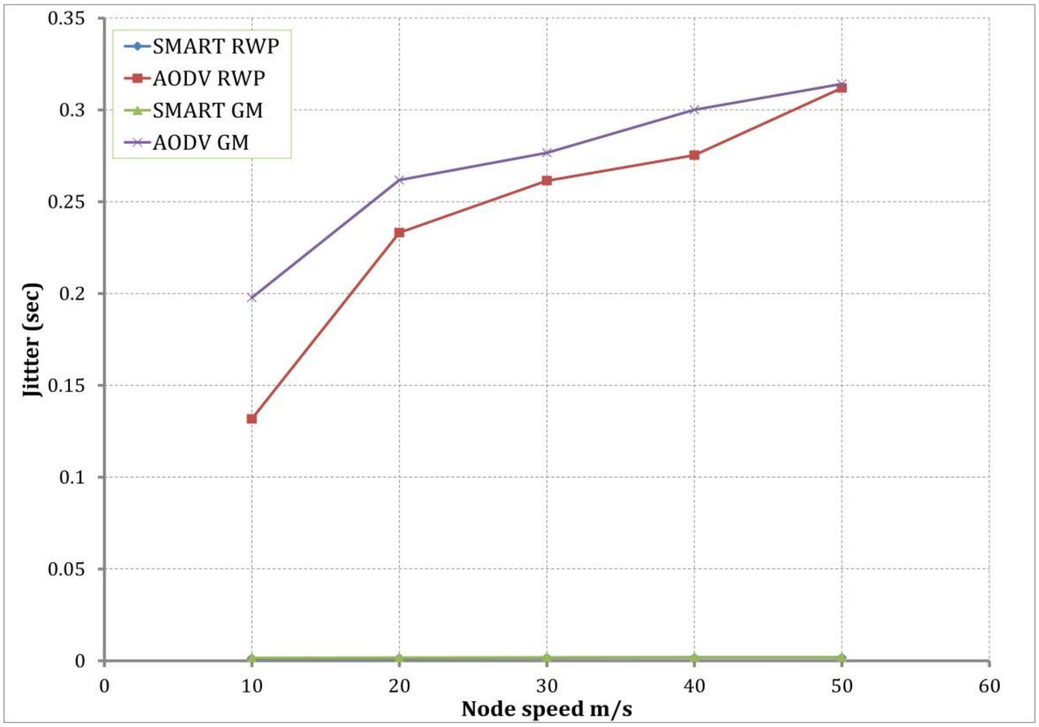


Figure 7.5 Average jitter under various speed values

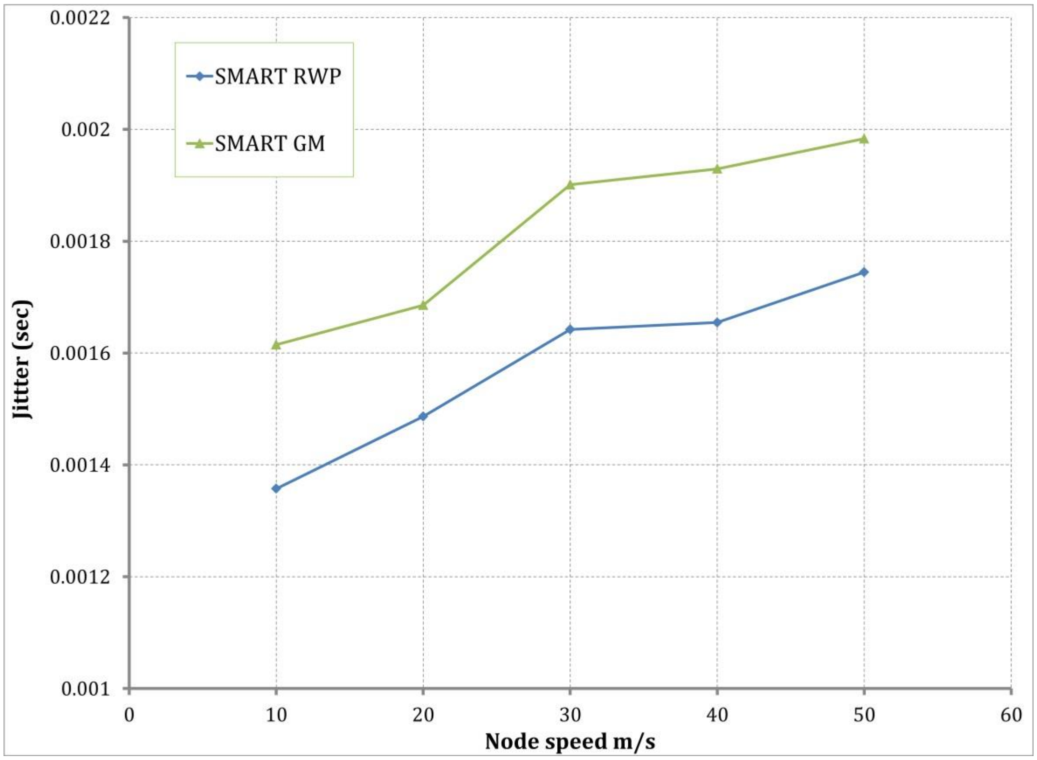


Figure 7.6 SMART protocol average jitter under various speed values

Smart data protocol is a hybrid routing protocols and it sends control messages throughout the entire data session in order to maintain the route between the source and the destination. This can produce more overhead in the network and costing the network to use more resources. Figure 7.7 shows the control packet overhead of both protocols under RWP mobility. Smart protocol generates more control overhead than AODV protocol.

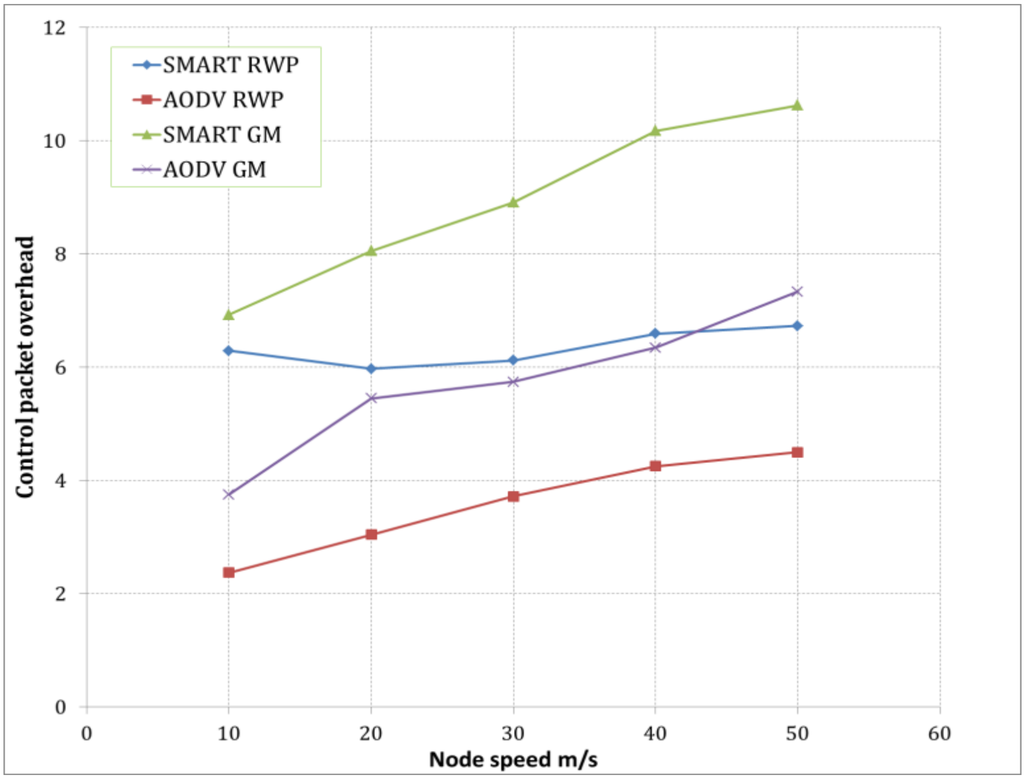


Figure 7.7 Control packet overhead under various speed values.

Figure 7.8 shows average power consumption to deliver each packet. It is clear that smart protocol consumes less power for packet delivery. Our proposed protocol tries to minimize cost per delivered packet. The packets are routed through nodes that have higher battery level at the same time it tries to avoid congested nodes. This will minimize the number of retransmission in the network which saves power.

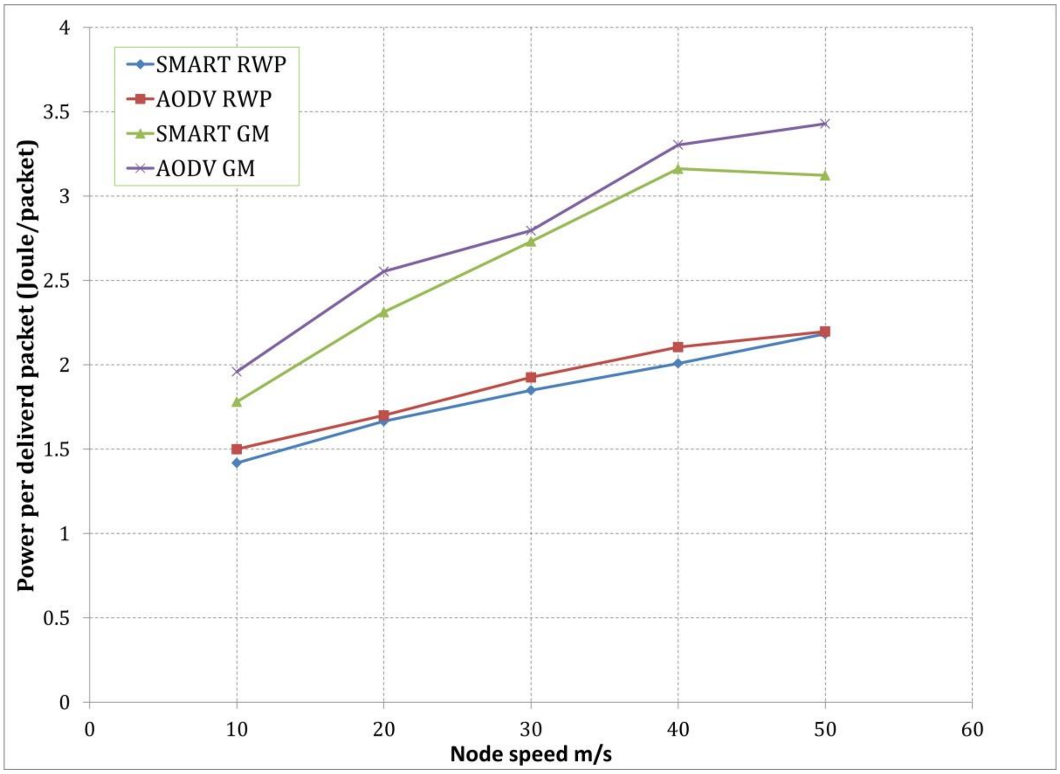


Figure 7.8 Average power consumed for delivering a packet under various speed values

The second scenario is used to study the effect of node density on the protocol performance. Figure 7.9 shows the effect of node density on network throughput. With lower node density of about 4 for 20 nodes in the network, the throughput is low. The throughput increases when the node density is increase and then it goes down as node density becomes about 19.6 for 100 nodes in the network. From the figure, it is clear that the proposed protocol overcome AODV protocol throughout all number of nodes.

Figure 7.10 and Figure 7.11 show the end to end delay for various numbers of nodes. When there are few nodes in the network, the number of route failure increases. For each route failure, the AODV route recovery requires more time which causes more delay because of broadcasting and queuing. Smart protocol usually do not buffer data packet, unless if the destination is unreachable. As the number of nodes increases in the network, the cost of broadcasting increases as well as the probability of collision increases.

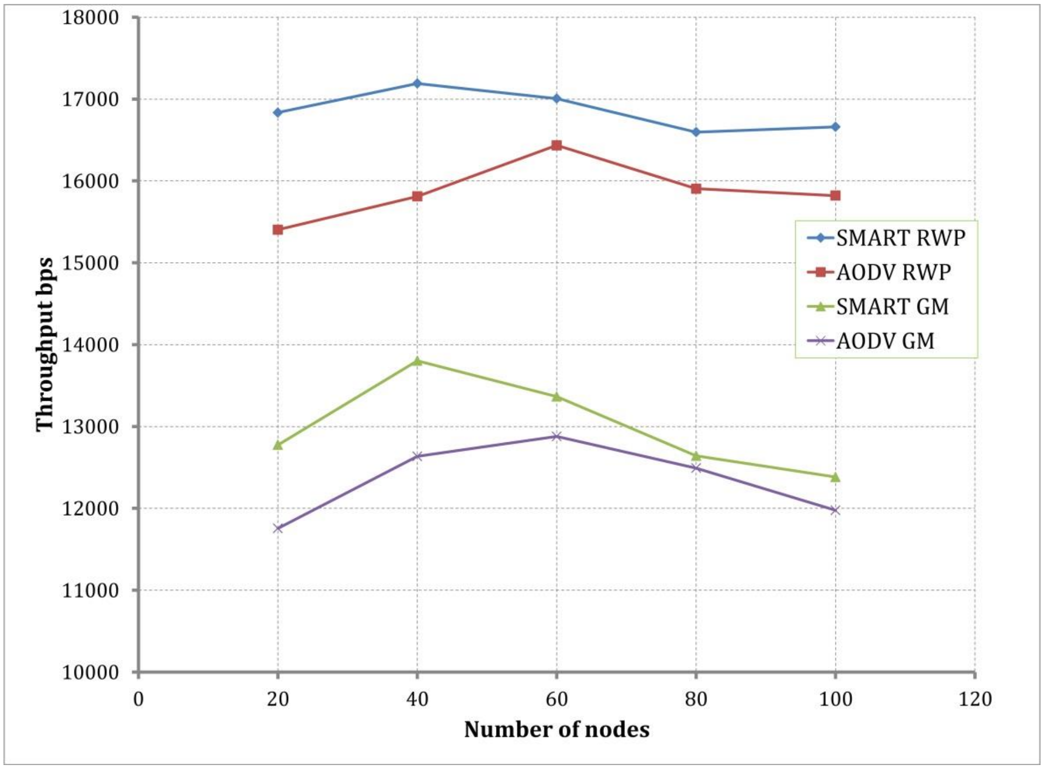


Figure 7.9 Network throughput under various numbers of nodes

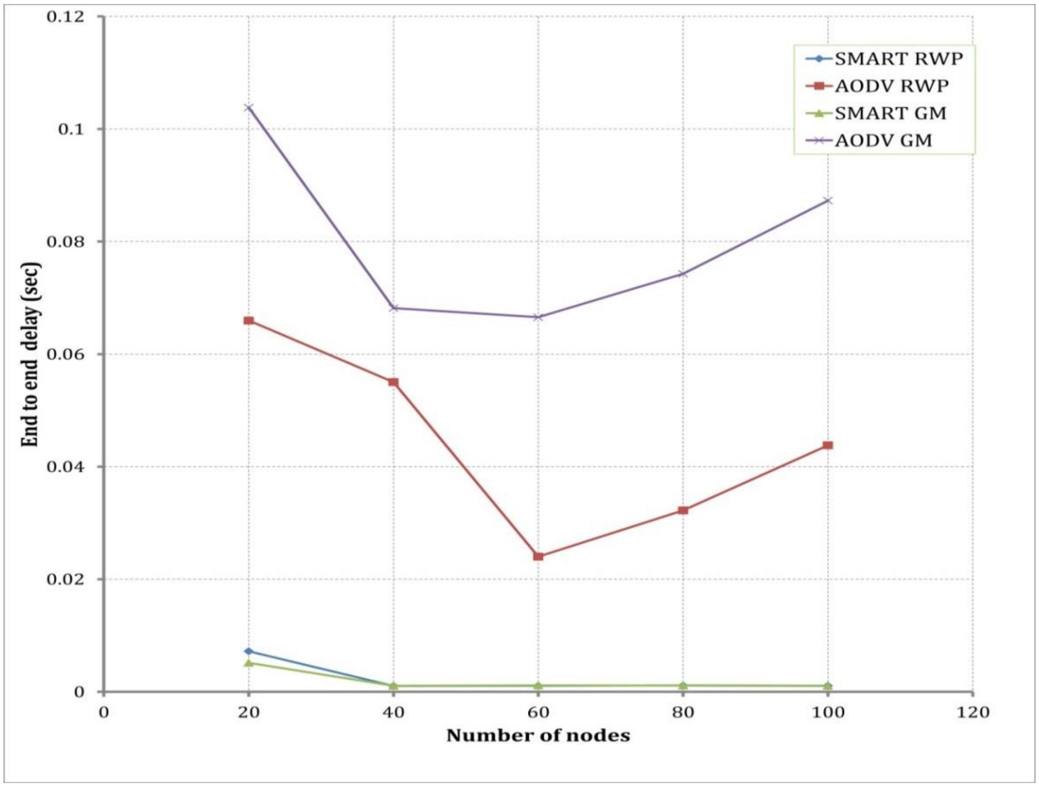


Figure 7.10 Network end to end delay under various numbers of nodes.

Smart protocol is more efficient in searching for better routes, and it minimizes the number of broadcasting in the network. It can be seen that the end to end delay at low node density is high. This occurs because at low node density the probability of network partitioning is high. The delay occurs usually as a consequence of queuing and broadcasting. Putting in mind, in the proposed protocol queuing occurs only at the original source when the destination is unreachable and all the nodes around the original source do not have a route to the destination. In other words all the nodes around the source have an altitude equal to one. This situation is rarely occurs unless the source node is isolated alone or with only few nodes. Limiting the punishment procedure to one node, described in section 7.3.5, will require many packets to set the altitude of all the nodes around the source to one. Moreover, the hello messages erode altitude of neighbour nodes which in a case if a node with an altitude less than one and if it broadcasts its hello messages to those neighbour that have been punished, it will result in decreasing their altitudes. This will also decrease the probability of bringing the altitude of all nodes around the source to one. More explanation has been given in previous chapters. Throughout all our simulation, it has been observed that the total number of broadcasting for any source in each simulation was always less or equal to three. This indicates that the probability of queuing is very low. In or point of view, we can see the effectiveness of the RFD algorithm as drops and smart packets are always moving and searching for a route rather than depending on a recovery mechanism as in AODV. At the same time as they search for the destination, the learning process is going on.

Finally, Figure 7.12 shows the throughput for 72 nodes network. The proposed protocol performs better than AODV protocol. Figure 7.13 shows the end to end delay of the protocols for the 72 node s network. Although the size of the network increases the route length, however the SMART protocol overrides the AODV protocol. The jitter in Figure 7.14 for both protocol show the same result where it is clear that SMART protocol outperforms the AODV protocol. finally the power consumed to deliver a packet in the SMART protocol is less than the power needed to deliver a packet in AODV protocol as shown in Figure 7.15.

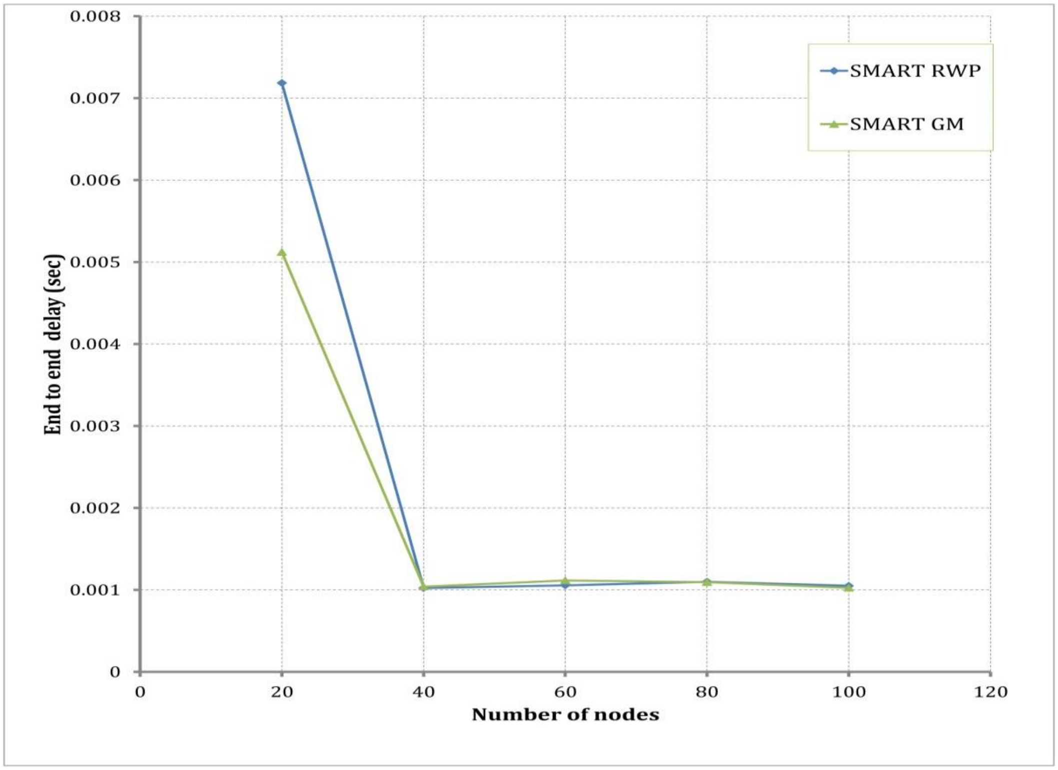


Figure 7.11 SMART protocol end to end delay under various numbers of nodes

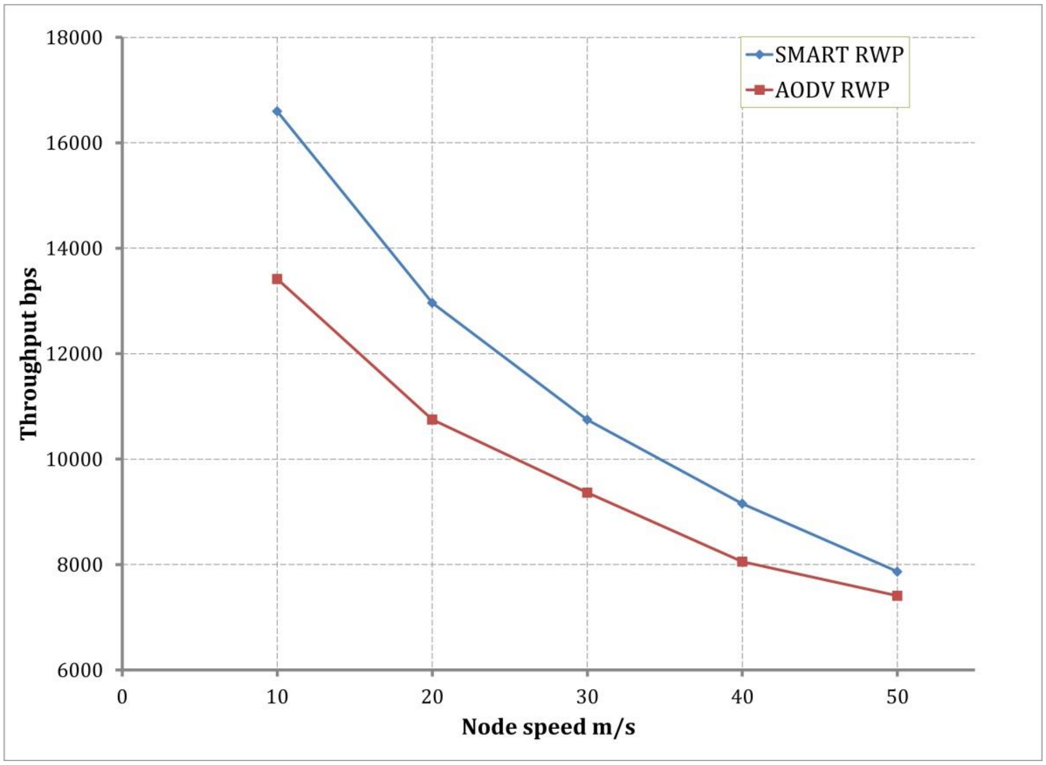


Figure 7.12 Average throughput under various speed values

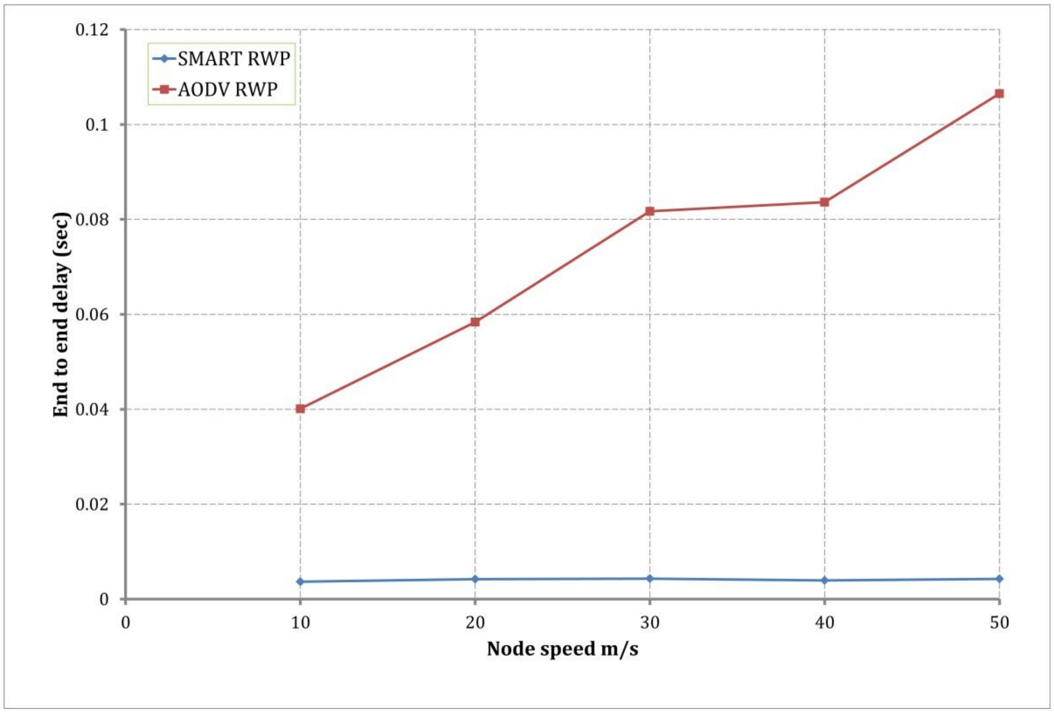


Figure 7.13 Average end to end delay under various speed values

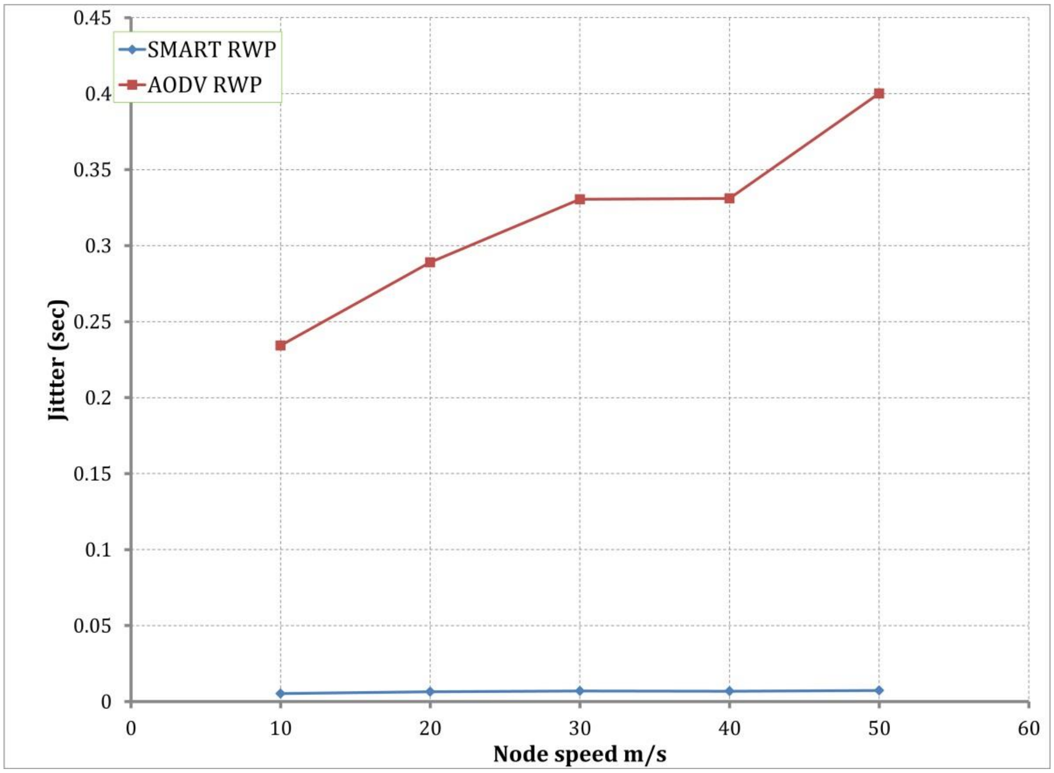


Figure 7.14 Average jitter under various speed values

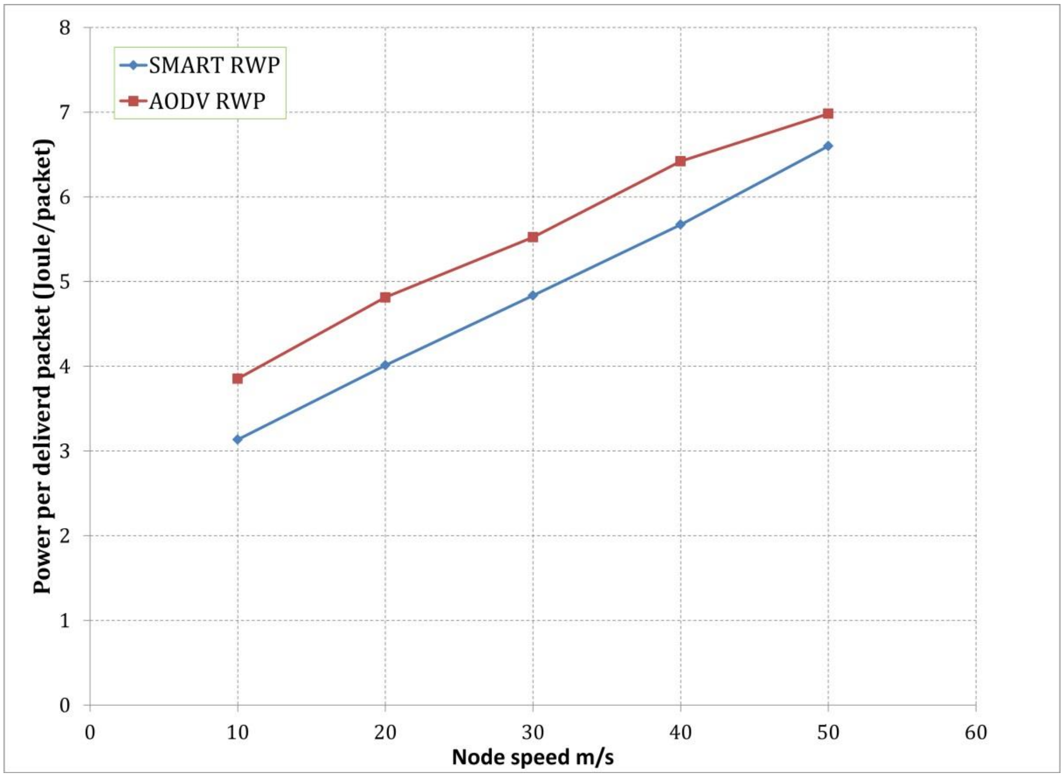


Figure 7.15 Average power consumed for delivering a packet under various speed values

7.5 Summary

In this chapter, smart packet swarm based routing protocol for mobile ad hoc networks we have been. The proposed protocol is power and congestion aware where packets are routed through less congested area of the network in order to save power. At the same time the protocol senses the remaining battery power of the nodes and prefers nodes with higher battery level. The protocol is based on River Formation Dynamics swarm algorithm. This algorithm is inspired from raindrops and how they form rivers, which in turn find best and shortest path to the sea.

The swarm algorithm adds a bifacial improvement to the protocol as it is based on one kind of agents that follow from source to destination. The protocol does not require feedback agents where the feedback is locally embedded in the RFD algorithm and the way that drops act. This feature makes it easy to convert data packets, which also move from source to destination, into smart packets. Smart data packet is a type of data packets that route itself though best instantaneous available route to the destination. As these packets move throughout the network they update routing information and contribute in the learning algorithm. Based on RFD algorithm, the extra amount of information added to the data is very small. Moreover the protocol adds this information to the IP header of the protocol which makes it flexible and compatible with the well know IP layered infrastructure.

Simulation results show an enhancement in the performance of the protocol. At the same time, the protocol shows less consumption of power for packet delivery.

References

[1] A. Boukerche, B. Turgut, N. Aydin, M.Z. Ahmad, L. Bölöni and D. Turgut, "'Routing protocols in ad hoc networks: A survey," *Computer Networks*, vol. 55, no. 13, 2011, pp. 3032-3080.

[2] E. Alotaibi and B. Mukherjee, "'A survey on routing algorithms for wireless Ad-Hoc and mesh networks," *Computer Networks*, vol. 56, no. 2, 2012, pp. 940-965.

[3] Jiageng Li, D. Cordes and Jingyuan Zhang, "'Power-aware routing protocols in ad hoc wireless networks," *Wireless Communications, IEEE*, vol. 12, no. 6, 2005, pp. 69-81.

[4] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum and L. Viennot, "'Optimized link state routing protocol for ad hoc networks," *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, 2001, pp. 62-68.

[5] C.E. Perkins, P. Bhagwat, C.E. Perkins and P. Bhagwat, "'Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers;" *SIGCOMM Comput.Commun.Rev.*, vol. 24, no. 4, 1994, pp. 234-244.

[6] D.B. Johnson, D.A. Maltz and J. Broch, "'DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," *Addison-Wesley*, 2001, pp. 139-172.

[7] C.E. Perkins and E.M. Royer, "'Ad-hoc on-demand distance vector routing," *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, 1999, pp. 90-100.

[8] P. Samar, M.R. Pearlman and Z.J. Haas, "'Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks," *IEEE/ACM Trans.Netw.*, vol. 12, no. 4, 2004, pp. 595-608.

[9] S. Singh, M. Woo and C.S. Raghavendra, "'Power-aware routing in mobile ad hoc networks," *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, 1998, pp. 181-190.

[10] K. Kaur and I.K. Aulakh, "'Power aware metrics and routing techniques in MANETs," *Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on*, 2011, pp. 369-373.

[11] C.E. Jones, K.M. Sivalingam, P. Agrawal and J.C. Chen, "'A Survey of Energy Efficient Network Protocols for Wireless Networks," *Wirel.Netw.*, vol. 7, no. 4, 2001, pp. 343-358.

[12] B. Panigrahi, A. Sharma and S. De, "'Interference aware power controlled forwarding for lifetime maximisation of wireless ad hoc networks," *Wireless Sensor Systems, IET*, vol. 2, no. 1, 2012, pp. 22-30.

[13] N.-. Wang, "'Power-aware dual-tree-based multicast routing protocol for mobile ad hoc networks," *Communications, IET*, vol. 6, no. 7, 2012, pp. 724-732.

[14] Wei Liu, Chi Zhang, Guoliang Yao and Yuguang Fang, "'DELAR: A Device-Energy-Load Aware Relaying Framework for Heterogeneous Mobile Ad Hoc Networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 8, 2011, pp. 1572-1584.

[15] Sun-Ho Lee, Eunjeong Choi and Dong-Ho Cho, "'Timer-based broadcasting for power-aware routing in power-controlled wireless ad hoc networks," *Communications Letters, IEEE*, vol. 9, no. 3, 2005, pp. 222-224.

[16] R. Bauza, J. Gozalvez and M. Sepulcre, "'Power-Aware Link Quality Estimation for Vehicular Communication Networks," *Communications Letters, IEEE*, vol. 17, no. 4, 2013, pp. 649-652.

[17] G.S. Sharvani, N.K. Cauvery and T.M. Rangaswamy, "'Different Types of Swarm Intelligence Algorithm for Routing," *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on*, 2009, pp. 604-609.

[18] I. Kassabalidis, M.A. El-Sharkawi, R.J. Marks II, P. Arabshahi and A.A. Gray, "'Swarm intelligence for routing in communication networks," *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 6, 2001, pp. 3613-3617 vol.6.

[19] G. Singh, N. Kumar and A. Kumar Verma, "'Ant colony algorithms in MANETs: A review," *Journal of Network and Computer Applications*, vol. 35, no. 6, 2012, pp. 1964-1972.

[20] Kwang Mong Sim and Weng Hong Sun, "'Ant colony optimization for routing and load-balancing: survey and new directions," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 33, no. 5, 2003, pp. 560-572.

[21] B. Kalaavathi, S. Madhavi, S. Vijayaragavan and K. Duraiswamy, "'Review of ant based routing protocols for MANET," *Computing, Communication and Networking, 2008. ICCCn 2008. International Conference on*, 2008, pp. 1-9.

[22] E. Gelenbe, Zhiguang Xu and E. Seref, "'Cognitive packet networks," *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on*, 1999, pp. 47-54.

[23] E. Gelenbe and R. Lent, "'Power-aware ad hoc cognitive packet networks," *Ad Hoc Networks*, vol. 2, no. 3, 2004, pp. 205-216.

[24] R. Lent, "'Smart packet-based selection of reliable paths in ad hoc networks," *Design of Reliable Communication Networks, 2005. (DRCN 2005). Proceedings.5th International Workshop on*, 2005, pp. 5 pp.

[25] P. Rabanal, I. Rodríguez and F. Rubio, "'Using River Formation Dynamics to Design Heuristic Algorithms," *Unconventional Computation Lecture Notes in Computer Science*, vol. 4618, 2007, pp. 163-177.

[26] Hifza Afaq , Sanjay Saini, "'On the Solutions to the Travelling Salesman Problem usingNature Inspired Computing Techniques," *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 4, July 2011, pp. 326 -334.

[27] P. Rabanal, I. Rodríguez and F. Rubio, "'Applying RFD to Construct Optimal Quality-Investment Trees," *J.UCS Journal of Universal Computer Science*, vol. 16, no. 14, May 2010, pp. 1882-1901.

[28] P. Rabanal, I. Rodríguez and F. Rubio, "'Applying River Formation Dynamics to the Steiner Tree Problem," *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*, 2010, pp. 704-711.

[29] P. Rabanal, I. Rodriguez and F. Rubio, "'A Formal Approach to Heuristically Test Restorable Systems,"in Theoretical Aspects of Computing - ICTAC 2009, vol. 5684. Martin Leucker and Carroll Morgan. , Springer Berlin Heidelberg, 2009, pp.292-306.

[30] P. Rabanal, I. Rodriguez and F. Rubio, "'Finding Minimum Spanning/Distances Trees by Using River Formation Dynamics,"in Ant Colony Optimization and Swarm Intelligence, vol. 5217. Marco Dorigo, et al. , Springer Berlin Heidelberg, 2008, pp.60-71.

[31] P. Rabanal and I. Rodriguez, "'Testing Restorable Systems by Using RFD,"in Bio-Inspired Systems: Computational and Ambient Intelligence, vol. 5517. Joan Cabestany, et al. , Springer Berlin Heidelberg, 2009, pp.351-358.

[32] OMNET++ community, OMNeT++ Network Simulation Framework, Accessed: July 2013, [Online] Available: http://www.omnetpp.org/.

[33] E.M. Royer, P.M. Melliar-Smith and L.E. Moser, "'An analysis of the optimum node density for ad hoc mobile networks," *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 3, 2001, pp. 857-861.

[34] C. Bettstetter, "'Smooth is better than sharp: a random mobility model for simulation of wireless networks," *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, 2001, pp. 19-27.

[35] L.M. Feeney and M. Nilsson, "'Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2001, pp. 1548-1557 vol.3.

[36] G.D. Caro, F. Ducatelle and L.M. Gambardella, "'AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications*, vol. 16, SEP 2005, pp. 443-455.

[37] S. Rajagopalan and C. Shen, "'ANSI: a swarm intelligence-based unicast routing protocol for hybrid ad hoc networks," *Journal of Systems Architecture: the EUROMICRO Journal*, vol. 52, no. 8, 2006, pp. 485-504.

Chapter Eight

RFDTrust: A Secure Ad Hoc Routing Protocol

8.1 Introduction

The distributed nature of mobile ad hoc network and the absolute control of each node over the data packet that passed through it, require a sustained benevolent behaviour from all nodes that are participating in the network in order for the network to deliver good performance. However different internal or external types of attacks could degrade network performance. Selfish behaviour is one kind of attacks where a node tries to maintain its resources by not sharing and cooperating with other nodes in the network. A number of secure routing protocols have been proposed to protect ad hoc network against different types of attack [1-3]. Cryptographic methods have been widely used to secure routing algorithm. Other types of secure routing protocols are based on reputation and trust. Trust is widely used human's life and is it an important concept that controls human's social interaction and behaviours. Trust based security protocol is better deal with malicious or compromised nodes than cryptographic based protocol. Many security protocols have been proposed based on reputation and trust system [4, 5].

Trust is based on observation and recording nodes' behaviours in a way identical to the trust system model in human beings where experiences are shared through networks of people and decisions are made based on these experiences. Using trust, a node can remember other nodes' behaviours, which provide good bases to prevent working with non-cooperative nodes. Moreover, trust provide a way to predict nodes' future behaviours and cooperate with nodes that support and enhance network performance.

This chapter introduces an RFD based trust routing protocol. The protocol detects black hole and grey hole attacks. Neighbour behaviour is evaluated based on node altitudes and the fact that in RFD protocols packet should mimic water drops and should always move toward the sea. An important feature of water drops is they move from high altitude toward lower altitude, therefore node that try to misbehave will be detected as not forwarding toward lower altitude. The proposed protocol is derived from previous RFD based protocols described in previous chapters. This is a multipath smart data packet protocol. This is hybrid protocol where it contains reactive and proactive technique to build routing and security tables.

8.2 Related works

In general, most ad hoc routing protocols are designed based on the full cooperation between all nodes that participate in the network, and the assumption that they are all trustworthy. Security is usually added to these protocols in order to secure data delivery and better protect them from different types of attacks.

Several researches focused on adding cryptographic security algorithm to ad hoc routing protocols. These works usually based on key management and encryption algorithms in order to prevent unauthorized node from joining the network [6].

A Secure On-Demand Routing Protocol for Ad Hoc Networks (Ariadne), introduced by Yih-Chun Hu and Adrian Perrig [7], is a secure extension to DSR routing protocol. As DSR is source routing protocol, the main objective of the security protocol is to ensure that the chain of nodes that the source node selected to deliver the message is note changed. It uses one of three ways of authentication. The first two types use shared secret keys. The keys are either shared between all pairs of nodes or between end communicating nodes combined with broadcast authentication. The third method of authentication is by using digital signatures. It adds one-way key chain to each message. The source node adds a Message Authentication Code (MAC) to the message. The receiver is assumed to have the last released key of the sender which should be distributed through key management services. Each intermediate node chains its hash key to the hash key in the message by repeatedly computing a one-way hash function. The hash keys authenticate the chain nodes and the originator of the message. A drawback of Ariadne is the increase of the message size. Moreover, the protocol suffers from the problems of key distribution, computation requirement. Time synchronization is also problems in the protocol.

Secure AODV (SAODV) is a security extension for AODV distance vector routing protocol [8]. The protocol uses digital signature to authenticate end to end nodes. As AODV uses number of hops as quality metric, the protocol also tries to protect the hop count in the RREQ and RREP messages from misbehaved decrement. The protocol also uses hash chain to secure hop count in the messages. SAODV have many drawbacks, the digital signature enlarges the size of messages, and the cryptographic operations are resource consuming.

Other cryptographic techniques also have been developed like Secure Link State Routing Protocol (SLSP) [9], Secure Efficient Distance vector routing for mobile wireless ad hoc networks (SEAD) [10], Adaptive SAODV (ASAODV) [11], and others [8, 12-14]

In spite of the effectiveness of cryptographic techniques and their high degree of security, they introduce different types of problems to ad hoc networks. Cryptographic techniques are computationally very expensive especially as nodes in ad hoc network are resource limited. The length of the encryption keys as well as the exchange and verifying of the keys introduce a heavy traffic load to the network. The mobility and other characteristics of MANET like distributed nature and open access nature makes it difficult to build a centralized or distributed trusted third party for cryptographic techniques. With cryptographic techniques, it is difficult for a node to reroute a message dynamically and forward it to other node according to the change in network security. Moreover there is no clear way of sharing and distributing security information and the method of convincing other nodes to discard non-trust nodes in the network. Finally, traditional cryptographic techniques are not effective against malicious internal attacks especially if a node has been compromised.

Trust is an alternative security technique. It is based on reaction to other nodes' behaviours. Blaze, Feigenbaum, and Lacy identified distributed trust management as an important component of security [15]. The authors describe the basic of distributed trust management based on a simple language for specifying trusted actions and trust relationships.

A selection of trust techniques have been added to existing ad hoc routing protocols. In [16] a trust model is introduced for DSR routing protocol. As DSR is a source routing protocol, trust is computed regarding all intermediate nodes between the source and the destination that are agreed on by source node to deliver the message. The algorithm is based on replying the source node with acknowledgment from the destination to inform the sender of successful reception. Whenever a destination replies by an acknowledgement to the sender, all intermediate nodes trust value are increased, otherwise if no acknowledgement received, the trust values of all intermediate nodes are decreased. The protocol cannot identify which node in the chain from the source to the destination has discarded the message. Trust and reputation system for DSR routing protocol also addressed in [17].

In [18] the author proposed watchdog and path rating technique in order to control forwarding mechanism. The protocol intended to work with the DSR protocol. The watchdog is based on monitoring next node using promiscuous mode and detecting if the next node has forwarded the packet to the second forwarding node. Accordingly, the trust value is computed and shared. However, the node will not announce directly any misbehaviour. If a node repeatedly misbehaved then it will reported to other nodes as misbehaving node. A *pathrater* is used to rate the path that will be selected according to trust values. Initially all node are set to 0.5 (neutral value). This value will be incremented by 0.01 every 200 ms if the node is in an active path and a message has been delivered to destination. The value is decremented by 0.05 if the node is misbehaving. The best route is selected according to average trust values of all nodes within the route.

In [19] a Simple Trust model for AODV protocol is introduced (ST-AODV). Trust value is calculated based on positive acknowledgements where a node monitors the channel using promiscuous mode to detect if the next node have forwarded the packet to the desired destination. As the base routing protocol, AODV, is single path protocol, the resulting trust protocol is a single path protocol. Based on (ST-AODV) the authors in [20] introduced a multipath trust routing protocol (AOTDV).

COllaborative REputation mechanism to enforce node cooperation in mobile ad hoc networks (CORE) introduced in [21]. CORE uses both first-hand information, collected from direct observation, and second-hand information, information shared from other nodes. The reputation is divided into three different types. Subjective reputation which is computed by direct observation, indirect reputation from other nodes, and functional reputation which is the behaviour of a node during specific task. CORE addresses selfish behaviour problem. Two entities are used in CORE, a requestor and a provider. The requestor is a network entity that asks for reputation value. The provider is the entity that evaluates the reputation value and returns it to the requestor. The reputation values are combined using different weights in order to evaluate recent reputation value of a node. Old observations are weighted higher than newer ones. Both interaction and observation are treated similarly.

CONFIDANT is another type of trust based routing protocol [22] introduced by Buchegger and Le Boudec. Like CORE, CONFIDANT uses both first-hand and second-hand information for updating reputation values. CONFIDANT is based on DSR protocol. Watchdog is used to monitor other nodes. A copy of sent message is queued and compared with forwarded message by the next node to determine any possible changes in the message. Whenever a malicious node detected, an alarm is send to other nodes. Alarm table is used to recorded alarms from other nodes. Another table called friend list is used which contain the addresses of nodes that should be informed if a malicious node is detected. Unlike CORE, new observations have higher weights than old ones. Personal observations, first-hand information, have higher weights than neighbour observations, second-hand information. An improve version of CONFIDANT is presented in [23] and called "A Robust Reputation System" (RRS). RRS introduced Bayesian framework with Beta distribution to update reputation values. A deviation test is used on the received second-hand information to check its deviation from node itself opinion.

Most of trust management systems are based on passive overhearing of sent packets. Statistical approaches are usually embedded in these systems to calculate the trust value. These techniques usually requires a lot of time in order to increase the probability of overhearing to qualified number of packets to compute the trust value. Moreover, these techniques cannot guarantee the delivery of the packet to next node. These techniques can only determine whether the node has transmitted the data packet or not. Several approaches have tried to address this problem by using 2ACK scheme [24-27]. In 2ACK scheme, both first and second forwarding nodes should return acknowledgment to the sender. For example, three nodes (A, B, and C), node A forwards the packet to node B, than node B forwards the packet to node C. In 2ACK scheme, node C should return a specific type of acknowledgment to node A through node B. This will indicate that the packet has correctly forwarded by node B. However, the use of 2ACK requires special kind of hash function and cryptographic technique to ensure that the middle node is not returning fake acknowledgments.

Cluster based trust management techniques have been widely used in Wireless Sensor Networks (WSNs) [28-31]. The architecture of WSN is based on groups of sensor nodes where each group had a cluster head. These cluster heads can communicate with each other and form another sub-network. Data packets are routed to base station through the cluster heads sub-network. These techniques usually try to minimize the communication overhead produced by trust algorithm. In general, each cluster head is responsible of computing the trust value of its group of sensor nodes. The cluster heads then share these values between them.

Ant-based Adaptive Trust Evidence Distribution (ABED) [32] is a distributed system of trust certificates. In ABED, each node require to have a certificate and uses cryptographic technique (public and private key) to sign a certificate. Ant in ABED is used to explore and collect nodes' certificates public keys. Authentication is assumed to be carried out prior to the setting up of the network. AntTrust [33] is another ant based routing protocol where ants are used for both route discovery and computing trust values of the nodes. Ant also used to monitor neighbours' behaviours.

In [34] a Light-weight trust-based routing protocol for mobile ad hoc networks is presented. The proposed method focuses on only one matric to compute the trust level, the packet forwarding behaviour. Node trust value is computed as weighted sum of self-observation and neighbours' observation, where neighbours' trust values are averaged on number of neighbours. The route final trust value is computed as the product of all nodes trust value in the route. The trust value of each node is estimated statistically from number of packets that has been forwarded to the overall number of packet that arrived to the node which should be forwarded.

In [35] the convergence time of reputation and trust system is addressed. The authors address the high dynamic nature of mobile ad hoc network by selecting nodes with higher centrality and high reputation as preferred sources for indirect reputation. Nodes are classified into different zones depending on the reputation information and the information on the centrality of the nodes.

In [36] the authors incorporate multiple values of trust and confidence in a vector auto regression based trust model. These values represent different behaviours of a node and are used to compute a final node trust value.

.

8.3 Attack model

MANETs have many characteristic that differentiate them from other networks. MANET is self-organization network where the collaboration of its nodes is crucial principle in order to deliver packets in the network. Thus, MANETs can work well if the nodes work in cooperative way. However, the characteristics of the network, like wireless environment, mobility, limited resources, and sometimes open access (depending on the application) introduce high threats to these networks. For example, the wireless environment makes it easy for eavesdropping attacks to be carried out against these networks. Due to the mobility and the wireless environment, and open access where node can join the network at any time, MANETs have been a target for various types of attacks which target different layers of their protocol [12, 37]. Accordingly, these characteristics could open the way for numerous types of security attacks against network layer and routing protocols [38, 39]. Different types of attacks have been discussed in chapter two.

A serious type of attack is performed by malicious nodes where nodes are compromised. These nodes can start denial of service attacks against the network. Cryptographic approach usually cannot detect this kind of attack. A trust based model should be built to address these attacks.

Different types of attacks can be initiated against RFD based routing protocols. Although these types of attack fall under the same categories of MANETs attacks, they use different approaches. A malicious node can start sink hole attack by advertising a fake altitude to convince other nodes that it has the best path to a destination. It should be noted that the same attack in different protocol may be implemented in different way, however having the similar effect on the network. For example, in AODV to implement sink hole attack, the attacker may need to send fake RREP messages to deceive a source node that it has a valid shortest path to a destination.

RFD based protocols are basically consists of two phases, route setup and route maintenance. Accordingly attacks against RFD based protocols could be categorized into two groups.

1- Attacks against route setup procedures.

These kinds of attacks try to disrupt the route creation procedure. In chapters five and six, two procedure for route setup have been proposed. In ant-drop procedure, discussed in chapter five, the attacker may discard the route request message, change time stamp, or generate fake backward ant-drop message. In pure RFD based protocol, described in chapter six, the attacker may also discard the route request message, change its altitude toward the source, and generate fake route acknowledgement.

2- Attacks against route maintenance procedure.

Route maintenance in RFD based protocol is carried out by drops and data packets. Both drops and data packets are depending on altitudes in their moving and searching for better paths. The first approach of any attack is to alter altitude tables. An attacker could discard all messages forwarded to it (black hole attack) or it may selectively discard some messages that have been forwarded to it (grey hole attack).

8.4 Proposed security protocol

8.4.1 General overview and explanation

Trust approaches are based on observation. Nodes in the network monitor other nodes behaviour. Two kinds of information are usually used in trust system, first-hand and second hand information. However there are main differences between the proposed approach and others due to the use of RFD algorithm.

The trust value reviewed in related works section is calculated by the observer node as a ratio between the total number of packets already forwarded by a node to the total number of packets that delivered to the node which should be forwarded. The value of trust is not exactly confidential as there is always a probability of miscounting the number of packets [34].

Figure 8.1 shows an example. Node A sends packets to node D through nod B and C. Node A monitors the behaviour of node B by overhearing passive acknowledgment of forwarded packets from B to C. Node A calculates trust value about node B by calculating the number of packet that have been forwarded from node B to node C to the total number of packets that A send to node B. However, node A cannot be certain about exact number of packets that have been forwarded by node B due to sheared media and possibility of collision while overhearing the channel. Moreover, packets could be dropped because the buffering queue in MAC layer is full. The number of packet that could be overheard by node A depends mainly on the traffic in the network. The higher traffic in the network, the higher probability for collisions. The trust algorithms must consider this situation and usually all of the reviewed protocols give certain level for uncertainty.

If node B is compromised, it can intentionally drop small ratio of the packets which affect the traffic but has less effect on the trustworthiness. On the other hand, if node B is a legitimate node, its trust value at node A will not get to 100% because of the collisions as well. Most protocols require long initialization time to collect reasonable information about other nodes.

Another problem is how to know which group of nodes should node B forward the packet to? In the example, node B should forward the packet to node C and should not forward it to node E. Node B could forward the packet to node E and node A will observe that its packet has been forwarded, but node E is not the best route. This attack is called silent route changes. This is why most reputation and trust security approaches tend to use DSR as a base routing protocol. DSR is a source routing protocol and the header contains the full path that the packet should use in order to reach the destination. By using DSR, trust algorithms are able to detect that a packet has diverted from its path. However, source routing protocol introduces many problems. Packet forwarding techniques could not be implemented in source routing protocol. The route is fixed and the source node set the route. If the quality of the route changed, only the source node can select new route. Link information should be returned to the source in order to select another route.

RFDTrust overcome these problems and can detect misbehaviour in different way. Beside the packet rate ratio, RFDTrust get benefit from the RFD algorithm. A packet in RFDTrust should always move from higher altitude to lower altitude. If a node sent a message to a node with higher altitude that indicates it is a malicious node. Returning to Figure 8.1, if node B is malicious node and start an attack by forwarding the packet to node E.

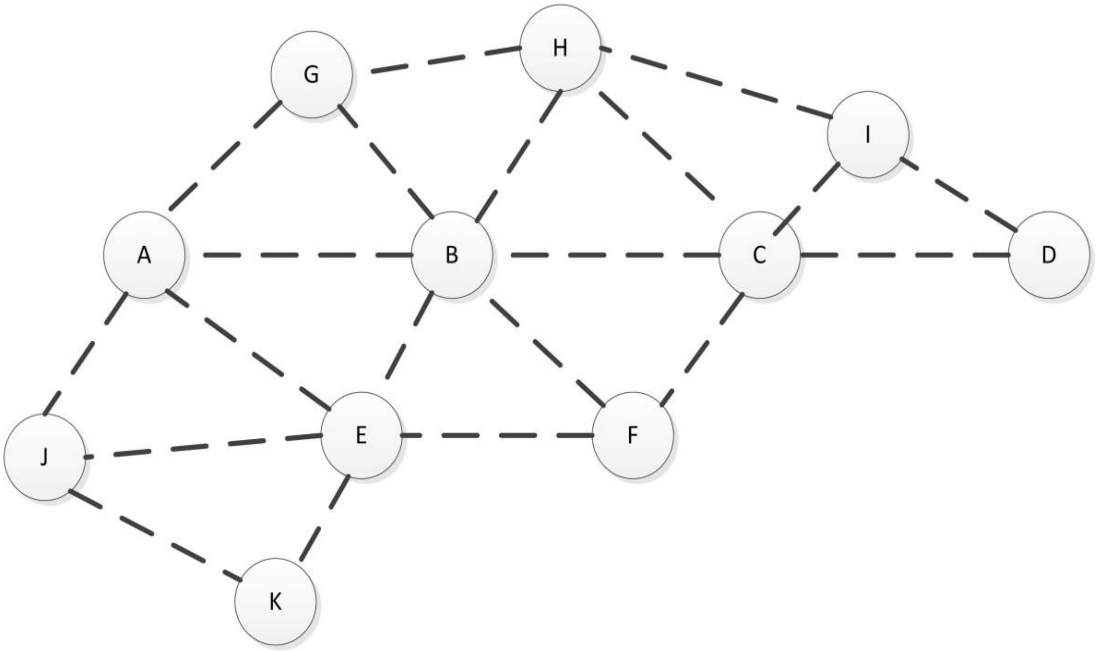


Figure 8.1 Example network

Node E has higher altitude than B itself. Node A is a neighbour to E and may have its altitude. Once the packet is forwarded, node B will be marked as malicious node. At the same time, node F is a neighbour for both B and E. It will also mark B as malicious node. Moreover, if node E itself is not a one of node B cooperative malicious nodes group, it will also mark node B as malicious node. Node A also receives second hand information about other nodes that are not in its range of transmission through its neighbours. RFDTrust stores these misbehaviours in separate record and packet forwarding rate in another record. Distrust value is stored and is affected by misbehaviour. Higher weight is given to the value of distrust when selecting next forwarding node.

Four different values are used to record trust. More about reputation and trust metrics will be explained in section 8.5.

Distrust: represents the rating of sending packets to higher altitudes.

Trust_forwarding: the rate of forwarding packets.

Trust_goodness: the rate of forwarding packet to lower altitude nodes.

Reputation: reputation of the node in others perspective.

It should be noted that a node cannot send to nodes with higher altitude as long as there are some neighbours nodes around the node itself. It is similar to the situation when a monitoring camera is mounted is a place, bad people will think twice before committing a crime. Even if the camera is a dummy one, as long as no one can tell, it adds security to that place. The same is true with ad hoc network; the malicious node cannot tell if any of its neighbours is in range with the next intended forwarding node. This will force the node to behave well or otherwise it may be detected.

The use of altitude gives many advantages to the protocol. Sink hole attack is detected easily as explained before. Black hole attack is detected by the packet forwarding ratio as well as time needed to forward a message that is computed in the smart data packet algorithm. Moreover, trust_goodness has been added to this system. If node B is a black hole, node A and E can compute the ratio of packet forwarding by node B. Node F is not in the transmission range of A and cannot cooperate in this operation. However node F does not see any forwarding from node B to node C, node Cs altitude is lower than A. Node F second hand information will show that node B has no trust_goodness and this will increase the probability of rejecting node B.

The scenario of black hole attack that performed by node C could be different. As this is a distance vector routing the attacker could forward the packet to a dummy address pretending it sends the packets. Node A and E will record a good data forwarding rate for node B as they observe that node B is forwarding the packets. However, as node B actually does not forward the packets; no node around it will give it high trust_goodness value. As a result when this information reaches node A as second hand information, it will change the probability of selecting node B. Moreover if node B was legitimate node and for any reason node A and E could not overhear all forwarded packets by B, node F will report a good trust toward node B as it will monitor its good behaviour in forwarding to node C. Grey hole attacks are detected using same algorithm.

Part of selective forwarding attacks are blocked using previous technique as node cannot forward a packet to higher altitude. However, an attacking node like node B can select an altitude close to the lowest altitude of its neighbours. The selected altitude could be equal or higher to lowest node altitude. In this case the node is actually part of the best route and it helps forwarding the packets to best next forwarding node, so this attack will be non-feasible.

If the attacker selects an altitude higher or equal to node H, in order to forward packets to node H rather than C, this will also be detected. Node H altitude will decrease as it is forwarding packets. Node G’s altitude also decreases due to distributed learning. Putting in mind that the protocol is multipath protocol and node G will also act as forwarding node for node A. As a result, node B should forward all of the packets or else node G will become best forwarding node for A.

Node B could forward to node G, this means that node G has less altitude than B. This means that node G is best forwarding node for A.

If node B intended to forward packets to node E, its altitude should be higher than the altitude of node E. in this case node B will be less desired to be selected as forwarding node.

It should be noted that a node could forward a packet to next node that is 5 % higher than the difference between the previous sender and the node itself according to smart data packet protocol. In this case monitoring nodes consider it legitimate forwarding if the difference is not higher than 5% and the altitude of the node increased as there will be more sedimentation in this case. Finally if a node suddenly changes its altitude many times it will be considered as malicious node.

8.4.2 Some features of the proposed protocol

- ***Reputation and trust parameters:***

Four parameters are used to track the behaviour of other nodes, Distrust, Trust_forwarding, Trust_goodness, and Reputation. Different weights are given to these parameters in order to forward packets. Node can directly capture other nodes' behaviours or indirectly influenced by the other nodes opinions. All nodes start up with neutral trust level and no pre-initial trust relationship exist between the nodes. Trust is accumulated through direct observation as well as indirect information exchange.

- ***Parameters representation:***

Security parameters are represented in similar way to altitudes. Any parameter can have a value between zero and one. Second hand parameters affect trust's value with a ratio depending on how much a node trusts the sender. Whenever a node discovers a new node it sets its new parameters to neutral value.

- ***Forgetting:***

Nodes' behaviours are not maintained forever. Security parameters are regularly updated and all parameter are tending to move toward their initial value. This gives more weights to recent activities. Moreover, it will allow forgetting past behaviours especially if the information gathered were uncertain and this gives a second chance for a node to behave well. The source of uncertainty may come from non-reliable link, and false reports from other nodes.

- ***Packet forwarding:***

RFDTrust is a multi-path smart data packet routing protocol. Packets are forwarded depending on many parameters. Routing metrics as well as first-hand information have higher priority than second-hand information.

8.4.3 Assumptions

- ***Bidirectional link*:**

The protocol is based on observation and monitoring in order to compute many parameters. In order for activities like passive acknowledgment overhearing to complete successfully, the link between nodes should be bidirectional. The transmission power of all nodes is equal. Omnidirectional antenna is used.

- ***Unique identity*:**

Any node within the network is assumed to have one unique identity. A node cannot change its identity and the identity is assumed to remain the same during the entire network lifetime. It is assumed that a node within the network cannot impersonate another node and thus it cannot damage others reputations or get benefits from other reputation. The proposed protocol does not address Sybil attack [40]. Different approaches have been introduced to defend against Sybil attacks [41-46]. It is assumed that the network has a method to ensure that a node has one unique identity throughout whole network lifetime.

8.5 Protocol models

RFDTrust is built over smart data protocol which uses RFD as base algorithm for the routing protocol. The main tasks of RFDTrust protocol is to monitor other nodes, gather behaviour information, and combine this information with the smart routing protocol in order to find best way to deliver data packets.

New table is added to the routing protocol in order to store trust information. Table 8-1 shows trust table. No_i represents node i in the network and n is total number discovered nodes by the node. TD_i represents distrust value, TF_i represents Trust_forwarding value. TG_i represents Trust_goodness value, and RP_i represents reputation value.

Table 8-1 RFDTrust table

Node	Distrust	Trust_forwarding	Trust_goodness	Reputation
No_1	TD_1	TF_1	TG_1	RP_1
No_2	TD_2	TF_2	TG_2	RP_2
No_n	TD_n	TF_n	TG_n	RP_n

8.5.1 Initialization

The protocol is started with empty trust table. Each time a new neighbour is discovered or information is shared by a neighbour regarding another new node, an entry to the new node is created and added to the table. The address of the node is stored and all parameters set to initial values. In case if the new node information comes though sharing information, the entry is created than the processing is done. Initial value is equal to zero except for reputation parameter is 0.5 .

8.5.2 Information gathering

Nodes are using monitoring technique in order to gather first-hand information. Using this technique, a node can detect normal or malicious behaviours. Three type of information is gathered using monitoring, distrust, trust_forwarding, and trust_goodness.

The first kind of observation is related to the fact that packet should be sent to lower altitudes. Any misbehaving node that sends to higher altitude will be detected and the value of distrust related to that node will be changed accordingly.

The second type of monitoring is done by observing the packet forwarding ratio of other nodes. The change will affect the value of trust_forwarding.

Finally a node may send a packet to another node that has lower altitude but the observer has no knowledge of the previous sender of the packet. In this case, the trust_goodness credit for that node increases.

8.5.3 Information sharing

Second hand information is shared periodically throughout the network. Each node periodically attaches its trust value to the end of hello messages. The value of trust is calculated as single trust value using a weighed sum of its three first-hand trust values in trust table plus the reputation [29, 47]

$$trust(j) = w_1(1 - TD_j\ ) + w_2(TF_j) + w_3(TG_j) + w_4(RP_j) \qquad 8.1$$

where w_1, $w_{2,}$ w_3, and w_4 are weights associated with these four trust components with $w_1 + w_2 + w_3 + w_4 = 1$. w_1 is set to 0.4 and the other three weight values are set to 0.2. Distrust and trust_goodness are more reliable than trust_forwarding. Moreover, the normal operation is to send a packet to lower altitude, while it is misbehaviour when sending to higher altitude. For this reason the weight of distrust is higher.

Accordingly, the initial trust toward a node is equal to 0.5 which represent neutral trust value according to following

Initial Trust =0.4*(1-0)+0.2*0+0.2*0+0.2*0.5

=0.5

8.5.4 Information modelling

Different methods have been proposed to model behaviour information [4, 47, 48]. In general the ratio of number of successful forwarding packets to the total number of packets is widely used [34]. In [47] window forwarding ratio is used. However this will reset the window each time or large first in first out queue should be used to store the time events throughout the window.

The proposed approach is different. First of all, the protocol deals with levels of altitudes. Secondly, we want to give new event higher influence than the usual one, and give the nodes the opportunity to build reputation quickly and want to damp them in order not to get to maximum reputation level easily. We use an exponential like function to model node behaviours into trust values.

Each time a node is observed sending a packet to higher altitude, the value of distrust is recomputed as follows

$TD_j=TD_j+0.1*(1-TD_j)$ 8.2

Knowing that TD starts from zero, this function will quickly detect misbehaviour actions and the function rises quickly. Figure 8.2 shows the function with respect to number of events.

The same function is used for trust_goodness parameters as follows

$TG_j=TG_j+0.1*(1-TG_j)$ 8.3

For the ratio of packet forwarding, two functions are used. For the successful forwarding

$TF_j=TF_j+0.1*(1-TF_j)$ 8.4

which increases the altitude.

For unsuccessful forwarding, the following invers function is used

$TF_j=TF_j-0.1*(TF_j)$ 8.5

It should be noted that for packet forwarding, the node stores a copy of sent message in a buffer. If it could not overhear the event of forwarding the packet from the other node within twice the time needed to forward a message, in its altitude table, it deletes the copy of the packet and marks the event as failure forwarding.

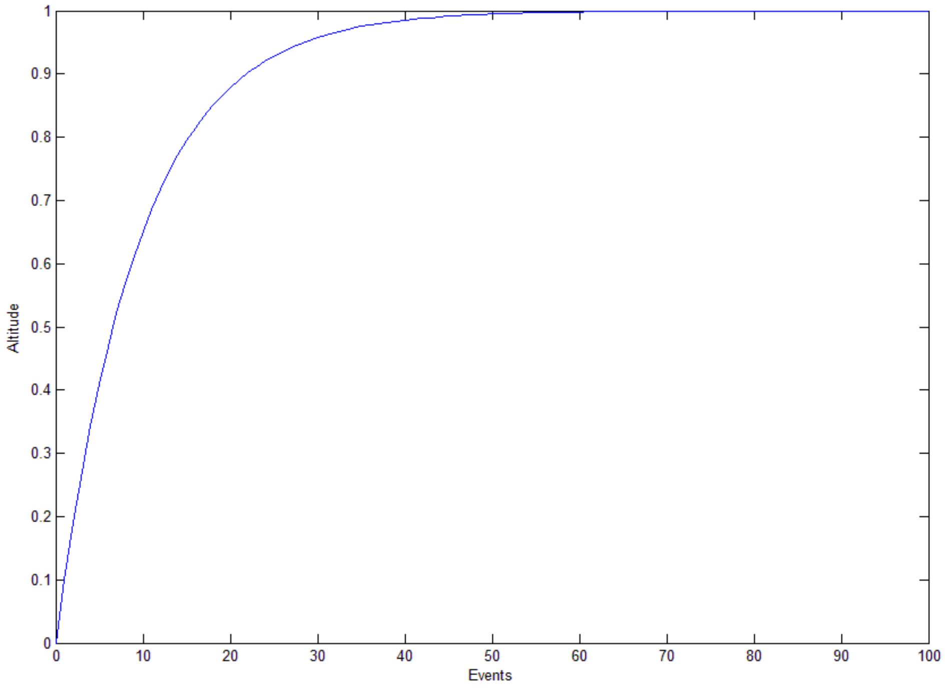


Figure 8.2 Relation between events and trust parameters altitude

An important situation should be addressed here, the freshness of the altitude table. Occasionally, a node monitors two nodes and observes that a node has sent a packet to a second node which has a higher altitude, but the record in the observer shows that the last update of altitude value for the second node is not very fresh. If last update is more than 0.5 second old then the node will wait until the second node sends a hello message or forwards the packet before considering the event as misbehaviour.

Another parameter that requires modelling is the reputation parameter. This is updated by second hand information. The amount of update is proportional to the ratio of belief in the sender. If the sender is not trusted (sender node trust value less than neutral trust value) the information is discarded. Otherwise, the update will be proportional as following

$$Rep_j = \begin{cases} (RP_{sender} - .5) * 2 * recRP_j & if\ RP_{sender} \geq 0.5 \\ RP_j & if\ RP_{sender} < 0.5 \end{cases} \qquad 8.6$$

where RP_{sender} is the sender node reputation, $recRP_j$ is the reputation of node j according to the sender node. RP_j is the reputation of node j in the trust table of the receiver. $Rep_j$is the amount of update to node j reputation.

Then an updating factor $\propto$ $(set\ to\ 0.9)$ is used to update the reputation value of node j

$$RP_j = \propto * RP_j + (1 - \propto) * Rep_j \qquad 8.7$$

Finally all these parameter are updated regularly by subtracting a constant value (0.01) from them which act as forgetting factor. This is introduces to overcome fault observations. (if RP is less than 0.5 the value will be added to RP)

8.5.5 Decision making

Routing in RFDTrust is based on RFD algorithm and is similar to smart data packet protocol. However, routing decisions are made based on altitudes as well as trust values. First, all nodes that have a positive distrust value are rejected from the selection. If no node with zero distrust is found then all node will enter the selection process. Routing then is done according to the following

$$\mathrm{P_n(j,k)_d} = \begin{cases} \frac{\mathrm{decreasingGradient_d(j,k)}}{\sum_{\mathrm{l \in V_{\mathit{n}}(j)}} \mathrm{decreasingGradient_d(j,l)}} & \mathrm{if\ k \in\ V_n(j)} \\ 0 & \mathrm{if\ k \notin\ V_n(j)} \end{cases} \qquad 8.8$$

$$\mathrm{decreasingGradient_d(j,k)} =$$

$$\frac{(\mathrm{altitude}_d(j)-\mathrm{altitude}_d(k))*\mathrm{trust}(k)}{T(k)} \qquad 8.9$$

where $P_n(j,k)_d$ is the probability of packet *n* at node *j* with destination *d* to select next node *k*. *T(k)* is the average time that node k needs to send a packet. *altitude$_d$(j)* is altitude of the node j toward destination d. $trust(k)$ is the trust value of node k (as in section 8.5.3) . *V(j)* is set of neighbour node of node *j*.

8.6 Simulation and Results

To evaluate the effectiveness of RFDTrust routing protocol under attack conditions, different simulations have been performed using the OMNet++ simulator. The RFDTrust performance is compered to smart data packet protocol which introduced in chapter five.

8.6.1 Simulation environments

Three scenarios have been used to test the proposed protocol. The first scenario is used to test the performance of the protocol for various number of attack nodes. The second scenario is used to test the performance of the protocol for various nodes speed. While in the third scenario the effect of node density is analysed.

The first scenario consists of 50 nodes randomly distributed in 600 * 840 m^2. Node density is chosen to be close to 7 in order to have good connectivity. Number of malicious nodes is changed from 0 to 20 nodes. The number of source-destination pairs is set to 10 and 20, nodes speed is set to 10 m/s.

In the second scenario the effect of nodes mobility on network performance is analysed. The speed of nodes is varied from 10m/s up to 50m/s. The network size and number of nodes remains as in first scenario. The number of malicious node is set to 10 and 20 nodes. The number of sources destination pairs is 20,

The third scenario is used to test the protocol performance under variable nodes densities. The number of nodes varied from 40 to 100 nodes using the same simulation area of the previous scenarios. At the same time, nodes speed is set to 10 m/s. The number of sources destination pairs is 20. The number of malicious nodes is set to 10 and 20 nodes.

The medium access control protocol is the IEEE 802.11 DCF. Packet size is 512 bytes. Each node generates a packet every 0.2 second. The network remains silent in the first second.

The nodes start sending data at third second and keep sending until the end of simulation, which gives one second for some hello packet to be generated before starting data session. Data traffic is generated using constant bit rate (CBR) UDP traffic sources. The random waypoint (RWP) mobility model is used to test the performance of the protocols [49]. Each node has a radio propagation range of 150m and channel capacity of 54 Mb/s. Hello time intervals is set to 0.5 second. Pause time is set randomly between 0.1 and 1 second. Simulation time was set to 200 seconds. The simulations are repeated for twenty times with different seeds.

8.6.2 Attack Pattern

Malicious nodes simulate the following types of active attacks:

Black hole attack: In this attack, the malicious node dumps all data packets, which it is supposed to forward.

Grey hole attack: The grey hole attack is similar to the black hole attack, however, the malicious node selectively forwards data packets at random intervals. Uniform random generator is used and when the value is greater than 0.5 the node will dump the packet.

Sink hole attack: node advertise as they have the best available route to the destination. This is done by advertising altitude values that is less then all neighbour nodes. The sink hole attack is combined with both black hole and grey hole attacks.

Throughout all simulation the average number of different types of attacks is kept equal. In other word, equal number of grey hole and black hole are presented in each test.

8.6.3 Metrics

To evaluate the proposed mechanism, the following metrics have been used

5- Throughput: is the measure of the total number of successful delivered data bits over simulations time for a specific node, averaged over the number of source-destination pairs.

6- End to end delay: is the measure of average delay of data packets. This is the time from sending the packet from the application layer at the source node to the time that the packet arrives to the application layer at the destination node, averaged over the number of source-destination pairs.

7- Jitter: is the variation of packet delay which is averaged over the number of source-destination pairs, averaged over the number of source-destination pairs.

8- Routing overhead is the total number of control packets sent divided by the number of data packets delivered successfully.

8.6.4 Results

Figure 8.3 highlight the effectiveness of the proposed trust protocol. The figure compares smart data packet protocol proposed in chapter five with RFDTurst protocol in the presence of malicious nodes. The standard smart data packet protocol generally cannot differentiate between normal and malicious nodes like the RFDTrust. However, the smart data packet can partially avoid malicious node like black hole and grey hole nodes. Whenever a malicious node damp the packet, based on smart data packet algorithm the sender will monitor the node to calculate the time delay. This is the time needed for a node to forward a packet and has been explained in chapter five. Accordingly, black hole and grey hole nodes will have more time delays, and their distance according to smart data protocol will be high as explained in chapter

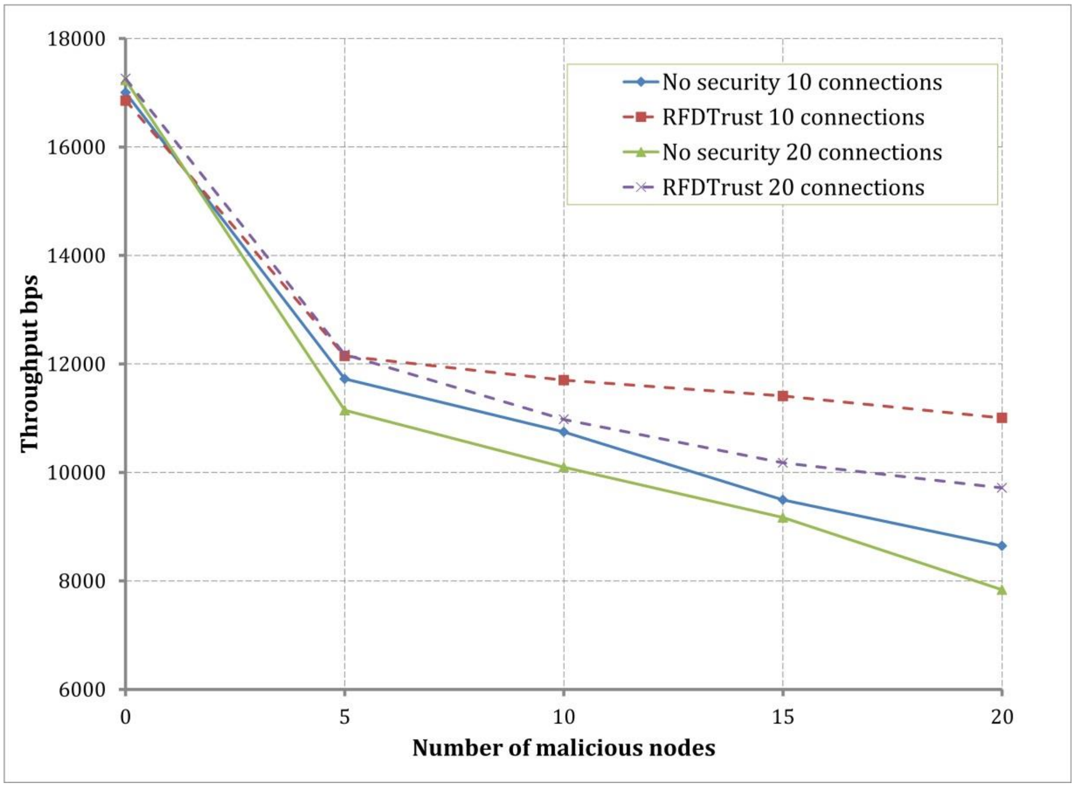


Figure 8.3. Average throughput under various numbers of malicious nodes for different number of connections

five. However, the proposed attack method is assumed to know this effect and combine sink hole with black hole and grey hole attacks. This will increase the probability of selecting these malicious nodes.

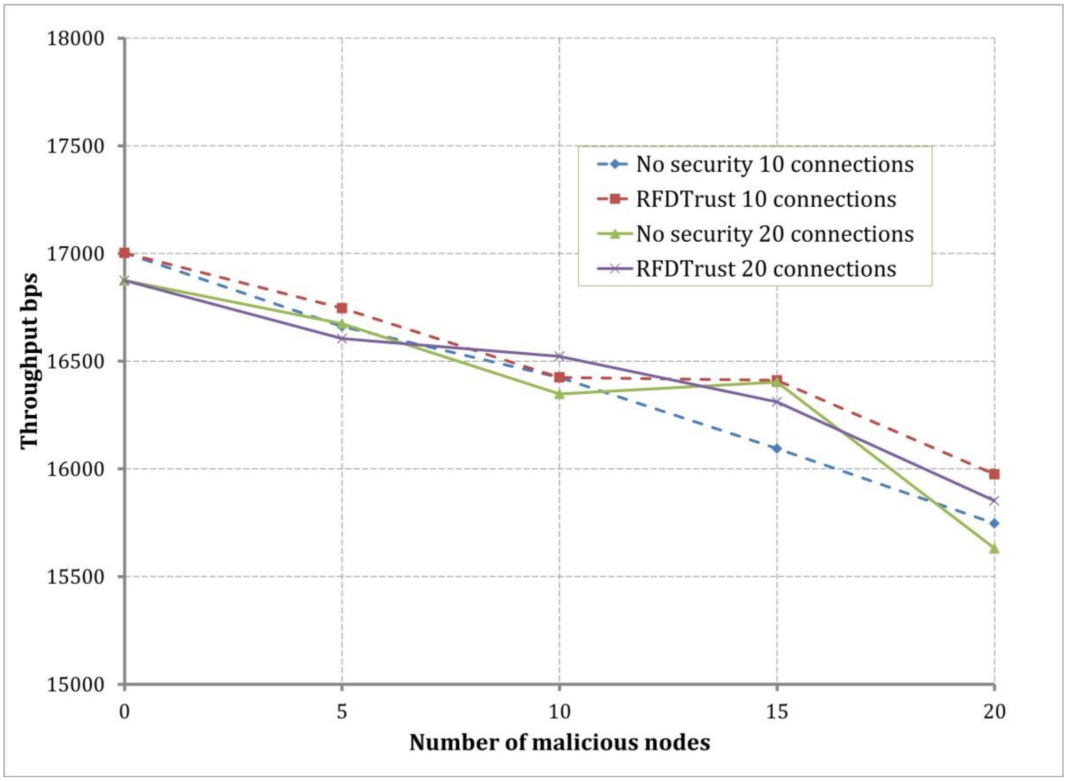


Figure 8.4 Average throughput under various numbers of malicious nodes with no sink hole attack

RFD algorithm forward packets based on altitude gradient, and sink hole attack address the main infrastructure of the algorithm. It is clear from the figure that presence of 5 malicious nodes degrades the performance about 30 %. Figure 8.4 shows the throughputs for the same scenario without the sink hole attack.

When the number of connection is low, the performance is better compared to higher number of connections. Couple of reason contributes in these results. First of all, the simultaneous number of packets in the network will be less. Moreover, some nodes become free and do not introduce any delay in the network as they only forward packets and do not add any traffic to the network. These normal nodes (non- malicious) will have less delay and will be more preferable to route packets through these nodes. The performance enhanced up to 27% for 10 connections and up to 16% for 20 connections.

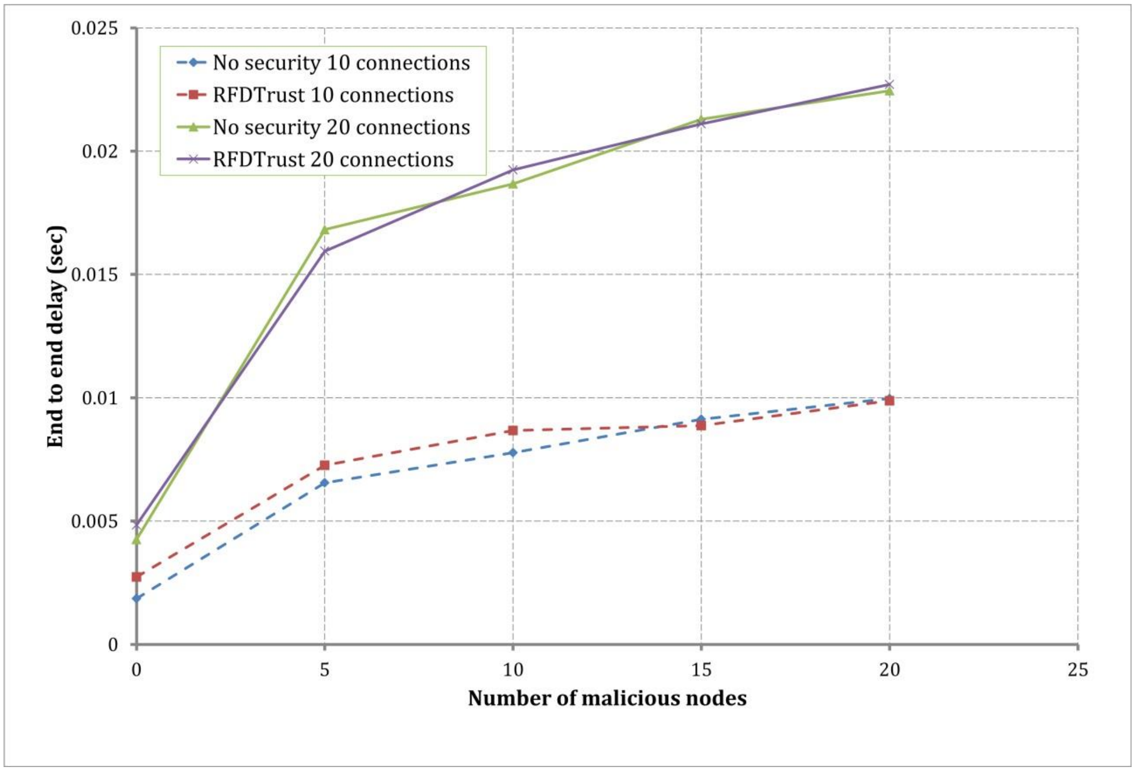


Figure 8.5 Average end to end delay under various numbers of malicious nodes for different number of connections

Figure 8.5 shows the effect of number of malicious node on the end to end delay. The RFDTrust protocol endeavors to find the most trusted paths in the network, so the packet may select a longer path toward its destination as long as the selected path is more secure. This increases the end to end delay in the network as number of malicious nodes increases. However, the delay for unsecured protocol is almost the same as the trust protocol. Only the packets that select longer route will reach the destination and which avoid malicious nodes that have longer time delay for forwarding packets.

Figure 8.6 shows the jitter for trusted protocol versus non trusted protocol. The jitter also increases as a result of select more secured path, which may be longer.

The packet overhead ratio is shown in Figure 8.7 demonstrates that RFDTrust has less overhead as it has higher throughput which is the main reason for these results.

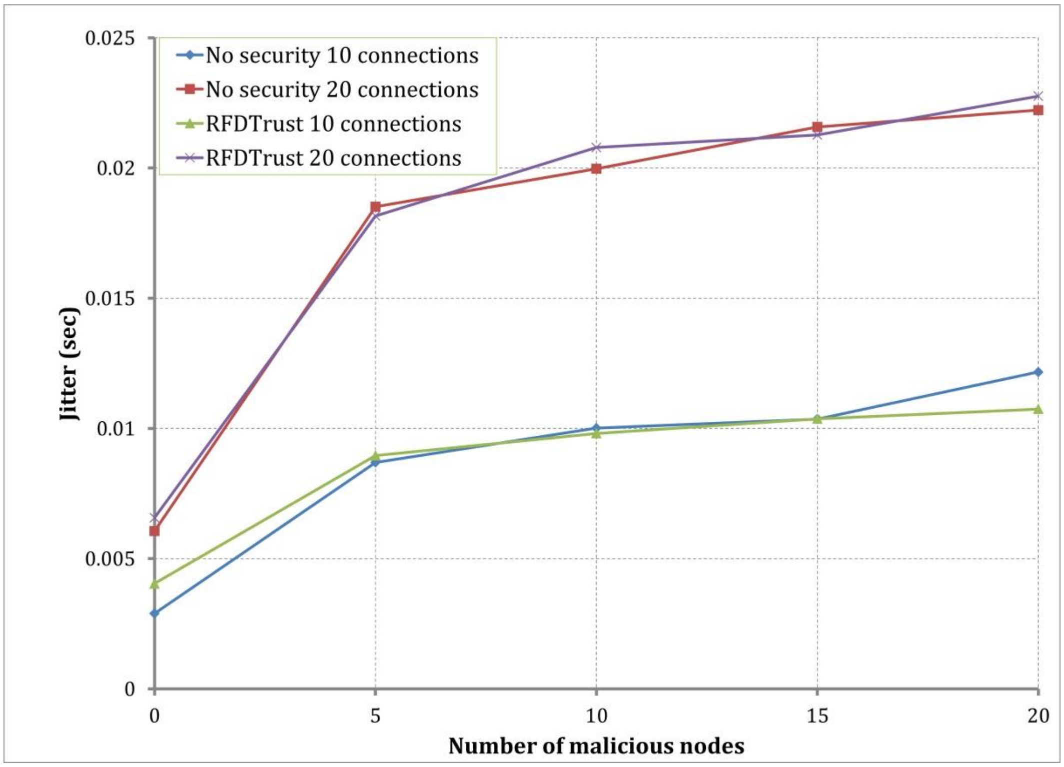


Figure 8.6 Average jitter under various numbers of malicious nodes for different number of connections

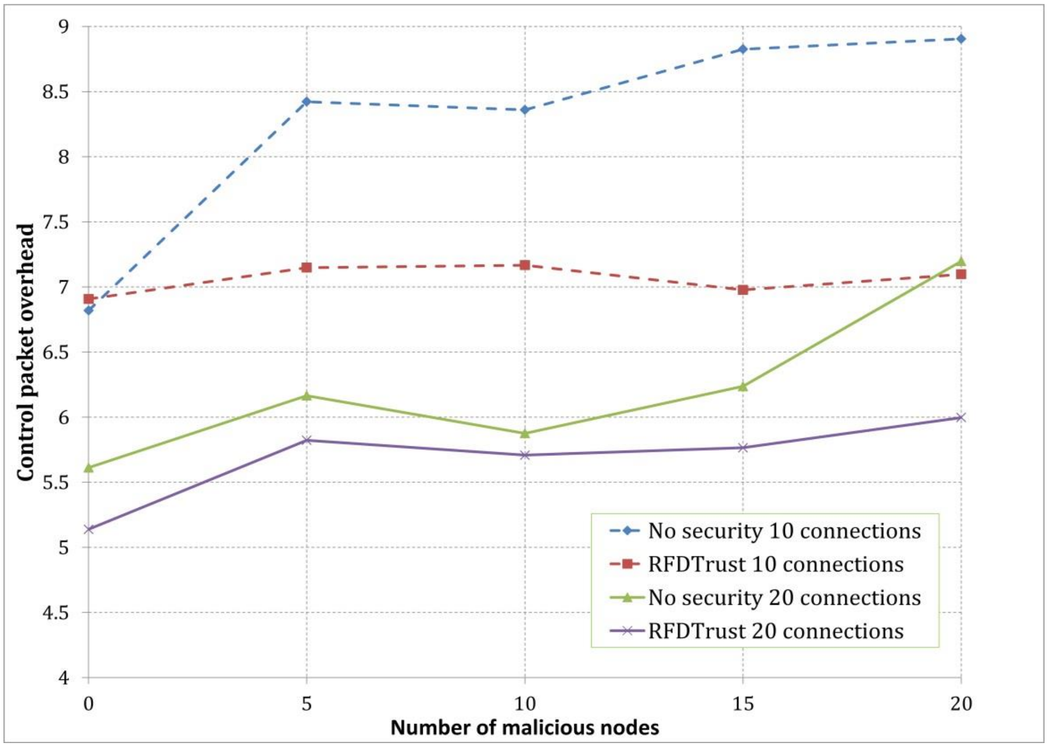


Figure 8.7 Average packet overhead under various numbers of malicious nodes for different number of connections

The effect of node mobility is shown in Figure 8.8. The throughput of RFDTrust protocol is compared with smart routing protocol. When the mobility increased, the throughputs of the protocols, secured and non-secured, will decrease. The effect of mobility on network performance using smart data packet routing protocol has been explained in chapter five. However, the presence of malicious nodes dramatically decreases the performance especially at high speeds. As the node become more mobile, the trust information will not have enough time to be built. Neighbours will not remain for long time to sufficiently detect their behaviours. Although second hand information could enhance the detection of black holes, the grey holes introduce more problems. Generally, grey hole are more difficult to detect as it changes its behaviour. In highly dynamic situation, the node may not have enough time to detect this behaviour. Within this small time, when the grey hole is a neighbour, its reputation may partially decreases. So when these small changes are shared, it will not be very effective. As the second hand information is combined with the sender reputation which also will be close to neutral value, the result will have less effect.

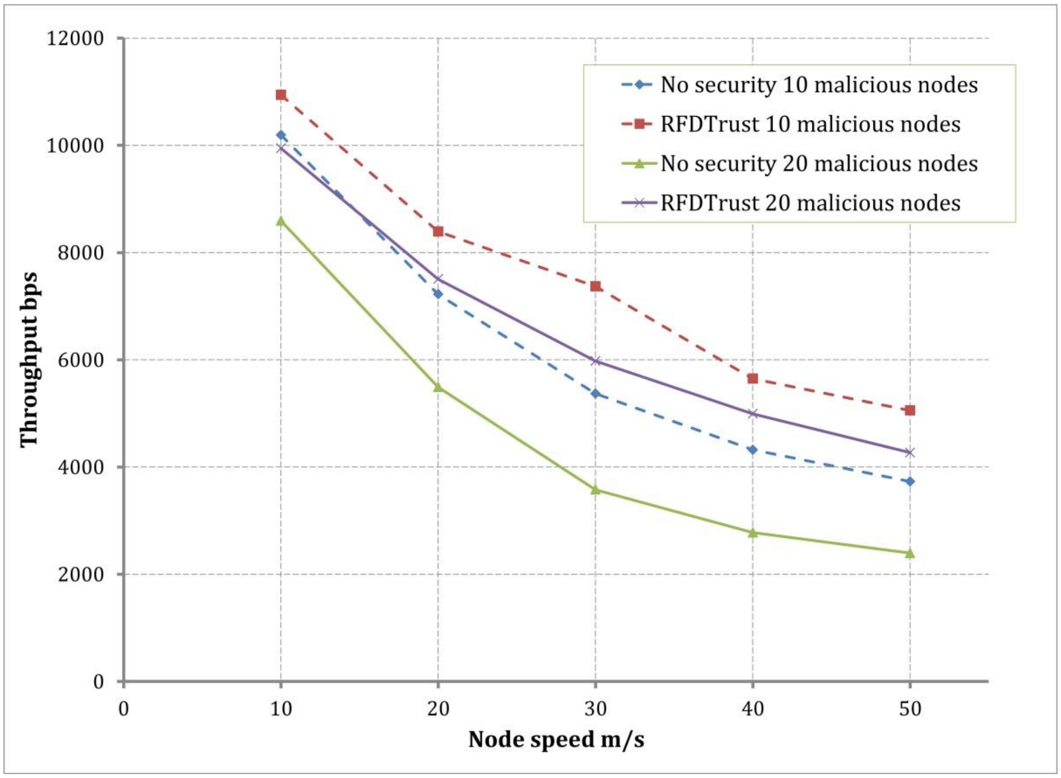


Figure 8.8 Average throughput under various speed values

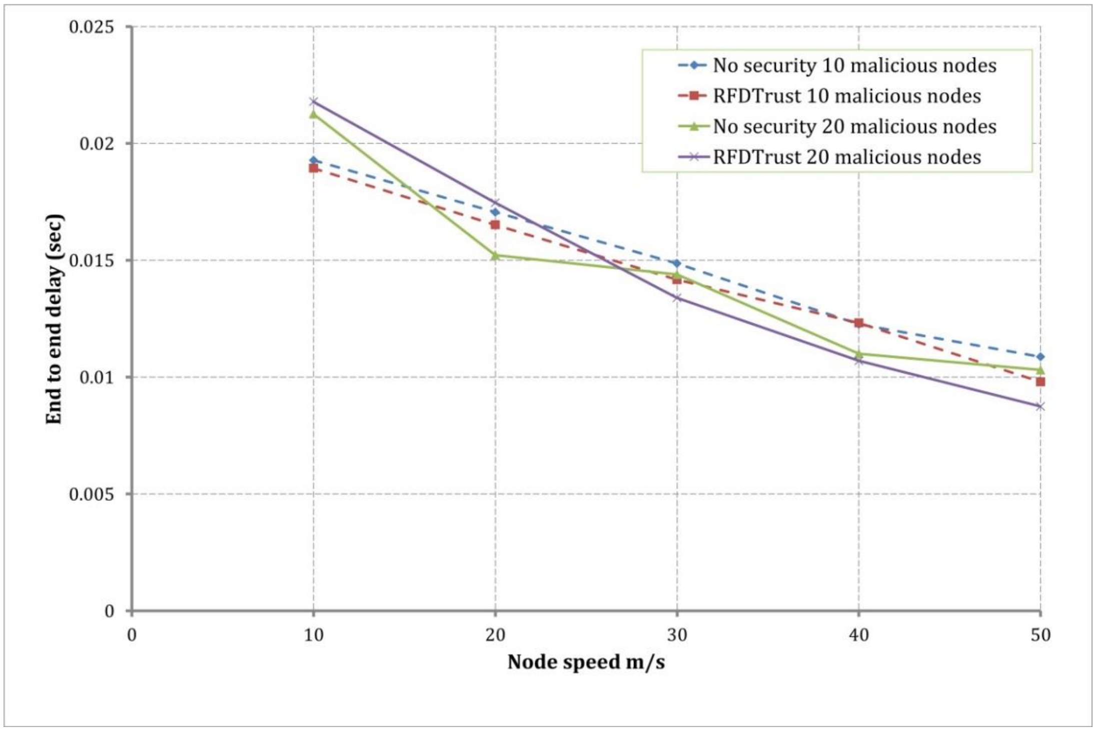


Figure 8.9 Average end to end delay under various speed values

The delay will also suffer from mobility and malicious nodes. Figure 8.9 shows the end to end delay for different node speed. As the speed increases, the network cannot build reputation table and which eliminate the effect of trust in routing equation 8.9. As a result, the packets that are delivered directly or through minimum number of intermediate nodes can reach their destination.

The effect of node density on network throughput is shown in Figure 8.10. A minimum of 40 nodes is chosen as there are 20 malicious nodes, and the number of connections pairs is 10 or 20. As the node density is increases, the throughput will decrease. The delay is shown in Figure 8.11. The effect of increasing number of nodes has been discussed in chapter five. The end to end delay at 60 nodes depicts the effect of nodes where minimum delay relates to node density and is between 40 to 60 nodes.

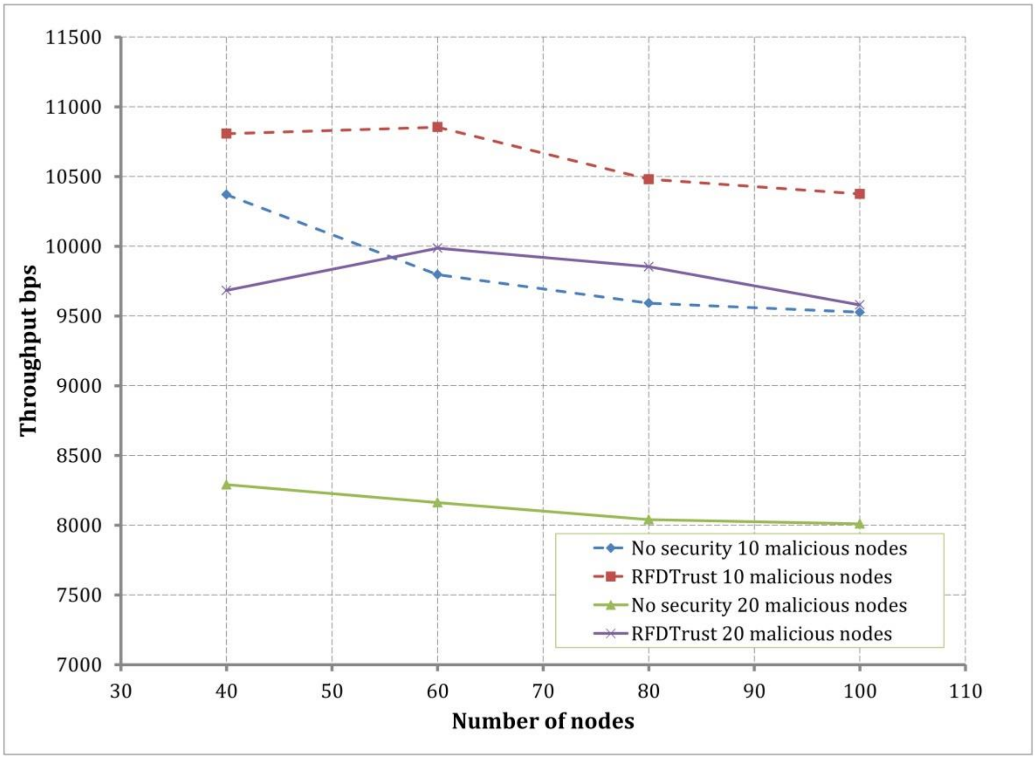


Figure 8.10 Average throughput under various numbers of nodes

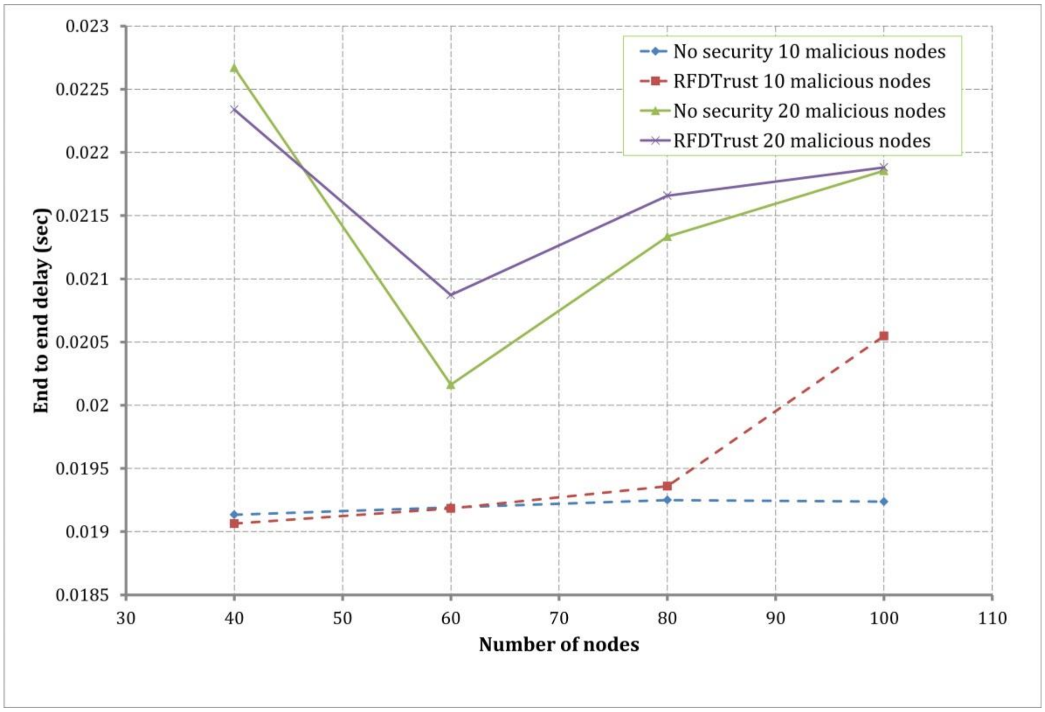


Figure 8.11 Average end to end delay under various numbers of nodes

8.7 Summary

This chapter introduces RFDTrust protocol. This is a reputation and trust based protocol where trust management is based on the RFD algorithm. The proposed protocol monitors nodes behaviours to detect misbehaviour nodes. RFD adds extra feature to monitoring algorithm where any monitoring node can detect if the monitored node has routed a packet correctly or not. Based on altitude differences, a legitimate node can only forward packet to nodes with less altitude. Simulation results show enhancement in throughput of RFDTrust compared to smart data packet protocol. Moreover, the packet overhead is decreased.

References

[1] H. Yih-Chun and A. Perrig, "'A survey of secure wireless ad hoc routing," *Security & Privacy, IEEE*, vol. 2, no. 3, 2004, pp. 28-39.

[2] Azzedine Boukerche, "' Algorithms and Protocols for Wireless and Mobile Ad Hoc Networks ,"Wiley-IEEE Press, 2008.

[3] L. Abusalah, A. Khokhar and M. Guizani, "'A survey of secure mobile Ad Hoc routing protocols," *Communications Surveys & Tutorials, IEEE*, vol. 10, no. 4, 2008, pp. 78-93.

[4] Han Yu, Zhiqi Shen, Chunyan Miao, C. Leung and D. Niyato, "'A Survey of Trust and Reputation Management Systems in Wireless Communications," *Proceedings of the IEEE*, vol. 98, no. 10, 2010, pp. 1755-1772.

[5] Jin-Hee Cho, A. Swami and Ing-Ray Chen, "'A Survey on Trust Management for Mobile Ad Hoc Networks," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 4, 2011, pp. 562-583.

[6] C.R. Erdal Çayırcı, "'Security in Wireless Ad Hoc and Sensor Networks,"John Wiley & Sons Ltd., 2009.

[7] Y. Hu, A. Perrig and D.B. Johnson, "'Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks," *Wirel.Netw.*, vol. 11, no. 1-2, 2005, pp. 21-38.

[8] M.G. Zapata and N. Asokan, "'Securing Ad Hoc Routing Protocols," *Proceedings of the 1st ACM Workshop on Wireless Security*, 2002, pp. 1-10.

[9] P. Papadimitratos and Z.J. Haas, "'Secure link state routing for mobile ad hoc networks," *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on*, 2003, pp. 379-383.

[10] Y. Hu, D.B. Johnson and A. Perrig, "'SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, 2002, pp. 3.

[11] D. Cerri and A. Ghioni, "'Securing AODV: the A-SAODV secure routing prototype," *Communications Magazine, IEEE*, vol. 46, no. 2, 2008, pp. 120-125.

[12] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu and Lixia Zhang, "'Security in mobile ad hoc networks: challenges and solutions," *Wireless Communications, IEEE*, vol. 11, no. 1, 2004, pp. 38-47.

[13] Hongmei Deng, W. Li and D.P. Agrawal, "'Routing security in wireless ad hoc networks," *Communications Magazine, IEEE*, vol. 40, no. 10, 2002, pp. 70-75.

[14] L. Abusalah, A. Khokhar and M. Guizani, "'A survey of secure mobile Ad Hoc routing protocols," *Communications Surveys & Tutorials, IEEE*, vol. 10, no. 4, 2008, pp. 78-93.

[15] M. Blaze, J. Feigenbaum and J. Lacy, "'Decentralized trust management," *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, 1996, pp. 164-173.

[16] C.D. Jensen and P.O. Connell, "'Trust-based Route Selection in Dynamic Source Routing," *Proceedings of the 4th International Conference on Trust Management, Springer-Verlag*, 2006, pp. 150-163.

[17] A.A. Pirzada, A. Datta and C. McDonald, "'Incorporating trust and reputation in the DSR protocol for dependable routing," *Computer Communications*, vol. 29, no. 15, 2006, pp. 2806-2821.

[18] S. Marti, T.J. Giuli, K. Lai and M. Baker, "'Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *MobiCom:Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, 2000, pp. 255-265.

[19] N. Griffiths, A. Jhumka, A. Dawson and R. Myers, "'A Simple Trust model for On-Demand Routing in Mobile Ad-Hoc Networks," *Intelligent Distributed Computing, Systems and Applications, Studies in Computational Intelligence*, vol. 162, 2008, pp. 105-114.

[20] X. Li, Z. Jia, P. Zhang, R. Zhang and H. Wang, "'Trust-based on-demand multipath routing in mobile ad hoc networks," *Information Security, IET*, vol. 4, no. 4, 2010, pp. 212-232.

[21] P. Michiardi and R. Molva, "'Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks," *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*, 2002, pp. 107-121.

[22] S. Buchegger and J. Le Boudec, "'Performance Analysis of the CONFIDANT Protocol," *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking \&Amp; Computing*, 2002, pp. 226-236.

[23] S. Buchegger and J. Le Boudec, "'A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks," *P2PEcon Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[24] K. Liu, J. Deng, P.K. Varshney and K. Balakrishnan, "'An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, 2007, pp. 536-550.

[25] M. Tamilarasi and T.V.P. Sundararajan, "'Secure enhancement scheme for detecting selfish nodes in MANET," *Computing, Communication and Applications (ICCCA), 2012 International Conference on*, 2012, pp. 1-5.

[26] S. Samreen and G. Narasimha, "'An efficient approach for the detection of node misbehaviour in a MANET based on link misbehaviour," *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, 2013, pp. 588-592.

[27] K. Vijaya, "'Secure 2ACK routing protocol in Mobile Ad Hoc Networks," *TENCON 2008 - 2008 IEEE Region 10 Conference,* 2008, pp. 1-7.

[28] Guoxing Zhan, Weisong Shi and J. Deng, "'Design and Implementation of TARF: A Trust-Aware Routing Framework for WSNs," *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, no. 2, 2012, pp. 184-197.

[29] Fenye Bao, Ing-Ray Chen, MoonJeong Chang and Jin-Hee Cho, "'Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection," *Network and Service Management, IEEE Transactions on*, vol. 9, no. 2, 2012, pp. 169-183.

[30] Xiaoyong Li, Feng Zhou and Junping Du, "'LDTS: A Lightweight and Dependable Trust System for Clustered Wireless Sensor Networks," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 6, 2013, pp. 924-935.

[31] H. Safa, H. Artail and D. Tabet, "'A Cluster-based Trust-aware Routing Protocol for Mobile Ad Hoc Networks," *Wirel.Netw.*, vol. 16, no. 4, 2010, pp. 969-984.

[32] Tao Jiang and J.S. Baras, "'Ant-based adaptive trust evidence distribution in MANET," *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on*, 2004, pp. 588-593.

[33] C.A. Melchor, B.A. Salem, P. Gaborit and K. Tamine, "'AntTrust: A Novel Ant Routing Protocol for Wireless Ad-hoc Network Based on Trust between Nodes," *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, 2008, pp. 1052-1059.

[34] N. Marchang and R. Datta, "'Light-weight trust-based routing protocol for mobile ad hoc networks," *Information Security, IET*, vol. 6, no. 2, 2012, pp. 77-83.

[35] S.R. Zakhary and M. Radenkovic, "'Reputation-based security protocol for MANETs in highly mobile disconnection-prone environments," *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on*, 2010, pp. 161-167.

[36] R. Venkataraman, M. Pushpalatha and T. Rama Rao, "'Regression-based trust model for mobile ad hoc networks," *Information Security, IET*, vol. 6, no. 3, 2012, pp. 131-140.

[37] B. Wu, J. Chen, J. Wu and M. Cardei, "'A Survey of Attacks and Countermeasures in Mobile Ad Hoc Networks," *Signals and Communication Technology, Wireless Network Security, Springer US*, 2007, pp. 103-135.

[38] A.K. Abdelaziz, M. Nafaa and G. Salim, "'Survey of Routing Attacks and Countermeasures in Mobile Ad Hoc Networks," *Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on*, 2013, pp. 693-698.

[39] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato and A. Jamalipour, "'A survey of routing attacks in mobile ad hoc networks," *Wireless Communications, IEEE*, vol. 14, no. 5, 2007, pp. 85-91.

[40] J.R. Douceur, "'The Sybil Attack," *IPTPS '01 Revised Papers from the First International Workshop on Peer-to-Peer Systems*, 2002, pp. 251-260.

[41] S. Hazra and S.K. Setua, "'Sybil attack defending trusted AODV in ad-hoc netwok," *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*, 2012, pp. 643-647.

[42] Xiaojuan Liao, Dong Hao and K. Sakurai, "'Achieving cooperative detection against Sybil attack in wireless ad hoc networks: A game theoretic approach," *Communications (APCC), 2011 17th Asia-Pacific Conference on*, 2011, pp. 806-811.

[43] S. Hashmi and J. Brooke, "'Authentication Mechanisms for Mobile Ad-Hoc Networks and Resistance to Sybil Attack," *Emerging Security Information, Systems and Technologies, 2008. SECURWARE '08. Second International Conference on*, 2008, pp. 120-126.

[44] Tong Zhou, R.R. Choudhury, Peng Ning and K. Chakrabarty, "'P2DAP — Sybil Attacks Detection in Vehicular Ad Hoc Networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 3, 2011, pp. 582-594.

[45] S. Abbas, M. Merabti, D. Llewellyn-Jones and K. Kifayat, "'Lightweight Sybil Attack Detection in MANETs," *Systems Journal, IEEE*, vol. 7, no. 2, 2013, pp. 236-248.

[46] Shan Chang, Yong Qi, Hongzi Zhu, Jizhong Zhao and Xuemin Shen, "'Footprint: Detecting Sybil Attacks in Urban Vehicular Networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 6, 2012, pp. 1103-1114.

[47] X. Li, Z. Jia, P. Zhang, R. Zhang and H. Wang, "'Trust-based on-demand multipath routing in mobile ad hoc networks," *Information Security, IET*, vol. 4, no. 4, 2010, pp. 212-232.

[48] A.A. Pirzada, C. McDonald and A. Datta, "'Performance comparison of trust-based reactive routing protocols," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 6, 2006, pp. 695-710.

[49] C. Bettstetter, "'Smooth is better than sharp: a random mobility model for simulation of wireless networks," *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems,* 2001, pp. 19-27.

Chapter Nine

Conclusions and future works

9.1 Introduction

This thesis has investigated different issues and problems in mobile ad hoc routing and security protocols. The main propose of this theses is to use and suggest new swarm based approaches to address these problems, aiming to improve the performance and security of ad hoc routing protocols.

The first approach uses techniques that rely on the nodes' position were a prediction system is introduced that detect link and overall end to end route qualities. The protocol jointly solves the problem of routing and channel assignment in MANETs as these two problems are depending on each other. The channel assignment scheme is distributed and dynamic where multiple non-overlapping channels are assigned to different interfaces to increase the throughput of the network. Ant agents where used to continually search for better routes, and new route set up and the channel re-assignment process occurs based on specific events related to predicted route life time. The selection of a route depends on link qualities between nodes within the route where it tries to select a route with higher data rate. Geographical information used to predict link quality and combined with mapping algorithm to reflect the dynamic structure of selected route.

The second approach introduced the use of the RFD algorithm in ad hoc routing. RFD offers many advantages toward implementing this approach. The main two reasons of using RFD are the small amount of information that required to be added to the packets and the main idea of the RFD algorithm which is based on one kind of agent called drop that moves from source to destination only. This will eliminate the need of feedback packets or agents to update the network and offers a suitable solution to change data packet into smart packets. A hybrid ACO and RFD protocol has been proposed where ACO approach is used to address the bottleneck of slow start up convergence of RFD algorithm, as well as to build multipath routes to the destination. The protocol is based on RFD algorithm and manifests routes as altitudes gradients. Drops which are the RFD agents are moving according to gradient differences between nodes. Data packets are acting like drops and can adapt learned information. The adaptation process of data packets and the local feedback technique in RFD give data packet the ability to detect network status and react according to this status, and guide themselves to better routes. Route maintenance is carried out by data packet themselves. Data packets continuously optimize the route between the source/s and destination/s.

The third approach introduced RFDMANET, a routing protocol which is mainly built based on RFD algorithm. Data packets have the ability to route themselves as well as cooperate in learning process. The protocol uses an acknowledgment packet to build multiple paths to the destination by creating downslope on the altitudes surface. RFDMANET is a hybrid protocol, containing both reactive and proactive procedures. Moreover, the protocol divides the information into two kinds, reactive and proactive, and this information updated by different types of packets. Mainly, hello messages are responsible for information diffusion and proactive information updates. This proactive information could be out of date and therefor routing data packets upon this information could unreliably forward those packets in the network. Separation could increase the reliability but at the same time it will decrease the amount of information that data packets will use to decide their route.

Power consumption by nodes in MANETs should be minimized epically as the power capacity of the nodes in MANETs is limited. In this context, a power and congestion aware routing protocol has been introduced. Packets are routed through less congested area of the network in order to save power. At the same time the protocol senses the remaining battery power of the nodes and prefers nodes with higher battery level. The protocol uses one set of tables to store both reactive and proactive information. Moreover, the protocol combines the idea of both smart data protocol and RFDManet and adds power awareness to them.

Finally, the idea of RFD algorithm and smart data packet is used to build a secure routing protocol. Misbehaving attacks, especially internal attack, is difficult to be detected by standard cryptographic system; therefor a reputation and trust security approach is introduced based on RFD algorithm. The protocol detects black hole and grey hole attacks as well as sink hole attacks. The main important feature of RFD algorithm is that water drops are always moving downward and therefore nodes with in the network cannot redirect packets to higher altitudes. Gradually, packets should reach their destinations as a result to this fact. Misbehaving node will be exposed and will be easily detected. The proposed security protocol adds this to the existing smart packet protocol and uses packet forwarding ratio as well to build more reliable and secure routing protocol.

9.2 Directions to future works

There are several recommendations which can be used for future research directions in using swarm intelligence in the field of ad hoc routing and security. The future research is outlined as follows:

The first direction is to use the RFD algorithm in multi-channel multi-interface protocol. This could enhance the performances of the network especially as no backward agents are required. Moreover there are many techniques that could be used to detect link quality without the use of geographical information. This will reduce the amount of information shared in the network. The use of fuzzy logic instead of using mapping technique could make the selecting next forwarding nodes decisions more easily.

The RFD algorithm is promising technique in mobile ad hoc routing. A further improvement could be added to the protocol. Controlling the ratio of smart data packets to dump data packets could lead to more optimization. RFD algorithm mimics topographical structure of the watershed surface, and tries to find shortest path between source and destination. Using geographical information to detect the distance between nodes could enhance the gradient function results, as the presented smart packet in chapter five uses congestion and time delay as a distance value. Another approach is to adaptively control erosion and sedimentation parameters. Moreover, as wireless mesh and sensor network have defined infrastructure, implementing smart data packet in these network could enhance their performance.

The result of chapter five showed the end to end delay of the network is increased at high data rates introduced as result of buffering at MAC layer. These results motivate the need of better coupling between the network layer and MAC layer. One of the approaches is to couple these layer better by selecting the erosion and sedimentation ratio depending on the number of buffer packets in the MAC layer.

The proposed RFDTrust protocol could be enhanced by using Bayesian approach for the reputation representation and mixing information. The protocol should be tested agents warm hole attacks, as we couldn’t implement worm hole attack in OMNet++ epically providing a tunnel between two mobile nodes.

List of publications

[1] S.H. Amin and H.S. Al-Raweshidy, '"Geographical Multi-Channel Multi-Interface Routing Protocol for Mobile Ad Hoc Network," Broadband, Wireless Computing, Communication and Applications (BWCCA), 2012 Seventh International Conference on, 2012, pp. 66-73.

[2] S.H. Amin, H.S. Al-Raweshidy and R.S. Abbas, '"Smart Data Packet Ad Hoc Routing Protocol," Computer Networks, http://dx.doi.org/10.1016/j.bjp.2013.11.015

Submitted papers

[1] S.H. Amin and H.S. Al-Raweshidy "RFDManet: A Swarm Based Routing Protocol Using Intelligent Data Packets", IEEE transections on mobile communication

[2] S.H. Amin and H.S. Al-Raweshidy , "Power and Congestion Aware Smart Data Packet Routing Protocol" Elsevier Journal, Ad Hoc Networks.