Design and Implementation of an On-demand Ad-hoc Routing Algorithm for a Positional Communication System

Tahmid Al-Mumit Quazi

Submitted in fulfilment of the academic requirements
for the degree of Master of Science in Engineering
in the School of Electrical, Electronic and Computer Engineering
at the University of Natal, Durban, South Africa

September 2003

As the candidate's supervisor I have approved this dissertation for submission.

Signed: _____

Name: Mr. Stephen McDonald

Date: 29 September 2003

Abstract

A mobile ad-hoc network is an autonomous network of mobile devices that are connected via wireless links. In such networks there is no pre-existing infrastructure and nodes are free to move in a random fashion. Due to this mobility mobile ad-hoc networks have dynamic topologies. A host in the network typically has limited bandwidth and energy resources. Routing is a major challenge in the development of such systems and there have been many solutions proposed in the recent past. The aim of this work is to design and implement a routing scheme for a Positional Communication System (PCS). The PCS is a network of mobile handheld pocket PCs connected via wireless interfaces. The system allows voice and data communication between nodes in the network.

This dissertation addresses the process of designing a routing protocol for an ad-hoc network. There have been many proposed algorithms that solve the routing problem in a mobile ad-hoc network. It is a difficult task to compare the performance of these protocols qualitatively as there are many parameters that affect network performance. Various simulation packages for networks of this type exist. One such package is the Network Simulator (NS-2). It is a discrete time event simulator that can be used to model wired and wireless networks. This dissertation presents NS-2 simulations that compare four recently proposed routing algorithms. From this comparison study it is shown that on-demand algorithms perform best in a mobile ad-hoc environment.

The dissertation then describes the design of a novel on-demand routing algorithm. The on-demand algorithms proposed thus far use a blind flooding technique during the route discovery process. This method is inefficient and creates excessive routing overhead. The routing protocol proposed in the dissertation implements a query localization technique that significantly reduces the network traffic. The protocol also introduces a load checking metric in addition to the metric used by most on-demand schemes, namely hop count. Simulation results show that such a scheme makes the on-demand routing algorithm more efficient and scalable than existing ones.

It is widely believed that prior to implementing a routing protocol in real world systems it is essential that it is tested and validated on a test-bed. The dissertation presents the

implementation of an on-demand routing algorithm in a Positional Communication System test-bed, where each handheld PC in the network runs an embedded Linux operating system.

Preface

The research work in this dissertation was performed by Mr. Tahmid Quazi under the supervision of Mr Stephen McDonald in the School of Electrical, Electronic and Computer Engineering at the University of Natal, Durban, South Africa. This work was sponsored by ARMSCOR.

The whole dissertation, unless specifically indicated to the contrary in the text, is the author's work, and has not been submitted in part, or in whole to any other university.

Acknowledgements

I would like to thank my parents as without their kind sacrifices I would not have had the opportunities to achieve all that I have in life. I am eternally grateful to them for their constant support throughout my academic work.

I would like thank my supervisor, Mr Stephen McDonald, for all his assistance in this research work. His passion for teaching is extraordinary. I am grateful that he believed in me when he offered to take me onboard the PCS project. Many thanks to go to ARMCOR for their generous support for this project. Although not directly involved with this project, Mr Peter Handley needs to be thanked as well. It was his "deja moo" joke that started this all! His generosity and zeal for research is exemplary.

I would like to thank the members of staff of our school for all the chats and kind advice in the time that I have been here. It is unbelievable how much I have learnt from this interaction. Finally I would like to thank my friends. I have been honoured to know some incredible people and it is because of these friendships, and all that I have gained from them, that I can honestly say that the last few years have been some of the best in my life. A friend that needs particular mention is my sister. I thank her for her patience in dealing so kindly with an obsessive engineer. I am grateful for all the intense conversations and all the laughter.

Contents

Abstract i
Preface iii
Acknowledgements iv
Contents v
Chapter 1 1
1.1 Background 1
1.2 The different architectures in wireless communications 1
1.3 The Mobile Ad-hoc Network (MANET) 4
1.4 Characteristics of a MANET 5
1.4.1 Dynamic Topology 5
1.4.2 Bandwidth Constraints 5
1.4.3 Energy Constraints 5
1.4.4 Host capacity 5
1.4.5 Limited security 6
1.5 Possible applications of Mobile Ad-hoc Networks 6
1.5.1 Commercial applications 6
1.5.1.1 Emergency Crisis management 6
1.5.1.2 Impromptu conferencing 7
1.5.1.3 Vehicular communications 7
1.5.1.4 Personal Area Networks (PAN) 7
1.5.2 Military applications 8
1.5.2.1 Tactical networks 8
1.5.2.2 Sensor networks 8
1.5.3 The Positional Communication System (PCS) 9
1.6 Challenges in ad-hoc networking 10
1.6.1 Wireless Interface 10
1.6.2 Power control 10
1.6.3 Security 11
1.6.4 Routing 11
1.7 Dissertation Layout 12
Chapter 2 14
2.1 Introduction 14

2.2 Basic routing schemes 14
2.3 Routing in ad-hoc networks 16
2.4 Classification of routing protocols 18
2.5 Pro-active routing protocols 19
2.5.1 Destination Sequenced Distance Vector (DSDV) 19
2.5.2 Cluster-Gateway Switch Routing (CGSR) 20
2.5.3 Global State Routing (GSR) 22
2.5.4 Fisheye State Routing (FSR) 23
2.6 Reactive routing protocols 24
2.6.1 Dynamic Source Routing (DSR) 25
2.6.2 Ad-hoc On-demand Distance Vector (AODV) 27
2.6.3 Temporally Ordered Routing Algorithm (TORA) 30
2.6.4 Associativity Based Routing (ABR) 34
2.7 Hybrid routing protocols 35
2.7.1 Zone Routing Protocol (ZRP) 35
2.8 Conclusion 37
Chapter 3 38
3.1 Introduction 38
3.2 Network Simulator (NS-2) Introduction 39
3.3 Mobile Networking in NS-2 40
3.3.1 The mobile node model 40
3.3.1.1 Address Classifier 41
3.3.1.2 Port Classifier 42
3.3.1.3 Routing Agent 42
3.3.1.4 Link layer 42
3.3.1.5 ARP module 42
3.3.1.6 Interface queue 43
3.3.1.7 Medium Access Control (MAC) 43
3.3.1.8 Network Interface 43
3.3.1.9 Radio Propagation Model 43
3.3.1.10 Wireless Channel 44
3.3.2 The Physical Layer Model 44
3.3.2.1 Free-space model 44
3.3.2.2 Two-ray ground reflection model 45
3.3.3 The Medium Access Control (MAC) 46

3.4 Protocols Simulated 47
3.5 Simulation Setup 48
3.5.1 Mobility Model and Network Traffic Generation 48
3.6 Metrics used 51
3.6.1 End-to-End Network Throughput 51
3.6.2 End-to-End Network Delay 51
3.6.3 Efficiency 51
3.7 Results and Discussion 51
3.7.1 Packet Delivery Ratio 52
3.7.2 Routing Overhead 53
3.7.3 Average end-to-end packet delivery delay 56
3.8 Other protocol comparison studies 57
3.9 Conclusion 57
Chapter 4 59
4.1 Introduction 59
4.2 Overview of AODV's on-demand methodology 60
4.3 Proposed improvements 61
4.4 Description of the proposed protocol 63
4.4.1 Route localization 64
4.4.2 Load Checking 67
4.5 Implementation of TAODV in NS-2 71
4.6 Simulation setup 71
4.7 Simulation results 72
4.7.1 Packet Delivery Ratio 72
4.7.2 Routing Overhead 76
4.7.3 Average end-to-end delay 80
4.8 The effect of TAODV's load checking algorithm 82
4.9 Related work 83
4.10 Conclusion 84
5.1 Introduction 86
5.2 Description of the test-bed 87
5.2.1 The handheld device 87
5.2.2 The wireless interface 88
5.2.2.1 802.11b Operating modes 88
5.2.2.2 The 802.11b Physical Layer 89

5.2.2.3 The 802.11b Data Link Layer 89
5.2.3 The Operating System 91
5.3 The Linux routing architecture 92
5.4 Requirements of on-demand routing 92
5.5 The Netfilter Structure 95
5.5.1 The hooks 96
5.5.2 The kernel modules 96
5.5.3 The actions 96
5.6 Description of the routing application 97
5.6.1 The kernel module 97
5.6.2 The queuing module 99
5.6.3 The user-space data packet handling module 99
5.6.4 Routing protocol packet handling module 100
5.6.5 Development status 101
5.7 Test setup and results 101
5.7.1 Ping Test 102
5.7.2 Real-time Voice over IP test 103
5.8 Conclusion 104
References 109

List of Figures

Figure 1.1: Structure of 2.5 G cellular network 2

Figure 1.2: Network configuration in Mobile IP 3

Figure 1.3: Screen-shot of the Graphical User Interface of the PCS 9

Figure 2.1: The routing architecture in CGSR 21

Figure 2.2: The scope architecture in FSR 24

Figure 2.3: Flooding of the route request to discover the route record in DSR 26

Figure 2.4: Propagation of Route Reply in DSR 26

Figure 2.5: Network flood of route request packets in AODV 28

Figure 2.6: Reverse Path setup in AODV 29

Figure 2.7: Forward Path setup in AODV 30

Figure 2.8: Flooding of QRY packet in the network 32

Figure 2.9: The setting up of the DAG in the network 32

Figure 2.10: Re-establishment of a route upon a link failure 33

Figure 3.1: Schematic of a mobile node in NS-2. 41

Figure 3.2: The two-ray ground reflection model 45

Figure 3.3: Packet delivery ratio performance of simulated protocols 53

Figure 3.4: Routing overhead of simulated protocols in terms of the total number of routing packets 55

Figure 3.5: Routing overhead in total number of bytes showing the performance of AODV vs. DSDV 55

Figure 3.6: Average end-to-end delay performance of the simulated protocols 56

Figure 4.1: The network configuration in LAR 66

Figure 4.2: Rebroadcast decision in TAODV 66

Figure 4.3: Packet delivery performance for the 16 node network 73

Figure 4.4: Packet delivery performance for the 50 node network 74

Figure 4.5: Routing overhead performance in a highly mobile 16 node network (pause time of 1 second) 77

Figure 4.6: Routing overhead performance in a stable 16 node network (pause time of 250 seconds) 78

Figure 4.7: Routing overhead performance in a highly mobile 50 node network (pause time of 1 second) 79

Figure 4.9: Average end-to-end delay performance for a 16 node network 81

Figure 4.10: Average end-to-end delay performance for a 50 node network 81

Figure 4.11: Performance of protocols in a highly mobile 16 node network 83

Figure 5.1: Picture of 3800 and 3600 Compaq iPaq's used for the PCS 87

Figure 5.2: Different modes in 802.11b networks 89

Figure 5.3: Hidden Terminal problem 90

Figure 5.4: Layout of the DSDV implementation in Linux 93

Figure 5.6: Structure of the TAODV routing daemon 98

Figure 5.7: Indoor test-bed network setup 102

List of Acronyms

ABR	Associativity Based Routing
AODV	Ad-hoc On-demand Distance Vector
ARP	Address Resolution Protocol
CBR	Constant Bit Rate
CCK	Complementary Code Keying
CGSR	Cluster-Gateway Switch Routing
CRC	Cyclic Redundancy Check
CSCW	Computer-Supported Collaborative Work
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DAG	Directed Acyclic Graph
DARPA	Defence Advanced Research Projects Agency
DBF	Distributed Bellman-Ford
DCF	Distributed Coordination Function
DLAR	Dynamic Load Aware Routing
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
FA	Foreign Agent
FHSS	Frequency Hopping Spread Spectrum
FIFO	First-In-First-Out
FSR	Fisheye State Routing
GPS	Global Position System
GSR	Global State Routing
GUI	Graphical User Interface

HA	Home Agent
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv4	Internet Protocol version 4
ISM	Industrial Scientific Medical
LAN	Local Area Network
LAR	Location Aided Routing
LLC	Logical Link Control
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
Mbps	Mega bits per second
MCS	Mobile Switching Centres
MN	Mobile Node
NAM	Network Animator
NASA	National Aeronautics and Space Administration
NS-2	Network Simulator
OS	Operating System
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PAN	Personal Area Network
PCS	Positional Communication System
PDA	Personal Digital Assistant
QPSK	Quadrature Phase Shift Keying
rfc	Request For Comment

RIP	Routing Information Protocol
RTS/CTS	Request-To-Send/Clear-to-Send
TAODV	Tactical AODV
TAODVD	Tactical AODV daemon
TCP	Transmission Control Protocol
TORA	Temporally Ordered Routing Algorithm
VINT	Virtual InterNet Testbed
VoIP	Voice over IP
ZRP	Zone Routing Protocol

Chapter 1

Introduction

1.1 Background

There have been significant advances in mobile computing technology recently. Mobile devices today are faster, have more memory and are far more power efficient than those in the past. Advances in battery technology has resulted in greater convenience in using such devices. As a result mobile devices such as laptops, notebook computers and Personal Digital Assistants (PDAs) are becoming cost effective and readily available, so much so that they are becoming part of everyday life. However, the novelty these days is not to just use these as stand-alone devices but to connect them for the purposes of data sharing and, more pertinently, for communications. The current drive in the computing and communications industry is to develop a mobile device that will give the functionality of both the PC and a communications device. Thus, running concurrently with the development of mobile devices has been the development of technology that enables such devices to communicate with each other and thus create networks of mobile devices. This has led to the innovation of devices and standards that allow mobile devices to communicate with each other over a wireless channel. Cost effective wireless modems and standards such as Bluetooth and the IEEE 802.11 wireless standards are products of such innovation. The availability of these necessary technologies has fuelled significant interest in wireless technology research in the recent past. This chapter gives a brief overview of the state of the art in wireless technology.

1.2 The different architectures in wireless communications

The architectures in wireless communications can be divided into two schools:

i. Structured network architecture
ii. Structure-less network architectures.

It is the first category that has changed the face of communication in the past few decades. This category itself can be split into two with the well established cellular communications

system and the newly developed wireless LAN architecture that has seen such a boom in recent years. The architecture of 2.5G and the proposed 3G cellular systems consists of the network being split into geographically unique cells with mobile hosts communicating with each other through a backbone of stationary network nodes called Base Stations, as shown in Figure 1.1. The communications between the Base Stations are then coordinated by base stations higher up in the hierarchical structure called Mobile Switching Centres (MCSs). The mobile hosts within a cell maintain connectivity to the wireless network by communicating with the nearest Base Station. There is a registration process that is performed whenever a mobile host is turned on within a cell. It is also necessary for the synchronization of data transfer between the mobile host and the base stations. When a mobile migrates from one cell to another, the seamless connectivity is ensured by a hand-off process. This involves the coordination of the signalling between the two base stations concerned in such a way that the mobile host is unaware of the transition.

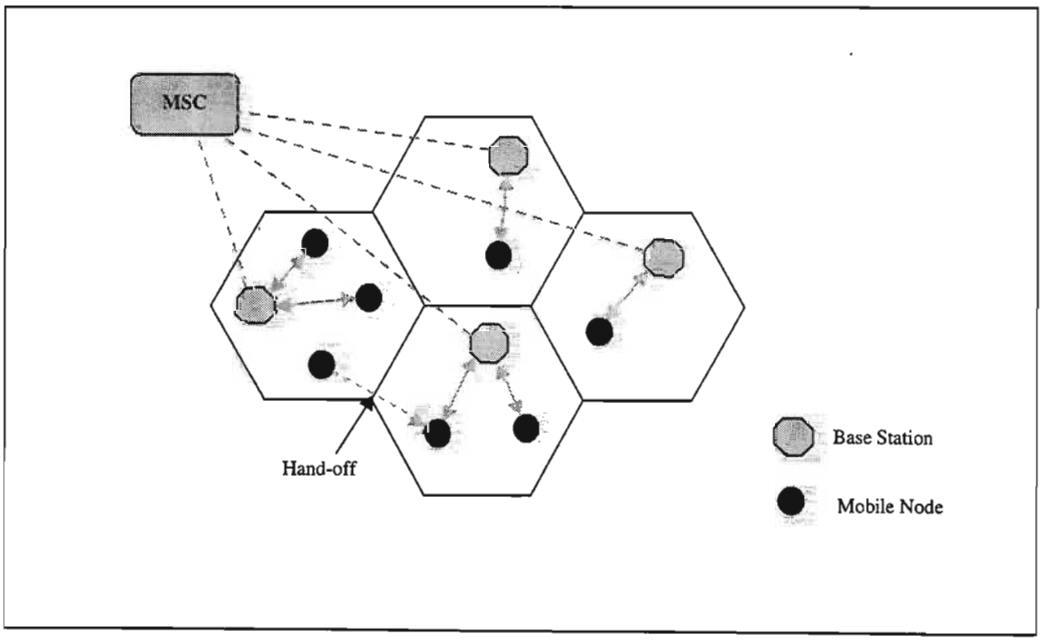


Figure 1.1: Structure of 2.5 G cellular network

The second type of structured network offers wireless network connectivity to mobile Personal Computers. This involves mobile PCs with wireless interfaces communicating with wireless hubs or Access Points. Today this architecture is popularly known as "hot-spots" with the broadband 802.11b "Wi-Fi" wireless LAN standard. The Access Points act as gateways that give the mobile users access to facilities such as the internet. However the underlying Internet Protocol (IP) used for this set-up was designed for the wired network. In

such a network, the nodes remain stationary with its IP address identifying the network to which it is attached. If the host with that IP moves to another autonomous network, with a different IP subnet, it would be impossible to communicate with the host as its IP address would not be recognised. Thus to cater for this scenario a modified version of IP called Mobile IP [Perkin98] was developed to support this type of mobility.

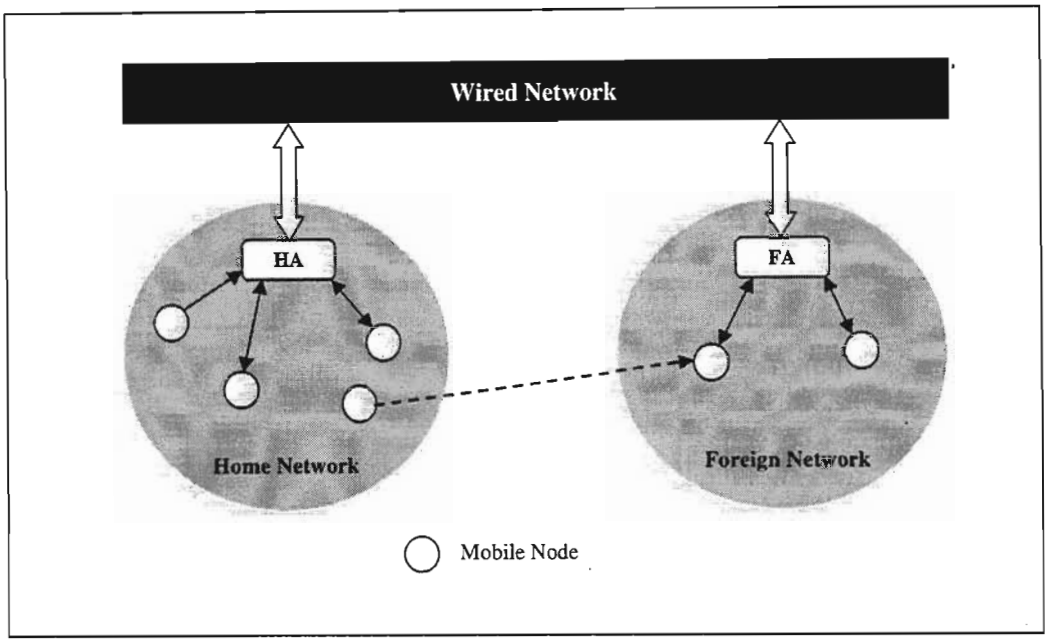


Figure 1.2: Network configuration in Mobile IP

In Mobile IP there are three defined entities: the mobile node (MN), the Home Agent (HA) and the Foreign Agent (FA), as shown in Figure 1.2. The MN is the mobile host that has the ability to move from one subnetwork to another and has two addresses associated with it: the *home address* and the *care-of address*. The network topology as seen by the MN is divided into the *home* subnetwork which is the network in which it can be reached. All the other subnetworks to which the mobile migrates to are considered *foreign* networks. The mobility agents HA and FA are thus the mobile nodes routers in the respective subnetworks. The mobility agents advertise their presence to mobile nodes by means of Agent Advertisements [Perkin98]. These beacons provide the mobile node with such necessary information as whether it is in its home network or in a foreign network. When in a foreign network, the mobile node obtains its address in such a network (its care-of address). The MN then registers this address with its Home Agent with the help of the Foreign Agent. All packets that are destined for the MN are received by the node's Home Agent which relays the packets to the Foreign Agent in a process termed *encapsulation* or *tunnelling* [Perkin98] by

the protocol. The Home Agent encapsulates each data packet destined for the MN with an additional IP header, with the destination field set to the MN's care-of address in the foreign network. This leads the packet to the Foreign Agent which simply removes the additional header and passes the rest of the packet payload to the mobile node.

The architectures described above, although novel, provide limited ubiquitous computing and communications. In such architectures the users, despite being mobile, are still bound to the core wired, stationary architecture. A user has to be in the vicinity of an Access Point or Base Station in order to be able to communicate with others. The second type of wireless network architecture was thus developed to offer true ubiquitous, mobile computing with no dependence on any fixed architectures. These networks are useful in scenarios where there are no fixed established architectures available or cannot be relied upon in terms of emergency of deployment. The essence of these structureless networks, called ad-hoc networks, is that users are able to communicate directly with each other or through host systems of other user in the network with multi-hop paths.

1.3 The Mobile Ad-hoc Network (MANET)

Mobile ad-hoc technology has attracted the attention of the communications field since the development of the Mobile Packet Radio Networks in research projects initiated by the US military in the 1970 and 1980s. The MANET is an autonomous network of mobile computers that are connected via wireless links. There is no pre-existing infrastructure and thus each node in the network may act as a host or as a router (an intermediate node) to allow connectivity between other source and destination hosts in the network. The term ad-hoc (meaning *for this only* in Latin) implies that the network is formed in an impromptu manner to meet an immediate and specific goal. Since the nodes in the network are mobile, the network topology can be configured in an arbitrary manner and can change dynamically. An ad-hoc network can operate in an isolated fashion or it can be connected to the wider internet via gateways.

1.4 Characteristics of a MANET

The applications of MANETs may vary widely. However such networks have some salient features which set them apart from other networks. These are documented by the MANET [Manet03] working group of the Internet Engineering Task Force (IETF) [Ietf03]. This documentation is in the form of a Request For Comment [rfc2501] and is summarised as follows.

1.4.1 Dynamic Topology

Due to the mobility of the nodes in a MANET, the network topology may be connected in any arbitrary manner and may change dynamically. Such a topology is randomly changing and is unpredictable.

1.4.2 Bandwidth Constraints

There has been advancement in the development of wireless modem technology which offers significantly higher data rates than in the past. However, the capacity of the wireless links is still significantly lower than the links in the wired environment. In addition, the radio communication in the wireless environment has to account for other issues such as multiple access, channel fading, noise and interference. This leads to a marked decrease in the realised throughput when compared to the radio's maximum transmission rate.

1.4.3 Energy Constraints

The mobile stations in an ad-hoc network rely on batteries or other exhaustible sources for power. There have been leaps in the development of battery technology for mobile computing. However, for ad-hoc network systems, the efficient use of this most important resource to a node is vital to its operation.

1.4.4 Host capacity

The envisaged applications of MANETs could use platforms other than laptops and notebook PCs such as Personal Digital Assistants (PDAs), and even smaller embedded devices. These devices will not offer the same amount of processing power and memory

resources. Thus these resources available to a node have to be taken into account in the design of ad-hoc network systems.

1.4.5 Limited security

Mobile ad-hoc network systems by their very nature are susceptible to security threats. There is the physical security threat with the possibility of the mobile devices being stolen or damaged in the field. There is also the threat of malicious hosts with possibilities of eavesdropping, spoofing, and denial-of-server attacks. To alleviate such issues existing link security techniques have to be considered in the design of MANET systems.

1.5 Possible applications of Mobile Ad-hoc Networks

The technology of ad-hoc networks has received much attention because of the flexibility offered by infrastructureless nature of such networks. MANETs are dynamic, self-starting, self organising and thus can be deployed easily and quickly. There is no cost, either financially or in terms of time, of setting up any network architecture as it is formed when the nodes come together. These networks can also be deployed in a variety of environments and situations, which are out of the reach of the rigid, structured communications architectures. This has led to there being a wide range of applications for ad-hoc networks as detailed in this section.

1.5.1 Commercial applications

1.5.1.1 Emergency Crisis management

In emergency situations such as natural disasters or accident scenarios communications, using ad-hoc networks would be invaluable. In the event of a natural disaster it is likely that if there was any existing communications architecture in the area, it would be in disarray. On the other hand, a self-organising dynamic ad-hoc network would restore communications easily and at a moments notice in such a time critical situation. Search and rescue systems in environments where there is no pre-existing communication architecture ad-hoc network provide an ideal, efficient solution.

1.5.1.2 Impromptu conferencing

There exist scenarios where collaborative computing becomes a highly useful facility if not essential. In conference venues, ad-hoc networks could provide the functionality for audio and video conferencing. Researchers equipped with mobile computers in the field could use MANETs to share information and communicate. In some business environments, ad-hoc networks would allow users to conduct interactive meetings in environments other than the office. This type of Computer-Supported Collaborative Work (CSCW) [Toh02] is beneficial in improving workflow and enhancing the productivity of collaborative design work.

1.5.1.3 Vehicular communications

With Global Positioning System (GPS) information being offered in automobiles today, it is not far fetched to imagine communications systems that would enhance such location based services. As the automobiles travel through an area, a communication system would collect information on weather conditions, possible road accidents and other road information such as locations of petrol stations, garages, hotels and tourist spots. If there was an architecture that allows vehicles to communicate with each other, the system running in the vehicles would be able to share such information. A MANET is ideal for this type of application.

1.5.1.4 Personal Area Networks (PAN)

Today wearable computing is very much a reality with the size of chips diminishing with each version revision. In the foreseeable future there will exist wireless nodes on a person in terms of a watch, pen or belt. It will be possible to store information in such devices, which may be shared when users with such devices meet in an ad-hoc fashion. With the advent of the intelligent home and office environment, the idea is to network mobile wireless nodes such as PDAs with other devices of the environment enabled with wireless interfaces. Thus a person entering his home with his PDA will automatically alert the alarm system to deactivate, instruct the temperature controlling system to set the appropriate temperature, activate his entertainment system etc. In the office environment a person's PDA will be able to interact with his desktop system and share necessary information such as memos, emails and data files in an ad-hoc manner without any wired connections.

1.5.2 Military applications

1.5.2.1 Tactical networks

The development of ad-hoc networks, from its incubation stages in the 1970 with DARPA in the US, has been done with military applications in mind. In the battlefield scenario the communications between personnel has to be robust and reliable. Structured communication architectures cannot be relied upon in such an environment, as such structures are vulnerable to physical attacks by an enemy. The self-organising nature of an ad-hoc network makes it ideal for the treacherous scenario. With such a network, the soldiers would be able to move forward to uncharted territory being able to communicate within their own battalion, with other such battalions and with their command and control centre. This form of situational awareness is vital in organising operations in a military effort and the ability to instantaneously establish field communications is invaluable. MANETs are by nature very robust and adaptive. The existence of the network is not hampered if one or a few of the nodes in the network disappear either due to damage or mobility. The network changes to adapt to the new topology in order to maintain connectivity. In addition, the multi-hopping functionality enables the network to cover a far wider physical region in the battlefield environment than previously possible.

1.5.2.2 Sensor networks

Wireless sensor networks are a well-known application of ad-hoc networks. They are used for monitoring and analysis of uncharted generally inhospitable terrain. They are deployed in places that are not easily accessed by humans. Sensor networks consist of many possibly disposable, low power devices equipped with programmable computing. In a military scenario, sensor networks are deployed in enemy territory where they gather information on enemy activities and relay such information back to the central control. However, sensor network applications are not limited to the military as they have found applications in environmental monitoring systems, security surveillance and distributed computing. A futuristic use of sensor networks was imagined in the motion picture "Minority Report" as sensory robots. With the development of artificial intelligence technology, ad-hoc communications can be the only feasible way for such systems.

The above mentioned applications are merely a subset of the possibilities of ad-hoc networks. In the near future MANETs will be deployed in a variety of applications and

environments. The applications are not limited to earth alone as in a recent project by NASA where MANETs are seen as a "natural" for space communications [Rice03]. It is conceivable that the communications between the modules of the unmanned missions to Mars will be done in an ad-hoc manner. It is likely that as the technology develops there will be applications for ad-hoc networks which have not even thought of today.

1.5.3 The Positional Communication System (PCS)

The application that is the basis of this research project is called the Positional Communication System (PCS) [PCS03]. It is a tactical network system that is to operate in a battlefield scenario. The purpose of the system is to provide situational awareness by enabling communication between soldiers in a battalion unit. Each node in the network is enabled with hardware and software that will enable voice and data communications. The voice communication will be in the form of Voice over IP (VoIP). The data will consist of image information from an image capture module and GPS information from a GPS module. One of hosts in a unit will have an interface to tactical long-range radio, which will allow communications with other units. A screen shot of the intended application GUI can be seen in Figure 1.3. It shows the current node's position at the centre with three other neighbouring nodes. The need for a self-commencing, self-organising and spontaneous network has led the research into a mobile ad-hoc network for the intended application.

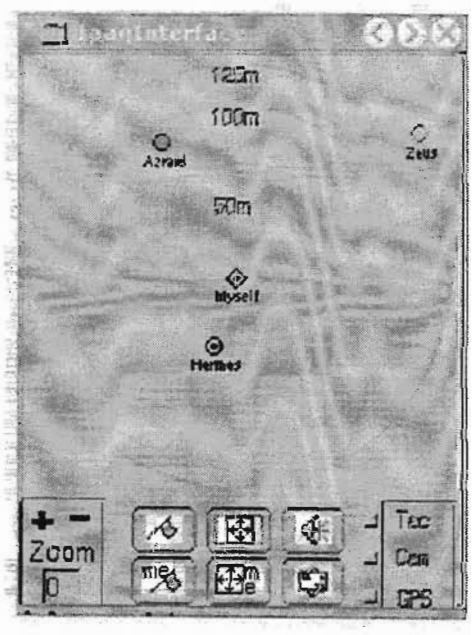


Figure 1.3: Screen-shot of the Graphical User Interface of the PCS

1.6 Challenges in ad-hoc networking

1.6.1 Wireless Interface

There are two technologies that are being considered for the wireless interface in ad-hoc networks: Bluetooth [Blue03] and IEEE 802.11 [IEEE99] standards. The production of Bluetooth is as a result a collaborative effort of several communications companies (Ericsson, IBM, Intel, Nokia and Toshiba). Their aim was to produce a solution that would give mobile devices access to a wireless channel for communication purposes. The standard is ideal for small devices with short range low power radio links. It operates in the unlicensed 2.4 GHz band using Frequency Hopping Spread Spectrum (FHSS). Low power Bluetooth devices are the norm today and provide a range of approximately 15 m, although higher powered devices with ranges of 150 m are available. The limitation of the standard is in terms of the speed of operation with only 2 Mbps data rates being offered.

The IEEE 802.11x[1] standard has achieved much greater success in ubiquitous computing applications. It offers functionality for structured communication of users with Access Points as well as peer-to-peer communications. The wireless standard also operates in the 2.4 GHz band with much higher data rates than Bluetooth. It has different modulation techniques that offer data rates of up to 11 Mbps. There is a newly released version of the standard with even higher data rates. The maximum range specified by the standard is also significantly greater, easily exceeding twice the distance achieved by Bluetooth. The MAC used overcomes many of the problems faced in the wireless medium and can allow many more users than that allowed by Bluetooth. However, none of the two standards address the routing issue in wireless communications. Their task is to offer a fairly reliable and efficient access to the wireless medium to higher layer network protocols.

1.6.2 Power control

The mobile hosts in an ad-hoc network are most likely powered by exhaustive resources such as battery power. It is therefore essential that any application running the ad-hoc network architecture uses some form of power control during transmissions. The transmit power not only has an impact on the battery life of a host, it also affects the range in terms of hops that a host's transmissions achieve. A higher transmission power would increase the

[1] x could be either a, b or the recently produced 802.11g

range and make routing simple but it would also negatively affect the traffic carrying capacity of the channel with the increased congestion. Thus intelligent power control in an ad-hoc network application is important consideration.

1.6.3 Security

One of the major challenges in the research into ad-hoc networks is the security of connections between hosts in the network. With free-space radio transmission in the wireless environment it is fairly easy for a malicious host to eavesdrop on a communication session. This could lead to unauthorised access, information theft, interference, jamming and service degradation. Due to the multi-hop nature of ad-hoc transmissions, it is a very difficult task to even detect such intrusion. The field of security for ad-hoc networks is at a very premature stage and this issue has to be thoroughly studied before ad-hoc network systems can be practically deployed in real world applications.

1.6.4 Routing

The design of routing protocols for ad-hoc networks has been a challenging task from the earliest stages of development of the technology. The routing is most often a complex matter given the unique characteristics of MANETs. As mentioned previously due to the mobile nodes in the network, it can change rapidly and unpredictably. A routing scheme for such a network has to be sufficiently quick in its routing information gathering process to maintain current, up-to-date information on dynamically changing network topology. However the routing protocol must achieve this while remaining conservative in its use of resources available to the hosts in the network. A routing protocol's conservative use of the channel bandwidth and battery power in its operation will result in efficient routing in the network and the host lasting longer in terms of power availability.

The design of ad-hoc routing protocols has received a lot of attention because of the above mentioned challenges. As a result there are a large number of schemes that have been proposed to solve the ad-hoc routing problem. The aim of this research work is to select and if necessary, design a suitable protocol that would be used in the PCS application described in Section 1.5.3. As with any research it is necessary to thoroughly study the currently documented techniques available in literature. The information derived on the characteristics and qualities of the techniques could then be used to gauge their usefulness. However, in order to select one routing scheme over another it is necessary to determine a methodology

that would allow a direct comparison of the protocols. In this regard a set of rules or criteria are generally chosen and the performance of each protocol in terms of each criteria (or metric) determined. In the past, the criteria have only allowed qualitative comparison of ad-hoc routing protocols. Although useful in providing insights into the functionality of the protocols, these comparison studies were inconclusive as they did not provide a true picture of a protocol's performance. What was required was a platform that would allow the quantitative performance comparison using a common, realistic environment where each protocol could be subjected to a set of repeatable scenarios. This led to the development of packages for the simulation of ad-hoc routing protocols. One of the most widely used packages in the ad-hoc research groups is the Network Simulator (NS-2) [NS-203]. There have been a number of comparative studies done using NS-2. However the conclusions drawn from these studies could not be applied directly to the PCS as such studies used different network configurations and different sets of comparison metrics. Thus, it became necessary to setup and run simulations in NS-2 that modelled the PCS network.

The initial objective of this research work was to simulate and compare four prominent ad-hoc routing schemes and determine which one best suited the PCS network. This study showed that on-demand algorithms are more suited to highly mobile ad-hoc networks than proactive algorithms. The contribution in this dissertation is a new on-demand protocol that shows significantly better performance than other proposed ad-hoc routing schemes. The improved performance is attributed to the protocol's routing mechanism that overcomes some of the disadvantages of recently proposed on-demand algorithms. The dissertation then presents an implementation of the proposed routing algorithm on a real world test bed using mobile handheld PCs running an embedded Linux operating system.

1.7 Dissertation Layout

The rest of the dissertation is organised as follows. Chapter 2 provides further details of the routing challenge in an ad-hoc network. The chapter then provides a survey of different proposed ad-hoc routing protocols. The ad-hoc routing protocols in literature are grouped into different classes and a selection of protocols from each class are described in detail. Chapter 3 presents the simulation of four prominent ad-hoc routing protocols. The NS-2 simulation package is described with emphasis on the wireless models used in the simulations. The mobility and traffic models used in the simulations are also described. The chapter then presents the results of the simulations and the conclusions derived from them. The proposed ad-hoc routing protocol in this research is presented in Chapter 4. The chapter

gives an overview of the methodology of the protocol on which the proposed protocol is based. The proposed improvements are then explained before the protocols mechanism is described in detail. The chapter concludes with NS-2 simulation results that present the performance comparison of the proposed protocol against the performance of the protocol upon which it is based. Chapter 5 describes the real world physical implementation of the proposed protocol. The implementation is done in the PCS test-bed. A hardware and software description of the test-bed is given in the chapter. The current routing structure in Linux is described and an explanation is given as to why it is not suited for on-demand routing without additional routing modules. The supplementary routing modules that make the implementation of the on-demand application a possibility are then briefly described. The inner working of the routing application is then presented. The chapter then presents current results from tests conducted on the test-bed to demonstrate the operation of the on-demand routing application. The dissertation finally concludes with future work mentioned in Chapter 6.

This work has been published in the following conferences:

- SATNAC 2002, Drakensburg, South Africa [Dearh02]
- IEEE MWCN 2003, Singapore [Quazi03b] (This has not been presented yet)
- SATNAC 2003, George, South Africa [Quazi03a]
- MICSSA 2003, Pretoria, South Africa [Quazi03c] (This has not been presented yet)

Chapter 2

Routing Protocols for Ad-hoc Networks

2.1 Introduction

Routing is the function of determining a path between a source and destination nodes in the network in order for them to communicate. The unique characteristics presented by a mobile ad-hoc network that were detailed in Chapter 1 make the design of a routing protocol for such networks a considerable challenge. This chapter presents the efforts made in ad-hoc protocol research work to solve the routing problem.

The chapter begins with the basic routing schemes used in general data networks. Section 2.3 presents the routing problem in ad-hoc networks and explains why the protocols used in the wired networks are inappropriate in the mobile ad-hoc network environment, and details the desirable properties of ad-hoc routing protocols. The classification of the MANET routing protocols is presented in Section 2.4. The routing protocols surveyed in this research work are presented in the next three subsections. It is the functional properties of each protocol that make them unique. Thus the task in this chapter is to describe the properties of each existing protocol, making observations on their advantages and disadvantages.

2.2 Basic routing schemes

There are three basic routing methodologies that are used in the design of traditional routing protocols:

- Distance vector routing
- Link state routing
- Source routing

This first category of routing scheme is also known as the Distributed Bellman-Ford [Berts87] algorithm. Each node in the network maintains a routing table that contains information on all the routes in the network of which it is aware. An entry in the table contains a vector, which consists of the address of the destination node, the address of the next hop node on route to the destination and the distance in terms of the number of hops

required to reach it. A distance vector routing protocol running at a node periodically broadcasts its routing information to its neighbours. The nodes that receive this information in turn use it to calculate their routes to destination nodes in the network. Although these algorithms are computationally efficient and easy to implement, there are disadvantages that limit their usefulness. There is the well known *count-to-infinity* problem [Berts87] which is caused by inconsistencies due to slow propagation of routing update messages in the network. This problem results in the formation of short and long-lived routing loops which has a significant impact on the routing performance. There is also the limit in the scalability of distance vector algorithms as the increase in the routing update message packets size increases with the increase in the number of nodes in the network.

A node running a Link-state protocol maintains in its routing table, a view of the entire network topology. Again each entry contains the destination and next hop address and the cost to reach the destination. The protocol ensures that each node maintains consistent topological information by periodically broadcasting its routing table to all other nodes in the network. This method, called flooding [Berts87] ensures that each node gets a copy of the link-state information from all other nodes. A node that receives the information, updates its view of the network topology by applying a shortest path algorithm for each destination. Since each node in the network calculates its routes independently to the other nodes there is little possibility of inconstancies that can cause count-to-infinity problems and routing loops. However the control message overhead caused by the flood of the routing information significantly reduces the bandwidth available for data routing. Link-state algorithms are often complex and are thus computationally intensive.

Source routing protocols include the path information for each route with each data packet sent. The source node determines the route and each data packet that is sent to the destination carries the complete path information. The routing loop problem faced by the other types of protocol is avoided with this approach. There is however a delay in determining the route which could impact routing performance in terms of packet delivery latency. In addition there is the added routing overhead with each packet sent.

2.3 Routing in ad-hoc networks

The commonly used routing protocols in the wired networks are Routing Information Protocol (RIP) [Rip03] and Open Shortest Path First (OSPF) [Ospf03]. RIP is a distance vector protocol while OSPF is based on the link-state routing philosophy. The two protocols, although quite efficient for routing data in the wired networks of the Internet, are entirely unsuited for application in the mobile ad-hoc networks. The dynamic nature of MANETs causes random and unpredictable changes in the routes in the network. The slow update rate of the wired protocols diminishes their ability to converge to a steady state for finding routes in the ever-changing topology. As mentioned in the previous chapter, the bandwidth available in the links of the wireless ad-hoc network is significantly less than that in the wired environment. The routing overhead incurred by the distance vector and link state protocols in terms of protocol control messaging thus becomes much of a factor in the ad-hoc network environment. Finally, the computationally expensive operations of the traditional wired protocols would be highly taxing on the scarce CPU, memory and battery power resources of the mobile nodes in an ad-hoc network.

The MANET working [manet03] group of the IETF has detailed a list of desirable qualitative properties of ad-hoc routing protocols [Corso99]. It is necessary that a newly proposed protocol meet some if not all of the standards in the list summarised below:

Distributed Operation: Route computations in the network must be done in a distributed manner as the centralised approach is inappropriate for the dynamic ad-hoc network. Centralised routing would create critical nodes in the ad-hoc network which is a highly undesirable scenario in MANETs.

Loop-freedom: Although not critical, a routing protocol should determine loop-free routes from source to destination. This prevents the wastage of resource due to a fraction of the packets looping in the network for an arbitrary amount of time.

Demand based operation: In order to make efficient use of available resources, a routing protocol should be adaptive to the network traffic produced. This means that the protocol should only react when it is required for it to do so and avoid periodic exchange of routing information. There is a possibility with protocols that use periodic exchanges that they

maintain routes that are never used. The clear drawback with the on demand approach is the increase in data packet delivery latency due to the route discovery process.

Proactive operation: In certain applications of ad-hoc networks the packet delivery latency due to the on demand based operation is unacceptable. In such scenarios the use of additional resources should be traded off for lower delays protocols that employ a proactive operation.

Unidirectional Link Support: the precarious radio environment can result in the formation of unidirectional links in the network. It is a desirable feature in a designed routing protocol that it detects and adapts to such types of links.

Power conservation: the mobile nodes in an ad-hoc network rely on limited battery power for their operation. Thus it is necessary that a routing protocol be conservative in its use of such resources. In addition, in order to conserve energy, a node may stop transmitting and/or receiving for arbitrary time periods. The routing protocol should support the "sleep" mode functionality in its operation.

Security: A node in a MANET is susceptible to security attacks in the form of snooping of network traffic, replaying transmissions, redirecting routing messages and manipulation of packet headers. These are all actions that can be easily carried out by a malicious node in the open radio environment. Thus it is necessary that a protocol provide a degree preventive security.

Multiple routes: A protocol that creates multiple routes between source and destination pairs could theoretically increase the traffic carried by the network. This would also decrease the number of reactions to topology changes and congestion in the network. The availability of alternative routes would make it unnecessary for the routing protocol to re-discover routes that have been broken.

Quality of service: In order to carry multimedia data traffic in an ad-hoc network it is essential that a routing protocol support some sort of Quality of Service. This becomes pertinent when time critical data such as voice is considered.

It should be noted that no ad-hoc routing protocol proposed thus far satisfies all the desirable requirements detailed above. Each protocol tries to solve a certain sub-set of the problem-set often trading off one requirement for another depending the characteristic of the ad-hoc network for which the protocol is being designed.

2.4 Classification of routing protocols

The task of routing involves making forwarding decisions for data packets depending on the routing state of the network. The routing protocol thus has a two-fold operation. The first is to collect information about the state of the network and secondly to use this information to create routes through which data packets are forwarded.

There are many protocols that have been proposed to solve the routing problem in ad-hoc networks. With the wide variety available there are different criteria that can be used to classify these protocols. One of the major criteria used is based on the idea of "when" a routing protocol collects information regarding the state of the network. There are protocols that constantly maintain the paths to all destinations and thus learn of the network topology before a forwarding request is made. These protocols are grouped into the ***proactive*** routing schemes. The second group of protocols in this classification, called ***reactive*** routing schemes, only become active after there is a request for a route. In most cases this type of protocol does not have a route to the destination before such a request is made as they purge routes that have not been used recently. The source node, in such cases, has to discover a route to the destination. Once a route is found it can be used for routing data traffic. The protocols execute route maintenance procedures in the event of a route to a desired destination breaking. There are different advantages and disadvantages with each type of routing scheme and so some protocol designs attempt to incorporate more than one philosophy. These protocols, termed ***hybrid*** routing schemes, use both proactive and reactive actions in their operation.

In addition to these broad criteria, routing protocols can be further classified according to the type of addressing used. Protocols that use *flat* addressing maintain an architecture where all the nodes in the network are on the same level. In *hierarchical* addressing the network is aggregated to form groups. This type of addressing is particularly appropriate in large networks where it is essential to reduce the control messaging overhead in the network.

The text in the sections that follow gives brief reviews of the well known protocols that have been proposed thus far. The novelty or main idea behind each protocol is presented and the optimisations are omitted for brevity. Such detail can however be acquired from the protocol proposal references cited in each section.

2.5 Pro-active routing protocols

These proactive routing protocols are often called table-driven ad-hoc protocols. They try to maintain complete routes from each source in the network to all other nodes. This information is generally cached in tabular form with one or more tables being used by the different protocols. In order to maintain a consistent view of the network at each node, the protocols continuously propagate updates of topological changes throughout the network. Proactive protocols were, in general, derived from the distance vector and link-state schemes of the wired network protocols. They have been adapted and modified to solve the problems that the static network protocols faced in the dynamic mobile ad-hoc environment.

2.5.1 Destination Sequenced Distance Vector (DSDV)

The Destination Sequenced Distance Vector (DSDV) [Perkin94] routing algorithm is the modification of the classic Distributed Bellman-Ford (DBF) algorithm. In a MANET any node in the network may be required to act as router and so each node maintains a routing table that lists all the nodes in the network of which it is aware. Each entry in the table contains the destination and the next hop addresses as well as the cost (in terms of hops) to get to the destination. The reason DSDV is an improvement on the original wired network protocol is that it avoids DBF's tendency to create routing loops. Each entry in the routing table and a protocol message update is marked with a *sequence number*. This number is maintained by the destination node of a route entry and is increased whenever the node publishes its routing information. The sequence number value is used by all other nodes in the network to determine the "freshness" of the information contained in a route update for the destination. Since the value is sequentially incremented, a higher sequence number implies that the routing information is newer.

In order to maintain routing information consistency in the network each router shares its routing table with its neighbours by means of routing updates. These updates are done both in a periodic and triggered fashion. The designers of the protocol proposed this method with the aim of alleviating the potentially large amount of network traffic that will be induced by

the routing updates. In a periodic update which occurs at predetermined regular intervals, a node broadcasts its entire routing table in a packet termed a *full dump*. *Incremental* routing update packets are used when triggered significant topological change. The change could be either due to node mobility or link breakages to next hop neighbours. The *incremental* update packets only contain those entries which have changed since the last periodic update. The triggered updates with the smaller packet sizes result in the reduced overhead incurred by the protocol. A route table update entry contains the destination address of a node, the cost to reach it and the highest known sequence number for the destination. When a node receives an entry for a particular destination with a higher sequence number its old entry is replaced with the newer route. In the case where a node has to choose between two entries with the same sequence number, it selects the path with the least cost. An intermediate node that detects a broken route to a destination assigns an *infinity* value to the route's path cost, increments the entry destination sequence number and immediately broadcasts the information as an update. Using this technique critical network topology information such as link breakages is disseminated quickly across the network.

2.5.2 Cluster-Gateway Switch Routing (CGSR)

Cluster-Gateway Switch Routing [Chiang97] uses the DSDV routing protocol as its underlying routing mechanism. It differs from DSDV in the type of addressing and network organization it uses to perform routing. Instead of the "flat" network, the protocol aggregates the nodes in the network to form clusters with one node in the cluster being assigned the responsibility of being the cluster-head. The cluster-head controls the communication in the cluster and communication with other clusters is done via gateway nodes, which are nodes that are members of two or more clusters. The authors of the protocol state that this architecture provides a convenient framework for the development of essential features such as code separation among clusters, channel access, routing and bandwidth allocation [Chiang97].

The protocol uses a distributed clustering algorithm to select specific nodes as cluster-heads. All nodes within the transmission range of a cluster-head belong to that cluster. It is paramount that the clustering algorithm used provides stability in the state of the routing architecture, since frequent cluster-head changes adversely affect the performance of other sub-tasks of the protocol such as scheduling and resource allocation, which rely on it. To provide this stability the authors propose a clustering algorithm called Least Cluster-head

Change (LCC) [Chiang97]. This algorithm stipulates only two conditions in which the cluster-head should change:

- Two cluster-heads come within range of each other
- A node becomes disconnected from any cluster

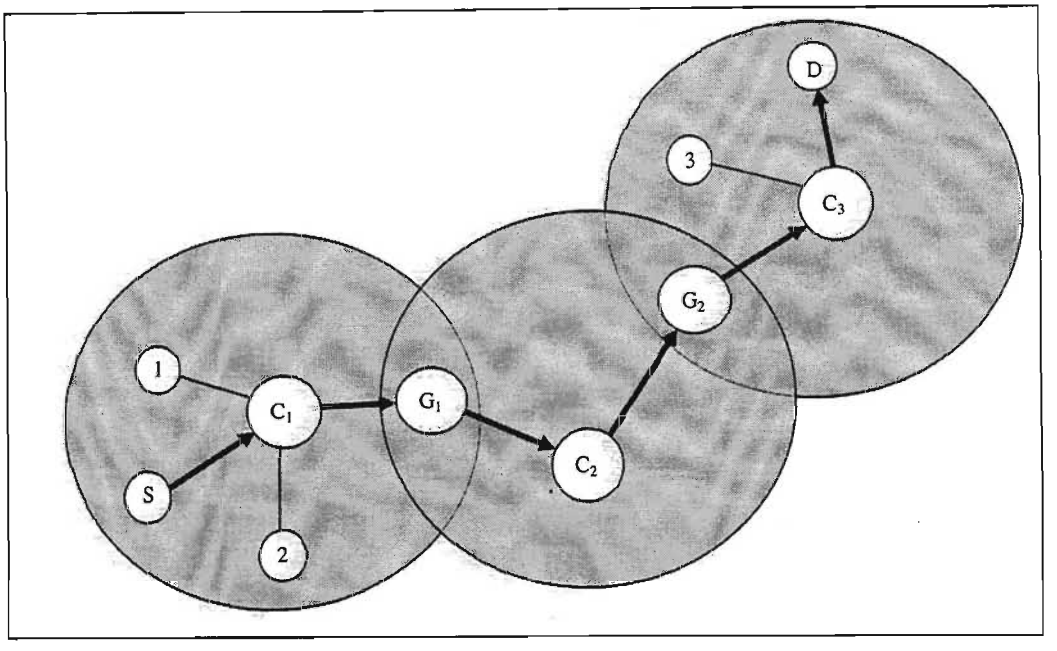


Figure 2.1: The routing architecture in CGSR

CGSR has much of the same overhead as DSDV, which is used as the underlying routing scheme. Its novelty is in the use of a hierarchical cluster-head to gateway routing approach to route traffic from source to destination. A packet being sent by a node in the network is first sent to the nodes designated cluster-head. The cluster-head forwards the packet to the gateway that will take the packet towards the cluster-head of the destination node. Thus the packet is routed alternatively through cluster-head and gateways ($C_1G_1C_2G_2C_3$ in Figure 2.1), until the destination cluster-head is reached. The destination cluster-head then passes the packet to the destination node. Thus each node in the network maintains two tables: a cluster member table which associates a destination node to a cluster-head and a routing table which shows the next hop to reach the destination cluster. These tables are periodically broadcast in a similar way to the DSDV algorithm.

The clustering scheme of this protocol makes it easy to implement enhancements such as power control, code scheduling and Medium access control between communicating nodes. However, there is additional overhead incurred in the creation and maintenance of the

clustered architecture. The clusterhead and gateway nodes have higher computation and communication burden than other nodes in the network. This could lead to the creation of critical nodes, the failure of which could result in poor network performance.

2.5.3 Global State Routing (GSR)

GSR [Chen98] is based on the link-state philosophy where each node in the network maintains the knowledge of the full network topology. The increase in the popularity of link-state protocols over distance vector schemes was primarily due to their ability to converge quickly and avoid routing loops. However, with the bandwidth constrained links in the ad-hoc network, the flooding technique used by traditional link state schemes would be far too resource intensive. GSR avoids the flooding by periodically exchanging its link-state table with its neighbours only.

Each node in the network stores the knowledge of the full network topology by means of a *neighbour* list and three tables. The *topology* table contains, for each destination entry, the link state information reported by the node and the timestamp (in the form of a sequence number) indicating when the destination node generated this information. The *next hop* table specifies the next hop on route to the destination, while the *distance* table holds the distance of the shortest path to the destination node.

On start-up a node has an empty neighbour list. This list is populated with information gathered by studying the sender field of each packet in the node's inbound queue. The routing information that is periodically broadcast by the neighbouring nodes, contains link state information which is used by the node to fill its own topology table. The protocol ensures that a node maintains the most-up-to-date link-state information by means of time-stamping each route entry. This is done by means of the destination sequence numbering mechanism, a concept borrowed from DSDV. The updated topology table is used to calculate the shortest path from a node to all the other nodes in the network. This computation is done using Dijkstra's shortest path algorithm [Berts87]. The next hop and the distance tables are then filled with the appropriate information.

The modified link state algorithm is better than the traditional distance vector and link state schemes. GSR displays faster convergence as it uses more accurate routing information and it avoids the network wide flooding of its link state table. However the drawbacks to the

protocol are the large size in the routing update message and the latency of the link state change propagation, which is highly dependent on the update period used by the algorithm.

2.5.4 Fisheye State Routing (FSR)

The Fisheye State Routing [Iwata99] is an implicit hierarchical version of the GSR scheme. The authors of FSR observed that the large packet size of the protocol update messages in GSR incurred an unacceptable amount of overhead in the resource constrained ad-hoc environment. Their aim was to find a way to reduce the overhead without seriously sacrificing on the routing accuracy. It was achieved by using a novel "fisheye" technique that was developed to reduce the amount of information required to represent graphical data [Iwata99]. FSR remains a link-state protocol but it maintains more accurate distance and path quality information on nodes closer to the current node than nodes that are far away. The level of detail decreases as the *logical* distance of the nodes from the current node increases.

FSR uses the fisheye technique in the method it uses to disseminate link state information. The protocol uses the same data structures as GSR and its periodic route updates also only go as far as a node's neighbours. However, the fisheye technique is used to reduce the size of the update packet. The protocol assumes hypothetical circles around each node in the network as shown in Figure 2.2. Each "circle" envelopes nodes that are reachable within a certain number of hops and these nodes define the scope of the source node. The decrease in the update packet size is achieved by using different exchange periods for different entries in the route cache at a node. Entries for nodes within a smaller scope are propagated to neighbours with the highest frequency while the remaining entries are broadcast at a lower rate.

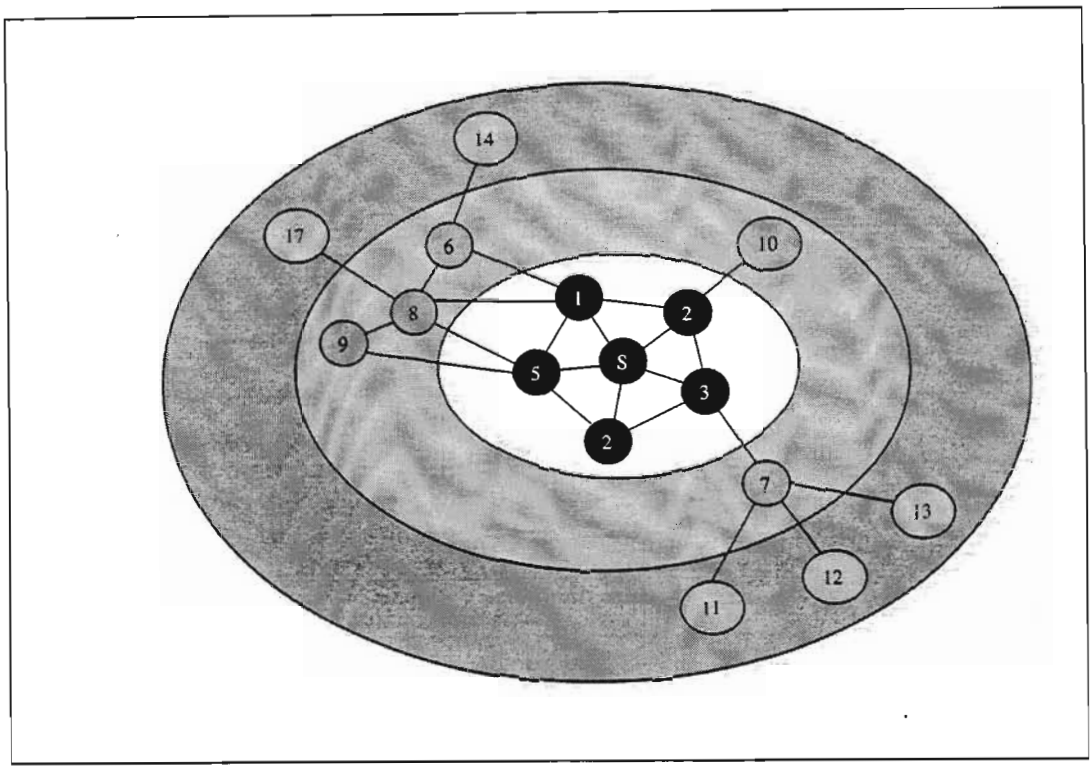


Figure 2.2: The scope architecture in FSR

This strategy ensures that topology changes for nodes near a node are propagated quickly, while there are latencies in transmitting information on the state of connections with nodes that are far away. This should not be seen as a disadvantage since, although inaccurate information is initially used for distant nodes, the packet is still accurately routed because the level of accuracy increases as the packet approaches the destination node.

2.6 Reactive routing protocols

Reactive routing protocols have also coined the term on-demand protocols since these routing schemes create and maintain routes only when such routers are in demand. There is no periodic update of routing information between the nodes in the network with reactive protocols and so it is most often the case that a requested route is not known apriori. When required a node in the network requiring a route has to perform some type of route discovery to find a suitable route. Once a route is found, the node can begin transmission of data packets towards the intended destination. If the conditions in the network remain similar to the instant the route discovery process created the route, the route can be used without disruption as long as it is needed. If however conditions do change, due to link breakages or

mobility, the source node has to repair the route or re-create it. Thus reactive routing protocols, in general, have a two phase operation: a route discovery phase and a route maintenance phase. The motivation in the design of this ad-hoc routing philosophy is to reduce the protocol routing overhead created by periodic updates of the table-driven schemes. The proactive schemes also use significant resources to maintain certain routes which have the possibility of never being used. This is avoided by the reactive schemes which only create and maintain routes when they are needed.

2.6.1 Dynamic Source Routing (DSR)

The Dynamic Source Routing (DSR) protocol is based on the concept of source routing in which a source node determines the complete sequence of nodes through which to forward data packets. A node sending a packet to a destination node explicitly lists the route to the destination in the header of the packet. The list identifies each "next hop" node that should be taken in order to get from the source to the destination. Each node in the network maintains a route cache that contains source routes that the node is aware of. The route cache is continually updated with old unused routes being purged and new routes inserted as a node learns about them.

Characteristic of an on-demand algorithm DSR has two procedures: route discovery and route maintenance. When a node requires a route to a destination its first action is to consult its route cache to determine if it already contains a route to the destination. If an unexpired route is found, the route is used for data transmission. However, if there is no route in the nodes cache, it initiates a route discovery process by generating and broadcasting a route request (RREQ) packet across the network. The RREQ packet contains the IP addresses of the source and destination nodes, a unique route request ID and a route record which will contain the addresses of the sequence of nodes for the route. To limit the number of route requests traversing the network, each node only processes a route request once. The source nodes address and the unique route request ID are temporarily cached and if the node receives another request with the same details it silently drops the packet.

When an intermediate node (any node other than the source and destination) receives a route request that it can process, its first action is to determine if its address is in the packet's route record. If the route record already contains the nodes address a routing loop has occurred and the packet is dropped. If there is no routing loop, the intermediate node inspects its route cache for an unexpired route to the destination. It generates and sends a route reply (RREP)

packet to the source node if such a route is found. If a route is not found in the route cache, the intermediate node adds its own address to the route record in the RREQ and broadcasts it to its neighbours. The route request packet is thus flooded in the network until either an intermediate node or the destination node itself replies to it. This process is shown in Figure 2.3. Note that the replying node, given a choice between two routes, chooses the route with the least hop count. The route reply packet is routed back to the source node by reversing the order of the next hops in the route record of the original route request packet. The route reply that is sent back to the source node with the route record included. This can be seen in Figure 2.4.

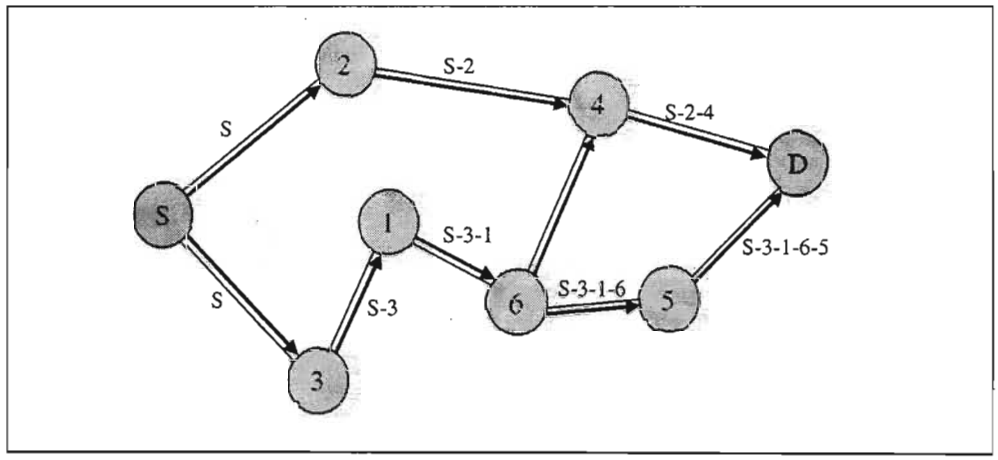


Figure 2.3: Flooding of the route request to discover the route record in DSR

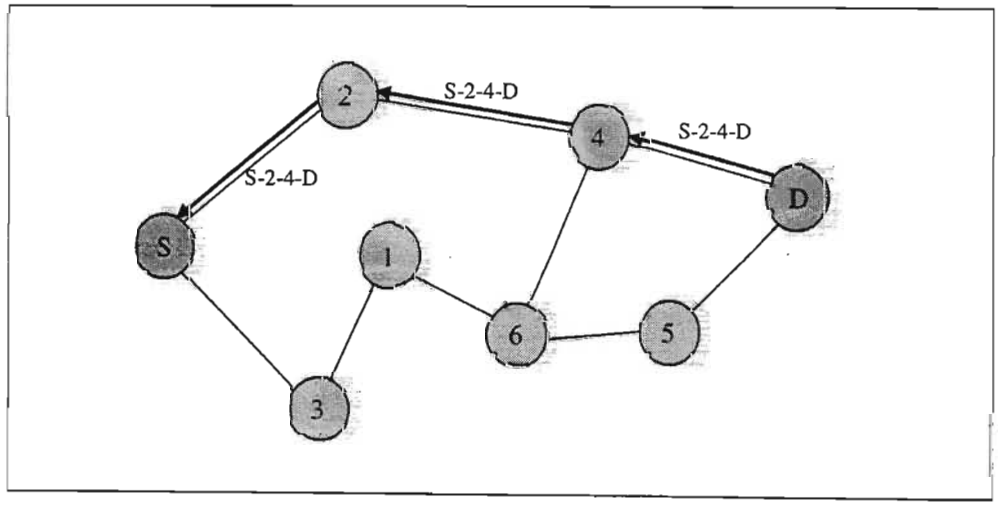


Figure 2.4: Propagation of Route Reply in DSR

The route maintenance procedure of the protocol monitors the operation of a route and is responsible for making the source node aware of any errors. If an intermediate node detects a failure to transmit a data packet to a downstream link it generates a route error (RERR) packet. When a route error is received by a node, the node in the route error is removed from the nodes route cache and all routes containing that node are truncated at that point. Link errors are detected by means of link layer feedback and/or data acknowledgements.

One of the many optimisations proposed for DSR is the operation of the protocol in a "promiscuous" mode. In this mode the network protocol receives all packets (RREQ, RREP; and RERR) that the node's wireless interface overhears. These packets are studied for useful source routes or route error messages after which they are discarded.

There are a few drawbacks to the operation of DSR. The protocol includes the entire route information in the data packet header which creates significant overhead as the route length increases. DSR also relies heavily on route caches to avoid repeated route discoveries. However, using stale route caches can adversely affect the performance of the protocol. If the routes are not updated a source node may use cached routes which are invalid due to mobility in the network. Intermediate nodes sending route replies using stale cached route could cause pollution of cached routes maintained at other nodes in the network.

2.6.2 Ad-hoc On-demand Distance Vector (AODV)

AODV [Royer99b] is another example of an on-demand route acquisition system where a route between two hosts in an ad-hoc network is only created when they wish to communicate. The protocol is similar to DSR in the route acquisition and route maintenance mechanisms. However the two protocols differ in that AODV stores the route information in a distributed fashion at each node on the route while DSR includes the route information in the header packet of each data packet that is transmitted. AODV maintains loop free routes at all times using sequence numbers. This mechanism is imported from the DSDV routing algorithm. Each node in the network maintains its own monotonically increasing sequence number, which is incremented whenever the node generates and sends a route request packet. The sequence number is used as a form of logical time-stamping and ensures that the most recent route is selected in the route discovery procedure.

AODV is a pure on demand algorithm and uses the route request/route reply cycle to discover routes to new destinations. The three main message types used by the algorithm are

route requests (RREQ), route reply (RREP) and route errors (RERR). The protocol comes into action whenever a new route is needed to a destination. AODV utilizes an enhanced version of the traditional route table to store and maintain routes to destination nodes. These routes, however, are cached only as long as they are being actively used. Thus in most cases routes to destinations are not known prior to a route being requested. The protocol initiates a *route discovery process* by generating and transmitting a RREQ packet. Each route request packet is uniquely identified by the source IP address and a broadcast ID. The packet is broadcast to the source's neighbouring nodes. A node receiving the route request first checks to determine if it has recently processed a RREQ with the same source IP and broadcast ID. If a match is found the RREQ is silently discarded. If on the other hand the request is new to the node, it records a *reverse route* entry to the source node in its route table (or activates an old one). If the node is not the destination node or an intermediate node with a current route to the destination, it broadcasts the route request packet to its neighbours. This process continues until a node is reached which meets the two conditions. In this manner the RREQ packet is disseminated using a network wide flood until a route is found (refer to Figure 2.5). The reverse path setup at each intermediate node is shown in Figure 2.6. The destination node D does not accept the route request packet from node 7 since it has already received a request with the same details from node 5.

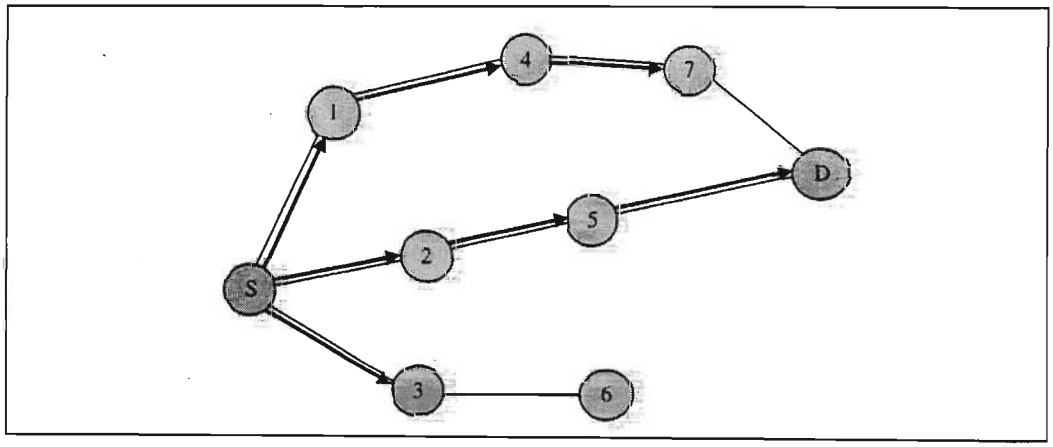


Figure 2.5: Network flood of route request packets in AODV

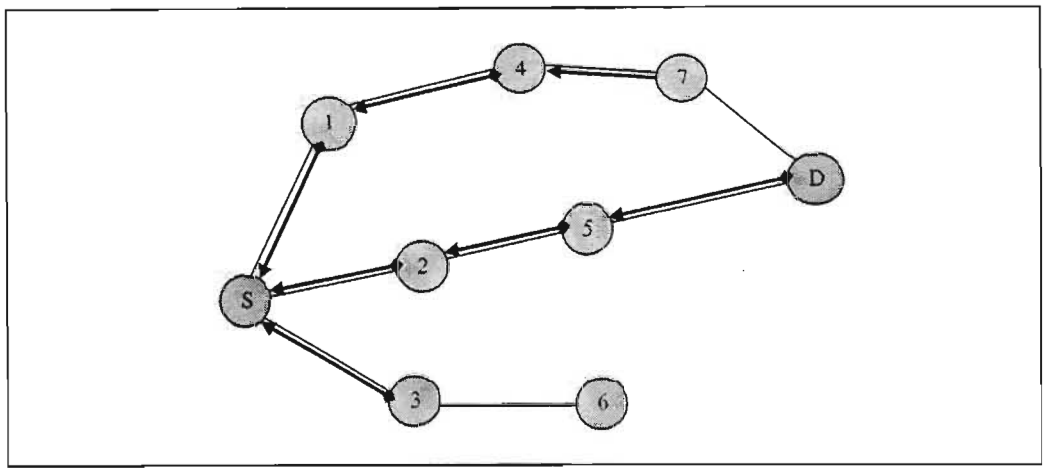


Figure 2.6: Reverse Path setup in AODV

The processing of the RREQ is different depending on whether the node is the destination node itself or an intermediate node with a current, active route to the destination. The decision on how current or "fresh" a route is, is determined by the value of the sequence number associated with the route entry. If the destination sequence number of the entry at the intermediate node is higher than that contained in the RREQ, the route is considered to be fresher. The intermediate node is permitted to reply to the route if such a condition is met. The destination node simply replies to the route request by generating and transmitting a route reply (RREP) packet. As seen in Figure 2.6, by the time the RREQ arrives at a node that can provide a route to the destination (or the destination itself), a reverse route is established to the source of the route request. The route reply that is transmitted travels along this route to get back to the source node. Each node through which the RREP packet hops sets up a forward pointer to the node from which the packet was received. The forward path set up can be seen in Figure 2.7. The hop count field in the RREP is incremented by each intermediate node that processes the message. When the reply reaches the source node, the hop count value presents the distance in terms of hop count between the source and the destination. Once the route reply arrives the route discovery process is terminated and the source can begin to send the packets that were queued for the destination.

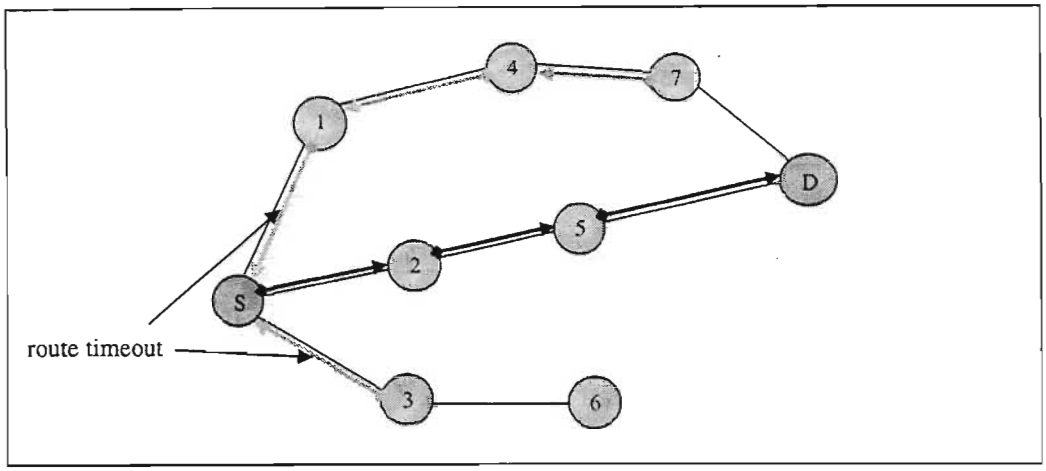


Figure 2.7: Forward Path setup in AODV

Once a route has been discovered, it is maintained as long as it is needed by the source node. If the network topology does not change there is little action on the part of the protocol. However route breakage, due to node mobility or link layer failure, results in execution of route maintenance procedures. There is an option in the protocol enabling the intermediate node to repair the route locally in a process termed *local repair* [Perkin02]. If this option is not used or is not suitable, the intermediate node sends out a route error (RERR) message to the affected source node. One of the data structures in an entry in the protocols route table is a list of neighbours that use the current node as the next hop to get to the destination. This is known as the *precursor list*. Thus when an intermediate node detects a route breakage to a destination, it transmits the RERR to all its upstream neighbours in its precursor list. The message is unicast if the list contains only one neighbour, otherwise it is broadcast. When a neighbouring node receives the RERR, it marks the route to the destination as unreachable and transmits a route error to the nodes in its precursor list. After the reception of a RERR message the source node re-initiates a route discovery if the route is still required.

2.6.3 Temporally Ordered Routing Algorithm (TORA)

The Temporally Ordered Routing Algorithm (TORA) [Park97] is based on the concept of link reversal. It was designed with the idea of reducing algorithmic reaction to topological change in a highly mobile ad-hoc network. It is a source initiated protocol that provides multiple routes between source and destination nodes. It detects network partitions quickly and reacts by deleting of invalid routes.

There are three basic functions in the operation of TORA:

i. Route creation
ii. Route maintenance
iii. Route erasure

The route creation process establishes a sequence of directed links from the source to the destination node. A logically separate route creation process is run by the source node for each destination with which it communicates. The algorithm creates a Directed Acyclic Graph (DAG) rooted at the destination [Park97]. In this routing structure, each node in the route is assigned a *height* metric and the links between neighbouring nodes in the DAG are assigned to be either upstream or downstream, depending on the height metric of a node.
The height metric is a quintuple comprising of the elements:

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

The first three elements of the quintuple collectively represent the reference level of the height metric while the remaining two determine (for each node) the difference with respect to the reference level. It should be noted that "time" is an important factor in the routing operation with the value of the element storing the time of a link failure. The authors of the protocol rely on the fact that all the nodes in the network will have access to synchronized time. This could possibly be provided external time source such as a GPS module.

The route creation is achieved through a query/response cycle. The route creation algorithm starts with the source node broadcasting a QRY packet searching for a route to the destination. This message is flooded across the network until the destination is reached. The destination node then responds by sending a UPD message. The height of the destination is set to 0 and all other nodes are assigned a NULL height value. This process is shown in Figure 2.8. A node that receives the UPD packet sets its height value to one more than that of the node from which it received the message. A node with a higher height is considered upstream. The values in brackets next to each node in Figure 2.8 represent the height value. The first value is reference level and the second is the delta with respect to the reference. Note that node 2 does not accept the QRY packet for the destination from node 3 and it has already processed a similar request from node 1.

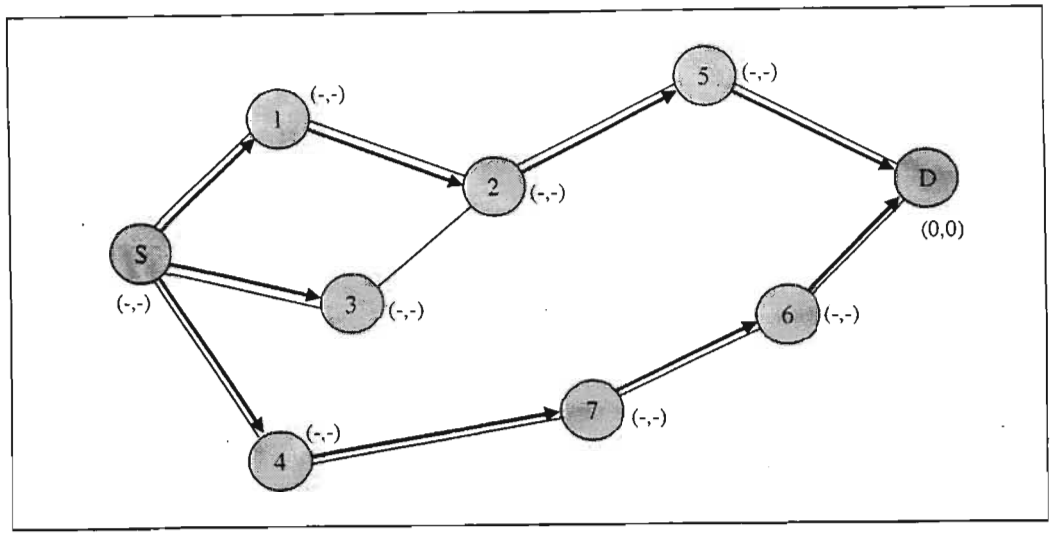


Figure 2.8: Flooding of QRY packet in the network

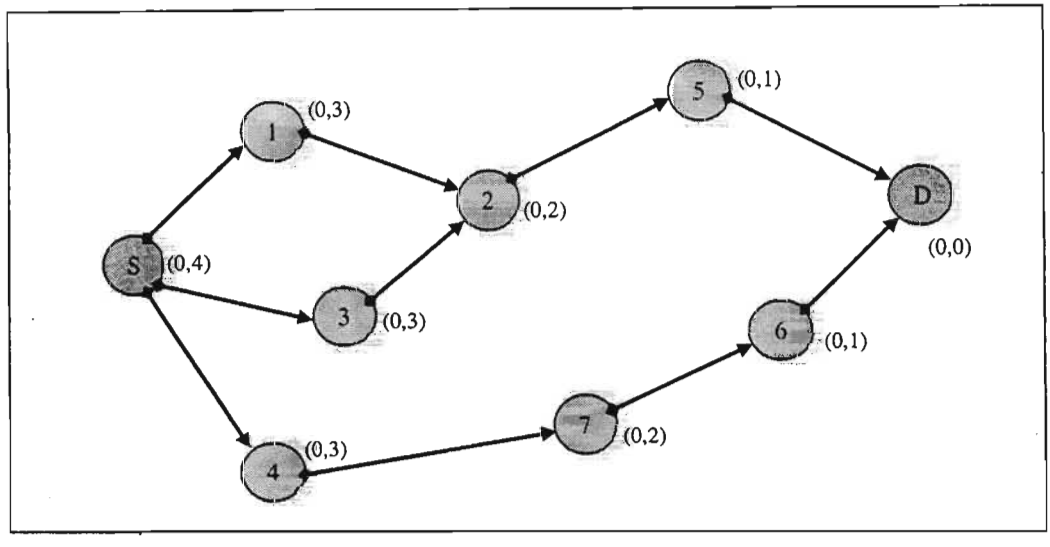


Figure 2.9: The setting up of the DAG in the network

When the UPD packet reaches the source, all the nodes between the source and destination have been assigned a height value (refer to Figure 2.9) and the source chooses the shortest route for data packet transmissions. This continues unhindered as long as the route is needed or until there is topological change. The novelty of the protocol is in the way it reacts to route breakages due to network mobility. When a node in the network moves, the DAG route

is broken and route maintenance function of the protocol is executed to re-establish a DAG for the same destination. If an intermediate node in the DAG loses its last downstream link due to a link failure, it selects a new global maximum height by defining a new reference level.

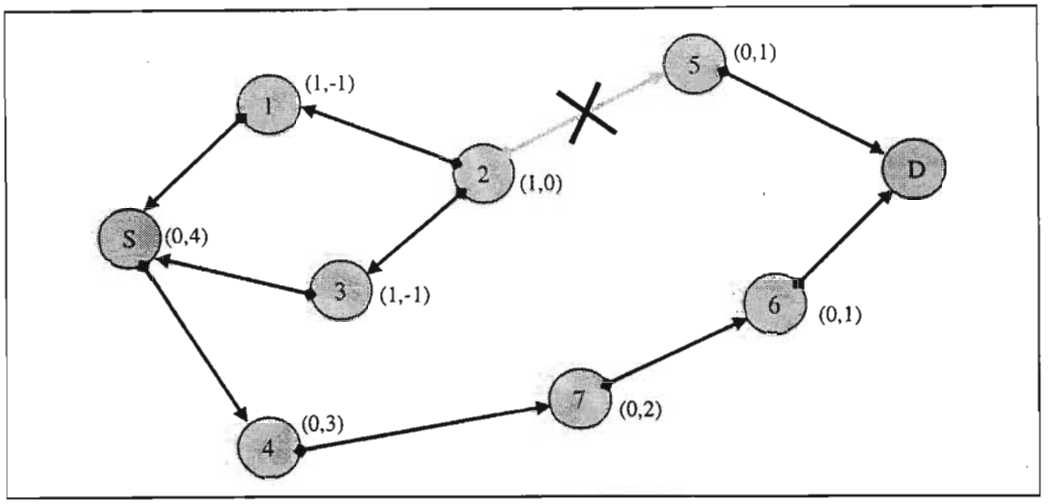


Figure 2.10: Re-establishment of a route upon a link failure

In Figure 2.10 the link break between nodes 2 and 5 results in node 2 selecting a new reference level. This results in link reversals which result in other nodes losing their last downstream links to the destination. These nodes perform a partial link reversal to reflect the changes and adapt to the new reference level. The link breakage in Figure 2.10 only affects a few nodes, thus minimizing protocol reaction to the change. However, in the event of losing all downstream links to the destination, the source node re-establishes a route by means of a new route discovery. The protocols route erasure phase consists of a network flood of broadcast clear packet (CLR) which erases invalid routes in the network.

There is the possibility of oscillations occurring in the operation of TORA. This is caused especially when there are multiple sets of coordinating nodes that are concurrently detecting partitions, erasing routes and creating new routes based on each other. These oscillations are similar to the "count to infinity" problem experienced by traditional distance vector algorithms, except that the problem is temporary and route convergence occurs eventually.

2.6.4 Associativity Based Routing (ABR)

The Associativity Based Routing[1] algorithm [Toh96] was one of the first ad-hoc routing algorithms to consider a routing metric other than the smallest hop count. It defined a new metric called the *degree of association stability*. This associativity is a measure of a nodes connectivity relationship with its neighbours over time and space [Toh02]. Each node in the network periodically transmits a beacon to its neighbours signifying its presence. A node caches an entry for each neighbour which records the number of beacons received. This information is stored in a variable termed 'associativity tick' that is incremented each time a beacon is received. A high associativity tick value for a neighbouring node implies a low state of mobility for that node. A stable link with a neighbour provides an ideal opportunity to select the node for routing purposes. The protocol introduces other Quality of Service (QoS) parameters such as load, signal strength and battery life in addition to the associativity ticks to determine the degree of routing stability. The routes determined using this metric are expected to be long-lived routes. These routes however are not necessarily the shortest in terms of hop count between the source and destination. The protocol breaks the traditional paradigm, which holds that the shortest path is ideal. Thus, although a longer path is sometimes chosen, with the high degree of stability, the route will be maintained with lower probability of having to execute route recoveries.

The operation of ABR can be divided into three phases [Toh96]:

- Route discovery
- Route reconstruction
- Route deletion

The route discovery phase involves a query and wait-reply (BQ-REPLY) cycle where a node desiring a route broadcasts a BQ (Broadcast Query) message to find nodes that have routes to the destination. Intermediate nodes (IN), on the reception of the BQ message, append their IP address and associativity ticks with its neighbours (as well as other stability parameters) before broadcasting it to its neighbours. The next succeeding IN erases its upstream node neighbours associativity tick entries and retains the entries that relate itself and the upstream node. Thus each BQ packet that arrives at the destination node contains addresses of the intermediate nodes (a record of the path taken) and their associativity ticks (a record of the

[1] This protocol has received a United States patent for its design

stability state of the INs). The destination node, after waiting an appropriate amount of time to discover all possible routes and their qualities, selects the best route via which to send a REPLY.

The protocol's route reconstruction phase (RRC) is used to deal with network topological change. Different functionality is used depending on which node(s) along the route move. Route breakages caused by the movement of the source node result in the execution of a new BQ-REPLY process. The source node also sends a route notification message (RN[1]) [Toh96] to all the downstream nodes which prompt them to erase the entries associated with the route. In the case of the movement of the destination node, the immediate upstream node erases its route and performs a localized query (LQ[H]). H in the query refers to the hop count from the upstream node to the destination. If a partial route to the destination is not found within a timeout period, the localized query process backtracks to the next upstream node. If this backtracking continues for more than halfway to the source, a notification packet is sent to the source. Reception of such a packet results in the source initiating a new BQ-REPLY cycle.

When a route that has been used for data transmissions is no longer needed, the on-demand algorithm specifies that the source initiate a route deletion (RD) broadcast. All the nodes along the route that receive this message appropriately purge the route entry from their routing table. The RD is disseminated as a full broadcast as opposed to a directed broadcast as the source may be unaware of route changes that occurred during RRCs.

2.7 Hybrid routing protocols

There are certain advantages to both proactive and reactive routing. The hybrid class of ad-hoc routing protocols attempt to combine both the techniques in order to produce a more efficient algorithm.

2.7.1 Zone Routing Protocol (ZRP)

The Zone Routing Protocol (ZRP) [Haas99] is a hybrid routing protocol that uses both proactive and reactive routing mechanisms. The protocol divides the network into several zones and the type of routing used depends on whether the destination node is within a source node's zone or outside of it. A zone is characterised by a parameter called *zone radius*, which is measured in terms of hops from a node. The zone radius is an adjustable

variable and provides an optimisation parameter for a particular ad-hoc network application. ZRP uses different routing schemes for routing within a zone and routing between zones. Nodes use a proactive algorithm to maintain routing information for nodes within their zone and reactively discover routes for destinations outside their zone. The protocol refers to the two routing mechanisms as Intrazone Routing Protocol (IARP) and Interzone Routing Protocol (IARP).

IARP provides the proactive functionality of ZRP. It operates within a zone and is responsible for maintaining routes to nodes in the zone through periodic routing table updates. The authors of ZRP do not specify a particular type of proactive algorithm and thus it can be either a distance vector or a link-state algorithm.

IERP forms the reactive routing component of ZRP and is used when routes are required across zones. The protocol refers to nodes that are exactly *zone radius* number of hops away from the source node as *peripheral nodes*. When a node requires a route to a destination that is not within its zone, it employs a ZRP service called bordercasting. In this process the source node, instead of broadcasting route queries to all its neighbours, directs the query to its peripheral nodes. The peripheral nodes in turn forward the request if the destination desired is not within their routing zone. This process continues until the desired zone is reached where either the destination node or some other node within the zone responds to the route request with a reply.

The critical parameter that affects the performance of ZRP is the zone radius. If the radius is chosen to be too large IARP will dominate and the protocol in general will experience the negative effects of proactive protocols. On the other hand many small zones will result in IERP being used frequently, which will mean that the protocol will have to endure the negative effects of reactive routing protocols. Thus it is imperative that the appropriate zone radius is selected for a particular network.

2.8 Conclusion

The routing protocols proposed thus far can be broadly categorised into two classes depending on the mechanism used to gather routing information. The proactive routing protocols are modified versions of the distance vector and link state algorithms used in the wired networks. These protocols maintain updated routing tables that contain routes to all reachable nodes in the network. This information is disseminated in the network by nodes broadcasting their routing tables to their neighbours. The reactive ad-hoc protocols create and maintain routes to destinations only when they are needed. These protocols repair or re-establish routes if the routes break when the network topology changes. The two classes of ad-hoc algorithms, and the algorithms within the classes, have their own advantages and disadvantages. Proactive routing protocols maintain routes to destinations even before they are requested and should theoretically route faster than on-demand algorithms that have delays due to the route discovery process. However, the periodic update messages used by the table-driven schemes create considerable routing overhead leading to congestion which negatively affects data packet throughput and creates delays. Reactive algorithms only create routes on-demand and thus save on the resources used for period update messages. These observations are, however, derived from brief qualitative comparison studies of the proposed protocols. In the task of selecting a routing algorithm for an ad-hoc network application, such studies are limited in their usefulness. What is required is a thorough quantitative performance analysis and comparison of the protocols on common, realistic platform. This is the theme of the work presented in Chapter 3 of this dissertation.

Chapter 3

Simulation of Ad-hoc Routing Protocols Using Network Simulator (NS-2)

3.1 Introduction

In ad-hoc network research, the design of routing protocols has received much attention recently. This is due the challenge the design poses with the dynamic nature of ad-hoc networks. In addition ad-hoc network hosts are mobile computers which have limited resources such as bandwidth and energy. A routing protocol must manage the network routing information in a quick and efficient manner, yet it must do so while remaining conservative its use of the resources available to the host. There have been many ad-hoc routing protocols proposed recently as described in Chapter 2. Each protocol is unique in its operation and characteristics. The aim of this research work is to develop a routing protocol for a Positional Communication System (PCS) network. Thus the objective of the work in this chapter is to compare four proposed algorithms in order to determine which specific protocol or type of protocol is suited for such an ad-hoc network application.

There are comparative studies of ad-hoc routing protocols that have presented only qualitative performance evaluations [Royer99a]. These studies compared proposed protocols using qualitative measures such as time and complexity in the operation of the protocols. However, such comparisons only present the features, differences and characteristics of the protocols. In [Royer99a] it was concluded that each protocol that is proposed has its own advantages and disadvantages and is suited for certain situations. Thus from such studies it is a difficult task to determine which algorithm or class of algorithm is best suited for all ad-hoc routing scenarios. In order to achieve this, what is needed is a common platform that would test different protocols under many similar repeatable scenarios. The development of simulation packages enables researchers to conduct such comparative studies.

In order to compare ad-hoc routing protocols in a simulation environment, what is needed is a packet level simulation using realistic scenarios. It is important to develop a simulation environment that, as far as possible, models the system being simulated. A measure of the

actual performance of a protocol can only be determined on such a simulation platform. The Network Simulator (NS-2) simulation package [NS-203] was designed with this in mind. It was initially developed for conducting simulations involving local and wide area wired networks. However recent work [Broch98] has extended the functionality of the package to include wireless and ad-hoc networks. Thus today it is possible to complete quantitative performance analyses of different ad-hoc routing protocols using NS-2. Using such information and common performance criteria, different protocols can be compared in a variety of simulation scenarios.

The NS-2 simulation package is presented in this chapter. There is a brief description of the wireless models used in NS-2, after which the simulation of four prominent ad-hoc routing protocols is presented. The aim of the simulations was to determine which of the four protocols performed well under a variety of network conditions.

The NS-2 package is introduced in Section 3.2. The wireless extensions of NS-2 are presented in Section 3.3. The model of the mobile node used in the simulator is detailed in Section 3.3, along with the physical and data link layers of NS-2. The protocols simulated are given in Section 3.4, with the motivation for the simulation work conducted in this chapter. Section 3.5 describes the simulation setup. The metrics used in the comparative study are detailed in Section 3.6. Finally Section 3.7 presents the results from the simulations. The chapter concludes by mentioning the findings of the comparative study.

3.2 Network Simulator (NS-2) Introduction

The Network Simulator [NS-203] is a simulation package developed at the University of California (Berkley) and USC ISI as part of the Virtual InterNet Testbed (VINT) [Vint03]. It is an object oriented, discrete event driven simulator developed to support network communication research in terms of protocol design, prototyping and comparisons. It can be used for simulations of terrestrial, satellite and wireless networks. The work by the Carnegie Mellon University (CMU) Monarch group [Broch98] extended the simulator to add wireless models. This allowed the package to be used in the simulation of ad-hoc networks. The simulator's software is mostly object oriented code, which has resulted in its simple and modular structure. It is for this reason, and its ease of extendibility, that has made the simulation package popular in the network communications research community.

NS-2 has a split design in its implementation. A scripting tool (OTcl) is used as the "front-end" and is used to write scripts for the creation of simulation scenarios such as node movements and network traffic. It is also used to configure objects and schedule events in the simulation. The "back-end" operation of NS-2 such as event handling and per-packet processing is executed using C++ objects. Thus, the more complex, repetitive and time consuming functionality is undertaken in this environment. The network routing protocols are implemented as C++ objects and are included in the NS-2 package. The user of the simulator can write such objects which can be added to the architecture of the simulator. This flexibility enables the user to easily enhance the simulation package as needed.

The logging of the output data from the simulations is done in the simulators *trace files*. These files are created with each instance of the simulation and log detailed information of packets generated during the simulation. The output files are used for post-processing, generally done with a scripting language or an analysis tool. In addition, NS-2 generates special trace files that are used by the visualization tool included in the distribution. The Network Animator (NAM) [NS-203] is a Tcl/Tk based animation tool for the packet level animation of a simulation run.

3.3 Mobile Networking in NS-2

As mentioned previously, mobile networking in NS-2 is based on the work done by the CMU Monarch group [Broch98]. This work extended the simulator to include structures for mobile hosts being connected to wireless channels. The "wired node" structure was modified to give each node the capacity to be mobile in a network topography. A physical layer and the IEEE 802.11b Medium Access Control (MAC) protocol was implemented. These extensions allow realistic simulation of wireless ad-hoc networks.

3.3.1 The mobile node model

The software of NS-2 is made up of objects, derived from base classes, which interact with other objects to give the simulator functionality. Complex network objects such as nodes and links are made up of basic network objects such as connectors and classifiers [Fall03]. The model for a mobile node consists of a compound object built from a variety of NS-2 objects. The mobile node is derived from the base class which creates *Node* objects for a wired network. It has the basic features of the *Node* object with the added functionalities such as the ability to move within a given topology and the ability to receive and transmit data over a

wireless channel. The schematic of the mobile node can be seen in Figure 3.1. An explanation of each element in the schematic follows the diagram. Each element is an object which combines to form the compound object.

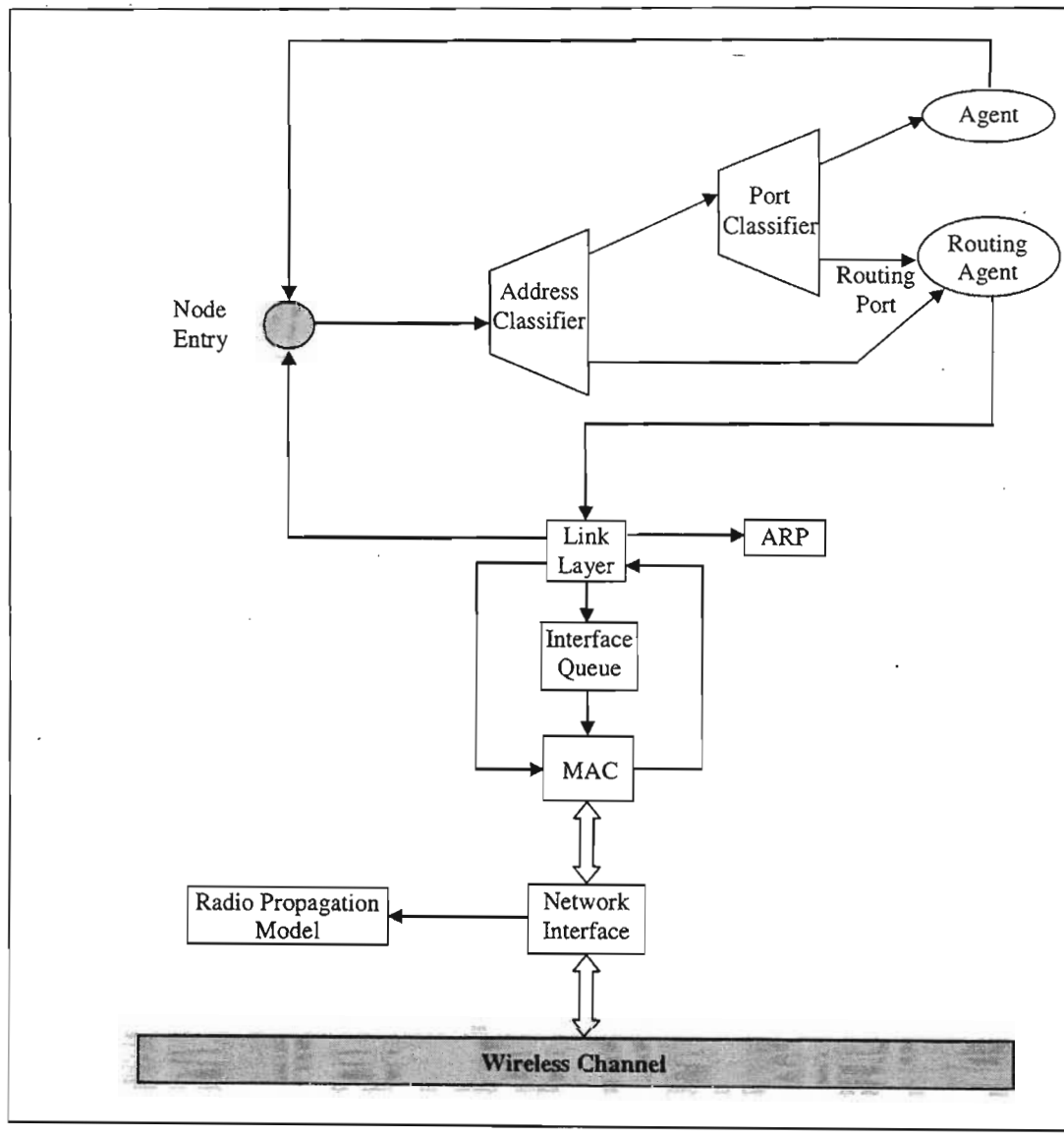


Figure 3.1: Schematic of a mobile node in NS-2.

3.3.1.1 Address Classifier

One of the basic network components in NS-2 is the classifier. It is a switching object that has multiple possible output data paths. The address classifier in the mobile node is responsible for passing a packet to either the port classifier or the routing agent. The packet

is handed to the port classifier if its destination is the current node. In all other cases the packet is passed to the default target of the classifier, which is the node's routing agent. The routing agent then makes a decision as to the fate of the packet. It will either forward the packet or drop it.

3.3.1.2 Port Classifier

This element determines which agent (protocol) attached to the node should receive a packet by inspecting the port number on the packet. Possible protocol agents are UDP, TCP or an ad-hoc routing protocol agent.

3.3.1.3 Routing Agent

In a mobile node the routing agent is responsible for both the gathering of the routing information (route table management) and packet forwarding. In the model of the wired (stationary) node the classifier executes the forwarding using information provided to it by the routing functionality at the node. The routing functionality determines routes to destinations and constantly updates the information held at the classifier. The routing agent in the mobile node processes a packet and, if a route is found, sets the *next hop* field to indicate the next hop that the packet should take in order to reach the destination.

3.3.1.4 Link layer

The link layer is given the task of converting network address (IP address) of a host to a hardware address. It achieves this by consulting an ARP module. The link layer also prepares the packets to be transmitted onto the wireless channel.

3.3.1.5 ARP module

Most routing protocols operate at the network layer of the OSI model, using IP addressing. Therefore, NS-2 includes an Address Resolution Protocol [Plumm82] implementation to resolve IP addresses to link layer addresses. The implementation of the module was modeled after the BSD Unix implementation of the ARP [Wrigh95].

3.3.1.6 Interface queue

This module temporarily stores packets that are to be transmitted by a host. There are certain parameters of the queue that are specified through the simulation setup scripts. Parameters include the queue length and type of queue. The type most commonly used in the simulation of ad-hoc routing protocols is the DropTail/PriQueue [Fall03] which allows routing protocol packets to be given higher priority. The packet is buffered in this module until a signal is received from the MAC layer.

3.3.1.7 Medium Access Control (MAC)

The MAC layer in the wireless model manages the access to the wireless channel. During the transmission of a packet, when the MAC layer acquires access to the channel, it retrieves the packet from the Interface queue and passes it to the network interface module. In the reception process, the MAC layer is handed a packet only after it has been correctly received. A packet is correctly received i.e. error free if the power level at which it was received is above the receive threshold [Broch98].

3.3.1.8 Network Interface

This module sends and receives packets over the channel. When a packet is transmitted, the network interface at the node passes to the appropriate physical channel object. This object then computes the propagation delay[1] from the sender to all the network interfaces attached to the channel. The delay calculated determines the instance at which a "packet reception" event occurs at the receiving interfaces. The arrival of the packet does not imply that a packet will be correctly received at all the nodes. The next module determines this once a packet is received.

3.3.1.9 Radio Propagation Model

The task of this module is to determine whether a packet should be received by the network interface. This decision is dependent received signal strength of the packet. The lower limit in the signal strength is determined by the settings of the network interface and the selected radio propagation model (Refer to Section 3.3.2).

---

[1] This depends on parameters such as distance between the nodes.

3.3.1.10 Wireless Channel

All the packets that are transmitted or received by the nodes in a simulation are distributed over the wireless channel, and this module of the simulator is responsible for such functionality.

3.3.2 The Physical Layer Model

The physical layer in NS-2 is modeled using radio propagation models. These models are used to calculate the signal strength of received packets during wireless simulations. Currently there are three radio propagation models implemented in the simulator:

- The free space model
- The two-ray ground reflection model
- Shadowing

These models are based on theories in radio engineering and physics. The first two models are described here, as they are used in the simulation work done in this work.

3.3.2.1 Free-space model

This model assumes a single, clear line of path between the transmitting and receiving nodes. The antennas are assumed to be isotropic i.e. radiates equally in all directions. The path loss in such a scenario is known as the free space path loss. (The path loss is the ratio of the power radiated by the transmitting antenna to the power at the receiving antenna). The received power, $P_r(d)$, at a distance d from a transmitter is calculated by the Friis transmission equation (equation 3.1) [Janas01].

$$P_r(d)=\frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{3.1}$$

In equation 3.1:

P_t is the transmitted signal power

G_t and G_r are the antenna gain of the transmitter and receive respectively.

λ is the wavelength (in the same units as d)

L is the system loss

As can be seen, the free-space model attenuates the signal as $1/d^2$ with distance d.

3.3.2.2 Two-ray ground reflection model

The two-ray ground reflection model takes into account the single line of sight path between the transmitter and receiver, and another factor known as the *ground reflection*. Consider the scenario in Figure 3.2:

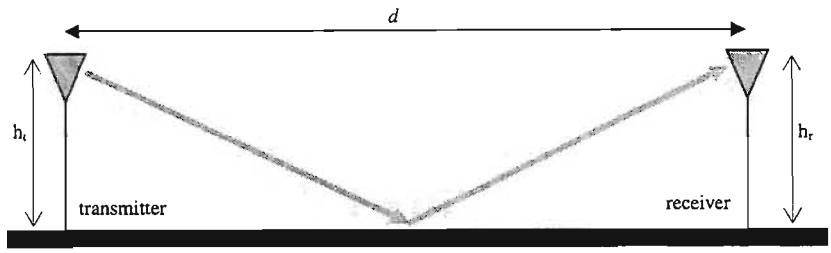


Figure 3.2: The two-ray ground reflection model

If the distance d between the transmitter and receiver antennas is much greater than the product of the heights of the antennas (h_t and h_r), the low angle of incidence of the radio waves will cause the earth to act as a reflector. There would be destructive interference between the line-of-sight signal and an out of phase reflected signal. The two-ray ground reflection model takes into account both the line-of-sight and reflected path to give a more accurate prediction of the received power at long distance. There is a higher path loss exponent for the relationship of the received power to the distance as can be seen in equation 3.2.

$$P_r(d)=\frac{P_t G_t G_r h_t^{\,2} h_r^{\,2}}{d^4 L} \tag{3.2}$$

However this model produces poor results at low distances. Thus the model in NS-2 uses a crossover distance, d_c, to switch between the free-space and two-ray ground reflection model. If the distance is less than d_c, then the free-space model is used, else the two-ray ground reflection model is used. The value of d_c, also known as the Fraunhofer distance [Janas01], is calculated using equation 3.3:

$$d_c = \frac{(4\pi h_t h_r)}{\lambda} \tag{3.3}$$

The communication range of each node in the simulated network is set using the characteristics of the network interface being modeled, as well as the radio propagation model. The characteristics of the interface are derived from the technical specifications of the wireless network interface being simulated. The following parameters pertaining to the network interface are set using the simulation startup scripts:

- Pt_ : the transmission power (Watts)
- Freq_ : the frequency of the radio transmissions
- CPThresh_ : the capture threshold (in dB), which determines how much stronger a signal must be than the one being currently received for capture to occur.
- CSThresh_ : the carrier sense threshold. This determines the minimum received power (in Watts) needed for the network interface to detect a transmission.
- L_ : the system loss
- RXThresold_ : the receive threshold. This determines the minimum received power (in Watts) required to receive a packet.
- Bandwidth_ : the bandwidth of the network interface.

3.3.3 The Medium Access Control (MAC)

The link layer of NS-2 implements the complete IEEE 802.11 standard [IEEE99] Medium Access Control (MAC) protocol. This enables the modeling of collisions and contentions of transmissions in the wireless medium. The protocol employs the Distributed Coordination Function (DCF) where both physical and virtual carrier sensing is used. The physical carrier sensing uses Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) to avoid transmission collisions. In addition, the "hidden terminal" problem in the wireless environment is overcome by means of the virtual carrier sensing mechanism. Further details of these terms can be followed in Chapter 5.

The MAC protocol executes a Request-To-Send/Clear-To-Send (RTS/CTS) exchange before a unicast packet is transmitted. This is used to test the activity status of the channel. If it is inactive, the channel is reserved for the transmission of the data packet. The receiving node, after the reception of each unicast packet, responds by sending an Acknowledgement (ACK)

to the sender. If the node transmitting the data packet does not receive the ACK immediately, it retransmits the packet a limited number of times until the ACK is received. The transmission of a broadcast packet is not preceded by the RTS/CTS exchange and is sent when the physical carrier sensing indicates that the channel is clear. There is also no Acknowledgment (ACK) for a broadcast.

3.4 Protocols Simulated

Currently there are four publicly available protocol implementations in NS-2:

- DSDV
- AODV
- DSR
- TORA

These are four prominent protocols that cover a range of design choices for ad-hoc routing protocols. DSDV is a classic table driven scheme and is a representative of the pro-active routing philosophy. The other three protocols represent the on-demand methodology. However they are very distinct in their operations. DSR is a pure source routing algorithm where the entire route is included in the header of a packet. AODV distinguishes itself from DSR in that routing information is stored in the routing tables of nodes in the network. TORA is different from AODV and DSR in its use of the Directed Acyclic Graph (DAG) structure to create routes between source and destination nodes.

The NS-2 implementations of these protocols were developed by the CMU Monarch group [Broch98] for the purposes of performing ad-hoc routing protocol performance comparisons. The details of the implementations are omitted here for brevity but can be found in [Broch98] and [NS-203]. The conclusions made by the original comparison work could not be considered for this comparison study for the following reasons:

- Their physical layer parameters reflected the characteristics of a Lucent WaveLan network interface card. The characteristics of this card are very different to the Cisco Aironet LAN card which is used in the system being modeled in these simulations. A major difference between the cards is that the Lucent card allowed a maximum bandwidth of 2 Mbps, which is increased to 11 Mbps in the Cisco card.
- The implementations used in the original work do not reflect the modifications and optimizations of some of the protocols. The protocol implementations available with the current version of NS-2 have newer versions of the protocols

- The mobility pattern used in these simulations is more accurate in modeling mobility in an ad-hoc network. (Refer to Section 3.5 which describes the mobility model)
- Broch et al did not study the packet transmission delay characteristics of the simulated protocols. With the applications that are envisaged for ad-hoc networks it is essential that such a metric is used in the comparison of the protocols.
- The study in [Broch98] (as with most comparison) was conducted with a relatively large (50 node) network. The study herein is aimed at the PCS network which consists of fewer nodes. Thus it was important to do the comparison study with a smaller network to reflect the network being modeled.

3.5 Simulation Setup

The simulation study would also aid in becoming familiar with running simulations using the NS-2 package. This experience was useful in the design, prototyping and comparison work for the proposed protocol in this dissertation. The details of this can be found in Chapter 4.
The simulation setup was configured to model the proposed Positional Communication System (PCS) network [PCS03]. This is a network of 16 handheld computers that allows voice and data communications between the hosts in the network. The wireless links between the hosts are made possible by Cisco Aironet wireless LAN cards operating in the Industrial Scientific Medical (ISM) frequency band. Thus the simulation setup herein consists of 16 nodes moving about in the network topography. The physical characteristics of the wireless interface were set-up to reflect the characteristics of the Cisco LAN card. This was done by assigning the appropriate values for the parameters of the network interface of the nodes in the simulation (Refer to Section 3.3.2.2).

3.5.1 Mobility Model and Network Traffic Generation

The purpose of the simulations was to test the ability of the protocols to maintain a high level of data throughput between communicating nodes in a dynamically changing network topography. The simulated network consists of 16 nodes moving in a network space of dimensions 700m X 380m. A small network space was used with the aim of avoiding network partitioning. A rectangular region was chosen as opposed to a square region since the latter case results in a smaller average number of hops between the source and destination nodes in the same area. The communication range for each node is set to 250m, which is an appropriate approximate value for the Cisco wireless LAN interface running the 802.11b standard. A typical mobility pattern file for a slow moving network using such a

setup showed that there were over 300 link changes and over 450 route changes during a simulation run.

The mobility model used for the simulations is a modification of the random way-point model [Broch98]. A mobility scenario in the random way-point model is characterised by a parameter called *pause time* which determines how long a node remains stationary at a point. In a simulation each node in the network begins by remaining stationary at a randomly selected location in the simulation space for *pause time* seconds. The node then moves to a randomly selected destination with a chosen speed. Once the destination location is reached the node remains stationary for the *pause time* period before repeating the process. The speed is uniformly distributed between zero and the maximum speed of the mobile nodes. The application of the PCS is aimed at soldiers in a battlefield environment. The maximum speed that can be practically reached by the soldiers in such an environment is approximately 8m/s.

It is noted in [Davie00] that there is a complex relationship between the node speed and the pause time in the Random Way-point model. A scenario with fast moving nodes and long pause times produces a more stable network than a case where there are slower nodes with shorter pause times. It is shown in [Davie00] that in scenarios with pause times above 20 seconds produce a stable network despite nodes moving at high speeds. Therefore, in the simulations in this dissertation the network mobility scenarios are classed into two sets:

- a highly mobile network with pause time of 1s.
- a slow moving stable network with pause time of 250s.

In each set the pause time remained constant as the speed is varied in each mobility scenario. Each scenario in a set was characterised by the node speed. The speed of the nodes moving from the current location to the destination location is chosen uniformly between *average* ±10% m/s, where *average* is set to 1, 2, 4, 6 and 8 m/s respectively. In the simulations, 10 mobility scenario files were used for each speed, thus each point in the presented graphs is an average of 10 simulation runs.

The choice of the modified random way-point model was motivated by the results obtained from simulations using the original random way-point model in [Quazi03a] and [Quazi03b]. The results showed that there was no discernable relationship between the protocol's performance and the level of mobility introduced by the model. In the simulations therein, the pause time was varied from 0 to 1000 seconds but the performance of the protocols remained fairly constant across the range. This was an indication that, although the pause

time increased from 0 seconds, the level of mobility remained unchanged with such a mobility model.

The communication model used in the simulations is similar to that used in [Broch98]. All traffic sources in the network were Constant Bit Rate (CBR) sources. CBR models UDP traffic and is more useful for routing protocol comparisons than TCP traffic. This is because a TCP traffic source employs its own flow control mechanisms, which hinders the actual assessment of the routing protocol running at a lower layer than TCP. In the simulations presented in this chapter the network communication consists of 8 traffic sources and 8 connections. Each source sends a data payload of 125 bytes at a transmission rate of 10 packets per second. Thus there are eight 10 Kbps communications channels across the network, which equates to an offered network load of 80 Kbps. The purpose of the set-up was to compare the performance of the protocols against each other and not to test their ability to deal with high traffic load, and thus a moderate load was used.

In the simulations in this dissertation, Variable Bit Rate (VBR) traffic was not considered and only CBR traffic sources were used. The reason for this is as follows. VBR traffic is generally produced by data communication such as text and image transfers across the network. In the PCS however, such application specific traffic has low priority when compared to the real-time traffic produced by the audio application of the system. Since the audio application is the primary objective of the system, all the simulations were done to determine the system performance with such type of offered traffic, namely CBR traffic. Applications producing VBR type of traffic are less time critical and thus would only receive network resources if there is little or no voice traffic. This would be implemented using some form of prioritization mechanism. Despite the lower priority given to data traffic, the performance of the PCS network has to be studied under such traffic. This investigation will be a priority in the future work of this project.

3.6 Metrics used

The following are the quantitative metrics that are used to evaluate the performance of the simulated protocols. These metrics are selected from the specifications provided by the MANET working group [Manet03] of the IETF [Corso99].

3.6.1 End-to-End Network Throughput

This metric, also known as packet delivery ratio, is defined as the percentage of the packets that are generated by the application layer agents at the source nodes that are received by the application layer sinks at the destination nodes. It is considered as an external measure of the effectiveness of a protocol.

3.6.2 End-to-End Network Delay

This is measured as the average end-to-end delay of data packet transmission. The end-to-end delay implies the time elapsed between the instance that a packet was generated at the source and when it was successfully delivered to the application agent at the destination. The delay measure takes into account the queuing and the propagation delay of the packets in the network.

3.6.3 Efficiency

The end-to-end throughput is an external measure of a protocol's effectiveness while efficiency is considered as the internal measure of its effectiveness. To achieve a given level of data routing performance, two different protocols will use different amounts of overhead, depending on their internal efficiency. Control and data traffic share the same channel, and since in an ad-hoc network there is limited channel capacity, excessive control traffic will adversely impact on the data routing performance. In addition, due to the nature of transmissions of the 802.11 MAC protocol, excessive routing overhead will cause delays in the transmission of data packets. The routing overhead is presented simply as the amount of routing data transmitted during a simulation.

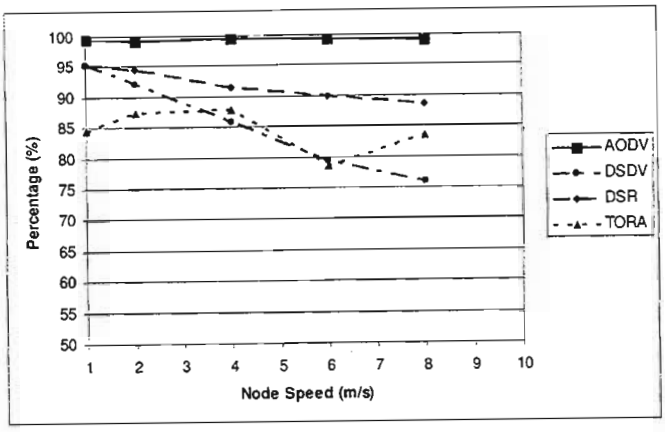
3.7 Results and Discussion

The results of the simulations are presented in the sub-sections that follow. The data throughput and delay are shown as packet delivery ratio and delay respectively. The

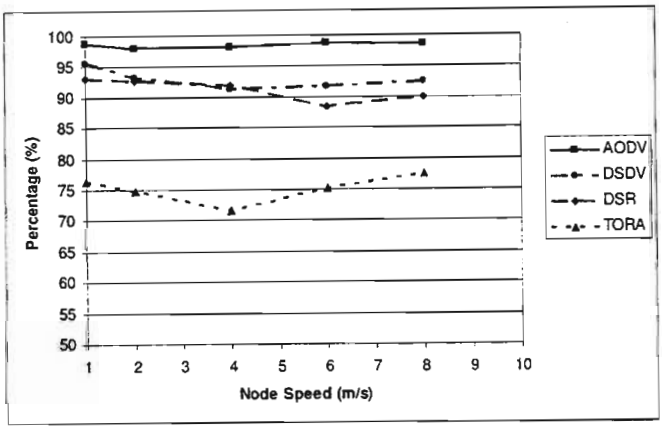
efficiency of each protocol is shown as the number of routing protocol packets used to determine and maintain routes during a simulation run. In each figure shown (a) represents the results for the highly mobile network configuration while (b) represents the stable network configuration.

3.7.1 Packet Delivery Ratio

The packet delivery ratio of the four simulated protocols can be seen in Figure 3.3. It can be observed from the packet delivery ratio graphs that DSDV finds it difficult to converge when the level of mobility in the network is high. In Figure 3.3a, DSDV's delivery ratio decreases linearly with increasing speed. In a highly mobile network the pro-active protocol is unable to maintain up to date routing information, and thus packets are dropped when they are sent along routes that have become outdated. The delivery ratio results for DSDV in Figure 3.3b shows that in a less mobile network the protocol fares better. The pure on-demand algorithms (AODV and DSR) maintain a high level of packet delivery ratio despite the level of mobility in the network. In both the scenarios AODV achieves nearly 100% delivery ratio for the varied speeds, showing the protocol's robustness in a moderately loaded network. The delivery ratio performance of DSR, however, decreases slightly when the network becomes stressed due to high mobility. This can be attributed to ineffective use of caching information in such conditions where outdated routes in cache are used to transmit data packets. The poor performance of the TORA on-demand algorithm is explained by the reasons that follow. As mentioned in Chapter 2 the protocol has the tendency to create short lived routing loops. This results in some packets being dropped as they are caught in such loops. However, the more significant reason is the routing overhead created by the protocol (as shown in Section 3.7.2). The high number of routing packets transmitted by the protocol creates high level of congestion in the network and this has significant effect on the protocols ability to successfully transfer data packets.



(a) Packet delivery ratio for highly mobile network (pause time of 1 second)



(b) Packet Delivery ratio for stable network (pause time of 250 seconds)

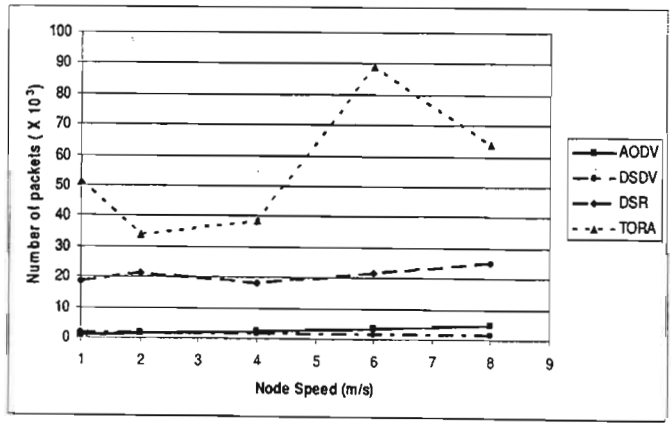
Figure 3.3: Packet delivery ratio performance of simulated protocols

3.7.2 Routing Overhead

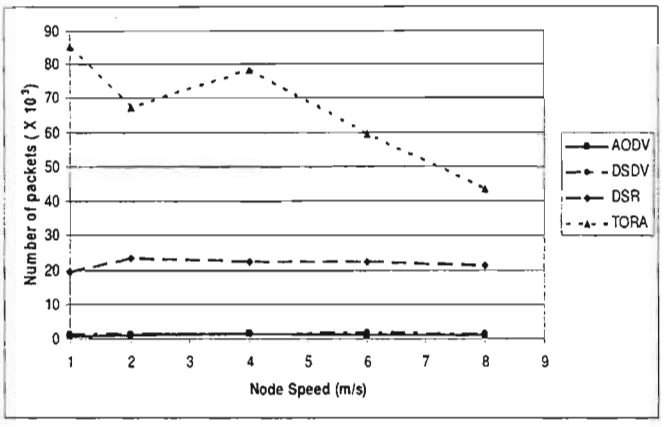
The routing overhead in Figure 3.4 is presented as the total number of routing protocol packets transmitted during a simulation. In terms of the routing overhead AODV and DSDV show the best performance amongst the four protocols simulated. The routing overhead characteristic of DSDV shows its pro-active nature. The protocol transmits a constant number of routing packets regardless of the node movement rate in the network. This is due to the periodic updates of the routing information. The on-demand protocols are more adaptive to the level of mobility in the network. With highly mobile nodes in the network there is increased number of link changes. In such cases the on-demand protocols exert considerable effort in route maintenance. This can be clearly seen in the routing overhead comparison between AODV and DSDV in Figure 3.5. In the graph AODV's routing overhead increases linearly with speed whereas DSDV's routing overhead remains fairly

constant. DSDV's slight increase with higher speeds is due to the triggered updates caused by the link changes in the network. DSDV's inefficient use the routing overhead is not its only disadvantage. Its routing overhead results in a there being a limit on the scalability of the protocol. This limit is due to the protocol's routing overhead increasing in the order of $O(n^2)$, with the increasing number of nodes n [Royer99a]. Thus in larger networks its routing overhead disadvantage would be significantly higher. It should be noted that the gradient of AODV's routing overhead graph in Figure 3.5b drops with higher network speeds. This result confirms the notion described in Section 3.5.1, where networks with high pause time values and high speeds in the random way-point model produce fairly stable network conditions. This is further confirmed in other results presented in this dissertation.

TORA's poor routing overhead performance is attributed to its use of an underlying protocol called the Internet MANET Encapsulation Protocol (IMEP) for its operation [Broch98]. IMEP is used to provide reliable, in-order delivery of all routing control messages from a node to each of its neighbours as well as the notification of the status of a node's link with a neighbour. This mechanism leads to the protocol's constant mobility-independent routing overhead i.e. the protocol would incur this overhead regardless of the level of mobility in the network. The mobility-dependent routing overhead is used to create and maintain the protocols routing structure (Directed Acyclic Graphs) as described in Chapter 2. The sum of these two factors causes the significantly higher routing overhead when compared to the other protocols. This adversely affects the protocols ability to route data packets in the network as seen in the packet delivery ratio results above.

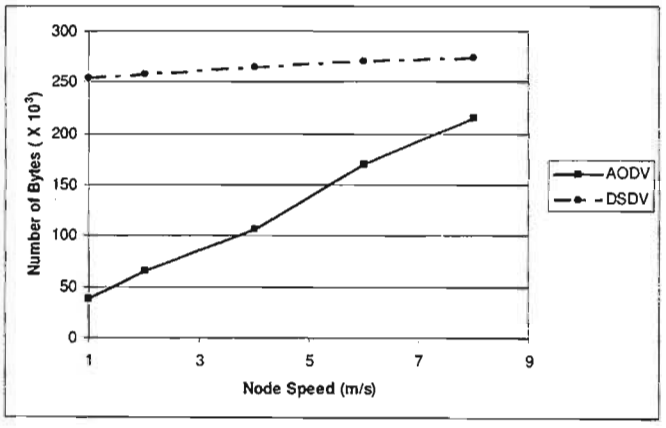


(a) Routing overhead for a highly mobile network (pause time of 1 second)

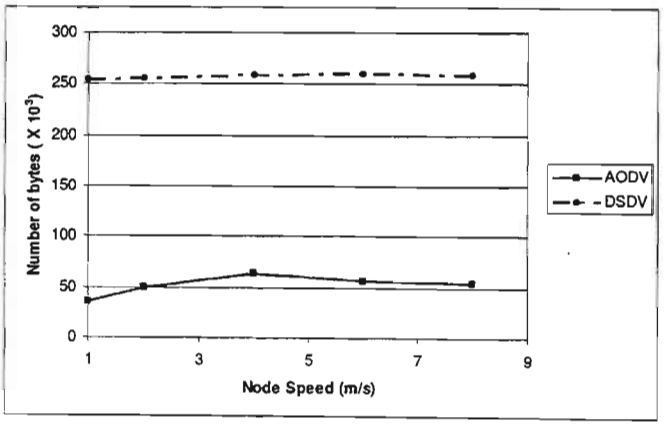


(b) Protocol routing overhead for a stable network (pause time of 250 seconds)

Figure 3.4: Routing overhead of simulated protocols in terms of the total number of routing packets



(a) Overhead comparison in a highly mobile network (pause time of 1 second)

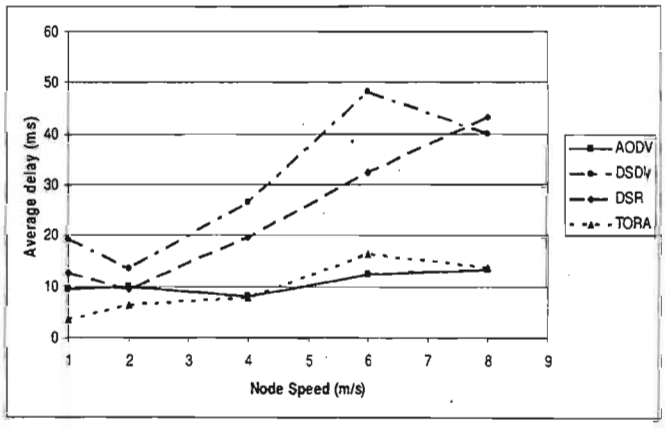


(b) Overhead comparison in a stable network (pause time of 250 seconds)

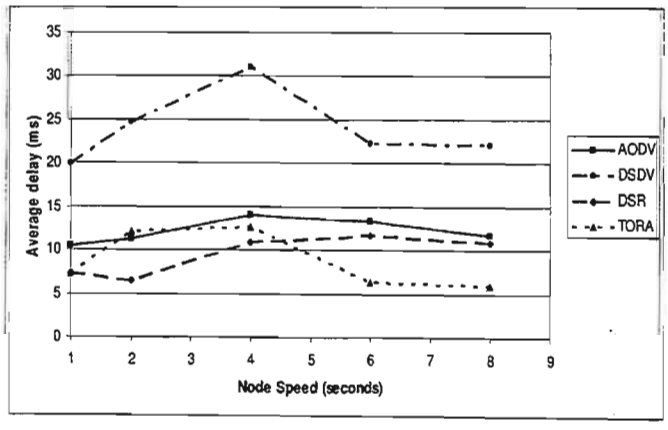
Figure 3.5: Routing overhead in total number of bytes showing the performance of AODV vs. DSDV

3.7.3 Average end-to-end packet delivery delay

The delay characteristics of the simulated protocols can be seen in Figure 3.6. Due to the pro-active philosophy of DSDV it is expected that the protocol's advantage over the other protocols would be in terms of the packet delivery delay. However both the scenarios simulated show that the on-demand algorithms perform better than DSDV. This can be attributed mainly to the protocols tendency to route packets using stale routes since new routes are not updated quickly enough, especially under highly mobile network conditions. TORA's good delay performance is due to the protocol maintaining multiple routes to a destination. This results in the protocol executing a fewer number of route reconstructions when compared to the other on-demand protocols. Both these graphs demonstrate the good performance of AODV in terms of delay.



(a) Delay result for highly mobile network (pause time of 1 second)



(b) Delay result for stable network (pause time of 250 seconds)

Figure 3.6: Average end-to-end delay performance of the simulated protocols

3.8 Other protocol comparison studies

The comparison study in [Jiang01] compared the STAR routing protocol with DSR and AODV. The GloMoSim simulation environment was used for this comparison work. The study in [Lee99], also using GloMoSim, compared the Distributed Bellman-Ford (DBR) algorithm with DSR and ABR. In [Das98], the Extended Bellman Ford, Shortest Path First (SPF) link state protocol, DSDV, AODV, DSR and TORA were compared. Here the MaRS (Maryland Routing Simulator) was used as the simulation package. This environment, however, was rather limited with no link or physical layer models.

The NS-2 simulator was the simulation package chosen for this research work. Since it is difficult to correlate performance results across different simulation packages, the focus of the comparison study survey in the dissertation was on those that used NS-2. The simulation work in this chapter was based on the work done in [Broch98]. The conclusions from other protocol comparison studies using NS-2 in [Das02] and [Johan99] are further referenced in Chapter 4.

3.9 Conclusion

It is evident from this work that the simulations performed provide useful metrics in order to determine which of the available protocols proposed in literature are most suited for an ad-hoc routing protocol targeted at the PCS application under development. Ideally what is required is for a protocol to achieve the maximum level of data throughput with the least amount of delay and use of network resources. The DSDV proactive routing protocol was unable to cope with high mobility situations where the periodic updates are inefficient in disseminating route updates quickly, thus stale routes are used. Under low mobility conditions DSDV achieved delivery ratios of up to 95% but as the mobility increased the delivery ratio figure decreased linearly, almost reaching 75% at the maximum simulated node speed. The protocol's use of network resources in terms of protocol overhead remained constant regardless of the level of mobility in the network. This result showed the algorithms inefficient use of available resources which is one of its disadvantages and limitations in terms of scalability. Amongst the on-demand algorithms TORA performed the poorest. One of the major drawbacks of the protocol is its routing overhead which is significantly higher than the other protocols. This is due to the protocol's reliance on the IMEP which TORA uses as its underlying protocol. This overhead creates congestion which undermines the

protocol's ability to successfully route packets. TORA achieves, at best, 85% throughput in the simulations conducted. However it is able to route its data packets with the least delay when compared to the other protocols (its end-to-end delay performance on average was under 10ms). This is because TORA maintains multiple routes for a particular destination and is able to react quicker when routes break. The AODV and DSR on-demand algorithms performed better than the other two protocols in terms of the packet delivery success. This was also shown in [Broch98]. AODV achieved close to 100% delivery ratio regardless of the level of mobility in the network. DSR's delivery success was above 90% for both the set mobility scenarios. However, its aggressive use of cached routes decreased its performance when the mobility in the network increased. AODV incurred less protocol overhead and also delivered the data packets with less delay, especially when the level of mobility in the network was high. These characteristics thus make the on-demand algorithm the best suited for a general ad-hoc network application.

Chapter 4

A Load Aware, Location Aided, Ad-hoc On-demand Routing Algorithm

4.1 Introduction

Recent comparison work via simulation in literature and the results presented in the previous chapter have shown that the on-demand ad-hoc routing protocols such as AODV and DSR generally show better performance than most proposed table driven algorithms. In particular, the AODV algorithm shows significant performance advantages in terms of the metrics used for protocol comparisons. Their good performance in terms of high delivery ratio, low routing overhead and delivery delay make them attractive for ad-hoc network applications. These characteristics result in on-demand algorithms being more scalable than table driven schemes.

The on-demand algorithms typically have a two phase operation: route discovery and route maintenance. Most of the proposed schemes use flooding in the route discovery phase of the protocol's operation. However, there is significant redundancy created using such a technique since a route request is sent to nodes far from the vicinity of the source and destination nodes. The broadcasting nature of the flooding method also causes collisions and contentions in the transmission of packets in the wireless channel. Therefore, if this flooding technique is made more efficient, these apparent disadvantages of on-demand algorithms would be reduced. This would result in further reductions in the routing overhead of on-demand routing protocols, and thus increase their performance advantage over table driven schemes.

A further characteristic that is common to most on-demand routing protocols is that the shortest path, in terms of the hop count, is used as the routing metric. These protocols select the shortest path to route traffic between the source and destination node with little regard for the ability of the routers in between them to carry such traffic. There exists no efficient mechanism to detect adverse conditions such as congestion on a route and hence use a route that is more suited. The authors of the AODV on-demand algorithm have noted [Das02] that

to make such protocols more efficient and scalable in MANETs, the load along a route had to be taken into account.

This chapter of the dissertation presents the effect of adding query localization to the route discovery process of an on-demand algorithm. The aim is to make the flooding technique more efficient. The protocol presented in this chapter also introduces a load metric, along with the hop count, as a decision criterion for route selection. The protocol performs load checking with the aim of balancing the traffic load in the network.

The next section of the chapter provides a brief description of AODV's on-demand scheme. Section 4.3 discusses the proposed improvements that led to the design of the new on-demand algorithm. The detailed description of the proposed protocol is given in Section 4.4. A brief discussion of the steps taken to implement the new protocol in the NS-2 package follows in Section 4.5. Sections 4.6, 4.7 and 4.8 present the simulation setup, metrics used and the results of the performance comparison of AODV and the proposed algorithm respectively.

4.2 Overview of AODV's on-demand methodology

An on-demand algorithm such as AODV [Perkin99], [Perkin02] only creates a route to the destination once a packet requesting a route is generated. This protocol generally has a lower routing overhead when compared to table driven schemes since these protocols only maintain routes that are actively used. When a request for an unknown destination is generated, the source node initiates a route discovery by sending a route request packet. This packet is broadcast to the host's immediate neighbouring nodes. Each intermediate node that receives the packet creates a reverse route entry in its routing table for the source node, with the next hop field in the entry pointing to the last hop taken by the route request packet. This reverse route is later used to forward route reply packets that are destined for the source node. The intermediate node then broadcasts the route request packet to its neighbours. Once the route request arrives at the destination, a route reply packet is unicast by the destination node using its reverse route entry for the source node. In AODV a route request is identified by means of the source address and a unique broadcast ID. Since a node processes a route request for a destination only once, the destination node ignores any further requests that it receives from that particular source node with the same broadcast ID. All intermediate nodes are then responsible for passing the route reply packet back to the source node. In AODV, an intermediate node replies to the route request on behalf of the destination node if it has an

active route to the host. Thus, during the route discovery phase, the protocol uses the flooding technique to ensure that the route request arrives at all nodes until a route to the destination is found.

4.3 Proposed improvements

Flooding is a robust method of getting the route request packet to every possible node in the connected component network. However, it is unnecessary for the route request to reach every possible node, especially those intermediate nodes not in the path of the source and destination. In a large and highly mobile network considerable routing overhead is incurred by the flooding method. This reduces the advantage, in terms of protocol routing overhead, which on-demand algorithms have over table driven ones. If the flooding could be made more efficient it would lower the routing overhead incurred. There would be further benefits such as reducing network congestion with fewer route request packets being transmitted in the network. Route request packets are generally broadcast packets and can have an adverse effect on data transmission over the wireless channel due to the broadcast storm problem [Tseng02]. The effect of this problem would be reduced with a more efficient flooding method. One way to make the flooding of the route request packets more efficient is to intelligently reduce the region in the network where the packet is flooded. Query localization in ad-hoc networks has been examined in different forms in [Casta02], [Skold01] and [Tripa02]. As mentioned in the introductory chapter of this dissertation, each host in the Positional Communication System (PCS) network is enabled with a Global Positioning System (GPS) module that provides the location information of each host. If a router (which could be any node in the ad-hoc network) has prior knowledge of the destination's location information, it could use this information to aid the query localization process. The method proposed in this work is based on ideas presented in the Location Aided Routing (LAR) algorithm [Ko98]. LAR uses location information to separate the network space into different regions for each communication session and the transmission of packets differ depending on the region in the network. What is proposed herein is that the location information be used to determine the proximity of an intermediate router to the destination node. Once this is determined, if an intermediate node is closer to the destination node than the node that passed it the route request packet, it forwards the packet to its neighbours. The packet is dropped if the intermediate node is found to be further away. The aim of the query localization is to bring the route request packet physically closer to the destination node with each hop and hence prevent it from traversing to unnecessary parts of the network. The

details of the location aided query localization can be found in the next subsection of this chapter.

Many of the on-demand protocols that have attained popularity use the hop count as the decision criterion for selecting a better route. The aim of choosing the shortest path, in terms of the number of hops, is to reduce the packet delivery delay. However, in high traffic load conditions, using the least hop count has a drawback in that there exists the possibility of creating congestion at certain nodes in the network. Due to their location in the network these nodes become critical nodes as they are in the shortest path for many routes. This adversely affects the delay performance, as packets remain queued at these highly loaded nodes for a considerable amount of time. The packet delivery ratio performance also deteriorates as packets are dropped from overburdened protocol queues of the critical nodes. There is also the possibility of power depletion at these nodes due to the burden of high load. These nodes would be continuously involved in various routes through them, and this could significantly exhaust the limited power available to them. In a recent simulation study [Das02] conducted by the authors of the AODV routing protocol, it was concluded that such on-demand protocols could benefit from using congestion related metrics such a route load to evaluate routes instead of always selecting the hop-wise shortest routes. The work in [Lee01] and [Wu01] emphasizes that doing load checking for routing traffic in a MANET has significant advantages over shortest path methods. Load checking not only prevents congestion at certain nodes in the network but also has the benefit of balancing the network traffic amongst the nodes in the network. When the load is balanced, there will be a reduced number of critical nodes in the network.

What is proposed in this research is the integration of the load metric and the hop count metric. The protocol queue length of an intermediate node is taken as a measure of load. The load metric, along with the hop count, is taken into account when deciding on the ideal route between the source and destination. The load checking is similar to the method used in the Dynamic Load Aware Routing (DLAR) protocol [Lee01] where the load metric takes into account the load at each intermediate node between the source and the destination node. The metric variable is updated at each intermediate node depending on the load condition at that node. However, unlike DLAR where the load is the primary metric, here both the load and hop count are used to select one route against another. In addition the scheme proposed herein prevents intermediate nodes that have their protocol queues close to 100% full from accepting route requests. Thus nodes that are already congested are not given the added

burden of accepting further routes. The dropping of route request packets reduces the routing overhead by preventing redundant route request packets clogging up routes that are already highly congested. This can be considered a form of call admission, which improves the overall performance of the on-demand algorithm.

4.4 Description of the proposed protocol

The protocol proposed in this dissertation, called Tactical AODV (TAODV), is a modification of the Ad-hoc On-demand Distance Vector (AODV) [Perkin02] routing protocol. Although the proposed improvements mentioned in the last section could be applied to most on-demand algorithms, AODV was chosen. This is because it was shown that AODV has the best performance under PCS appropriate network conditions when compared to other protocols in simulation comparisons done herein (Chapter 3 and in [Broch98], [Das02] and [Johan99]). TAODV is similar to AODV in that the routing information for each route to a destination is maintained in a distributed fashion in the routing tables of the nodes in the network. The protocol only creates routes to destination nodes when such routes are requested by the generation of data packets for the destination. Routes are only maintained as long as they are being actively used. There is a timeout period for each route, and if a route is not used in that period it is considered to be inactive and is purged. If a source node does not have a route to the destination, it initiates a route discovery. The data packets for the destination host are transmitted once a route is found. If the route is broken during the communication session between the source and destination node, it is repaired before further transmission can continue.

The first step of the route discovery process of TAODV is the generation and broadcast of a route request packet. The contents of the route request (RREQ) structure can be seen in Table 4.1.

TAODV uses AODV as the underlying protocol and the finer details of the two protocols are similar. Details such as the structure of a routing table entry, location cache entry and protocol's route error handling are omitted here. Described in the next two sections is the implementation of the route localization algorithm using location information and the load checking scheme characteristic of TAODV.

Data Type	Field	Comment/description
u_int16_t	rq_hop_count	Number of hops from the source to the node processing the request packet
u_int32_t	rq_bcast_id	Id of the current route request packet
u_int32_t	rq_dst	IP address of the destination for which the route request was generated
u_int32_t	rq_src	IP address of the source node
u_int8_t	q_thresh_count	Counter that is incremented depending on the load condition of the intermediate nodes that process the route request
Double	src_loc_x	x-coordinates of the source node
Double	src_loc_y	y-coordinates of the source node
Double	rq_distance	Approximate distance to the destination node
Double	li_timestamp	The timestamp of the location information that was used to calculate the distance to the destination
Double	rq_timestamp	The timestamp of the route request packet

Table 4.1: Fields of the *RREQ* structure used in the route request packet of TAODV

4.4.1 Route localization

The route localization used in TAODV is an optimization of the flooding technique used by on-demand algorithms. If available, the location information of the destination node is used to determine if an intermediate node (acting as a possible router) should rebroadcast a route request packet. It will only rebroadcast the packet if it is deemed to be closer to the destination than the node from which it received the route request packet. This method aims to prevent route request packets from traversing to unnecessary sections of the network i.e. going to nodes that are not in vicinity of the path between the source and destination pair. Preventing route request packets from reaching such areas will result in a reduced protocol routing overhead.

The dissemination of a node's location data occurs in an on-demand manner. There is no periodic transmission of the location data, it is thus not necessary to change the basic routing mechanism of the protocol to accommodate the route localization algorithm. Other nodes in

the network will only know of a nodes whereabouts if they have communicated with it, or acted as a router for any of its routes. The location information of a source and destination node is piggy-backed with each route request and route reply packet respectively. Refer to the appropriate fields of the route request (RREQ) and route reply (RREP) structures in Table 4.1 and Table 4.2.

The route localization is implemented as follows. When the route request is generated for the destination node, the source node inspects its location cache to see if it has a location entry for the destination. This is likely if it has either communicated with the destination previously or acted as router for it. If the location entry is found, the positional information of the destination (its x and y coordinates) is used to calculate the distance to it using equation (4.1). There is no account of height in this distance measure as currently NS-2 only supports a flat two dimensional grid. This could however be an optimization in the implementation of the proposed routing protocol in the real world test-bed.

$$D_{sd} = \sqrt{(\Delta x)^2 + (\Delta y)^2} + \lambda \tag{4.1}$$

D_{sd} represents the distance from the source to the destination node.

Δx and Δy is the difference between the x and y coordinates of the source and destination nodes respectively.

λ is a factor that takes into account the approximation of the distance measure and is given by equation (4.2).

$$\lambda = v \times (t_c - t_d) \tag{4.2}$$

t_c is the current time.

t_d is the timestamp of the location information. This holds the value of the time instance the destination node published its positional information.

v is the specified maximum speed that a node can move.

The application for this protocol is the Positional Communication System which is aimed at foot soldiers in a battlefield situation. The maximum speed used in the design of the protocol

is the maximum practical speed that a soldier can move in such an environment (the maximum speed used in the simulations was 8m/s). The measure given by equation (4.2) is the worst case scenario in terms of the distance to the destination node. It is a measure that is similar to the *expected zone* concept proposed in the Location Aided Routing (LAR) protocol [Ko98].

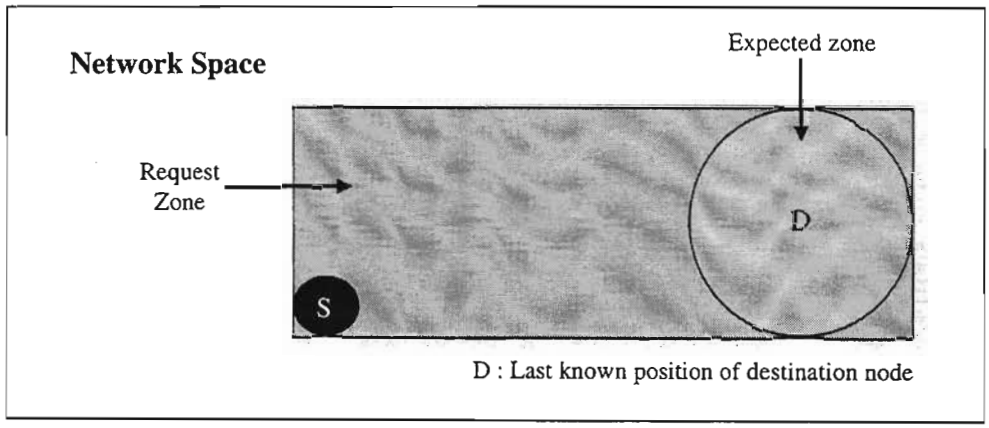


Figure 4.1: The network configuration in LAR

As can be seen in Figure 4.1 the expected zone is defined as the region within the request zone that is expected to hold the current location of the destination node.

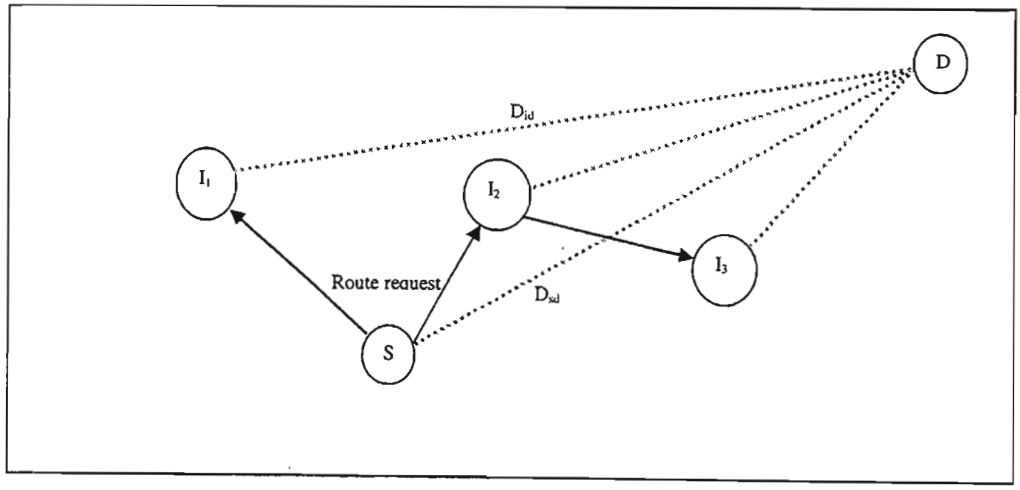


Figure 4.2: Rebroadcast decision in TAODV

The distance calculated by the source node (D_{sd} in Figure 4.2) is included in the route request packet broadcast to its neighbours. The timestamp of the location information (t_{sd}) that was

used to calculate D_{sd} is also one of the fields in the packet. When a node between the source and destination receives the packet, its first action is to query its location cache for an entry for the destination node. If an entry is found, the timestamp of the entry (t_{id}) is compared with the location information timestamp in the route request packet (t_{sd}). If t_{id} is newer than t_{sd}, it implies that the intermediate node's location information for the destination is more current. The intermediate node then calculates the distance from the source to the destination (D_{sd}) using its information for the destination and the location information of the source from the route request packet. The distance field in the route request packet is updated accordingly. If t_{id} is not newer, the distance field remains unchanged. The intermediate node then calculates its distance to the destination node and compares this value with the source to destination distance. If D_{sd} is found to be larger than D_{id} (the intermediate node is closer to the destination than the node from which the route request arrived), the intermediate node broadcasts the route request packet. The distance metric to the destination is equated to D_{id}, narrowing the localization region with each hop. If this condition is not met the packet is dropped. During the localized route request process, if any node has no entry for the destination in its location cache, it does not execute the localization algorithm. The node instead broadcasts the route request as it is done in a blind flood. If the location aided route request fails to secure a route to the destination after a certain number of attempts, subsequent route requests are done using the blind flood method. If a route cannot be found using the network wide flood within a route request timeout period, it is assumed that destination is unreachable and the route discovery is terminated.

4.4.2 Load Checking

The load checking operation of the protocol is done at each of the intermediate nodes that processes a route request. The length of the protocol queue at each node is taken as a measure of the load at a node. The protocol queue is a first-in-first-out (FIFO) queue in which packets that are awaiting routes are temporarily stored. Packets could be awaiting routes either due to route discoveries being attempted for unknown destinations or for broken routes to destinations that are being repaired.

When an intermediate node receives a route request, the first decision that is made at the node is whether it will get involved in the route. The node inspects its protocol queue and if it is near to its capacity the intermediate node rejects the route request by dropping the packet. This will prevent already congested nodes from further overload. After the initial load check, if the intermediate node decides to act as a router for the source, it first creates a

reverse route in its route table entry for the source node. This entry will be used to unicast reply packets back to the source node. The intermediate node then executes the load checking algorithm. The method used in this work is similar to one of the route selection procedures in the Dynamic Load Aware Routing (DLAR) protocol [Lee01]. A load variable in the route request packet (*q_thresh_count* in Table 4.1) counts the number of nodes between the source and destination that have their protocol queues loaded with packets above a certain threshold τ. This variable is initialized to zero by the source node when it generates the packet. Prior to broadcasting the route request packet, an intermediate node inspects the length of its protocol queue. If the queue length is above the threshold value τ, it increments the *load* variable in the packet. The variable keeps its previous hop value if the queue length is below the threshold value.

The load checking process is executed with each hop that the route request packet takes on route to the destination node. TAODV specifies that the route request should travel to the destination node and that no intermediate node is permitted to reply to the route request. Minimum hop count algorithms such as AODV and DSR specify that, if intermediate nodes have a cached route to the destination, they should send a route reply on behalf of the destination node. However, there are certain disadvantages to this methodology. Intermediate nodes replying to route requests generate a flood of route reply packets, which causes significant routing overhead. Route reply packets are unicast transmissions and use the RTS/CTS/DATA/ACK exchange of the 802.11 MAC protocol. Thus a route reply flood can create high level of congestion in the wireless channel. The least hop count method also results in certain routes in the network overlapping which creates congestion at certain nodes, creating bottlenecks.

A further disadvantage of intermediate nodes replying on behalf of destination nodes was found from a real world implementation of the AODV protocol [Royer00]. It was noted that when an intermediate node sends a route reply to the source on behalf of the destination node, the route is unknown to the destination node. Since it does not receive the route request packet it does not learn of the route to the source node. It is possible that all route requests are replied to on behalf of the destination node, and consequently it will never learn of a route to the source node. This could result in poor performance if the source node wishes to establish a TCP connection with the destination. This discovery has led to the modification of the AODV protocol in the form of *gratuitous* route replies, which are sent to the destination node informing it of the route to the source [Perkin02]. Thus, TAODV

specifies that only the destination node responds to route requests by sending a route reply, eliminating the need to resolve such issues. This method, having the benefit of reducing routing overhead, is essential in the acquisition of the most up to date load information on route to the destination. The contents of the route reply (RREP) structure can be seen in Table 4.2.

The heuristic used to select routes is a combination of the load information and the hop count. There are essentially three conditions that determine whether a route is better than another.

Case 1: A route is considered better if the *load* variable of a newly arrived route is lower than that of the previous route.

Case 2: The *load* variable is equal between the routes being compared. In such a situation the route with the lower hop count is deemed a better route.

Case 3: If the newly arrived route has a higher hop count than the previous route, it is only recognized as a route with a better metric, if the *load* variable is lower. In this case, although a longer route in terms of hop count is chosen, it is a route that is less loaded. The protocol was simulated with and without this case and it was observed from the results that the inclusion of the case in the heuristic improves its performance. It would be expected that the end-to-end delay performance would suffer due to longer routes being preferred. However, it was noted in [Das02] that the correlation between the end-to-end delay and the number of hops is usually small. This is due to delays caused by various buffering and queuing, and the time spent gaining access to the radio medium in a congested node is more significant than a less congested single hop.

Data Type	Field	Comment/description
U_int16_t	rp_hop_count	Number of hops from the destination to the node processing the request reply
U_int32_t	rp_dst	IP address of the destination for which the route request was generated
U_int32_t	rp_src	IP address of the source node
U_int8_t	q_thresh_count	Counter that is incremented depending on the load condition of the intermediate nodes that process the route reply
Double	Dst_loc_x	x-coordinates of the destination node
Double	Dst_loc_y	y-coordinates of the destination node
Double	li_timestamp	Equated to the time the reply is sent
Double	rp_timestamp	Field used to compute the route discovery latency

Table 4.2: Fields of the *RREP* structure used in a route reply packets in TAODV

In AODV duplicate route request packets at all nodes are silently discarded. However since this is also done at the destination node, there is no mechanism to accept further, possibly better, routes. Furthermore, the authors of AODV mention that this technique automatically favours the least congested route over the shortest route [Das02]. In TAODV, the load checking algorithm ensures that the least congested route is always selected. In addition, the destination node accepts more than one route request from the source. When the destination first receives a route request, it replies by generating a route reply packet (filling in the appropriate fields in the RREP structure). Further replies are only generated and sent only if the route request packets come along routes that have better routing metrics. The route reply packet is unicast to the source node via the reverse route that was set up by the initial process of the route discovery. The arrival of the route reply packet prompts the source node to transmit the data packets it has in its protocol queue, using the newly discovered route. If a route reply with a better route arrives subsequent to this, the source node according changes the transmission route to the destination.

4.5 Implementation of TAODV in NS-2

The implementation of a routing protocol is generally done in the C++ environment of the NS-2 simulation package. The routing protocol implemented here is a network object instantiated from a routing agent class called TAODV. The class has a constructor that creates instances of each node in a simulation. Each node in the simulation has associated data structures such as a routing and location information tables which are used in the routing process. A node also has its own protocol queue object which is used to temporarily buffer packets awaiting routes. The location information of a node, in terms of its x and y coordinates in the simulation space, is determined from the appropriate member variables of the *MobileNode* object associated with each node. These coordinate values are then used to calculate distance between nodes in the network. The load at each node is easily determined by inspecting the *length* member variable of the node's protocol *queue* object.

In order to allow the TAODV routing agents to communicate with each other, a protocol specific packet type and header had to be added to NS-2. The packet type PT_TAODV was added to the simulator's packet handing functionality to enable referencing to TAODV routing protocol control packets in simulations. Finally the tracing support in NS-2 was modified to support the new protocol's packet *type*. As a result the output trace and nam files from each simulation could be used in simulation post-processing to determine the performance of TAODV.

4.6 Simulation setup

The purpose of the simulations in this chapter was to compare the performance of AODV and the proposed protocol, TAODV. The first simulated network consists of 16 nodes moving in a 700m X 380m network space. A second larger network consisting of 50 nodes is simulated to demonstrate the scalability of the proposed protocol. The 50 node network has a simulation space of 1400m X 450m. The communication range for each node is set to 250m.

The mobility model used for the simulations is identical to that used in the previous chapter. The modified random way-point model [Broch98] is used to create mobility in both the networks. The mobility pattern files are grouped into two sets, depending on the *pause time* value for a pattern file. The first set can be described as a highly mobile network with the *pause time* value of the mobility model being 1s. The second set consists of a slow moving,

stable network with the *pause time* value of 250s. In each set the pause time remained constant as the speed of the nodes was varied with each mobility scenario (refer to Chapter 3).

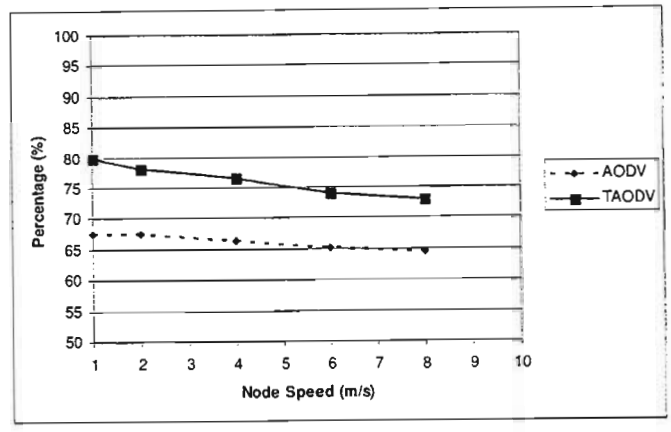
As in Chapter 3, Constant Bit Rate (CBR) traffic sources were chosen with random source and destination pairs across the network. The data payload for each transmission was 125 bytes. The purpose of these simulations was to test the performance of the protocol under considerable load. In the smaller 16 node network, there were 16 sources and 32 communications connections between the nodes. The offered network load with this setup, using a packet rate of 10 packets/second, is 320 Kbps. The network load is four times the figure used for the simulations in Chapter 3. In the 50 node network there were 20 sources and 30 connections. A lower packet rate of 5 packets/second was used and thus the offered load was 150 Kbps.

4.7 Simulation results

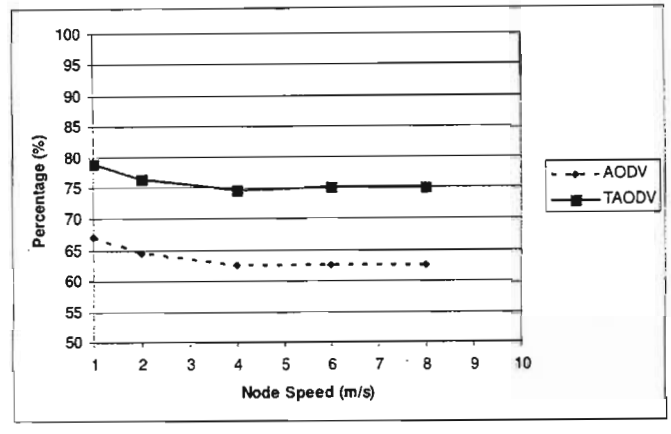
The comparison of the protocols was done using the same metrics used in the simulations of Chapter 3.

4.7.1 Packet Delivery Ratio

The packet delivery ratio results for the two different network configurations are shown in the graphs that follow. The packet delivery ratio for both the networks in the highly mobile scenario is presented in Figure 4.3a and Figure 4.4a. It can be observed from these two graphs that the protocol's performance decreases with increasing speed. The packet delivery ratio diminishes at high speeds as both the protocols drop packets when there is considerable topology change and links to next hops are consistently broken. The route repair process involves route re-discoveries, which if not achieved within a certain number of attempts causes further data packets to be dropped. The curves for the two protocols display a similar shape which is expected due to similarities in their operations. However, the proposed protocol's delivery ratio is consistently better than AODV in all scenarios.



(a) Packet delivery ratio for highly mobile network (pause time of 1 second)

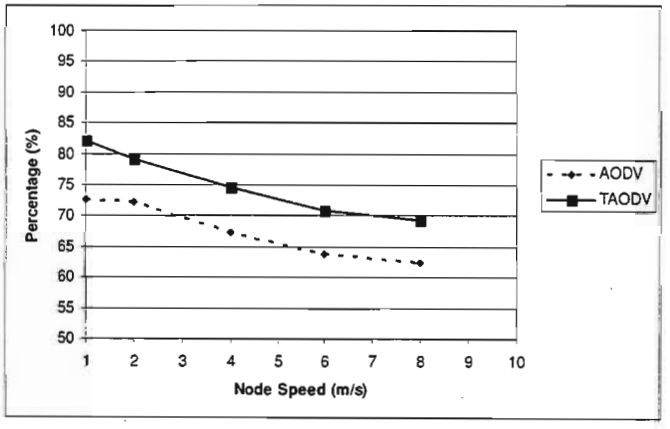


(b) Packet delivery ratio for a stable network (pause time of 250 seconds)

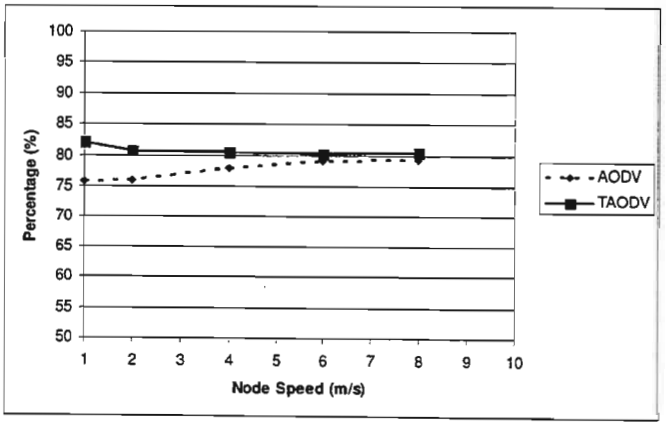
Figure 4.3: Packet delivery performance for the 16 node network

TAODV's packet delivery ratio results show particular advantage over AODV in the 16 node network configuration. There is (at least) a 10% gap in the packet delivery ratios achieved by the two protocols with varying mobility speeds. The routing of network traffic by the shortest-path AODV algorithm creates congestion at certain nodes in the network. The protocol queues of the routing agents are of a limited capacity, and when overloaded, packets are dropped. Since TAODV avoids the creation of such congestion scenarios, the dropping of packets from overloaded protocol queues is significantly reduced. This observation was made from the study of the output *trace* files of the simulations. The number of packets dropped due to a node's interface queues being overloaded was determined and it was noticed that AODV dropped far more packets due to this than TAODV.

The packet delivery ratio for the larger 50 node also shows the advantage presented by the proposed protocol. In the highly mobile network TAODV again shows a marked improvement in the packet delivery ratio. This concurs with the simulations results presented in [Quazi03b]. In this work, the mobility model used the original (unmodified) random way-point model and under low pause time value scenarios, TAODV achieved higher packet delivery ratios. Figure 4.4b shows that AODV reduces the advantage gap only when the level of mobility in the network decreases. This is specially the case in the scenario where the node speeds are high in the stable network. As mentioned in Chapter 3 this scenario presents a highly stable network. However, these results show that TAODV is not only more scalable than AODV, but is better suited to deal with high levels of traffic and mobility in a mobile ad-hoc network.



(a) Packet delivery ratio for highly mobile network (pause time of 1 second)



(b) Packet delivery ratio for a stable network (pause time of 250 seconds)

Figure 4.4: Packet delivery performance for the 50 node network

The quantitative packet throughput performance of the protocols in both the network configurations raises the question of network capacity in the ad-hoc network. In the simulation results presented in Chapter 3, the offered load in the 16 node network was 40 Kbps and AODV achieved nearly 100 % packet throughput in all mobility situations. In the results presented in this chapter, the packet delivery ratio is approximately 80% at best for both the network configurations. The results presented in [Das02] for AODV, using a setup of 50 nodes and 30 communication sessions, showed similar performance values. The decrease in the performance for the 16 node network when compare to that achieved in Chapter 3 is explained by the fact that the offered load in the simulations in this setup is increased four fold. The increased load creates significantly more congestion which hampers the ability of the protocols to route data as successfully.

Although relatively high, the offered load and resultant throughput achieved by the protocols is far below the network capacity. Thus each transmitting node is able to get only a fraction of the nominal channel bandwidth available. This is however justified in [Das02] with the following reasons:

- The bandwidth consumed by the delivered data packets is equal to the delivered throughput times the average number of hops traversed.
- Additional bandwidth is consumed by the data packets that are dropped, depending on the number of hops travelled before the drop occurs.
- A protocol's routing overhead consumes a significant portion of the bandwidth, in addition to the MAC control packets.
- The 802.11b MAC protocol's RTS/CTS/Data/ACK exchanges used for the reliable delivery of unicast packets significantly slow down the transmission of data packets. It was found in [Das02] that in stressful conditions (i.e. high mobility and/or load) the number of RTS is often twice as much the number of CTSs received. A high degree of errors, due to collisions or link loss, causes the frequent retransmissions of RTS.

4.7.2 Routing Overhead

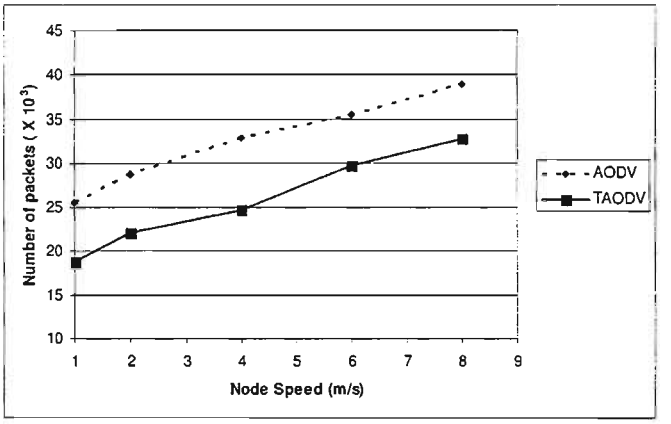
The routing overhead is presented in terms of:

- The total number of routing packets transmitted during a simulation
- The total number of bytes used for the routing process during a simulation.

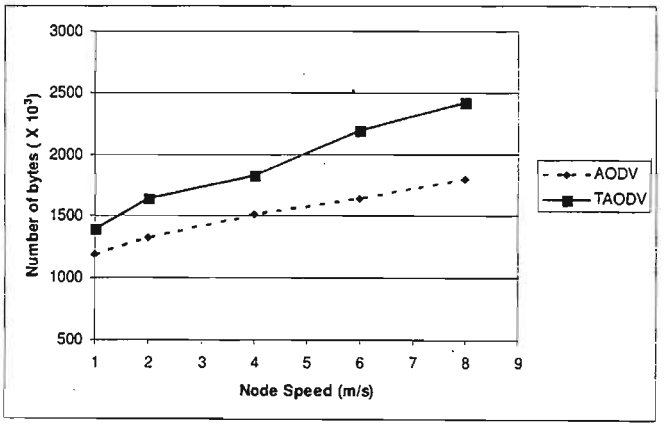
All the results presenting overhead in terms of routing packet overhead show that TAODV has a significantly lower overhead with the varying speeds in the simulations. The reduction in the overhead can be attributed to the route localization action of the algorithm; it uses a more efficient method of flooding the route request packets during the route discovery phase of the protocols operation. The route localization aims to prevent route request packets reaching unnecessary portions of the network. The effect of the algorithm can be seen since both on-demand algorithms use the same route discovery procedure, with the only difference being TAODV's route localization algorithm. Both the protocols show similar shape in their routing overhead curves (Figures 4.5 to 4.8). The routing overhead increases with the level of mobility in the network. This is due to the fact that routes are constantly changing and the protocols have to execute route re-discoveries in order to maintain them. It should also be noted that the gradient of the routing overhead curve decreases as the speed increases in the stable network scenario (particularly noticeable in Figure 4.6). This again reaffirms the observation made in Section 3.5.1 where it is mentioned that a network scenario with high pause time values and high speeds in the random way-point model produces a more stable network than the same scenario with lower speeds. The lower level of mobility can be deduced from the fact that both protocols use less routing overhead at the higher node speeds.

The routing overhead in terms of bytes shows that TAODV's routing overhead is less in the 50 node network configuration. However, this advantage is reversed in the 16 node network configuration. This is due to the larger packets sizes of the route request and route reply packets used by TAODV. This increase in size is as a result of the inclusion of positional information data of a node which is used for the route localization algorithm. The effect of the increases packet size is not noticed in the byte overhead graphs for the 50 node configuration (Figure 4.7b and Figure 4.8b) since the routing overhead improvement in terms of routing packets is considerably higher. However, it should be noted that due to the nature of the MAC protocol used in 802.11b standard, the cost of gaining access to the wireless medium is far more significant than the per byte transmission cost [Das02]. Thus

reducing the overhead in terms of routing packets has a greater impact on the overall performance of a protocol than the per byte routing overhead reduction.

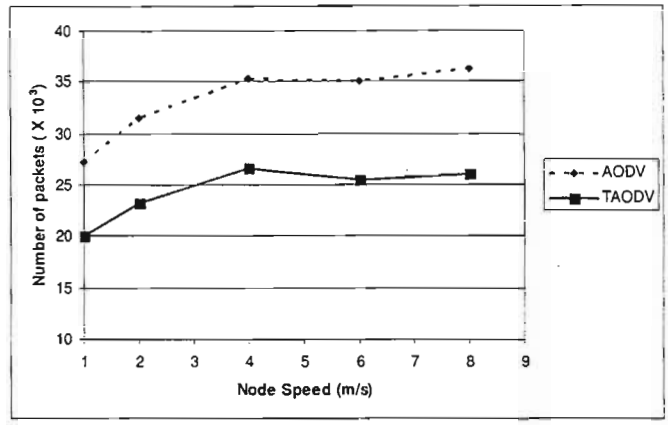


(a) Routing overhead in total number of packets

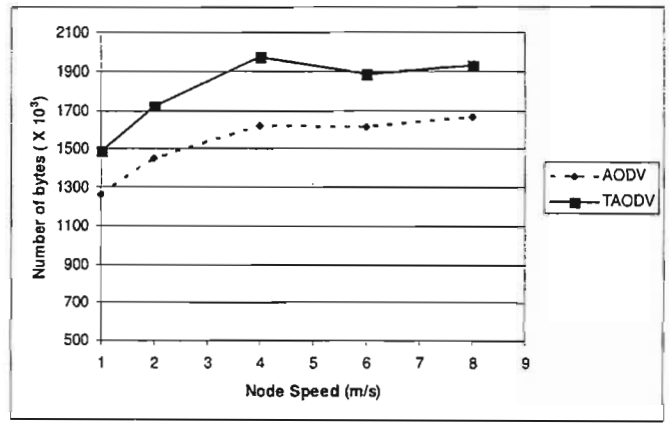


(b) Routing overhead in total number of bytes

Figure 4.5: Routing overhead performance in a highly mobile 16 node network (pause time of 1 second)

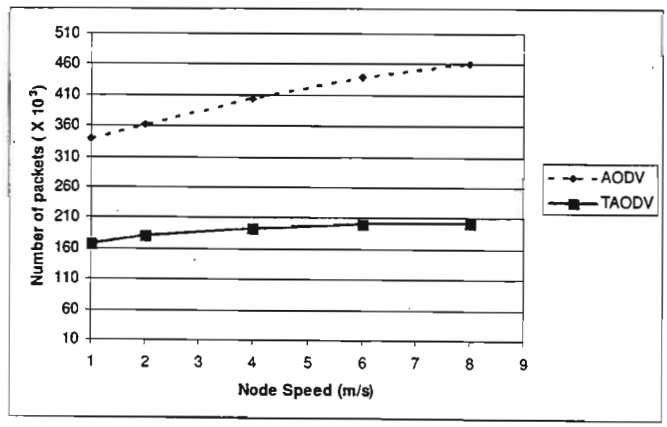


(a) Routing overhead in total number of packets

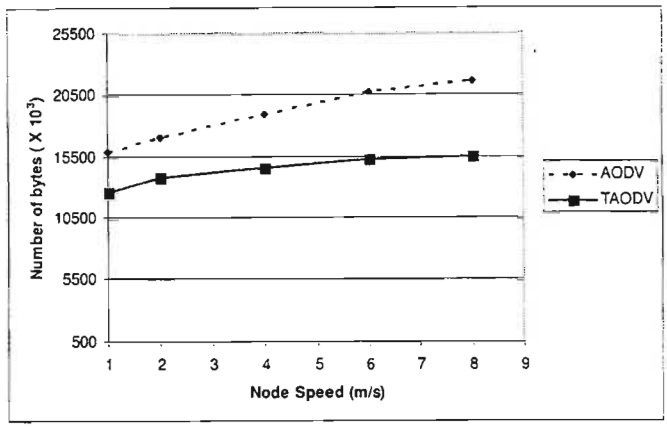


(b) Routing overhead in total number of bytes

Figure 4.6: Routing overhead performance in a stable 16 node network (pause time of 250 seconds)

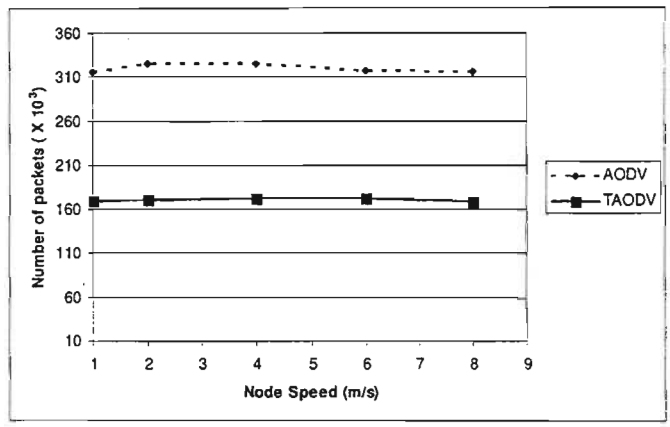


(a) Routing overhead in total number of packets

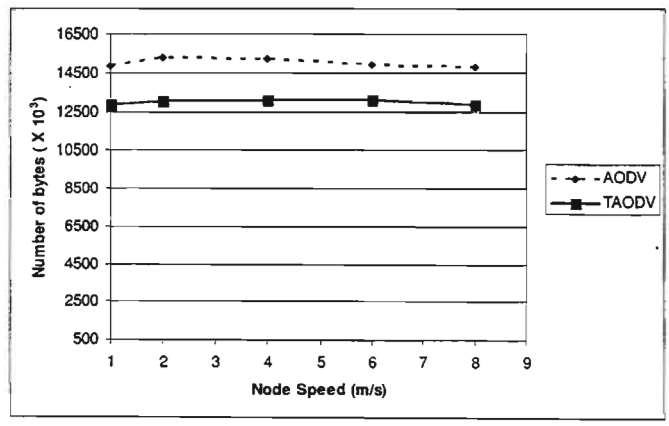


(b) Routing overhead in total number of bytes

Figure 4.7: Routing overhead performance in a highly mobile 50 node network (pause time of 1 second)



(a) Routing overhead in total number of packets



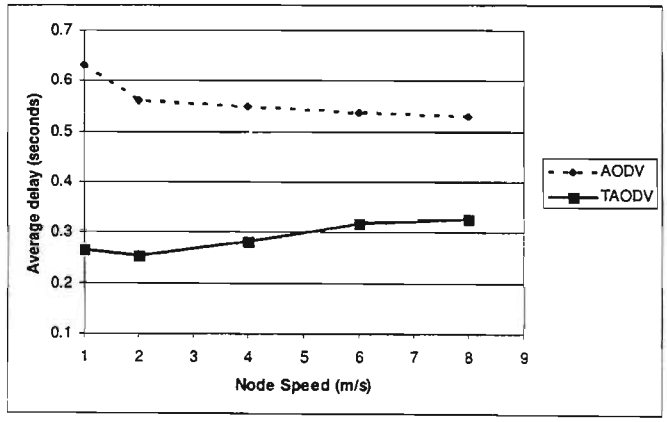
(b) Routing overhead in total number of bytes

Figure 4.8: Routing overhead performance in a stable 50 node network (pause time of 250 seconds)

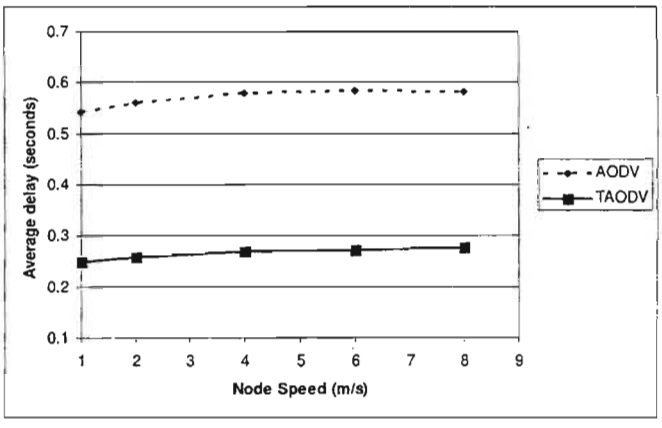
4.7.3 Average end-to-end delay

The average end-to-end packet delivery results show that TAODV delivers packets with less delay than AODV in all of the scenarios considered for the simulations. The delay is significantly lower in the 16 node network configuration where there is a considerable network traffic load. In both the high mobility and the low mobility scenario sets, the delay shown by TAODV is at least 200ms lower. In the larger 50 node network the delay performance difference exceeds 100ms for the high mobility network scenario. This is similar to the results shown in [Quazi03b]. The quantitative value for the delay in the 50 node is considerably higher than the 16 node network, and this is expected since (on average) the routes in this network will have a higher number of hop counts. The magnitude of the end-to-end delay for AODV is close to that presented in [Das02] for a similar network setup.

The reduced delay is shown by TAODV is expected as the protocol prevents the creation of congested critical nodes in the network. Such congestion results in packets being queued at overburdened protocol queues. Since these queues are implemented in a first-in-first-out (FIFO) structure, such packets will remain queued for a considerable amount of time. It should be noted that TAODV's route selection procedure will choose a route with a higher hop count if the route load is lower. It would be normally expected that traversing through extra hops would increase delay. However the delay results presented in the graphs that follow show that delay due to queuing at nodes in a congested route is considerably higher than delay due to less loaded routes with higher hop counts. The load balancing executed by TAODV results in network traffic being shared amongst the nodes in the network. This has a marked effect in reducing the latencies in the delivery of data packets.

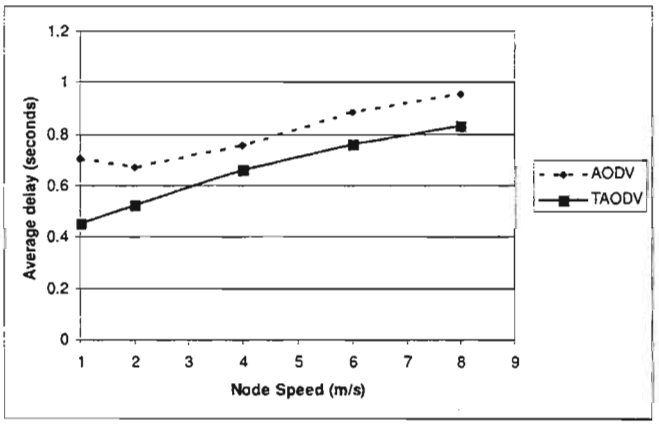


(a) Delay performance for a highly mobile network (pause time of 1 second)

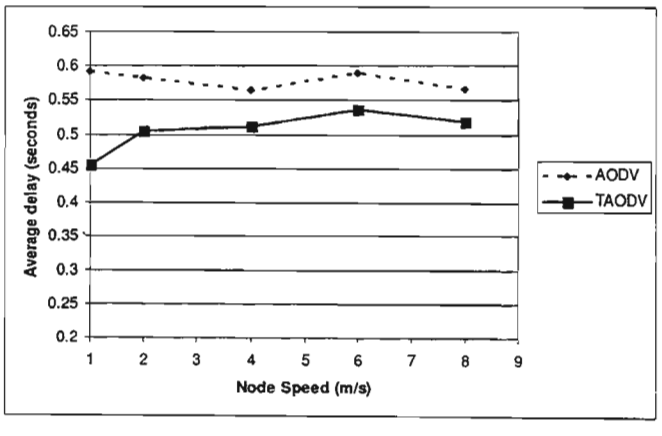


(b) Delay performance for a stable network (pause time of 250 seconds)

Figure 4.9: Average end-to-end delay performance for a 16 node network



(a) Delay performance for a highly mobile network (pause time of 1 second)

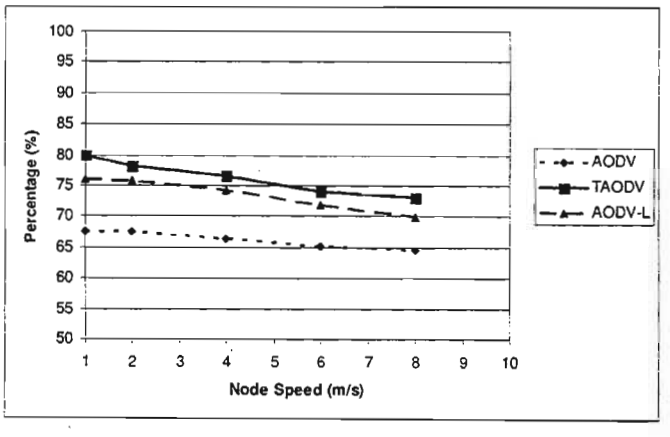


(b) Delay performance for a stable network (pause time of 250 seconds)

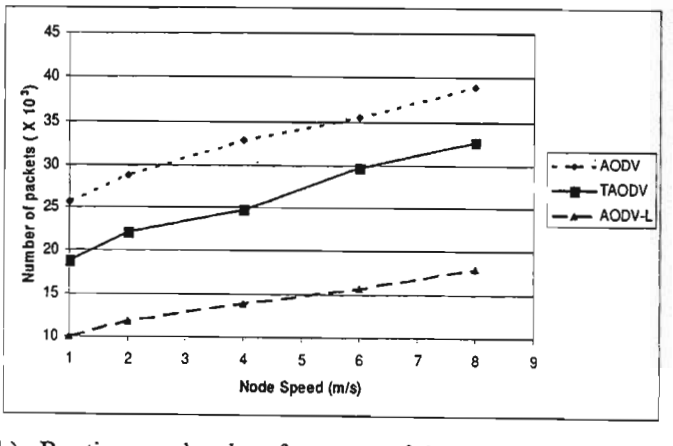
Figure 4.10: Average end-to-end delay performance for a 50 node network

4.8 The effect of TAODV's load checking algorithm

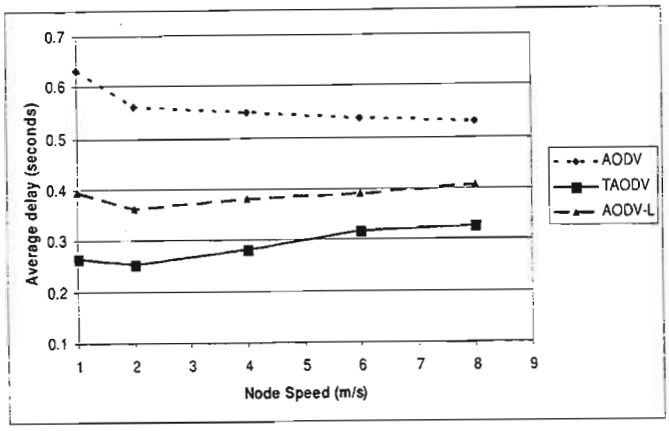
In order to demonstrate the effect of TAODV's load balancing scheme on its improved performance, a protocol was implemented that only implemented the protocol's query localization algorithm. The AODV protocol was modified to only include the query localization optimization of TAODV. The performance of the new protocol (labelled as AODV-L in Figure 4.11) was compared to that of TAODV and AODV. A highly mobile 16 node network setup was used. In terms of routing overhead (shown in number of packets), the result in Figure 4.11(b) shows that AODV-L outperforms both TAODV and AODV. The advantage over AODV is expected since AODV-L uses the location aided query localization which significantly reduced routing overhead. TAODV however incurs higher routing overhead than the new protocol as it uses additional routing mechanisms to implement its load checking algorithm.



(a) Packet Delivery ratio performance of the simulated protocols



(b) Routing overhead performance of the simulated protocols



(c) Average delay performance of the simulated protocols

Figure 4.11: Performance of protocols in a highly mobile 16 node network

The benefits of the load checking algorithm can be seen when TAODV is compared to AODV-L using the application specific metrics, namely packet delivery ratio and average delay. These results are shown in Figure 4.11(a) and 4.11(c). The load checking algorithm improves the packet delivery ratio by approximately 5 % and improves the delay by almost 100 ms at certain mobility speed.

4.9 Related work

At the time of writing the dissertation, the author was aware of the work in [Tripa02] and [Tripa01] as the only implementation of the location aided query localization as an optimization on an on-demand algorithm. This optimization, based on the concepts of the Location Aided Routing (LAR) algorithm, was implemented on AODV and the resultant protocol was called Location assisted AODV (LODV). In [Tripa01] the simulations used to compare the LODV protocol with AODV were done using NS-2, with a 50 node network. The three metrics used were Packet Delivery Fraction, Routing Load and Average Delay. The results for similar offered load to that used in this dissertation showed that routing load of the Location assisted AODV protocol was lower than AODV. In a simulation with 40 sources the routing load was 1.5 times better at high mobility scenarios. However, the application specific metric results showed that their proposed protocol had little advantage over AODV in such scenarios. In the 40 sources case the Packet Delivery Fraction and Average Delay were noticeably worse.

The results of the simulations using a highly mobile 50 node network in this dissertation showed that the proposed protocol (TAODV) was better than AODV using all three

performance metrics. The routing overhead was considerably less which conforms to the conclusion in [Tripa01], where the query localization optimization enhanced the internal efficiency of AODV. However, the advantage shown by TAODV in terms of the packet delivery ratio and average delay over AODV can be attributed to the TAODV protocol's load checking algorithm.

The designers of the Location Aided Routing (LAR) algorithm mention the possibility of using their scheme for such purposes in [Ko98]. In their implementation in [Ko98], they use the LAR scheme as optimization on a blind flooding algorithm used during the route discovery process. They compare their proposed schemes (LAR Scheme 1 and 2) with a flooding algorithm and show the merits of the schemes over flooding. The work in [Ko99] shows the increase in routing overhead efficiency due to location-based geo-casting over a flooding based geo-casting scheme. However since in both cases they do not implement the optimization on any particular protocol, they do not present the effect of the optimization on other important performance metrics such as data throughput or routing delay.

As mentioned in Section 4.3, the work in [Lee01] and [Wu01] has demonstrated the benefits of implementing load checking algorithms for routing in an ad-hoc network. However, to the knowledge of the author, there has not been a proposed ad hoc on demand algorithm that combines both a location aided query localization and load checking algorithm in its routing mechanism, and this has been the aim of this research work.

4.10 Conclusion

It has been shown in this chapter that, despite being well suited for routing data in a mobile ad-hoc network, on-demand algorithms proposed thus far use mechanisms that reduce the advantage that they have over table driven schemes. Most on-demand algorithms use blind flooding technique in the route discovery phase of their operation. The work presented in this chapter shows that if a more efficient flooding technique is used, the performance of on-demand algorithms can be improved. The query localization method of the proposed algorithm in this chapter shows a significant performance improvement in terms of routing overhead over the protocol upon which it is based. When the number of protocol routing packets is considered the proposed protocol incurs over 20% lower overhead than AODV in the 16 node network configuration. This advantage increases to over 50% in the high mobility 50 node network scenario.

The protocol also performs load checking in order to balance the data traffic among the nodes in the network. This leads to TAODV avoiding the dropping of packets at the overloaded interface queues of certain nodes in the network. As a result when compared to AODV, TAODV achieves over 10% increase in the packet delivery ratio under high mobility scenarios. The congestion avoidance helps to reduce the packet delivery latencies experienced by the proposed protocol. The increased efficiency of TAODV in using less protocol overhead to achieve its routing also prevents congestion, which contributes to the improved performance in terms of packet delivery ratio and end-to-end delay.

The purpose of the simulations in this chapter was to compare the performance of the proposed protocol with that upon which it is based. When considering the 16 node network setup, it is unlikely that the utilization of the PCS network will introduce traffic load levels that are as high as those used in these simulations. However, the results show that, with its location aided and load aware algorithm, TAODV will reach saturation in terms of packet delivery throughput at a significantly higher offered load than AODV. The performance of the proposed protocol in terms of the other application specific metric, namely end-to-end delay, shows that TAODV routes data packets with less latency. In the 16 node setup the maximum delay experienced by the packets was approximately 300 ms, which is within acceptable limits for audio applications. The performance advantage of the proposed protocol in a network with a larger set of nodes shows that it is more scalable than AODV. However the end-to-end delay performance of both the protocols indicates that it will not be feasible to run an audio application over such a large network.

Chapter 5

The Implementation of an On-demand Ad-hoc Routing Protocol

5.1 Introduction

It has been shown that simulations are good tools for determining the performance of ad-hoc routing protocols in a variety of scenarios. Due to the difficulties in comparing ad-hoc routing schemes, the conclusions derived from comparisons done using simulations are valuable. However, before a protocol can be used in real world systems it is vital that it is validated through implementation on a test-bed. This is because there are many simplifications and assumptions made in the simulation environment, which do not necessarily hold in the physical world. In addition the models used by the simulation packages are limited in their accuracy and can only give an approximate representation of the real world environment. Thus it is necessary to verify the functionality of an ad-hoc routing protocol by means of a physical implementation on a test-bed.

This chapter deals with the implementation of an on-demand ad-hoc routing algorithm on a test-bed network consisting of handheld mobile pocket PCs. The development of such test-beds has been made possible by the availability of cost-effective mobile computing hardware, and supporting software. Section 5.2 briefly describes the components of the test-bed used in this work. Section 5.3 deals with the routing architecture of the operating system used in the test-bed, namely Linux. The next section describes how such an architecture, although suited for the implementation of pro-active (table driven) routing protocols, does not support the requirements for the implementation of on-demand schemes. Section 5.5 describes a newly developed Linux kernel module, which when added to the operating system, provides the appropriate support that makes implementing an on-demand ad-hoc routing protocol possible. Section 5.6 describes the application developed that executes the routing protocol on a host in the test-bed. The chapter concludes with description of a number of test set-ups and results obtained therefrom, which are used to demonstrate the operation of the on-demand protocol proposed in this dissertation.

5.2 Description of the test-bed

5.2.1 The handheld device

The test-bed used for the purposes of this research is the Positional Communication System described in the introductory chapters of this dissertation. The mobile handheld device is the heart of the system. The device has the functionality that enables voice and data communication between hosts, as well as providing wireless network connectivity between them. The handheld device chosen for the PCS system is the HP (formerly Compaq) iPaq pocket PC [HP03] (A picture of the devices is shown in Figure 5.1). This product is the most advanced in its range and provides the necessary hardware and processing speed required for the application envisioned for the PCS.

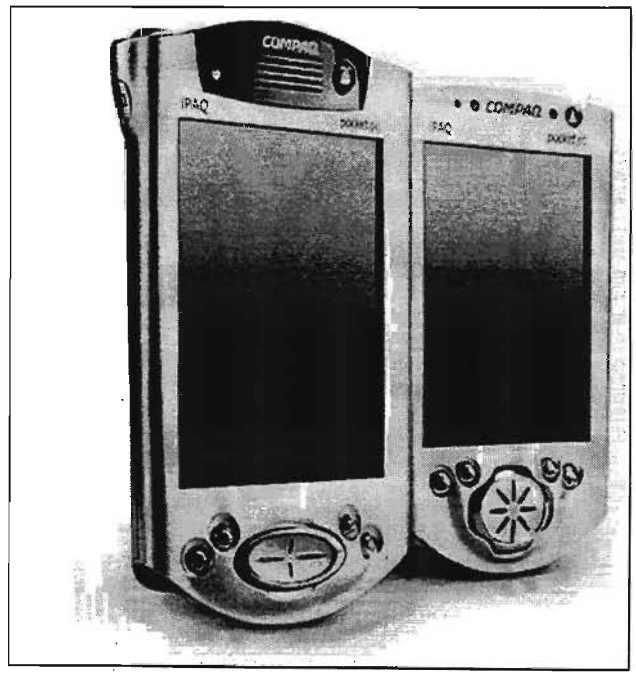


Figure 5.1: Picture of 3800 and 3600 Compaq iPaq's used for the PCS

A brief technical specification of the HP iPaq 3800 series:

- 206 MHz Intel StrongARM SA-1110 32 Bit RISC processor
- 64 MB of DSRAM
- 32 MB FLASH ROM
- Expansion pack system that allows for Type II PCMCIA cards to be connected. The wireless LAN card that provides the wireless link between hosts in the network is inserted in this slot.

- Microphone and speaker (for audio functionality)
- Serial and USB port. This functionality of the device is used for interfacing additional modules of the PCS system such as an image capture device and a GPS module.

The next two sub-sections describe the wireless interface and the operating system used in the test-bed. The functional specifications of these modules impact the design choices taken during the implementation of the on-demand routing algorithm. The wireless interface is described in detail as it is the underlying communication layer that is used by the routing application. The 802.11b is described to provide an overview of the functional characteristics of the standard, and what should be expected in terms of performance. The choice of the operating system is vital to the development of an application and the reasons for selecting Linux are presented in Section 5.2.3.

5.2.2 The wireless interface

The wireless interface of a host in the PCS network is provided by the Cisco wireless LAN adapter (Aironet 350) [Cisco03]. This card runs the IEEE 802.11b standard for wireless LANs. This standard is widely used for creating wireless LANs (hot-spots) that are accessed using mobile devices such as PDAs and laptops. The 802.11b standard [IEEE99], popularly known as Wi-Fi, uses an unlicensed portion of the radio spectrum in the 2.4 GHz ISM (Industrial, Scientific, Medical) band. It was produced by the IEEE Standard Association's Standard Board as an improvement on the 802.11a standard, and allows data rates up to 11Mbps. The standard defines only the bottom two layers of the OSI (Open Systems Interconnection) reference model, namely the Physical and the Data Link layers.

5.2.2.1 802.11b Operating modes

There are two modes of operation specified in the 802.11b standard. In the *infrastructure mode* the network consists of at least one access point connected to the wired infrastructure and a set of wireless hosts (as shown in Figure 5.2). The access point acts as a bridge between the wireless and wired network environments. In the *ad-hoc (peer to peer) mode*, the hosts in the network communicate directly with each other without association with any type of infrastructure. In the mobile ad-hoc PCS network, the wireless LAN cards are configured to operate in the ad-hoc mode.

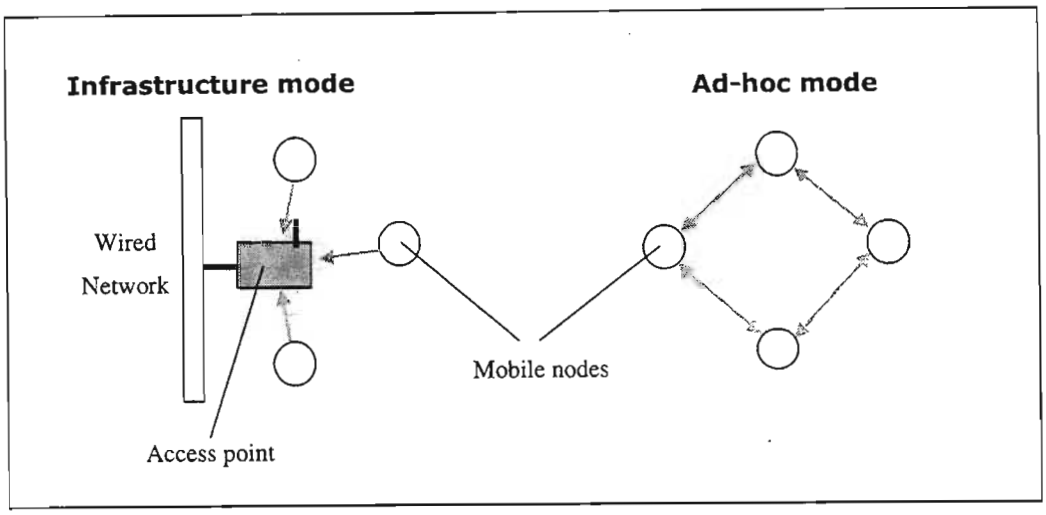


Figure 5.2: Different modes in 802.11b networks

5.2.2.2 The 802.11b Physical Layer

The IEEE 802.11a standard, the predecessor of 802.11b, achieves data rates of 1 Mbps and 2 Mbps using either the frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS) spreading techniques. However, 2 Mbps is the highest achievable data rate using the FHSS technique. To overcome this limitation, the IEEE introduced the 802.11b standard, which supports two new higher speeds: 5.5 Mbps and 11 Mbps. This was achieved by using DSSS as the sole physical layer spreading technique, and using a new advanced coding technique called Complementary Code Keying (CCK) with Quadrature Phase Shift Keying (QPSK) [Henty01]. In addition the standard specifies that the data rates be automatically adjusted depending on the noise conditions of the wireless channel. This is termed *dynamic rate shifting* [IEEE99]. Since 802.11b is backward compatible to its predecessor, it is able to offer the lower data rates of 1 Mbps and 2 Mbps. The dynamic rate shifting automatically adjusts the data rates to these lower values of 5.5, 2, 1 Mbps when either the devices running 802.11b move out of range for 11 Mbps transmission or when there is considerable interference. The devices revert to using the higher data rate when the channel condition becomes appropriate or when they are in range for such transmission.

5.2.2.3 The 802.11b Data Link Layer

The data link layer of the standard consists of two sub-layers:

- The Logical Link Control (LLC)
- Medium Access Control (MAC)

The MAC layer is designed to support multiple users on a shared medium by having a transmitting host sense the medium before accessing it. It performs a *physical carrier sense* with the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol, also known as the Distributed Coordination Function (DCF). A host intending to transmit on the wireless channel first senses it for activity. If the channel is inactive, the host transmits the data frame after waiting an additional, randomly selected period of time. If the receiving node receives the packet, it transmits an Acknowledgement (ACK) packet to the sender. The process is complete once the sending host receives the ACK. If the sending host does not detect the ACK, either because the data packet was not received intact or the ACK was not received, it assumes that a collision has occurred. This results in the sending host waiting an additional time period before it attempts retransmission. Thus this explicit acknowledgement of each data packet handles interference and other radio related problems in the wireless channel.

The physical carrier sensing is not sufficient to remove the possibility of collision due to two hosts attempting to access the wireless channel simultaneously. Collisions are caused by what has been called the "hidden terminal" problem. This problem is demonstrated in Figure 5.3.

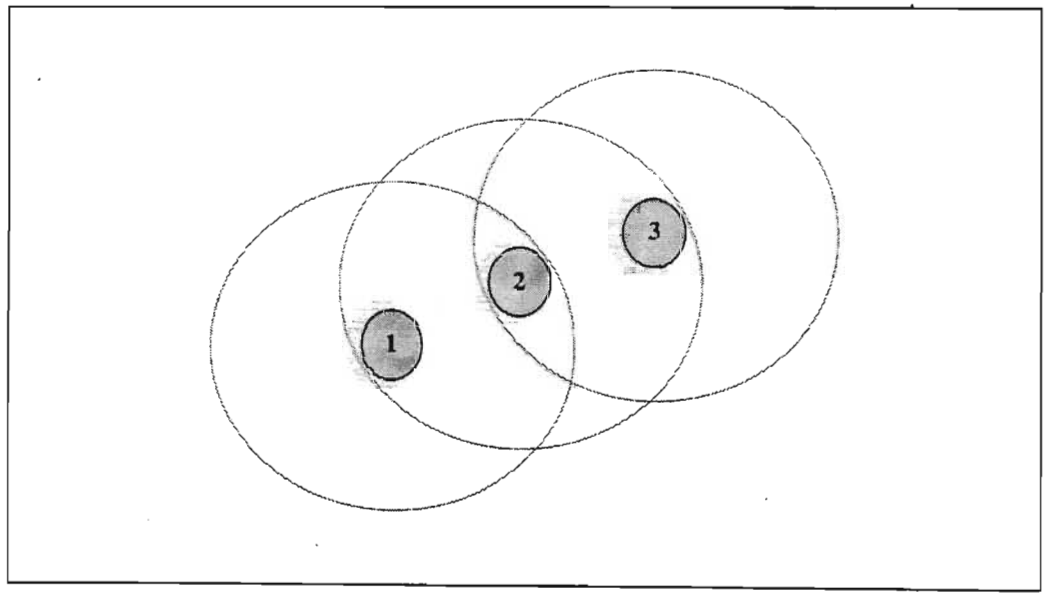


Figure 5.3: Hidden Terminal problem

The diagram in Figure 5.3 shows a network consisting of nodes 1, 2 and 3. The transmission range of each node is indicated by the lighter line around each node. As shown, nodes 1 and 3 are not in range of each other, but are in range of node 2. Now if 1 and 3 attempt

transmission simultaneously, they will not detect each other's transmission and assume that the channel is free. However, the result will be a collision in transmission at node 2. In order to avoid collisions due to this problem, the 802.11b standard implements a *virtual carrier sense* protocol. This protocol at the MAC layer causes the sending host to transmit a Request -to-Send (RTS) and wait for Clear-To-Send (CTS) reply from the receiving host. All the hosts in the transmission range of the receiving host will also receive the CTS. This will prompt them to delay any intended transmissions, allowing the sending host to transmit a data packet and receive an ACK without any chance of collisions. Thus a four-way handshake is specified by the standard for each transmitted data packet with the RTS/CTS/DATA/ACK sequence.

The IEEE specified two further features in the MAC layer with the intention of adding robustness to wireless transmissions. The first such feature is the inclusion of a Cyclic Redundancy Check (CRC) checksum with each data packet, ensuring that it is not corrupted in transit. The second feature is the fragmentation of large packets before transmission. In highly congested, high interference scenarios larger packets have a greater chance of becoming corrupted during transmissions. Packet fragmentation allows the larger packets to be split into smaller units before they are sent over the wireless medium. This reduces the necessity for retransmission and improves overall network performance. The MAC layer performs the reassembly of the packet fragments at the receiver host, making the fragmentation process transparent to higher-level protocols.

5.2.3 The Operating System

There were essentially two choices in selecting the operating system for the test-bed, namely Windows CE or an embedded Linux OS. In order to implement ad-hoc network routing, it was imperative that the operating system allowed the ad-hoc routing application to manipulate the information held by its routing architecture. To the knowledge of the author there is no freely available source code or any documented programming interface available for the Windows CE protocol stack. Thus, attempting to run the intended PCS application on a Windows CE platform would be difficult. This led to the choice of Linux for the intended task. The open source nature of the operating system as well as the documentation support available makes it ideal for the development endeavours undertaken with the PCS. Linux provides full access to the IP protocol implementation and the OS routing structures. This makes the manipulation of such structures for ad-hoc routing an achievable task. There are a number of embedded Linux distributions available each varying in its look and feel (with the

kernel implementation remaining standard). The *Familiar* distribution was chosen for this test-bed since its development is primarily focused on the HP (Compaq) iPaq pocket PC platform. The distribution is constantly updated with each iPaq model and is readily available from the hanhelds.org [Handh03]. This website is hosted and maintained by HP and is a rich source of documentation and user-group support for system developers who use the iPaq packet PCs.

5.3 The Linux routing architecture

Linux, as with many other clones of the UNIX operating system, provides two modes of operation in the operating system. There exists the *user-space* where applications written by the user/developers are executed. All operations specific to the operating system are executed in the *kernel-space*. The interaction between the kernel and the user space occurs by means of "system calls" made to the operating system.

The routing architecture in Linux is also modularised and follows a similar set-up. The routing functionality is split into two sub-sections [Peter00]:

i. Packet forwarding
ii. Packet routing

The packet forwarding process is called when a packet needs to be routed to a host. The system consults the host's routing table and sends the packet towards the destination host using the information in the entry for the destination (if found). This functionality resides in the kernel space since the process, which is executed for each packet passing through the system, needs to be fast and efficient. The packet routing process refers to the functionality of gathering information required for building the forwarding or routing table. This process is what is commonly referred to as the routing algorithm or protocol, and can involve complex decision-making. It is due to the complex nature of the task that it resides outside of the kernel space in user-space. This methodology of separation has meant that the modern operating system is efficient and flexible. As attractive as it appears, however, the current architecture is not suited to support the implementation of an on-demand ad-hoc routing algorithm. The reasons for this are described in Section 5.4.

5.4 Requirements of on-demand routing

It has been shown in [Gupta02] that pro-active protocols can be implemented fairly easily within the architecture described above. The table driven ad-hoc protocol is efficiently

executed as a user-space daemon[1], similar to the way protocols such as RIP [Rip03] and OSPF [Ospf03] are run in the wired network architecture. Figure 5.4 shows an overview of a proactive routing protocol (DSDV) implemented on the Linux platform [Gupta02].

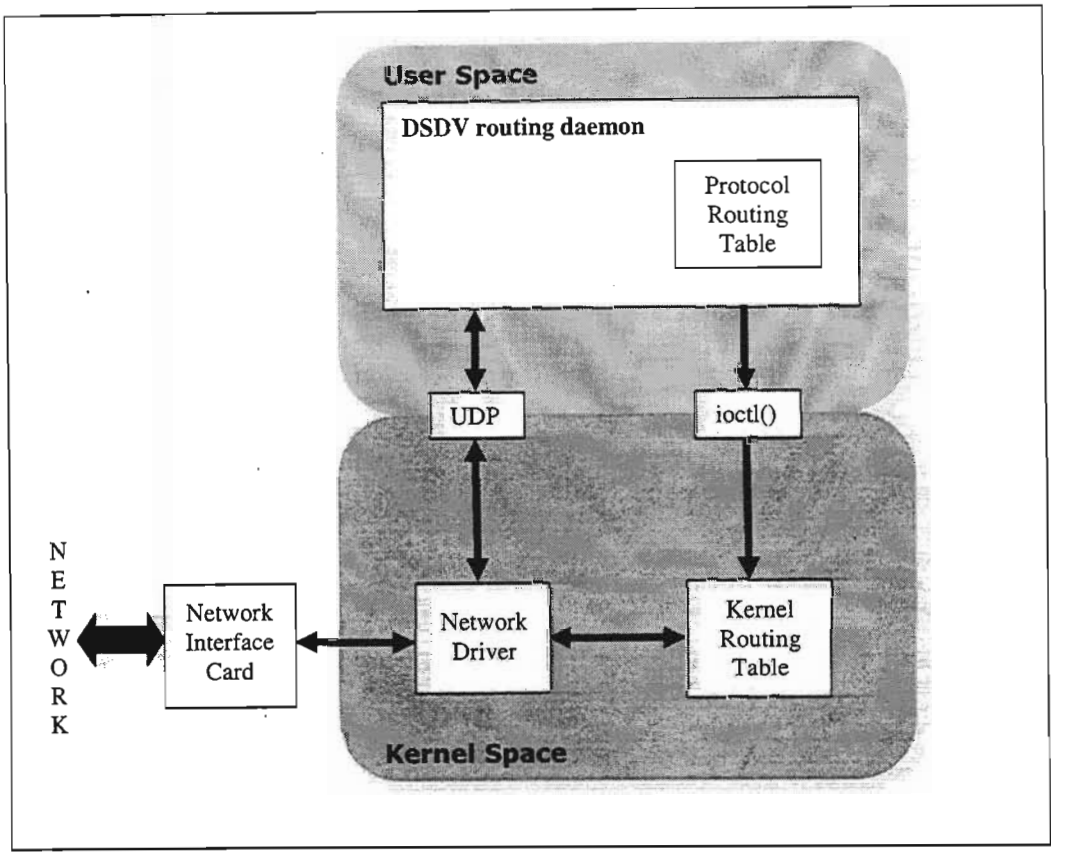


Figure 5.4: Layout of the DSDV implementation in Linux

The protocol routing daemon maintains a local copy of the forwarding or routing table called the *dsdv route table*. The necessary information for all the routes in the network that the host is aware of is contained in this table. It is updated and modified whenever the host receives routing updates from its neighbours via its wireless interface. The host also broadcasts its own routing table periodically or when triggered by a significant change to the information contained in the table. The kernel's routing table is updated at the appropriate time intervals with the information from the protocol's route table by means of the Linux *ioctl()* system calls. These commands cause the kernel's routing information to be modified by the routing daemon.

[1] A daemon is a user-space process that runs as a background operation with no interface to the user.

The task of implementing an on-demand algorithm, however, is not accomplished as easily. The reason for this is the difference in the routing models used by on-demand and pro-active schemes. In on-demand protocols routes are discovered and maintained only when they are needed. It is most often the case with such a mechanism that a route will not be known *apriori*. When a request is made for an unknown route, it will be necessary to initiate a route discovery process in order to find a route to the destination node. This action is not possible with the current Linux routing architecture. If a route to the destination host is not found in the kernel's routing table, the system immediately drops the packet. In order to allow the operation of an on-demand routing daemon, two mechanisms are needed:

i. The routing daemon needs to be informed of the route request
ii. The packet that requested the route has to be temporarily buffered. The period that the packet is buffered should be long enough for the daemon to discover the route (if the destination is reachable)

There is a timer associated with each entry of the routing table of an on-demand routing algorithm. This timer indicates the last instance that the route was used. This information is used by the routing daemon to distinguish active routes from stale ones. If a route has not been used in a specified timer period, the daemon considers such a route outdated and thus obsolete. The appropriate action is to purge the route information from the route cache, both in the user-space daemon and the kernel space. However, if there is no mechanism that allows the kernel to inform the daemon of the last time a route is used, the daemon's route table maintenance procedure will prematurely label entries as obsolete and purge them. The current Linux routing architecture offers no such mechanism and thus the caching functionality of an on-demand algorithm will not function as intended.

In order to perform the on-demand functionality, what is required from the routing daemon is the ability to monitor each packet that passes through the routing system at a host. Given this ability, the daemon would be able to inspect each packet and take appropriate action depending on the routing information that is available. If a packet requests a route that is already known to the daemon (and thus the kernel), it passes the packet to the kernel. In kernel space it is processed by the kernel's packet forwarding functionality. If, however, the route requested is not available then the daemon initiates a route discovery process and temporarily buffers the packet until a decision is made on the route request i.e. either the route will be found or it will be determined that the destination requested is unreachable. Since all the packets that pass through the system are processed by the daemon, it will gather

information regarding the last time a route was used. It is the flexibility of the Linux operating system that has enabled the development of a software architecture that allows the packet monitoring handling functionality required by on-demand protocols. The *Netfilter* [Netfi03] kernel module was developed (by the Netfilter core team [Netfi03]) for general packet mangling but is ideally suited for the implementation of an on-demand ad-hoc routing protocol. This module is described in the Section 5.5.

5.5 The Netfilter Structure

The netfilter architecture is a recent addition to the Linux networking stack. It is available as a kernel module in the Linux kernel versions beyond 2.4 [Russel02]. It is a framework for modifying the routing data structures specific to a packet that arrive at a host, are routed by the host or are generated by host. This action has coined the term packet mangling in the terminology of the Linux world. This packet mangling is achieved by the netfilter structure outside of the Linux socket interface, again emphasising the flexibility of the Linux operating architecture. There are three major components to the netfilter infrastructure as described in the following sub-sections.

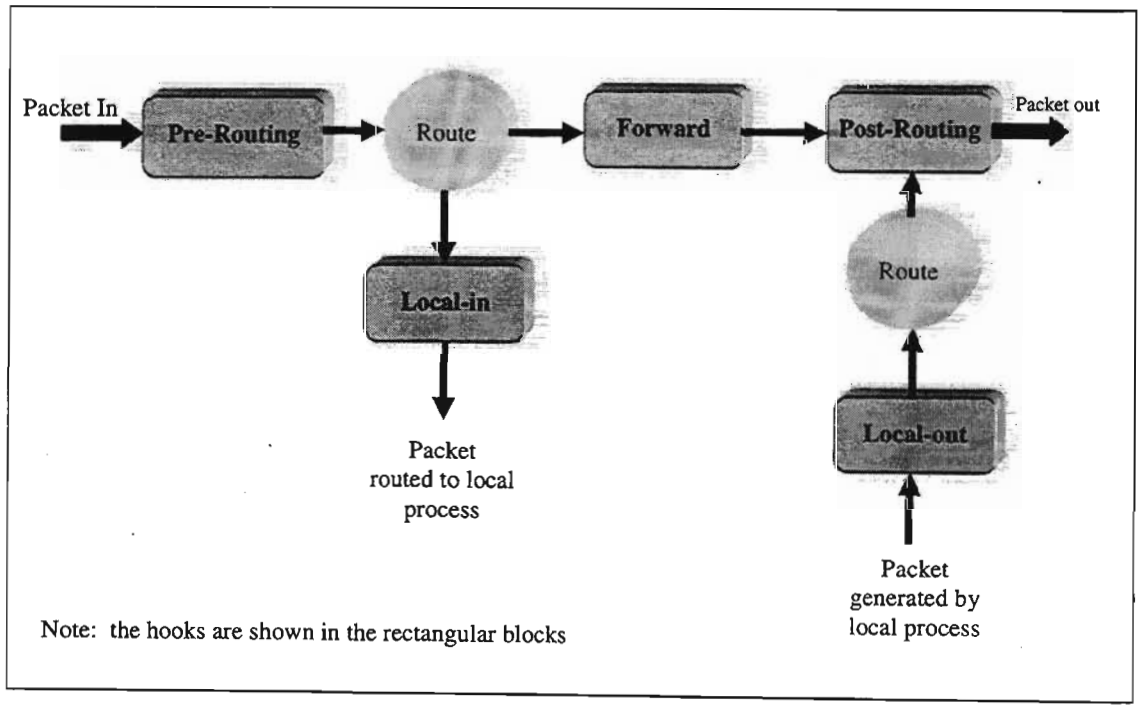


Figure 5.5: Netfilter hooks for IPv4

5.5.1 The hooks

Each protocol running at a host (e.g. IP) defines "hooks" which are well-defined points through which a packet traverses as it goes through the network protocol stack. Whenever a packet arrives at these strategically placed points the protocol calls the netfilter framework with the packet and the "hook" number at which the packet was intercepted. The position of the hooks in the IPv4 can be seen in Figure 5.5 (the hooks are shown in the darker blocks). The hooks for each protocol are defined in the protocol specific header files, thus in the case of IPv4 it is found in the "linux/netfilter_ipv4.h" kernel header file.

5.5.2 The kernel modules

These modules are system processes that can be loaded to give a running Linux kernel additional functionality. The kernel modules specific to *netfilter* register themselves and listen on the different predefined hooks for each protocol. When a packet arrives at a "hook" any modules that are registered for the specific protocol and "hook" are notified and appropriate action is taken.

The netfilter structure defines one of four possible actions:

i. **NF_ACCEPT** : Allow the packet to pass through the hook to be processed by the system
ii. **NF_DROP** : Drop the packet
iii. **NF_QUEUE** : Queue the packet for user-space handling
iv. **NF_STOLEN** : Request by the netfilter structure to ignore the packet

5.5.3 The actions

This component of netfilter deals with undertaking the actions requested by the kernel modules. It provides the software functionality to enable the queuing of packets in the kernel space for a user-space application.

5.6 Description of the routing application

The routing application developed that implements the on-demand routing algorithm proposed in this dissertation uses the netfilter framework for it's functionality. Similar to most routing applications it runs as a user-space daemon. Thus once initiated it needs no further input from the user. The application is labelled TAODVD[1] and is based on the AODV-UU [Aodv-uu03] implementation of the AODV routing protocol. The TAODVD application is primarily written in C and uses the modules described in the following subsections. The daemon deals with data and routing packets separately. Routing packets include all packets that are used by the routing protocol for the purposes of creating and maintaining the routing information at a host. Data packets are all packets that are generated by applications residing in the higher layers. The packet handling functionality is modularized and can be seen in Figure 5.6.

5.6.1 The kernel module

As mentioned in Section 5.5, a routing protocol's kernel module registers the hooks with *netfilter*. Since the routing is done over IP, the module registers the application to "listen" on hooks specific to IPv4. When a packet arrives at any of the registered hooks, the module is notified and it has a set of rules which determine the fate of the packet. The TAODV kernel module registers to listen on the following IPv4 hooks:

i. NF_IP_PRE_ROUTING
ii. NF_IP_LOCAL_OUT
iii. NF_IP_POST_ROUTING

The NF_IP_PRE_ROUTING hook found at the onset of the kernel routing structure, is used to capture all of the host's incoming packets. At this point no routing has been done on the packet. The NF_IP_LOCAL_OUT hook grabs all the packets that are generated locally by the host while the NF_IP_POST_ROUTING hook is used to re-route all packets for which routes have been found and are about to be sent.

[1] The "D" at the end of the name implies daemon

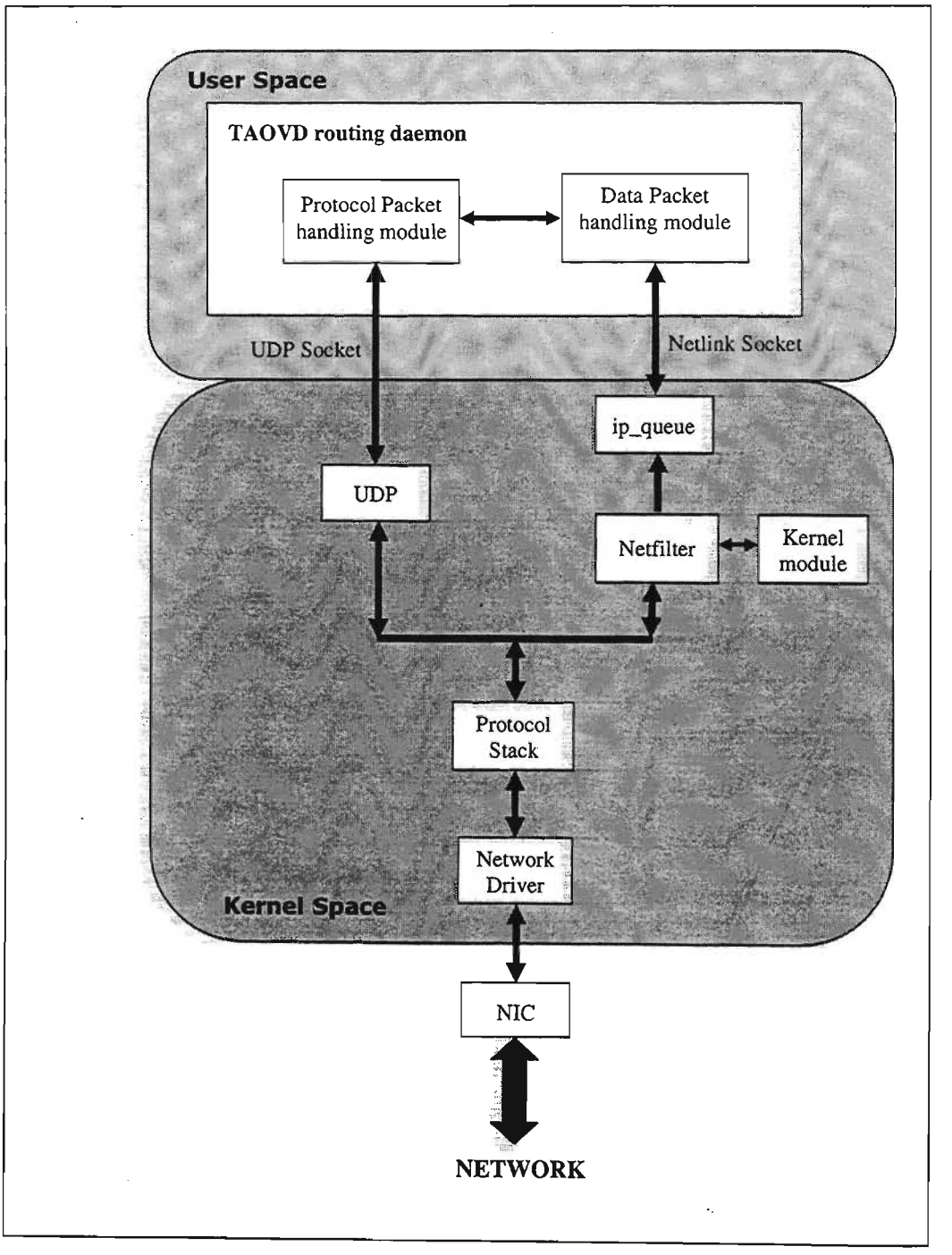


Figure 5.6: Structure of the TAODV routing daemon

The re-routing is used to ensure that the latest information available to the kernel's routing table is used for the forwarding of a packet. The action taken at the first two hooks is to *"queue"* the packets for further processing in user-space. The action taken at the third hook is to *"accept"* the packet so that the kernel's forwarding function can deal with it.

5.6.2 The queuing module

The netfilter architecture can be used to queue packets to user-space where the calling application can do further processing on the packet. In order to enable this queuing functionality, the operating system specifies that a *queue handler* has to be specified by the application that registers on the netfilter hooks. The queue handler is a kernel module that will perform the mechanics of passing packets to and from user-space. For IP routing, the standard queue handler is the *ip_queue* kernel module. The module uses a special type of socket known as Netlink sockets to allow communication between the kernel and user space applications. Once the ip_queue module has been loaded into the running Linux kernel, IP packets may be queued for user-space processing via the QUEUE target (action) of the netfilter rules. The user-space application communicates with the ip_queue module using an API (Application Programming Interface) developed by the netfilter core team called libipq [Morris01].

5.6.3 The user-space data packet handling module

This module determines the fate of data packets that are queued to user-space using the netfilter structure. All packets that are queued by the routing daemons kernel module go through this structure. Once a decision has been made on the path of a packet, this module returns the decision to the queuing module by means of the *verdict* command. If a data packet is destined for the current host or is a broadcast packet, an ACCEPT verdict is sent to the queuing module. This allows the queuing module to re-inject the packet into the kernel's routing structure to be processed by the operating system. If a packet does not meet either of the above conditions, the packet will be processed by the routing daemon which will choose to either forward, queue or drop the packet.

The first action taken by the daemon is to consult the protocol's internal routing table. If an active route to the destination is found, the packet is forwarded along the route. If however, the packet was generated by the local host and there is no known route to the destination, the packet is queued in the daemons protocol packet queue. This data structure maintained by

the daemon uses a packet unique packet ID to buffer the packet in a first-in-first-out (FIFO) queue. As it was done in the NS-2 simulations environment, the length of the protocol queue is taken as a measure of load at a host. Once the packet requesting an unknown destination has been queued, the data packet handling module prompts the protocol's routing module to initiate a route discovery process. When the protocol has made a decision as to the fate of a packet, the *verdict* along with the unique packet ID is returned to the queuing module which carries out the specified action.

5.6.4 Routing protocol packet handling module

Routing protocol packets are generated by the daemon for the purposes of gathering and maintaining routing information at the host. There are three types of packets used in this regard:

i. Route request (RREQ)
ii. Route reply (RREP)
iii. Route error (RERR)

The RREQ and RERR packets are broadcast packets while a RREP is a unicast packet. As can be seen in Figure 5.6 the routing daemon handles routing packets separately from data packets. While the data packets are passed to the user-space via the netlink socket, routing packets are sent through a protocol specific UDP (User Datagram Protocol) socket. The routing daemon, on initialization, creates a protocol socket which is bound to a port specific to the routing application. Once this is done, the daemon is able to receive and transmit protocol messages by reading and writing to the protocol socket respectively. The routing protocol module interacts with the data packet handling module to perform the routing at the host. In the scenario where a route is requested to a destination and the route is not known to the host, the routing protocol module initiates a route discovery by generating and sending a route request packet (RREQ). If, and when, a route is found to the destination, the routing protocol module instructs the data packet module to dequeue and send all the packets buffered for the destination node using the newly discovered route. If, however, the destination cannot be reached, (or in the event of a link breakage to a next hop neighbour) the routing protocol module generates a route error (RERR) message. This message is then broadcast to the node's immediate upstream neighbours. The data packet handling module is notified of failure of the route discovery attempt. This reception of such a message is an instruction to the packet handling module to purge all packets stored in the protocol queue for the now unreachable destination.

5.6.5 Development status

The current implementation on the PCS test-bed executes the load aware algorithm of TAODV. At the time of this writing the location aided route localization of the routing protocol was not implemented. This is planned as the next step in the implementation of TAODV in the future work. However, it should be noted that the route localization of the on-demand protocol is an optimization on the route discovery technique. Thus, although critical to the improved performance of the protocol, especially in terms of routing overhead, it is not essential for the routing functionality. The protocol has been designed such that if location information is not available, for reasons such as GPS information cannot be acquired, the normal blind flood is executed in the route discovery process. The GPS module of the PCS has been developed and tested, and in the future of the implementation work the necessary data structures will be added to the routing daemon to enable the location aided route localization functionality.

5.7 Test setup and results

The purposes of the tests run on the test-bed were to demonstrate the multi-hop functionality of the on-demand routing application. The tests presented at this stage were not exhaustive as a small network with very light traffic load is used. The tests were run in an indoor environment with a network consisting of three nodes. The network was setup on the fourth floor of the School of Electrical, Electronic and Computer Engineering (SEECE) building in a setup shown in Figure 5.7. A stationary node (SN) was at Position 1 which is in reach of Position 2 but out of reach of Position 3. An intermediate node (IN) was situated at Position 2 while a roaming mobile node (MN) was at the outer most reach of the stationary node at Position 3. The intermediate node would act as the router for the multi-hop configuration.

The range of each node was experimentally determined by using the Ping application of the Linux Familiar distribution that is being run at each node of the PCS network. This program, originally developed for fault detection on the Internet, uses Internet Control Message Protocol (ICMP) Echo Request and Echo Reply packets to determine whether a given system is working and available [Satch00]. A node constantly "pinged" its peer while moving away, until it reached a point where there was no further response. At this point, the node was considered to be out of reach of the node with which it was communicating. Position 3 was chosen to be far beyond the possible reachable limit of a node at Position 1 as this would

ensure that any transmission between nodes at the respective positions would occur through the router at Position 2.

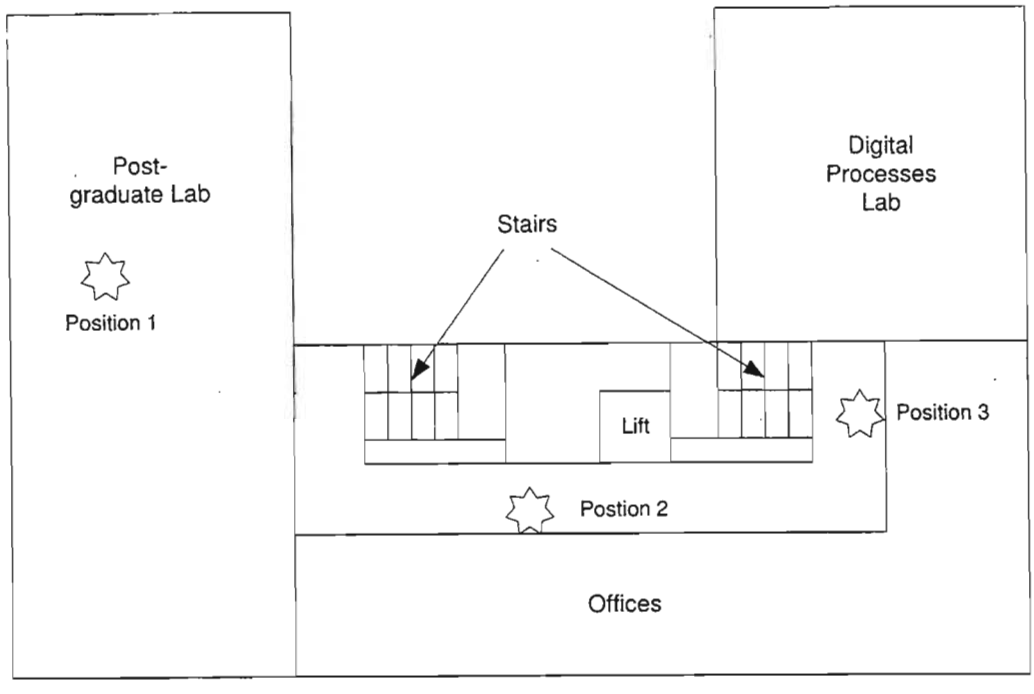


Figure 5.7: Indoor test-bed network setup

5.7.1 Ping Test

This test was done to determine the round trip packet delivery success using the Ping application. Each Echo Request packet sent includes a timestamp that is used to calculate round trip delays. The data traffic for the test consisted to the mobile node (MN) sending 125 byte pings to the stationary node (SN). The ping program supplied with the Familiar Linux distribution is limited in the options available to a user. The Ping application, available with Linux for desktop PCs, has the option to vary the number of packets sent per packet. The packet/seconds rate for the Ipaq could not be changed and therefore the default rate of one packet/second was used. Although this was a very low traffic load, it is useful as an initial test to determine if the routing application allows multi-hopping. The first set of tests conducted was a static network test with the SN at Position 1, the router or IN at Position 2 and the MN at Position 3. The packet delivery ratio is calculated by dividing the number of received packets by the number of ping requests. A set of 5 tests was attempted with 100 ping requests sent with each attempt. The averaged delivery ratio achieved was approximately 99.5% with the average round trip delay statistic being 24.72 ms. The second set of tests added mobility to the network configuration. The tests began with the first ping

request being sent while the MN is either at Position 1 or Position 3. In the former case the MN started in the vicinity of the SN and moved at an approximate speed to 1 m/s to Position 3. In the latter case, the MN began its route at Position 3 and moved at 1 m/s until it was near the SN. In the first case the MN was initially one hop away from the destination and moved to be two hops away while in the second case the scenario was reversed. The MN, in order to communicate with the SN, would have to discover a route through the IN whenever it was two hops away. The result obtained is an average of 4 test runs, two with the MN moving away from and two moving towards the SN. The delivery ratio displayed in the tests was 99% with an average round trip delay of 22.4 ms.

5.7.2 Real-time Voice over IP test

In order to demonstrate the routing speed of the routing daemon, a time critical application in the form of a real-time voice over IP program was used. A voice over IP (VoIP) application can be loosely described as a system that samples and segments a voice signal into frames. These frames are stored as voice packets which are then transmitted using the UDP/IP protocol suite to the intended destination host. At the receiver, the packets are received (over IP) on a UDP socket, queued and played back. The voice application used for the tests was custom developed with sampling parameters to provide toll quality speech. The sampling frequency was set to 8 KHz with 256 levels of linear quantization using an 8 bit Pulse Code Modulation scheme. A single channel (mono) recording was used with a packet segment size of 128 bytes. At this sampling rate the rate at which packets were sent was 62.5 packets/second. A streaming audio file was played onto the microphone of the SN which sent the packets to the MN where it was played back. The send and receive applications were commenced with the MN in the vicinity of Position 1 in the network setup and then it moved at the approximate speed of 1 m/s to position 3. The MN was stationary at this point for a few moments before it moved back to its starting position. At this point the applications were terminated. The sending and receiving applications, before exiting, recorded the number of voice packets sent and received in a text file. These values were used to determine the packet delivery ratio.

In each test run, the voice could be clearly heard at the MN from the start to the end of the test. The exceptions were at the instances in the MN's journey where it moved from being one hop to two hops away from its destination. During this transition the MN had to initiate a route discovery process. This caused a short delay of which resulted in a few crackles in the playback of the voice. These delays were almost instantaneous and the disturbance was

momentary. Approximately 50 000 packets were sent in each test and the average delivery ratio achieved over 3 runs was 98%. With the given quality of voice, the 2% loss was hardly noticeable. Further investigations need to be done to determine the quality degradation with higher delivery losses.

The packet transmission rate used in the voice application was significantly higher than that used in the simulations done in this dissertation. This is due to the fact that the VoIP programme used no compression technique and thus consumed a bandwidth of 64 Kbps. This is hardly feasible in the bandwidth constrained ad-hoc network and a suitable compression codec will have to be used for the intended audio conferencing application of the Positional Communication System. There are compression codec standards available such as GSM 06.10 [GSM03], the ITU's G.723.1 and G.729A [ITU-T03] which offer data rates comparable to the rate used in the simulations. The simulation results indicate that it would be possible to run an audio conferencing application using the appropriate voice compression codecs; however this has to be verified with tests on the test-bed. An audio conferencing application for the PCS is currently being developed and these tests will be conducted in the future work of the project.

5.8 Conclusion

This chapter has motivated the need for implementing an ad-hoc routing protocol in a test-bed in order to validate its design. The text in the chapter described the implementation of the proposed on-demand routing protocol on a mobile ad-hoc test-bed. The test-bed consisted of HP iPaq pocket PCs running an embedded Linux operating system. Having studied the routing architecture in Linux it was observed that it was a relatively simple task to implement a proactive, table driven algorithm in such an architecture. The operating system's routing structure, however, was not conducive to the implementation of an on-demand algorithm. This is due to the lack of support provided by Linux for the on-demand routing mechanism. Such algorithms discover routes only when they are needed and thus mechanisms for enabling the search for routes and packet buffering are needed. In order to provide such functionality and thus allow the implementation of the proposed on-demand algorithm, it was necessary to augment the current Linux routing architecture with a kernel module called Netfilter. This module allows the user-space queuing of packets, and thus the on-demand routing application running in the user-space of the operating system can monitor each packet that passes through the system. The routing application takes the responsibility of the routing functionality at a host, building and maintaining its own routing information.

This information is then used to instruct the kernel's routing mechanism on the availability of routes when they have been acquired.

The chapter described the operation of the Netfilter system and detailed the routing application that implemented the proposed on-demand algorithm on a Linux platform. The routing application was tested using the Ping program and a custom written Voice over IP application to demonstrate its multi-hop functionality.

Chapter 6

Conclusion

The recent advancement in mobile computing devices such as Notebook computers, laptops and PDA's has resulted in mobile computing becoming an everyday reality. The development of wireless technology has made connecting such devices to form networks for data sharing and communications feasible. The current wireless network structures offer only limited mobility and this dissertation has highlighted the reasons for ad-hoc networks being the next step towards truly ubiquitous computing and communications.

Developing solutions for mobile ad-hoc networks is a significant challenge because of their unique characteristics such as the networks having dynamic topologies, and nodes in the network having limited resources. Bandwidth and energy are included in the list of exhaustible resources available to a node. Despite the challenges there have been a wide variety of applications envisaged for ad-hoc networks. The military applications such as sensor networks and tactical networks are the primary focus of this dissertation. In particular, the Positional Communication System is being developed for situational awareness in the modern battlefield.

One of the main obstacles in ad-hoc network technology is the routing problem. Due to the challenges previously mentioned the design of ad-hoc routing protocols has received a great deal of attention recently. There have been many proposed solutions to the routing problem. The dissertation presented a classification of the algorithms showing the different philosophies used in the design of ad-hoc routing protocols.

The purpose of this research was to select or possibly develop a routing algorithm that would be suitable for its intended application, namely the PCS tactical network. Qualitative performance analysis is limited in indicating which ad-hoc routing philosophy and more specifically which routing algorithm is best suited for a general ad-hoc network application. What is required is a quantitative performance comparison of the proposed protocols in a common, realistic platform. In order to achieve this, simulation packages are available and are widely used in research. One of the most popular simulation packages is the NS-2

simulator which was recently extended to allow simulation of multi-hop ad-hoc networks. In this dissertation NS-2 was used to simulate four prominent ad-hoc routing schemes, namely DSDV, AODV, DSR and TORA. It was concluded from the simulation study that, with a given general set of performance criterion, the AODV protocol is most suited to the current application. It achieved the highest packet delivery success amongst the protocols simulated and it delivered the packets with the least latency. The protocol also used the least amount of network resources in terms of protocol packet overhead. However, from the study of the protocol (and other similar on-demand schemes), some aspects of its operation could be altered which would significantly improve its performance.

Nearly all of the proposed on-demand ad-hoc routing schemes use a blind flooding technique during the route discovery process. This method, although very robust, creates significant redundant routing overhead. The protocol packet overhead, generally in the form of broadcast packets, creates congestion which reduces the performance advantage on-demand algorithms have over other types of protocols. The routing algorithm proposed in this dissertation uses a query localization algorithm to make the flooding technique more efficient. Since each node in the PCS is enabled with a GPS module, the use of location information for query localization was a logical step. A router when faced with a forwarding decision with a query packet only forwards the packet if it closer to the destination than the node from which it received the packet. Thus the route request packet is taken closer to the destination with each hop that it takes.

In addition to the query localization, the proposed protocol implements a load checking algorithm which avoids congested, overloaded nodes in the network in its routing mechanism. The algorithm attempts to balance the data traffic amongst the nodes in the network preventing the creation of overburdened, and thus critical, nodes. The performance of the proposed protocol was compared to the algorithm on which it is based. The results from these simulations show that TAODV routes packets with greater success than the existing AODV protocol. The packet delivery ratio and delivery delay achieved by the protocol showed a marked improvement on AODV. The routing overhead incurred by TAODV was also significantly less and this is attributed to the query localization algorithm used by the protocol.

The dissertation then motivated the need to implement an ad-hoc routing algorithm in a physical test-bed before it can deployed in real-world systems. The current routing

architecture of the Linux platform on which the proposed algorithm was implemented is ideally suited for the implementation of proactive or table driven ad-hoc algorithms. The reason for this is that the routing architecture was designed with the protocols of the wired networks in mind and the table driven algorithms are modified versions of such protocols. In order to implement on-demand algorithms it was necessary to modify the current routing mechanisms of Linux. This was achieved through the packet mangling functionality of a Linux kernel module called Netfilter. The dissertation described how an on-demand routing application can be implemented using this extension to the Linux kernel's routing functionality. The dissertation presented preliminary tests on the PCS test-bed running the proposed on-demand algorithm. The aim of the tests was to demonstrate the multi-hopping functionality of the routing application. They were done using the Ping program and a custom developed Voice over IP (VoIP) application. The tests showed near perfect results which is to be expected from a network of only three nodes and limited mobility. The voice over IP test showed that the intended audio conferencing application of the PCS would be feasible provided the application uses intelligent voice compression algorithms to reduce bandwidth usage. Once this is achieved, the simulation results in the dissertation show that such an application can be operated over the ad-hoc network test-bed.

The future work in the development of the proposed algorithm includes improvements to both the query localization and the load checking algorithm. Currently the location information used for the query localization is disseminated in an on-demand manner. Further techniques, which could possibly make the dissemination process faster and more efficient, have to be investigated. The load checking algorithm used in TAODV uses the protocol queue length as a measure of load at a node. Additional load measures such as local signal to noise ratio and the load at neighbouring nodes will be used in the future versions of the protocol.

The implementation of the TAODV routing daemon has to be extended to support the query localization algorithm. As mentioned in Chapter 5 of the dissertation the GPS module hardware and software has been developed. Thus it's incorporation into the operation of the routing daemon is the next step in the development task. The preliminary tests were encouraging in demonstrating the multi-hopping functionality of the routing application. However, further rigorous tests are required which will be used to quantify the performance of the application on the intended test-bed. A voice conferencing application, currently being developed, will be used for this purpose.

References

Aodv-uu03 "AODV Implementation at Uppasala University", http://www.docs.uu.se/hedrik/aodv Last Accessed: August, 2003.

Berts87 D. Bertsekas and R Gallager, "Data Networks", Prentice Hall, 1987, ISBN: 0131968254.

Blue03 Bluetooth Special Interest Group (SIG) [online] http://www.bluetooth.com/ Last Accessed: August 2003.

Broch98 J. Broch, D. A. Maltz, D. B. Johnson, Y-C. Hu and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols", *Proceedings of ACM/IEEE MOBICOM '98*, October 1998, pp. 85–97.

Casta02 R. Castaneda, S. R. Das and M. K. Marina. "Query Localization Techniques for On-Demand Routing Protocols in Ad Hoc Networks", Wireless Networks 8, 137-151, 2002.

Chen98 T-W. Chen, M. Gerla, "Global State Routing: A New Routing Scheme for Ad hoc Wireless Networks," *IEEE International Conference on Communications(ICC 98)*, Jun. 1998, pp171-175.

Chiang97 C. Chiang, H. Wu, W. Liu and M. Gerla, "Routing in Clustered, Multihop, Mobile Wireless Networks," The IEEE Singapore International Conference on Networks (SICON), 1997, pp 197-211.

Choe02 M. Choe, V. Sheetal, A. Bourkerche, "A Route Discovery Optimization Scheme Using GPS system", Annual Simulation Symposium, 2002, pp 20-26.

Cisco03 Cisco, "Cisco Aironet 350 Series Client Adapters, Data Sheet", Cisco Systems, Inc.2003.

Corso99 S. Corson, J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", available from http://www.ietf.org/rfc/rfc2501.txt

Das02 S. R. Das, C. E. Perkins and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks", *Proceedings of IEEE INFOCOM'2000*, March 2000. pp. 3-12.

Das98 S.R Das, R. Castaneda, J. Yan and R Sengupta, "Comparative Performance Evaluation of Routing Protocols for Mobile, Ad Hoc Networks", Proceedings of the 7th International Conference on Computer Communications and Networks, 1998. pp 153-161.

Davie00 V Davies, "Evaluating Mobility Models Within an Ad Hoc Network", M.Sc. Thesis, Department of Mathematical and Computer Sciences, Colorado School of Mines, Colorado, 2000.

Dearh02 N. J. Dearham, T. A. Quazi and S. McDonald, "Comparative Assessment of Ad-Hoc Routing Protocols", *Proceedings of the South African Telecommunications Networks and Applications Conference (SATNAC)*, 2003. [Available in CD-ROM].

Fall03 K. FALL AND K. VARADHAN. *The ns Manual (formerly ns Notes and Documentation)*, April 2003.

GSM03 GSM 06.10 compression algorithm, "GSM 06.10 lossy speech compression"
[online] http://www.cs.tu-berlin.de/~jutta/toast.html
Last Accessed: September 2003.

Gupta02 B. Gupta, "Design, Implementation and Testing of Routing Protocols for Mobile Ad-hoc Networks", M. Sc. thesis, Graduate College of the University of Illinois at Urbana-Champaign, Urbana, Illinois, 2002.

Haas99 J. Haas, M. Pearlman, M, "Determining the Optimal Configuration for the Zone Routing Protocol," *IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks*, vol.17, no.8, Aug. 1999, pp.1395-1414.

Handh03 Handhelds.org, http://www.handhelds.org/
Last Accessed: August, 2003.

Henty01 B. E. Henty, "Throughput Measurements and Empirical Prediction Models for IEEE 802.11b Wireless LAN (WLAN) Installations", M.Sc. thesis, Faculty of the Virginia Polytechnic Institute and State University, Virginia, 2001.

HP03 HP iPAQ Pocket PC, "iPAQ Pocket PC family"
[online] http://www.hp.com
Last Accessed: September 2003.

IEEE99 IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", *IEEE 802.11-1999*, 1999.

Ietf03 The Internet Engineering Task Force,
http://www.ietf.org/home.html
Last Accessed: August 2003.

ITU-T03 ITU-T Recommendation Publications, "International Telecommunication Union",[available online],
http://www.itu.int/rec/recommendation.asp?type=products&parent=T-REC-g
Last Accessed: September 2003.

Iwata99 A. Iwata, C. Chiang, B. Pei, T. Chen, M. Gerla, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks*, vol.17, no.8, Aug. 1999, pp.1369-1379.

Janas01 R Janaswamy, "Radiowave Propagation and Smart Antennas for Wireless Communications", *Kluwer Academic Publishers*, 2001, pp. 22.

References

Jiang01 H. Jiang, J. J. Garcia-Luna-Aceves, "Performance Comparison of Three Routing Protocols for Ad Hoc Networks", *Proceedings of the 10th International Conference on Computer Communications and Networks*, 2001, pp. 547-554.

Johan99 P. Johansson, T. Larsson, N. Hedman and B. Mielczarek, "Routing protocols for mobile ad-hoc networks – a comparative performance analysis", *Proceedings of ACM/IEEE MOBICOM '99*, August 1999, pp. 195-206.

Ko98 Y. B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad-hoc Networks", *Proceedings of ACM/IEEE MOBICOM '98*, October 1998, pp. 66-75.

Ko99 Y. B. Ko and N. H. Vaidya, "Using Location Information in Wireless Ad Hoc Networks", Vehicular Technology Conference, 1999 IEEE 49th, Volume: 3. pp 1952-1956

Lee01 S.-J Lee, M. Gerla, "Dynamic load-aware routing in ad hoc networks".IEEE International Conference on Communications ICC 2001.Volume: 10 , 2001, pp. 3206-3210

Lee99 S.-J. Lee, M. Gerla, C-K Toh, "A Simulation Study of Table-Driven and On-Demand Routing Protocols for Mobile Ad Hoc Networks", In IEEE Network, Vol 13, no 4, Jul/Aug 1999, pp. 48-54.

Manet03 Mobile Ad-hoc Networks(MANET),
http://www.ietf.org/html.charters/manet-charter.html
Last Accessed: August 2003 (replaced manet with manet03)

Morris01 J. Morris, "Manpage of LIBIPQ", Linux Programmer's Manual, 2001

Netfi03 Netfilter: Firewalling, NAT and packet mangling for Linux 2.4,
http://www.netfiler.org
Last Accessed: August 2003.

NS-203 *The Network Simulator - ns-2*, http://www.isi.edu/nsnam/ns/ Last accessed:August 2003.

Ospf03 Cisco, "Open Shortest Path First", Online Document, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ospf.htm Last Accessed: August, 2003.

Park97 V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", *Proceedings of INFOCOM '97*, April 1997, pp. 1405-1413.

PCS03 The Positional Communication System, http://www.ee.und.ac.za/pcs/ Last accessed: August 2003.

Perkin02 C. Perkins, E. M. Royer AND S. R. Das. Ad hoc On-Demand Distance Vector (AODV) Routing, InternetDraft, draft-ietf-manet-aodv-11.txt, *work in progress*, June 2002.

Perkin94 C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Computer Communications Review*, October 1994, pp. 234-244.

Perkin98 C. E. Perkins, B Woolf and S. R. Alpert, "Mobile IP Design Principles and Practices", Prentice Hall PTR, First Edition, January 1998, ISBN: 0201634694

Perkin99 C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing", *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, February 1999, pp. 90-100.

Peter00 L. L. Peterson and B. S. Davie, "Computer Networks A Systems Approach", Morgan Kaufmann, San Francisco, CA, 2000, ISBN: 1558605770

Plumm82 D. C. Plummer. An Ethernet address resolution protocol: Or converting network protocol addresses to 48.bit Ethernet addresses for transmission on Ethernet hardware. RFC 826, November 1982.

Quazi03a T. A. Quazi and S. McDonald. "A Ad-hoc On-demand Routing Algorithm for a Positional Communication System". *Proceedings of the South African Telecommunications Networks and Applications Conference (SATNAC),* 2003. [Available in CD-ROM].

Quazi03b T. A. Quazi and S. McDonald. "A Load Aware, Location Aided, Ad-hoc On-demand Routing Algorithm". *Proceedings of the Fifth IFIP TC6 International Conference on Mobile and Wireless Communications Networks,* MWCN 2003, pp. 150-154.

Quazi03c T. A. Quazi and S. McDonald. "Design and Implementation of an on-demand routing algorithm for a Positional Communication System". *Proceedings of the Military Information and Communications Symposium of South Africa, 2003.<accepted>*

rfc2501 "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", The Internet Society, Request For Comments 2501, 1999.

Rice03 Rice Computer Science: "NASA funds Johnson's $1.2 million wireless network project", http://www.cs.rice.edu/News/items/dbj-nasa.shtml
Last Accessed: August 2003.

Rip03 Cisco, "Routing Information Protocol", Online Document, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/rip.htm
Last Accessed: August, 2003

Roger01 S. Rogers, A Boukerche, "Performance of GZRP Ad Hoc Routing Protocol", Journal of Interconnection Networks 2(1), 2001, pp 31-48

Royer00 E. M. Royer and C. E. Perkins. "An implementation study of the aodv routing protocol". *Proceedings of the IEEE Wireless Communications and Networking Conference*, 2000 [PDF document with no page numbers from proceedings].

References

Royer99a E. M. Royer and C. K. Toh, "A Review of Current Routing Protocols for Ad-hoc Wireless Mobile Networks", IEEE Personal Communications, April 1999, pp. 46-55.

Royer99b E. M. Royer and C. E. Perkins , "Ad-hoc On-Demand Distance Vector Routing", *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, February 1999, pp. 90-100.

Russel02 P. Russel and H. Welte, "Linux netfilter Hacking HOWTO", Linux HOWTO documentation, 2002.

Satch00 S. T. Satchel, H. B. J Clifford, "Linux IP Stacks Commentary", Coriolis Open Press, 2000, ISBN: 1-57610-470-2

Skold01 M. Skold, "Using location information to reduce the route search in tactical radio networks", Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force.IEEE, Volume: 1, 2001. 367 -371 vol.1

Toh02 C.-K. Toh, "Ad Hoc Mobile Wireless Networks: Protocols and Systems", Prentice Hall PTR, 2002, ISBN: 0-13-007817-4

Toh96 C. K Toh, "A novel distributed routing protocol to support ad-hoc mobile computing," Proceedings of IEEE 15th Annual International Conference on Computers and Communications, 1996, pp 480-486

Tripa01 S Tripathi, "Augmentation and Optimizations of AODV, Thesis for the degree of Master of Technology, Indian Institute of Technology, Kanpur, 2001.

Tripa02 S Tripathi, M. Ghosh, *R.K.* 1. "An implementational study of certain heuristics for the performance enhancements of AODV", Parallel, Distributed and Network-based Processing, 2002. Proceedings. 10th Euromicro Workshop on 2002 395 -402

References

Tseng02 Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, J.-P. Sheu The Broadcast Storm Problem in a Mobile Ad Hoc Network, Wireless Networks 8, 153-167, 2002.

Vint03 *The VINT (Virtual InterNetwork Testbed) Project homepage*, `http://www.isi.edu/nsnam/vint/index.html` Last accessed: August 2003.

Wrigh95 G. R. Wright and W. R. Stevens. "TCP/IP Illustrated, Volume 2: The Implementation", Addison-Wesley, Reading, Massachusetts, 1995.

Wu01 K. Wu, J. Harms, "Load-sensitive routing for mobile ad hoc networks". Computer Communications and Networks, 2001. Proceedings, pp. 568-575.