UNIVERSITY OF STRATHCLYDE
DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING

LOAD DISTRIBUTION AND ENERGY AWARENESS IN MANETS USING MULTIPATH ROUTING

CHRISTOS TACHTATZIS

A thesis presented in fulfilment of the requirements for the degree of Doctor of Philosophy

2008

Declaration

I, Christos Tachtatzis, hereby declare that this work has not been submitted for any other degree at this University or any other institution and that, except where reference is made to the work of other authors, the material presented is original.

Abstract

Wireless ad-hoc networks are characterised by lack of infrastructure and frequent topological changes. Traditional routing protocols seek only single paths to the desired destinations, while multipath routing obtains multiple paths for only marginal additional overhead. This work argues that multipath routing is advantageous, even allowing for the additional overheads, because of the improved network load distribution. The merits of multipath routing are shown through extensive performance evaluation considering packet delivery ratio, average end-to-end delay and routing efficiency, for both mobile and static scenarios.

The second aspect of the thesis addresses energy awareness. When a single connection is considered, multipath routing can potentially consume more total energy compared to its unipath counterparts because some traffic can traverse longer (in terms of hop count) paths. On the other hand, unipath routing concentrates nodal traffic over the same single path; resulting in unfairness for the intermediate hops and uneven energy consumption which, in turn, can result in network partitioning. Here, it is argued that multipath routing extends the network lifetime, because the routing protocol can make more sophisticated decisions to avoid node exhaustion. Two novel energy aware routing schemes, which select optimal paths to homogenise the energy map of the network, are presented. A range of performance evaluation techniques are employed to demonstrate the merits of the proposed schemes, and it is shown that the approach of homogenising the network energy map mitigates against the effects of inevitable node outages cause by energy exhaustion and prolongs network lifetime.

Acknowledgements

First and foremost, I would like to thank my supervisor Dr. David Harle. I am very grateful to David for providing me the intellectual stimulation, his consistently well focused, yet nonetheless visionary, critique of my work, and for his constantly "open door" attitude and willingness to talk through new ideas. I would also like to thank my external and internal examiners, Professor Jonathan Pitts and Professor John Dunlop for their valuable suggestions and advice which helped further enhance my thesis.

After my time at the Communication Division, I have learnt that good research cannot be achieved without the interaction between researchers. I am very grateful to all my friends and colleagues at the University of Strathclyde for their friendship and collaboration. I would like to especially thank Andrew S. T. Lee and Kurian Oommen who were unlimited source of inspiration; Robert Atkinson for his assistance and encouragement; Gordon Morison, Colin Arthur, Michael Gilroy and Konstantinos Sasloglou for their moral support.

Last but not least, I would like to thank my parents Gavriil and Foteini and my brother Konstantinos whose encouragement, support and unconditional love, upon which I fully relied on throughout the period of my studies. I am also indebted to Myrto Laskaridou who gave me her loving support, patience and understanding throughout this period.

Acronyms

ACK	Acknowledgement
AODV	Ad-hoc On-Demand Distance Vector
ARQ	Address Resolution Protocol
B-MAC	Berkley-MAC
BER	Bit-Error-Rate
BG	Beacon Group
BPSK	Binary Phase Shift Keying
BP	Beacon Period
CAP	Contention Access Period
CBR	Constant Bit Rate
CCK	Complementary Code Keying
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
CFP	Contention Free Period
CMMBR	Conditional Max-Min Battery Routing
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CSMA	Carrier Sense Multiple Access
CTS	Clear-To-Send

DCF	Distributed Coordination Function
DCM	Dual Carrier Modulation
DIFS	Distributed (coordination function) Interframe Spacing
DP	Data Period
DRP	Distributed Reservation Protocol
DS-UWB	Direct Sequence Ultra Wideband
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
DVS	Dynamic Voltage Scaling
DYMO	Dynamic MANET On-demand
EBG	Extended Beacon Group
ECMA	European Computer Manufacturers Association
EIFS	Extended Interframe Spacing
FCC	Federal Communications Commission
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FER	Frame Error Rate
FFD	Full Function Device
FHSS	Frequency Hopping Spread Spectrum
FSK	Frequency Shift Keying
GOD	Global Operations Director
GTS	Guaranteed Time Slots
HR-DSSS	High Rate Direct Sequence Spread Spectrum
ID	Identifier

IETF	Internet Engineering Task Force
IE	Information Element
IFS	Interframe Spacing
IR	Infrared
ISM	Industrial, Scientific and Medical frequency band
M-RWP	Modified Random Waypoint mobility model
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MB-OFDM	Multi Band Orthogonal Frequency Division Modulation
MBCR	Minimum Battery Cost Routing
MDR	Minimum Drain Rate
MDSR-NDR	Multipath Dynamic Source Routing with Node Disjoint Routes
MIMO	Multiple-Input-Multiple-Output
MMBCR	Min-Max Battery Cost Routing
MPR	Multipoint Relay
MSR	Multipath Source Routing
MTPR	Minimum Total Transmission Power Routing
NAV	Network Allocation Vector
OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimised Link State Routing
OQPSK	Offset Quadrature Phase Shift Keying
OSI	Open Systems Interconnection
PAN	Personal Area Network
PAR	Project Authorisation Request

PCF	Point Coordination Function
PDA	Personal Digital Assistant
PDF	Probability Density Function
PHY	Physical Layer
PIFS	Point (coordination function) Interframe Spacing
PPM	Pulse Position Modulation
PSU	Pipeline Stage Unification
PTD	Point-To-Diagonal
PTZ	Point-To-Zero
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
QoS	Quality of Service
RERR	Route Error
RFC	Request for Comments
RFD	Reduced Function Device
RNG	Random Number Generator
RREPLY	Route Reply
RREP	Route Reply
RREQ	Route Request
RTO	Retransmission Timeout
RTS	Request-To-Send
RTT	Round-Trip Time
RWP	Random Waypoint mobility model
S-MAC	Sensor-MAC

SDMA	Space Division Multiple Access
SIFS	Short Interframe Spacing
SIG	Special Interest Group
SNIR	Signal-to-Noise and Interference Ratio
SNRT	Signal-to-Noise-Ratio Threshold
SNR	Signal-to-Noise-Ratio
SPR	Strongest Path Routing
TCP	Transport Control Protocol
TC	Topological Control
TDMA	Time Division Multiple Access
TFC	Time-Frequency Code
TGnSync	Name of an IEEE 802.11n proposal
TORA	Temporally-Ordered Routing Algorithm
UDP	User Datagram Protocol
UWB	Ultra Wideband
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
WWiSE	Word-Wide Spectrum Efficiency
Z-MAC	A Hybrid MAC for Wireless Sensor Networks

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The introduction of both the mobile phones and wireless computer networks in the late 20^{th} century has significantly changed and enhanced our social and work behaviour. Users are no longer tethered to a physical location in order to make a voice or video call, perform some computing function or do both at the same time. Mobile telephone users can travel for miles and still remain connected as long as they remain within their area of coverage. The emerging era of mobile communications and mobile computing will usher in a vision of pervasive computing in which our communications and computing needs are no longer physically tethered to a particular geographic location or bounded to any specific device or platform.

Fixed line telecommunications networks encompass significant construction and maintenance costs when compared to their unwired counterparts. Installing fixed communications links require substantial infrastructure (in terms of cabling, conduits, and ducts) and complex right-of-way agreements. On the other hand, fixed wireless networks require significantly smaller footprints in order to achieve a communications network of comparable connectivity. However, the continual evolution of platforms with computational and communications capabilities surpassing present-day mobile communications devices are enabling the development of wireless networks where no pre-planned infrastructure exists at all.

For such scenarios, users form a network to communicate either among themselves or with another network through a gateway. These ad-hoc networks are challenging and characterised by their lack of infrastructure and centralised administration. The definition of an ad-hoc network can be summarised as [1]:

> *"An ad-hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure*

or centralised administration. In such an environment, it may be necessary for one mobile node to enlist the aid of other hosts in forwarding a packet to its destination, due to the limited range of each mobile host's wireless transmissions."

In ad-hoc networks, mobile hosts might appear and disappear without any notice and the network must be able to cope with such events. Examples of such networks could be a conference centre, where a limited number of base stations are installed and all the attendees are equipped with laptops. The users that are close to the base stations are able to access the Internet, but users distant from the base stations may not be capable to access network resources. In such a case, users can form an ad-hoc network to communicate among themselves or use Internet resources. Users that are located outside the coverage area can communicate with other users in an ad-hoc manner and use the Internet resources via the users who are located within the coverage area. Of course, the formation of such a network assumes that the users are willing to forward traffic on behalf of other users.

Wireless Sensor Networks (WSNs) represent a significant application for ad-hoc networks. Recent advances in processing, storage, energy and communication technologies allow compact implementation of sensors, capable of detecting and monitoring temperature, chemical contamination, seismic, auditory, light activity and other desirable information of the environment. In addition to the monitoring of an area, it is also necessary to facilitate the collection method of the data to a point where it will be possible to perform further processing of the data either by humans or automated software. Monitoring might be essential for unexpected circumstances within hazardous fields, which lack inter-networking infrastructures and the necessity of the network is temporary. Beside such catastrophic scenarios, sensor networks could be used in more peaceful applications for the monitoring of wind farms for fault provisioning, vehicle traffic to improve traffic conditions, natural habitats to understand wild life and many more.

1.1 Multipath Routing

Routing is the process of selecting paths in computer networks, along which the traffic is to be sent. In Mobile Ad-hoc Networks (MANETs) routing is particularly challenging because hosts move or become inactive and this results in frequent topological changes. Wireless ad-hoc networks are capacity constrained and the high overheads of traditional wired routing protocols make their use

prohibitive. Additionally, devices are constrained in memory and processing capabilities and routing table maintenance for every node in the network is usually not practicable. Reactive wireless routing protocols operate in an on-demand manner and discover paths when necessary. This approach tends to reduce network overhead for maintaining up-to-date routing information and scales better than traditional pro-active routing protocols.

On-demand routing consist of two phases, a route request propagation where the source node floods the network with requests to reach the destination of the discovery, and a route reply phase where the destination replies to the the source node indicating the route(s) available for communication. Unipath routing does not exploit the fact that the route request propagation phase has already been performed and does not discover multiple paths to the destination. This results in a higher frequency of route discoveries, which in turn, increases network congestion and energy depletion.

In terms of network congestion, multipath routing has the potential to increase network performance by distributing the network load to multiple paths. Multipath routing protocols in the literature modify the route request propagation process in such a way that network overhead is increased and subsequently the benefits of multipath routing are counteracted by the increased overhead. Other routing protocols in the literature attempt to maintain low route discovery overhead, but the multiple paths discovered are not exploited simultaneously, but instead, the alternative paths are used for reliability purposes. In this thesis, a multipath routing scheme is proposed, which discovers multiple node-disjoint paths to the destination. It is shown that the scheme offers higher packet delivery ratio, lower average end-to-end delay and increased routing efficiency for static and mobile scenarios.

The second aspect studied in the thesis is energy awareness; traditional energy aware routing schemes attempt to prolong the network lifetime by selecting paths based on the highest residual energy. The approach followed here is based on the homogenisation of energy consumption in the network. Two novel routing schemes are proposed that extend the network lifetime and maintain network connectivity for longer. Finally, it is shown that there is a trade-off between having knowledge with high precision to perform routing operations and the overhead carried by the network to collect and maintain such high precision knowledge.

1.2 Overview of the Thesis

The remaining of this thesis is organised as follows: Chapter 2 introduces the common technologies and background information related to mobile ad-hoc networks. Chapter 3 discusses performance evaluation techniques used in this thesis and the assumptions made throughout this work. The chapter also proposes a new multipath, node disjoint routing scheme and a simple scenario is used to describe the methodology used. In Chapter 4, a comprehensive performance evaluation of the proposed multipath routing scheme is performed for static and mobile scenarios of variable mobile intensities and the merits of multipath routing are illustrated over a unipath counterpart.

In Chapter 5, multipath routing is studied from the energy perspective and the chapter proposes two new routing schemes, which prolong the network lifetime based on the homogenisation of the energy consumption in the network as opposed to the more traditional approach where the paths are chosen based on the highest energy. The proposed routing schemes assume that the residual energy of all other nodes in the network is available with no latency and this assumption is removed in Chapter 6. Additionally, Chapter 6 proposes a signalling mechanism which is used by network nodes to collect and maintain energy related information. The precision of the information is varied and the energy-aware protocols proposed in the previous chapter are examined to evaluate their performance under partial knowledge with reduced precision and added latency. Finally, Chapter 7 concludes the thesis, presents its contributions and gives directions for future research.

Chapter 2

Background

This chapter introduces the concepts and technologies underpinning Mobile Ad-hoc Networks (MANETs). Technologies such as ZigBee, Bluetooth, Ultra Wideband (UWB) are not always seen as directly related to multipath routing, but they are crucial to understand the space and limitation of MANETs. The chapter also describes common routing protocols used in wired networks and explains why such protocols can not be used in the wireless domain. Finally, the chapter presents a classification of routing protocols in the wireless ad-hoc network environment and describes a series of Request for Comments (RFC) routing protocols in the Internet Engineering Task Force (IETF) MANET working group.

2.1 Ad-hoc Network Operating Principles

An ad-hoc network can be constructed without the need of pre-planed infrastructure. For example, consider a collection of nodes that are equipped with wireless network interfaces. The minimum number of peers required for the formation of an ad-hoc network (or any network in general) is two. Then, each node beacons their neighbours and the communication between them can start. The nodes communicate between themselves in a point-to-point fashion (single-hop) or when the distance between peers is greater than the range of their wireless interface then the aid for intermediate nodes is required to accomplish the communication (multi-hop).

2.1.1 Single-hop communication

When two adjacent peers seek to communicate with each other, they tune their wireless interface to the same frequency. In addition, the nodes must be within

each other's range in order to achieve successful communication. Factors that affect the successful single hop communication of the peers are described:

- **Physical characteristics of the transceivers:** Each of the nodes[1] is equipped with a transmitter capable to transmit with a maximum power P_t. This power is one of the key parameters defining the maximum transmission range. Additionally, each node is equipped with a receiver with a particular receiver sensitivity. The receiver sensitivity describes the ability of the receiver to sense activity in the channel. If the power of the received signal is below the receiver sensitivity, then the receiver is not able to detect any transmissions and assumes the channel to be idle. On the other hand, when the received signal strength is above the receiver's sensitivity, the receiver senses activity in the channel and attempts to decode the received signal. When a message is propagated in the channel (air) it is attenuated and distorted according to the characteristics of channel. In brief, the amount of attenuation and distortion of the signal is affected by the environmental conditions such as rain, fog, gas, pollution etc. Other phenomena that might significantly affect the signal propagation are fading effects (small scale fading in short distance scenarios, multipath fading in closed environments, etc.) and Doppler effects when transceivers are mobile.

  The metric that describes the quality of the received signal in communication systems is the Signal-to-Noise-Ratio (SNR)[2], and is significantly affected by the channel characteristics (environment), thermal noise of the transceiver components and the modulation scheme in use. For a system, where the SNR is known, it is possible to calculate the probability of symbol error and hence Bit-Error-Rate (BER) [2, 3]. When the SNR increases, the BER decreases and hence the probability of receiving a bit in error decreases. Consequently, the probability of receiving a complete message in error also decreases. However, even if the SNR is very high, there is still a finite chance that a complete message can not be received without error.

  Finally, in communication systems, there are techniques to recover from errors, such as Forward Error Correction (FEC), which have the penalty

[1]The terms node, peer, host are used interchangeably.

[2]There are variants of this ratio such as Signal-to-Interference-Noise-Ratio, and such terms are used according to the context. For example, if interference exists in the environment, the denominator of the ratio is the addition of the noise level and the interference level.

of transmitting longer messages. FEC introduces some redundancy in the message, in order to allow recovery. It is sometimes necessary to introduce this redundancy since it is more expensive to re-transmit the complete message than to add a few redundant bits. However, such techniques are not in the remit in this thesis and, hence, are not discussed further.

- **Concurrent transmissions:** Consider a node who transmits a frame to another adjacent node. The transmission is successful only if, for the duration of the transmission, no other nodes initiate transmissions in the same geographic region. If other nodes (within the geographic region) initiate transmissions while another transmission is taking place, they provoke interference to the first transmission and both communications fail, i.e. transmissions collide. For this reason, the communication between peers must be coordinated. Such coordination is managed by the MAC protocols which are described in Section 2.3.6.

2.1.2 Multi-hop communication

When two nodes seek to communicate with each other and the one is not within range of the other, then intermediate nodes are employed to establish communication between them; if such a set of intermediate nodes exists and are willing to aid the communication. The multi-hop communication is constrained by all the operating characteristics of each single-hop communication (as described in Section 2.1.1), plus the functionality being available to allow the node to discover paths for the communication and forward traffic on behalf of initiator of the communication.

Each node must be aware of the network topology either in full or partially. There are different techniques to maintain such awareness and are discussed further in Section 2.4. Assuming that the node is fully or partially aware of the network topology, then it must decide which path to use to route its traffic to the destination of the communication. This functionality is handled by the "so called" routing agent which is able to decide the optimum path for routing the traffic to the destination. The path may be optimum in various respects such as end-to-end delay, energy consumption, congestion and etc. Finally, all intermediate nodes must be capable of forwarding the traffic to the next hop along the path.

2.2 Propagation Models

The propagation model is an important component, when simulating ad-hoc networks. The most common propagation models are described below.

2.2.1 Free (Friis) space

The free space propagation model [4] is the simplest model and assumes that there is only one propagation path between transmitter and receiver, known as the line-of-sight path. The received signal power is inversely proportional to the square distance between transmitter and receiver. If the transmission power is P_t and the distance between transmitter and receiver is d, then the received signal power P_r is given by the equation:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{2.1}$$

G_t and G_r are the antenna gains of the transmitter and the receiver respectively and L the system loss ($L \geq 1$). Finally λ is the wavelength of the transmission ($c = \lambda \cdot f$ where c is the speed of light and f the frequency of the transmission). For simulation purposes, the antenna gains G_t, G_r and the system loss L are selected equal to 1 ($G_t = G_r = 1$ and $L = 1$) [5].

2.2.2 Two-Ray Ground

The two-ray ground propagation model gives more accurate prediction compared to the Free Space model, for long distances only [5]. This model considers the line-of-sight propagation path plus a ground reflection path. The formula for the received signal strength is:

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \tag{2.2}$$

where h_t and h_r the heights of the antennas of the transmitter and the receiver respectively. L is also selected to equal to 1 ($L = 1$).

Equation 2.2 results in values closer to reality when long distances are considered, but for smaller distances the Free space model is more accurate. It is necessary to define when each of the two models is selected and this depends upon the distance. Therefore, for a cross-over distance $d_c < d$ the two-ray ground Equation 2.2 is more appropriate. On the other hand, the Free Space

Equation 2.1 is more appropriate when the cross-over distance d_c is greater than d $(d_c > d)$. The cross-over distance is calculated using the equation:

$$d_c = \frac{4\pi h_t h_r}{\lambda} \tag{2.3}$$

2.2.3 Shadowing Model

Both of the above mentioned models consider the communication range as a perfect circle and both use deterministic equations to calculate the received signal strength. In reality, the received power is affected by fading multi-path effects and therefore the communication range may vary. The Shadowing model uses a random variable to determine the reception power of the signal and hence the communication range does not form an idle circle. However, the random variable is changed according to the environment which is simulated. For example, the parameters of the random variable are different when an outdoor experiment is performed compared to an indoor experiment.

The shadowing model consists of two parts the path loss model and the variation of the received power. The first part calculates the mean received power $(\overline{P_r(d)})$ as function of the distance d. The close-in distance d_0 is used as a reference and $\overline{P_r(d)}$ is computed relative to the $P_r(d_0)$ using the equation.

$$\frac{P_r(d_0)}{\overline{P_r(d)}} = \left(\frac{d}{d_0}\right)^{\beta} \tag{2.4}$$

β is called the path loss exponent and is empirically determined for the environment of the experiment. For free space propagation $\beta = 2$. For outdoor environments β lies within the interval $(3 - 5)$. Larger values for β represent environments with more obstacles and corresponds to a more rapid decrease in the average received power as distance increases. The value of $P_r(d_0)$ is computed using the Free Space model equation 2.1. Since the path loss is usually measured in dB, the equation 2.4 can be written as:

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)}\right]_{db} = -10\beta log\left(\frac{d}{d_0}\right) \tag{2.5}$$

The second part of the shadowing model represents the variation of the received power at a certain distance. This random variable (X) is log-normal and, when measured in dBs, the variable (X_{dB}) follows a Gaussian distribution with mean zero $(\mu = 0)$ and standard deviation σ_{dB}. The σ_{dB} is also called the

shadowing deviation and is obtained experimentally in a similar manner to β. Therefore, the final equation that describes the shadowing model is:

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)}\right]_{db} = -10\beta log\left(\frac{d}{d_0}\right) + X_{dB} \tag{2.6}$$

Typical values for β and σ_{dB} are shown in Table 2.1.

Environment		β
Outdoor	Free Space	2
	Shadowed urban area	2.7 to 5
In building	Line-of sight	1.6 to 1.8
	Obstructed	4 to 6

Environment	σ_{dB} (dB)
Outdoor	4 to 12
Office, hard partition	7
Office, soft partition	9.6
Factory, line-of-sight	3 to 6
Factory, obstructed	6.8

Table 2.1: Typical values for path loss exponent β and shadowing deviation σ_{dB} [5].

2.3 Ad-hoc Network Architecture; Node Perspective

The network architecture could be described either in a distributive fashion, where each layer of the protocol stack is described by a stand alone standard or by using an integrated approach. For example, the IEEE 802.11 standard describes the Physical and the Data Link Layers of the OSI model. Whereas technologies such as Bluetooth [6] and ZigBee [7] specify the whole protocol stack. Each of these two approaches has its own advantages and disadvantages but any judgement of each approach is out of the scope of this text.

The ad-hoc networks are characterised by their lack of infrastructure. The formation of the network is performed over the same shared medium, the air. The generic OSI layer model of source to destination communication is shown at Figure 2.1.

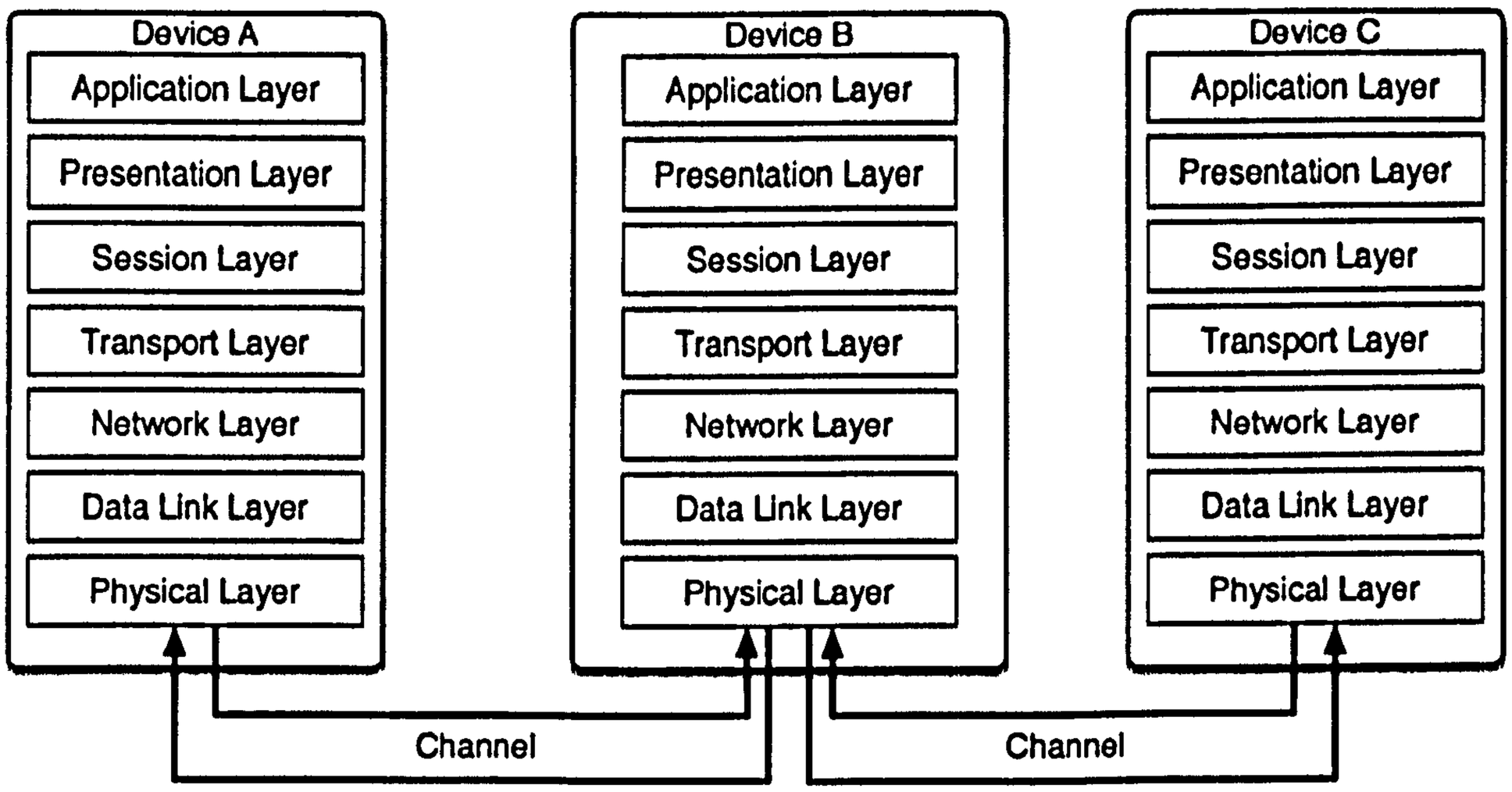


Figure 2.1: OSI layer model.

2.3.1 Physical Layer

The physical layer defines all the electrical and physical specifications of the network devices. The specifications define the medium (air in ad-hoc networks), transmission and reception characteristics of the transceivers, modulation techniques used, bit rates and operational frequencies. There is a variety of standards defined by the IEEE community which are discussed later in this text, such as IEEE 802.11 (Wireless Local Area Network (WLAN)), IEEE 802.15.1 (Bluetooth), IEEE 802.15.4 (ZigBee).

2.3.2 Wireless LAN - IEEE 802.11 (PHY)

In 1997, the IEEE adopted the IEEE 802.11 standard - the first standard to describe Wireless Local Area Networks (WLANs). The standard defines the Physical Layer (PHY) and Medium Access Control (MAC) (Data Link) layer of the protocol stack, regarding connectivity of Wireless Local Area Networks (WLANs) and is very similar to the IEEE 802.3 Ethernet [8]. On the other hand, IEEE 802.11 takes into account the uniqueness of the physical channel that is

used in the WLAN operation, such as range limitation, unreliable medium, dynamic continuously changing topologies and the inability of the devices to "hear" every other device in the network. The IEEE 802.11 committee, taking these issues to consideration, defined the standard [9].

The IEEE 802.11 standard defines five transmission techniques. The first is Infrared (IR) method which operates at 300-428,000 GHz using Pulse Position Modulation (PPM) and supports two speeds of 1 Mbps and 2 Mbps. The IR method is considered to have a good security level since it requires line-of-sight positioning of the peers. However, IR communication is affected by sunlight and offers comparatively low capacity.

The second method defined by the standard is Frequency Hopping Spread Spectrum (FHSS). This method operates at the 2.4 GHz Industrial, Scientific and Medical frequency band (ISM) where licensing is not required. The FHSS operates at 79 channels 1 MHz wide each. A pseudo random number generator with the same seed, is used in all station to coordinate the sequence of frequency hopping and all the stations are synchronised in time. In addition, the time that the network remains tuned to each frequency is known by all stations and is constant. This time is referred to as dwell time. The standard defines that dwell time should be adjustable, but below 400 msec [10].

Since all stations are using the same seed for the pseudo random number generator and all peers know the dwell time, they can switch frequencies at the same time and maintain connectivity.

FHSS operates in 1 Mbps or 2 Mbps rates and provides a fair allocation of the unlicensed band. It also provides a moderate level of security since the radio frequency of communication is changed and the intruder needs to know the seed of the pseudo random number generator plus the dwell time to eavesdrop a communication. The disadvantage that of FHSS is the low capacity offered.

The third method specified is Direct Sequence Spread Spectrum (DSSS); it also operates at 2.4 GHz ISM band and supports rates of 1 and 2 Mbps. In this technique each bit is encoded using 11 chips called the Barker Sequence. The modulation scheme used is Phase Shift Modulation. DSSS's drawback is also low bit rate support.

The fourth method is Orthogonal Frequency Division Multiplexing (OFDM). Its operation is also known as IEEE 802.11a. It operates in 5 GHz ISM band and uses 52 frequencies with supported data rates up to 54 Mbps; 48 of these frequencies are used for data transmission and 4 for synchronisation purposes. OFDM splits the signal to many narrow bands and uses complex encoding schemes

based on Phase Shift Modulation for speeds up to 18 Mbps, whereas the encoding schemes used for speeds in the range of 18 Mbps to 54 Mbps, are based upon Quadrature Amplitude Modulation (QAM). Finally, OFDM has good spectrum efficiency in terms of bits/Hz and good immunity to multi-path fading. Another advantage is that it is not affected by interference from microwave ovens, cordless phones or Bluetooth (which operate at the 2.4 GHz band).

Finally, High Rate Direct Sequence Spread Spectrum (HR-DSSS) is known as IEEE 802.11b. It operates at 2.4 GHz ISM band and uses 11 million chips/sec. The modulation scheme used is Complementary Code Keying (CCK) and supports data rates of 1, 2, 5.5 and 11 Mbps. The range of IEEE 802.11b is 7 times higher than the one offered by the OFDM (IEEE 802.11a) method.

In March 2000, the IEEE 802.11 committee created a new task group to develop a standard that will operate at the 2.4 GHz band and be able to offer speeds up to 54 Mbps. Finally, in June 2003, the IEEE 802.11g standard was approved and combines characteristics from IEEE 802.11b and IEEE 802.11a standards. It uses OFDM with CCK. The data rates supported are 1, 2, 5.5, 11 Mbps with CCK modulation (in order to maintain compatibility with the IEEE 802.11b) and data rates of 6, 9, 12, 18, 24, 36, 48, 54 Mbps using OFDM modulation [9].

Table 2.2 summarises the characteristics of IEEE 802.11a,b,g standards, in terms of bit rates, operating frequency bands and modulation techniques used.

Standard	Bit Rate (Mbps)	Frequency (GHz)	Modulation Scheme
802.11a	6-54	5	OFDM
802.11b	1, 2, 5.5, 11	2.4	DSSS with CCK
802.11g	54	2.4	OFDM with CCK

Table 2.2: Popular IEEE 802.11 standards.

In January 2004, IEEE announced the formation of the new 802.11 Task Group n. The purpose of the group is to develop a standard for WLAN, where its real data throughput (throughput without overhead) will exceed 100 Mbps and that will offer better operating distances than current networks. 802.11n extends

existing 802.11 standards and provides support for Multiple-Input-Multiple-Output (MIMO) techniques to increase the range and data rates by exploiting the spatial diversity. In 2004, six proposals were made for the eventual 802.11n [11]. However, two dominant, similar proposals have emerged, namely TGnSync and Word-Wide Spectrum Efficiency (WWiSE). Both proposals are similar. The TGnSync gives more emphasis on increasing peak data rates while WWiSE emphasises spectral efficiency improvements. Both proposals support operation in the legacy 20 MHz wide channel and optional operation in double-width 40 MHz channels for additional throughput. Both 802.11n use a variety of modulation schemes according to the bit rate, configuration of transceivers. The modulation schemes are Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), two Quadrature Amplitude Modulations (QAMs) (16-QAM, 64-QAM) and OFDM with a different number of bits encoded by each subcarrier. All major manufactures have released 'pre-N', 'draft-N' or 'MIMO-based' products.

2.3.3 Bluetooth - IEEE 802.15.1 (PHY)

Bluetooth technology was developed as a replacement for cables that connect one device to another. The technology initiated originally by Ericson Mobile Communications in 1994 with the aim to provide a low power and low cost radio interface solution for the communication between mobile phones and their accessories. Later, in 1998, Ericson Mobile Communications and another four companies Intel, IBM, Toshiba and Nokia Mobile Phones formed a group called the Special Interest Group (SIG) with the intention to develop a de-facto standard for the interface of communication between different devices of different manufacturers of portable computers, mobile phones and other devices. In 1999, the IEEE 802.15 working group (Wireless Personal Area Network (WPAN) group), based on the Bluetooth standard provided by the SIG, started to develop a standard; IEEE 802.15.1 [12]. However, the Bluetooth standard defined by the SIG and the IEEE are different. The Bluetooth SIG defines the whole protocol stack from the physical layer up to the application layer, whereas the IEEE 802.15.1 refers only to the physical and data link layers. The rest of the protocol stack is not relevant for the IEEE group.

For the physical layer, Bluetooth defines a typical range of about 10 meters, with a possibility to extend to 100 m. The typical transmission power is between 1 to 100 mW and Bluetooth operates over the ISM band of 2.4 GHz [12]. The

frequency band is divided into 79 channels with each channel occupying bandwidth of 1 MHz. The modulation scheme used is Frequency Shift Keying (FSK), with data rate of 1 Mbps; a lot of the spectrum is consumed by overhead. Bluetooth also uses FHSS with 1600 hops/sec and dwell time of 625 *μsec*. All the nodes switch at the same time and the master node directs the hop sequence (see Section 2.3.7.1)

As mentioned in Section 2.3.2, IEEE 802.11 operates at the same frequency (2.4 GHz) with Bluetooth. Therefore, all 79 channels can be affected by WLAN operation. However, Bluetooth has a dwell time of 625 *μsec* while IEEE 802.11 has 400 msec and therefore it is more likely that Bluetooth transmissions will dictate the spectrum use, since the frequency changes are faster and therefore enforce occupancy of the spectrum [10].

Currently, there is no obvious solution for the interference created and both working groups of IEEE 802.11 and 802.15 are trying to resolve this issue. One solution would be to put the IEEE 802.11a network infrastructure in place since it operates at 5 GHz band but this comes with the drawback of decreased operation range. The other solution will be to allow either IEEE 802.11 or Bluetooth to operate at the same region. Both of these solutions have their drawbacks.

2.3.4 ZigBee - IEEE 802.15.4 (PHY)

IEEE 802.11 aims to deliver a standard capable of longer operation range and higher data rates without significant concerns about energy efficiency. On the other hand, Bluetooth aims to deliver services at the Wireless Personal Area Network (WPAN) level, and its main concern is to provide Quality of Service (QoS) guarantees, rather than energy efficiency. Bluetooth is capable of providing medium data rate to services ranging from cellphones to Personal Digital Assistant (PDA) communications and have the ability to operate with QoS guaranties for voice applications.

In December 2004, IEEE 802.15.4 committee and ZigBee alliance approved a new standard (known as ZigBee 1.0) that focuses on low-cost, low-power, short-range, very small device sizes; data rate requirements do not dominate their attention. The IEEE 802.15.4 concentrates at the two lower layers of the protocol stack, physical and data link layers, whereas the ZigBee alliance concentrates on layers above; network to application layers.

ZigBee 1.0 operates at the 2.4 GHz band which does not require licensing worldwide. An additional band is allocated for America at 915 MHz, whereas in

Europe this additional band is at 868 MHz. The data rates supported are 250 kbps at 2.4 GHz, 10 kbps at 915 MHz and 20 kbps at 868 MHz. ZigBee uses DSSS with Binary Phase Shift Keying (BPSK) modulation at frequencies 868 and 915 MHz, and DSSS with Offset Quadrature Phase Shift Keying (OQPSK) at 2.4 GHz.

There is a single channel allocated for the band of 868 MHz and 10 channels for the band of 915 MHz. Specifically, the single channel of the 868 MHz band occupies the spectrum 868-868.6 MHz whereas the 10 channels of the 915 MHz band occupy the spectrum of 902-928 MHz. Finally, for the band of 2.4-2.4835 GHz there are 16 channels. This provides the ability for ZigBee to relocate within this spectrum and hence avoid interference from Bluetooth, WLAN, cordless phones, microwave ovens. Additionally, ZigBee standard allows dynamic channel selection, which further aids colocation of technologies.

2.3.5 Ultra Wideband - ECMA-368 (PHY)

Ultra Wideband (UWB) is a radio frequency technology that transmits digital data across a very wide spectrum. The concept of UWB transmission systems is not new. The first known UWB transmission made by Heinrich Hertz in 1887 and refined by Guglielmo Marconi in 1901. Use of ultra wideband does not allow spectrum sharing and the Federal Communications Commission (FCC) was forced to regulate the spectrum. Before 2002, UWB application was mainly radar and used by military. However, in February of 2002, FCC authorised the unlicensed use of the spectrum 3.1-10.6 GHz [13], which allows low power UWB transmissions (-41.3 dBm). This increased the interest in Ultra Wideband technology as it allows faster short range wireless communications. A new IEEE group formed, IEEE 802.15.3a, in an attempt to standardise the physical and data link layers for ultra wideband systems in November of 2001. The IEEE 802.15.3a attempted to combine proposals by two industry alliances WiMedia and UWB Forum, but the attempt failed to materialise and the Project Authorisation Request (PAR) withdrawn in January 2006. At the same time, WiMedia Alliance request the European Computer Manufacturers Association (ECMA) to standardise its proposal which was finally approved in December 2005 [14] and it is known as ECMA-368.

The proposal of UWB Forum was based on Direct Sequence Ultra Wideband (DS-UWB) technique. In this scheme, narrow UWB pulses and time-domain signal processing are used to transmit and receive information. The

data modulation is BPSK and can achieve data rates of up to 1 Gbps [15].

ECMA-368 the physical layer (PHY) for wireless Personal Area Networks (PANs) utilising the 3.1 to 10.6 GHz band with supporting data rates of 53.3, 80, 106.7 160, 320, 400, and 480 Mbps. The spectrum is divided to 14 bands, 528 MHz wide each. The first 12 bands are grouped to 4 band groups each consisting of 3 bands. The last 2 bands are grouped into a single band group. All ECMA-368 compliance devices implement the PHY and MAC for the first band group [14].

ECMA-368 uses Multi Band Orthogonal Frequency Division Modulation (MB-OFDM)[3] to modulate and transmit the information. There is a total of 122 subcarriers. 12 pilot subcarriers which are used to allow for coherent detection, 10 guard subcarriers and 100 data carries. Coded data is spread using Time-Frequency Codes (TFCs). The total number of logical channels in ECMA-368 is 30. The information modulation is dependant upon the rate. QPSK is used for the rates 53.3 to 200 Mbps and Dual Carrier Modulation (DCM) is used for rates 320 to 480 Mbps.

2.3.6 Medium Access Control protocols - Data Link Layer

The MAC protocols are used to coordinate the nodes sharing the same medium; the air. There are a variety of ways to share the channel among the multiple users. On one hand, there are fixed assignment channel access methods, such as Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA), and Space Division Multiple Access (SDMA). The alternative mechanisms are random access methods; such a protocol is IEEE 802.11.

For the first type of MAC protocols, each user is allocated a slot and its responsibility is to transmit only within this slot. For example, in TDMA-like protocols the user is allocated a time slot, while for FDMA-like protocols the user is allocated a frequency slot, for CDMA the users transmit simultaneously on time and frequency but each uniquely encodes its data using a code. Their unique code is their allocated slot. Finally, for SDMA the receivers use smart antennas to distinguish the direction of a particular transmitter. The multi-path phenomena and the interference of the other transmitters are cancelled. Usually in these scenarios, transmitter arrays are used but further discussions fall outside the scope of this document.

[3]Comparison of DS-UWB and MB-OFDM approaches is out of the scope of the thesis.

For the second category of MAC protocols, access to the channel is achieved randomly. When a node seeks to send traffic to another node, it first listens to the channel and, if the channel is idle, the node then starts data transmission. However, there is the possibility that other nodes, whose operation ranges overlap, seek to transmit data at the same time having sense the channel as idle. In this case, a collision occurs. Consider Figure 2.2 where nodes are arranged in a chain topology. Each node is only within the transmission range of adjacent nodes (chain topology).

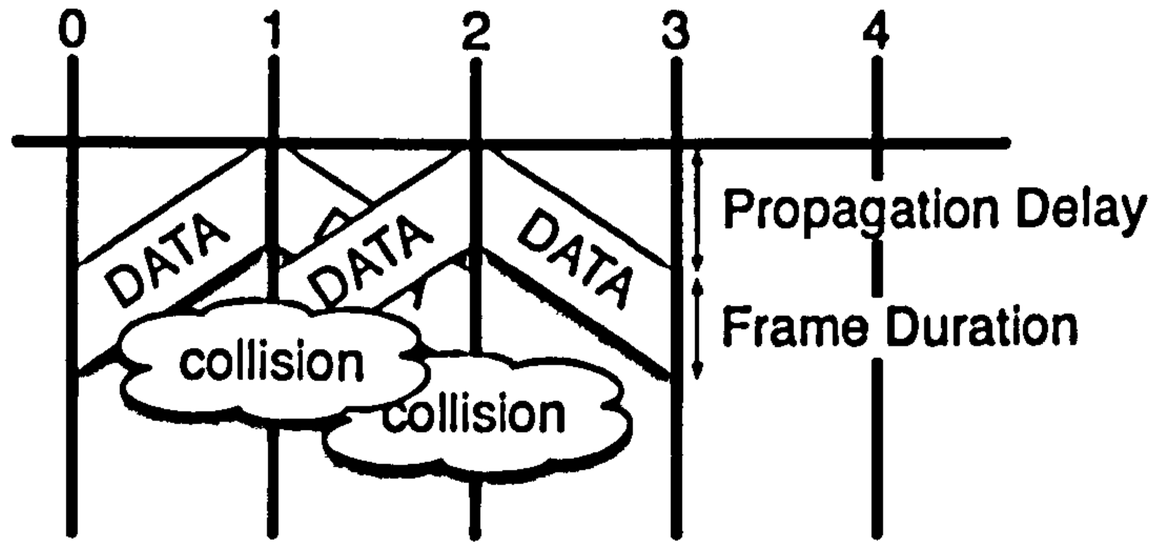


Figure 2.2: Collision scenario in a simple CSMA MAC.

Sensing channel occupancy is known as Carrier Sense Multiple Access (CSMA). The use of the carrier sense will certainly reduce the number of collisions compared to an unpoliced protocol but will not guarantee a collision free environment. Most collisions occur in a CSMA scenario are due to the inability of the transmitters to "hear" what the receiver "hears".

There is a well known problem called the hidden terminal problem (Figure 2.3). In the wired networks, this problem does not exist because the transmitter node and the receiver node are attached to the same piece of cable; not the case in the wireless world. One technique to reduce such collisions is the use of the pre-transmission handshaking messages such as those used in the IEEE 802.11 [9] and MACAW [16], (i.e. Request-To-Send (RTS) and Clear-To-Send (CTS)).

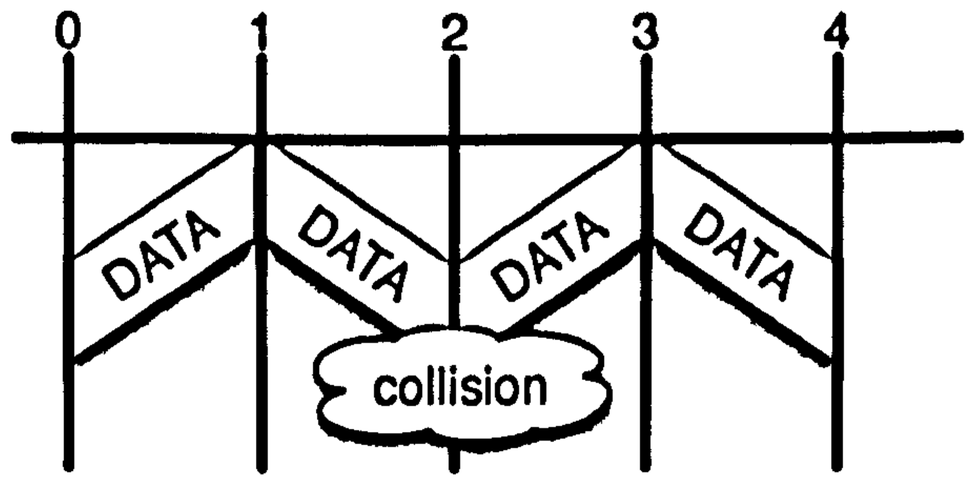


Figure 2.3: Hidden terminal problem in wireless systems using simple CSMA.

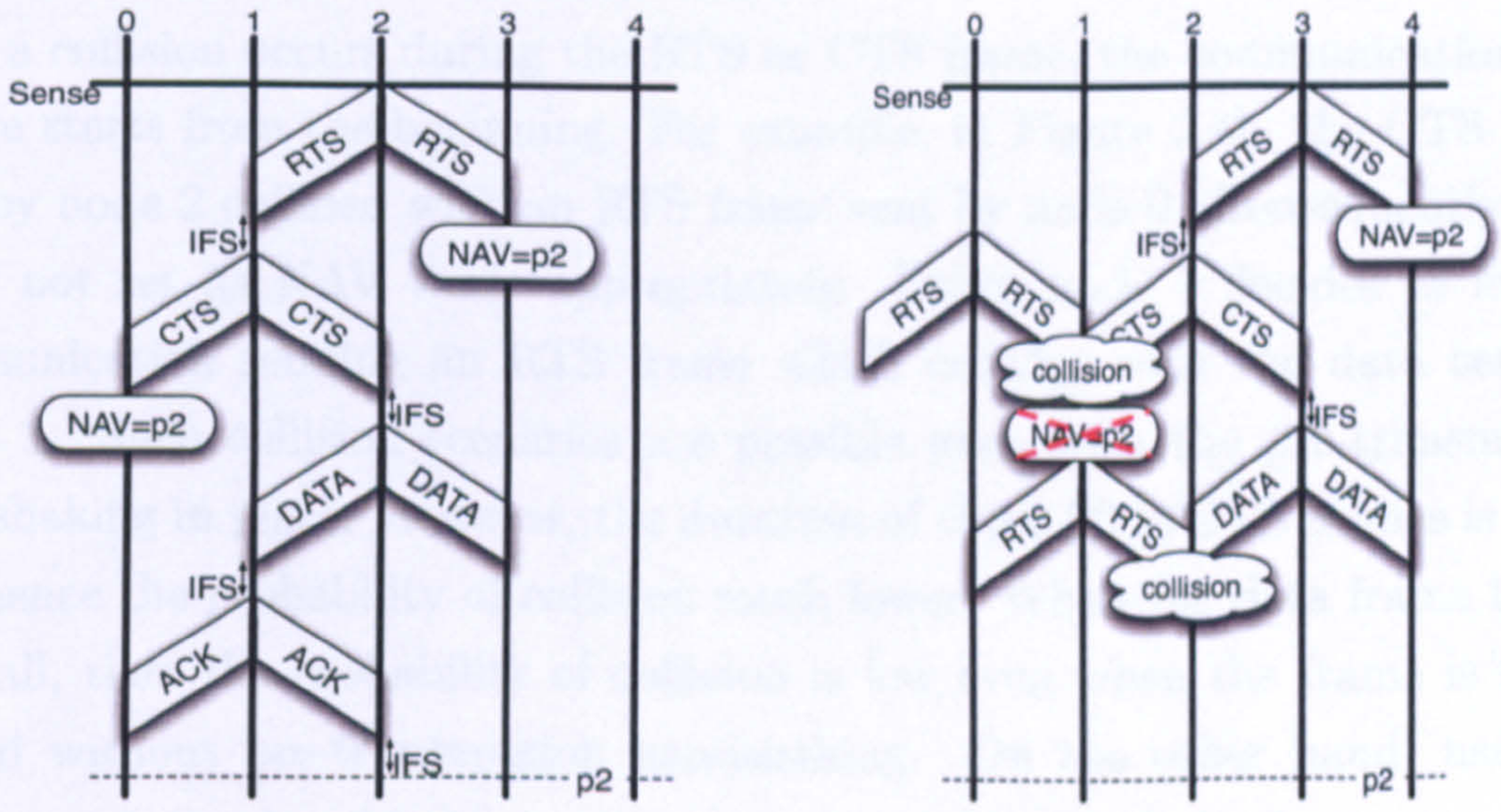
2.3.7 WLAN - IEEE 802.11 (MAC)

IEEE 802.11 and IEEE 802.3 (Ethernet) have common characteristics, but at the data link layer, IEEE 802.11 has some distinct differences compared to Ethernet. With Ethernet, a station who seeks to transmit a data frame, first, senses the channel and, if it is occupied by another station, it then waits for the channel to become idle and then transmits its frame. At the same time, the station listens to the channel and it should receive its own message. If the received data is not the same as that sent, it is assumed that a collision has occurred caused by another station attempting to transmit at the same time. The station then backs off for a random interval of time and a new transmission attempt starts, repeating the same procedure. This technique is known as Carrier Sense Multiple Access with Collision Detection (CSMA/CD). In the wireless world, the transmitter does not have the ability to detect the collision, because the collision occurs spatially on another geographic location where the destination of the transmission is located; the hidden terminal problem described before. Such collisions occur because not all stations are within radio range of each other and, hence, they sense the channel as idle, when in actual fact it is occupied by another station, i.e. transmissions which occur in a part of the cell may not be received by stations located in another geographic region of the cell.

To avoid the hidden terminal problem, IEEE 802.11 operates using two modes. The first is Point Coordination Function (PCF), which uses a base station to control the activity in the network area, and the second is Distributed Coordination Function (DCF), which does not use any central management (like the PCF mode), but instead the channel access is achieved in a distributive manner. The IEEE 802.11 standard decrees that all IEEE 802.11 compliant devices must support DCF, while PCF is optional. When the network operates in PCF mode, the base station polls the other nodes in the network to determine

if they have any data for transmission and then coordinates the transmissions accordingly. PCF mode is out of the scope of this document since ad-hoc networks are the subject of this study, and the lack of any fixed infrastructure is assumed. The other operation mode (DCF) can either use the standard Carrier Sense Multiple Access (CSMA) or an enhanced version of it; known as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In the latter, pre-transmission handshaking is used to avoid collisions from simultaneous data transmission. According to IEEE 802.11, devices can be configured to use pre-transmission handshaking either always, never or only on frames longer than a specified length. The parameter attribute that controls the devices behaviour is "dot11RTSThreshold". In the first case, the RTS/CTS is not performed at all and any device seeking to transmit data in the channel first senses the channel and, if the channel is free, the device starts its transmission immediately.

A typical acknowledged pre-transmission handshaking sequence is shown in Figure 2.4a.



(a) Successful communication

(b) An example of an unsuccessful operation.

Figure 2.4: CSMA/CA operation scenarios.

If a node is configured to operate using pre-transmission handshaking (either always or only when frames are longer than the specified length) it will execute the following steps. When a node seeks to communicate with another node, it first senses the channel; if the channel is idle then the node must transmit a Request-To-Send (RTS) frame which is implemented as a broadcast frame (Figure 2.4a). The RTS is a short frame containing information about the source,

destination and duration of the transmission. Upon reception of an RTS frame from the other nodes, only the node for which the transmission is intended generates a Clear-To-Send (CTS) frame; implemented again as a broadcast frame. All other nodes that have received the RTS frame set a back-off timer known as the Network Allocation Vector (NAV) and are not permitted to transmit for as long as the timer remains active. As soon as the originator of the request receives the CTS frame, it starts sending its data to the desired node. All other nodes that have received the CTS frame also set their NAV timer. All the nodes that have an active NAV timer are not permitted to transmit until their NAV expires. If the data frame transmitted requires acknowledgement; the destination node upon successful reception of the data frame issues an Acknowledgement (ACK) frame back to to the transmitter to indicate successful reception. In the mean time, the NAV timer of all nodes that received the RTS or the CTS frames expires and they are then allowed to initiate new connections (Figure 2.4a). This approach of pre-transmission handshaking reduces the probability of collisions since the RTS and CTS frames are short[4].

If a collision occurs during the RTS or CTS frame, the communication procedure starts from the beginning. For example, in Figure 2.4b, the CTS frame sent by node 2 collided with an RTS frame sent by node 0. Consequently node 1 did not set its NAV timer appropriately. Later, node 1 decides to initiate communication sending an RTS frame which collides with the data send by Node 3. Such collision scenarios are possible even with the pre-transmission handshaking in place. However, the duration of these RTS/CTS frames is small and hence the probability of collision much lower. When the data frame length is small, then the probability of collision is low even when the frame is transmitted without pre-transmission handshaking. On the other hand, usage of pre-transmission handshaking when data frame length is small may introduce unnecessary congestion to the channel. When devices are expected to operate in a lightly loaded environment, this value should be set to a maximum (i.e. disable pre-transmission handshaking), and when the devices are expected to operate in a heavily loaded environment the attribute should be set to a minimum (i.e. use pre-transmission handshaking for all frames). An alternative proposal would be to have a dynamic "dot11RTSThreshold" attribute [17].

Finally, IEEE 802.11 provides four types of Interframe Spacing (IFS). Discussion of when each interframe spacing is used is out of scope of this document and more details can be found in Section 9.2.3 of the standard [9].

[4]The RTS and CTS frame lengths are 20 octets and 14 octets long respectively.

- Short Interframe Spacing (SIFS); most notably used between RTS/ CTS/ DATA/ ACK sequences.
- Extended Interframe Spacing (EIFS); most notably used in DCF mode when the last transmission did not result in correct reception.
- Point (coordination function) Interframe Spacing (PIFS); most notably used to obtain channel access. The medium is sensed for PIFS in PCF mode before the device starts its back-off window.
- Distributed (coordination function) Interframe Spacing (DIFS); most notably used to obtain channel access. The medium is sensed for DIFS in DCF mode before the device starts its back-off window.

2.3.7.1 Bluetooth - IEEE 802.15.1 (MAC)

The basic network structure in a Bluetooth network is a piconet. When two Bluetooth devices come within range of each other, a piconet is formed. One device acts as a master node while the other acts as slave nodes. Each piconet can have up to 7 active slave nodes and supports up to 255 parked nodes. Parked nodes are those who are part of a piconet but are in sleep mode. Multiple piconets can exist in the same region. Interconnection of the piconets is achieved via a bridge node and such networks are known as scatternets.

In a piconet, the master node coordinates the channel access that is achieved in a traditional TDMA manner. The only communication allowed in a piconet is between the master and slave nodes. When a slave node seeks to communicate with another slave, then the master of the piconet is used as a relay. This means that the source slave node sends its traffic to the master node, who then forwards the traffic to the destination slave. Therefore, the master node has higher capacity demands as a result of its responsibility to act as a relay.

The master node coordinates the channel access and divides the time into 625 μsec time slots with each time slot able to carry 625 bits. The master node uses all the even numbered time slots for its transmission needs. The odd numbered time slots are allocated to the slave nodes for their communication towards the master node. The master node coordinates also the frequency hopping sequence of the piconet. The frequency hoping is organised in frames of 1, 3 or 5 time slots. For a frame of 1 time slot, the number of bits available for data transmission can be calculated as follows. A frequency hopping settling time of 250-260 μsec is required[5], in order to allow the radio circuits to stabilise. Subtracting this

[5]Higher cost circuits can settle faster than that.

time from the duration of the time slot; the time remaining corresponds to time equivalent for transmitting 366 bits. From the 366 bits another 126 bits are used for the access code and header information and only 366-126=240 bits are left for actual data transmission.

When 5 slots are combined together to a frame, only one settling time is required. Therefore, the actual bits transmitted are 5×625=3125 bits in five time slots, with 2781 being available to the baseband layer. Therefore, contiguous (longer) frames are more efficient than single slot frames.

2.3.8 ZigBee - IEEE 802.15.4 (MAC)

The network organisation in ZigBee is as follows. The devices can either be Full Function Devices (FFDs) or Reduced Function Devices (RFDs). Each network must include at least one FFD which acts as the network coordinator (usually referred as the Personal Area Network (PAN) coordinator). Because of the reduced functionality, RFDs should only be used for simple applications. An FFD can communicate directly with other FFDs or RFDs, while an RFD can only communicate with other devices only via an FFD. The network topologies can vary from star, to cluster tree and mesh as shown in Figure 2.5. A comparison of RFDs and FFDs is shown in Table 2.3.

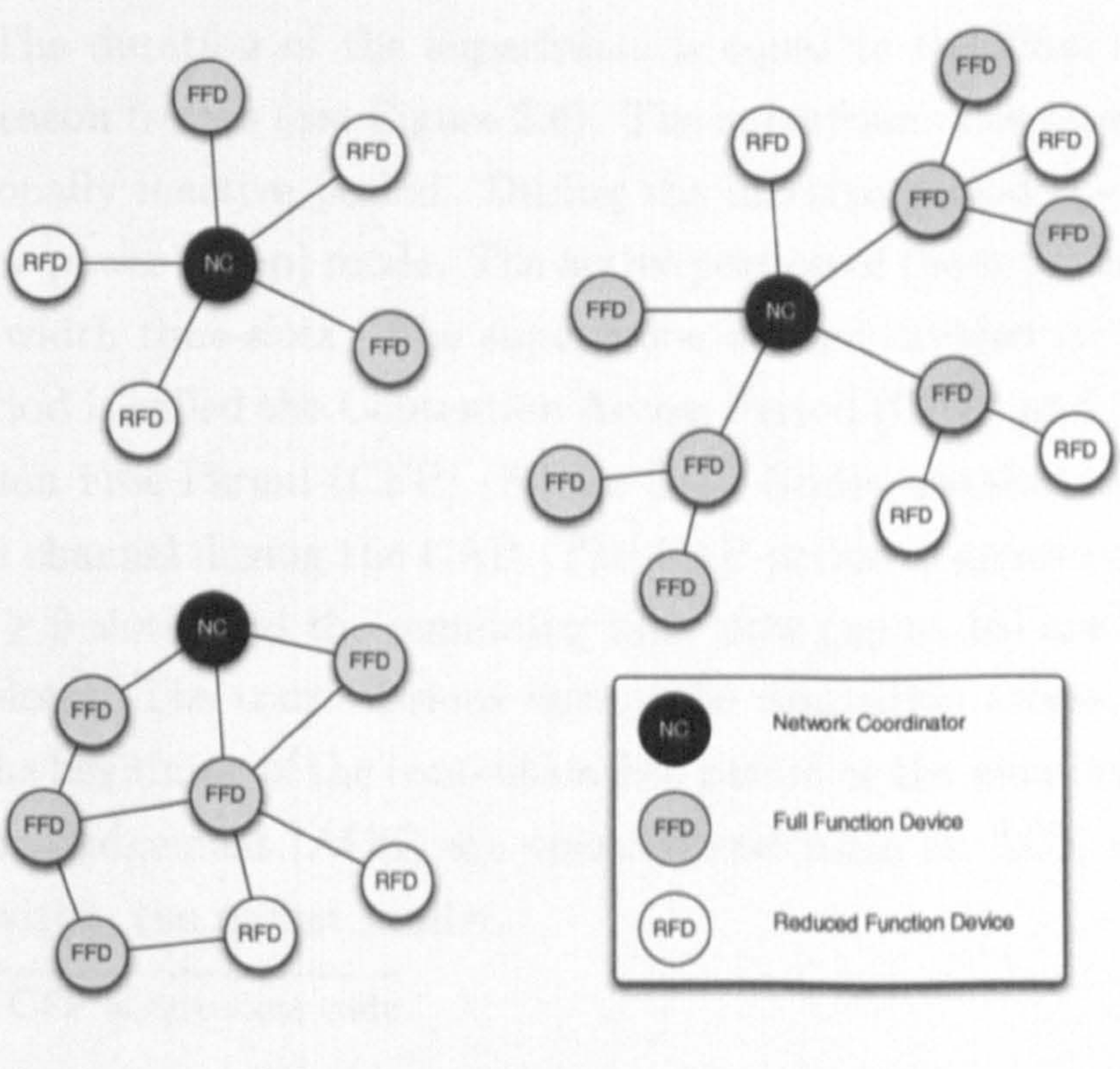


Figure 2.5: ZigBee topologies.

Reduced Function Device (RFD)	Full Function Device (FFD)
Limited to star topology	Can function in any topology
Cannot become network coordinator	Network coordinator capability
Talks only to network coordinator	Talks to any network device
Simple implementation design	More sophisticated design
Usually battered powered	Usually mains powered

Table 2.3: Comparison of RFDs and FFDs in a ZigBee network.

The IEEE 802.15.4 defines two type of networks; beacon-enabled and non-beacon-enabled networks. The PAN decides the network type and, in a beacon-enabled network, sends periodic beacons in predetermined intervals, that are multiples of 15.38 msec up to a maximum of 252 sec. When non-beacon operation is enabled, the PAN will not send any beacons. One reason for selecting a non-beacon network type may be because the PAN wishes to save energy and enter sleep mode.

For beacon-enabled networks, the IEEE 802.15.4 defines the superframe structure. The duration of the superframe is equal to the time between two successive beacon frames (see Figure 2.6). The superframe has an active period and an optionally inactive period. During the inactive period the coordinator may go to low-power (sleep) mode. The active portion of the superframe consists of 16 equal width time-slots. The superframe is then divided to two periods. The first period is called the Contention Access Period (CAP) and the second is the Contention Free Period (CFP) (Figure 2.6). Nodes use slotted CSMA/CA to access the channel during the CAP. The CAP period is greater or equal to 9 slots ($CAP \geq 9$ slots) and the remaining time slots (up to 16) are left for CFP ($CFP \leq 7$ slots). The transmissions during the contention access period must end before the beginning of the contention free period or the next beacon frame[6]. Packet Acknowledgement (ACK) are optional and when an ACK is required it is specified within the packet header.

[6]When the CFP is zero slots wide.

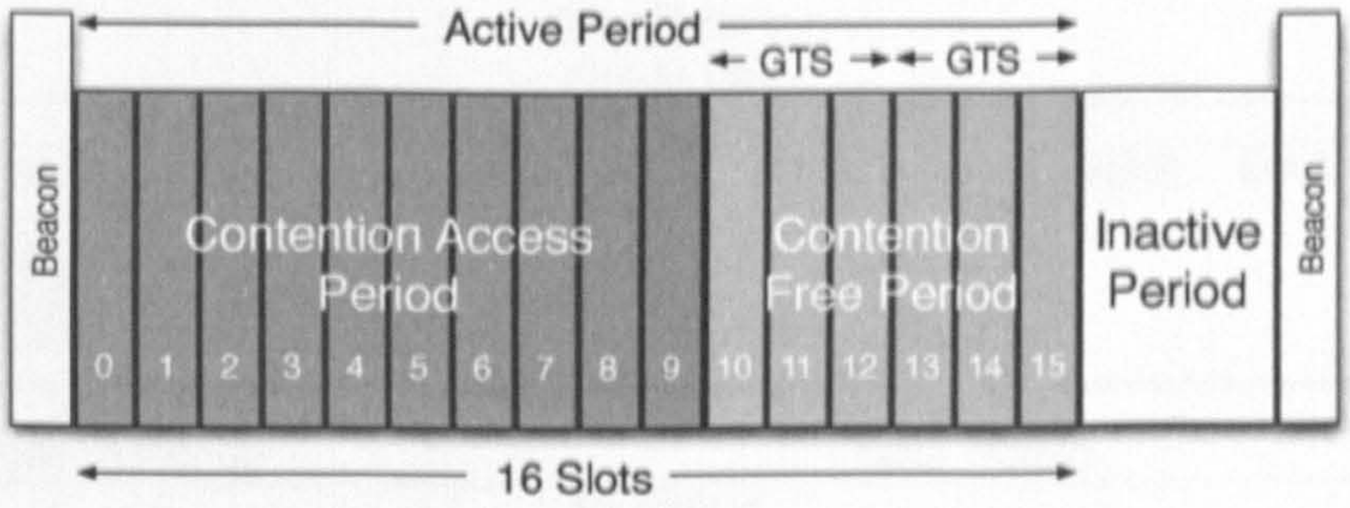


Figure 2.6: IEEE 802.15.4 superframe.

The CFP consists of Guaranteed Time Slots (GTS) for various nodes in the network. The GTS are allocated by the PAN coordinator and this information is contained within the beacon frames. The above described operation is a TDMA access method during the CFP where the PAN decides who is accessing the channel. During the CAP, the channel is accessed using slotted CSMA/CA method. This method is somewhat different from the CSMA/CA mechanism that is used by IEEE 802.11 DCF (Section 2.3.6). The back-off time in this mechanism is defined in terms of slots and aligned with the start of the beacon transmission. Every time a device seeks to transmit data frames during the CAP it locates the next back-off slot and then waits for a random number of back-off slots before trying to transmit its data. If the channel is idle during the attempt to transmit, then the device can start transmitting its data frame. If the channel is busy, then the device will wait for another random number of back-off slots. The only time when CSMA/CA shall not be used is when sending beacons or Acknowledgement (ACK) frames.

If the PAN coordinator decides that the network type will be nonbeacon-enabled network, it does not transmit beacons. Devices access the channel using the traditional CSMA/CA mechanism like the one described in Section 2.3.6.

2.3.9 UWB - ECMA-368 (MAC)

ECMA-368, unlike Bluetooth and ZigBee does not force distinction to devices, such as piconet master and slave nodes or full and reduce functionality devices. All devices operate independently and have the same capabilities. However, ECMA-368 forces a timing structure known as the superframe. The superframe length is 65,536 μsec and is divided to 256 Medium Access Slots (MASs). Each MAS is 256 μsec long. ECMA-368 requires that devices are synchronised with adjacent devices with a tolerance of ± 1 μsec. Figure 2.7 shows the superframe

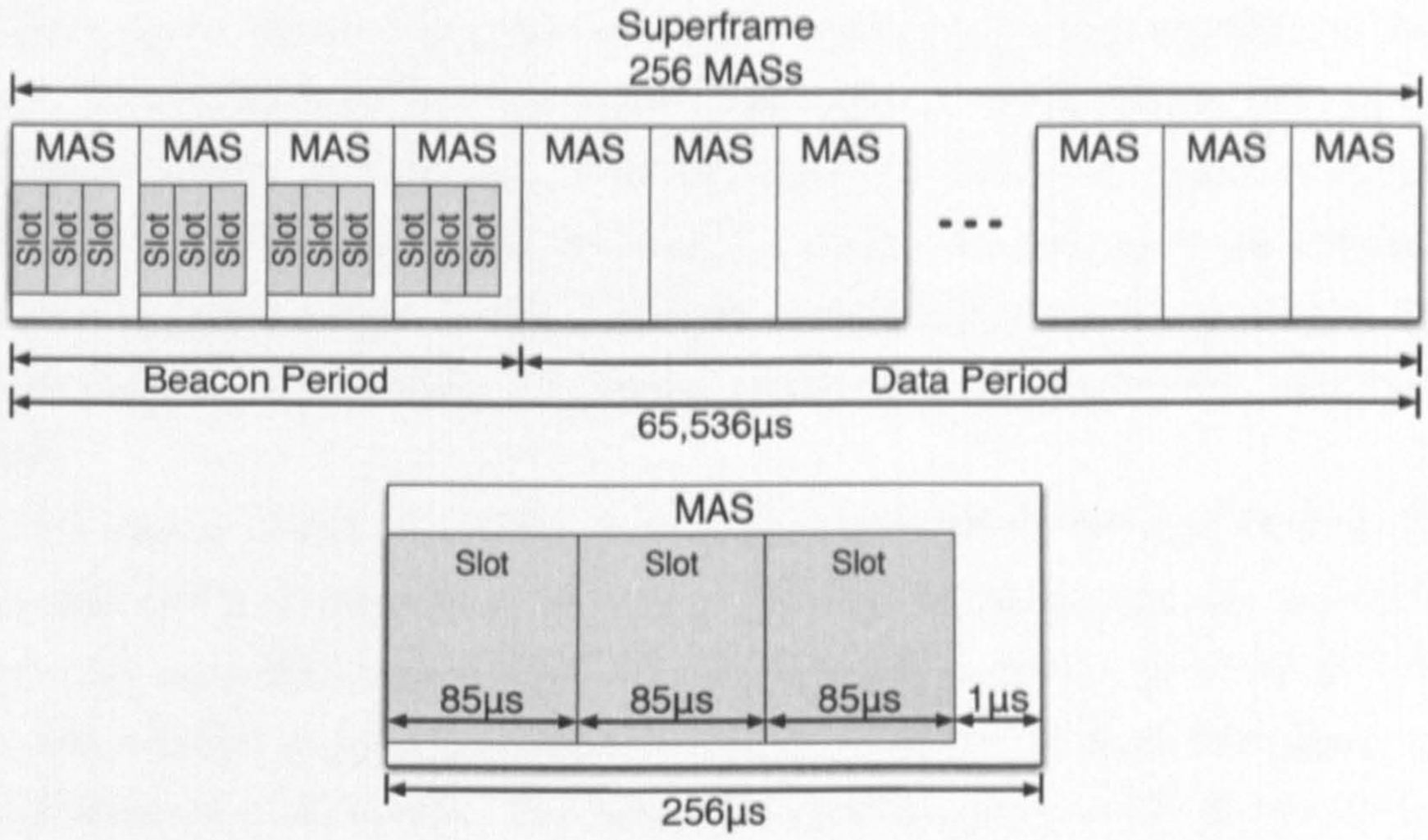


Figure 2.7: ECMA-368 superframe structure.

structure.

Devices are required to transmit beacon frames at the beginning of every superframe. The remainder of the superframe is used by devices for data communication. The beginning of the superframe is known as the Beacon Period (BP) and the remaining section of the superframe is known as the data period (DP). The BP is variable and its length depends upon the number of devices in the region of the network.

Each MAS during the beacon period is further divided to 3 beacon slots. Each beacon slot is 96 μsec long[7]. The length of the beacon period is measured in slots and must be no longer than 8 slots (maximum extension window) after the last occupied slot and no longer than 96 slots in total.

All devices, upon joining the network, attempt to get a random slot in the beacon period extension. If the beacon slot is "grabbed" successfully, the device transmits its beacons in that slot until a beacon collision is detected by the device. In that case, the device attempts to choose a new slot at random from the extension window. In order to minimise the beacon period length, ECMA-368 employs a contraction mechanism; again out of scope of this document. The data period is also variable, and devices exchange data using contention based and contention free methods. These access methods can be mixed, unlike ZigBee where the contention based and contention free methods are separated.

[7]3 slots, 96 μsec long result to 255 μsec. Which is not equal to 256 μsec, which is the duration of the MAS.

The beacon frames transmitted at the beginning of the superframe aid the devices to synchronise and exchange information. Information can be in the form of broadcast or responses to probe requests. For each type of information, an Information Element (IE) is defined. There are currently 19 information elements defined within the ECMA-368 specification which, amongst others, provide means to negotiate resources, probe device capabilities, monitor link quality.

Two logical group of devices are defined in ECMA-368; the Beacon Group (BG) and the Extended Beacon Group (EBG). These groups are defined with respect to an individual device. If the individual device is labelled with A, then the beacon group of devices A is the group of devices belonging to the neighbourhood of device A. The extended beacon group is the group of devices that belong to A's neighbourhood and the device belonging to A's neighbours neighbourhood.

An ECMA-368 compliant device employs the Distributed Reservation Protocol (DRP) to negotiate resources with devices in its beacon group. Devices can make hard, soft or Prioritised Contention Access (PCA) reservations.

Hard reservations guarantee exclusive channel access (contention free) to the reservation owner. If the reservation owner does not use all of its reserved channel time, it must relinquish any unused channel time thus allowing other devices to compete for the released channel time.

Soft reservation is a contention-based access but the device owning the reservation has a smaller back-off window than other contestants and hence has higher probability of accessing the slot.

For the final reservation type, PCA, no device has preferential access and all devices compete for the channel with the same success probability.

2.4 Routing - Network Layer

In traditional inter-networking terminology, the routing process can be described as the method of resolving a path for a data packet to travel from a specified source to a specified destination. By consensus, routing functionality is located logically within Layer 3 (also known as the Network Layer) in the standard OSI Layer Model [18]. In general, routing operations are performed by communication devices called routers. Routers can select paths based on a variety of cost-based criteria, i.e. length of a path, user priority, etc. Firstly, in this section, the wired network routing concept is introduced. The latter part of this

section will then proceed to classify wireless ad-hoc routing protocols: the main focus of this thesis.

2.4.1 Routing in wired networks

Routing in wired networks can be categorised as being either static or dynamic. In static routing, fixed paths are used to route packets to their desired destination. In IP networking, the destination is selected on a hop-by-hop basis, rather than computing the entire path in advance. Each node, upon reception of a packet, consults a statically-constructed routing table and forwards the packet appropriately to an adjacent node. Of course, such an approach demands that the node has a consistent view of how to reach all destinations in the network. In static routing, a system administrator constructs the routing tables for each node. The chief advantage of static routing is its simplicity and it is appropriate for small networks; with larger networks it is likely that such routing table maintenance is not feasible. For example, consider static routing in a network with 200 segments and 12 routers: in this scenario it is necessary to compute each next hop neighbour for each segment, which results in approximately 2,400 routes [19]. In static routing the price of simplicity is paid by the lack of scalability, since big routing tables are difficult to construct and maintain manually. The routing tables must be edited manually by the system administrator in order to adapt to such changes.

On the other hand, in dynamic routing, route maintenance is more convenient when large routing tables are considered; the mechanism by which routing tables are maintained is through the use of protocols responsible for exchanging routing information between routers thus human interaction in the routing process is not required. Supporting dynamic routing allows networks to grow larger, more quickly, and the routers are able to adapt to topological changes made, when failures or configuration changes[8] occur. This is particularly true in the wireless domain, where nodes can move at will and changes in the network topology are very frequent. In dynamic routing, each router announces to its neighbours the destinations it can reach. The other routers, upon reception of these routing tables, can capture any useful information found in the advertised routing tables; and then update their own routing tables accordingly.

However, the improved scaling properties of dynamic routing are offset by an increased cost of implementation complexity. This increased complexity requires

[8]From the routing protocol's point of view there is no real difference between a failure and a configuration change.

more sophisticated hardware for the router implementations. Such sophisticated implementations are feasible in the wired domain but this is usually not true for ad-hoc networks where the hardware capabilities of the nodes are limited due to their low cost. In addition to the higher hardware complexity, dynamic routing applies higher load to the network resources (transmission capacity), which essentially reduces the capacity available to the users. This is particularly important in the wireless domain where capacity is more precious. Independent of the domain (wired or wireless) a well designed dynamic routing protocol will attempt to minimise the routing overhead carried by the network as far as possible.

A third category is the hybrid routing scheme where both static and dynamic routing schemes are employed in different parts of the network. One common scenario is the deployment of static routing in the access network and dynamic routing within the core network. Under such a scenario, static routing is used for access network; communications devices used in access networks must be cost-efficient and thus lack the ability to support complex routing operations. Additionally, capacity available within an access network is usually lower than within a core network and therefore usage of the capacity for exchanging routing information impacts the performance of the network. By contrast, in the core network, the number of connections is very high with many topological changes, rendering static routing infeasible.

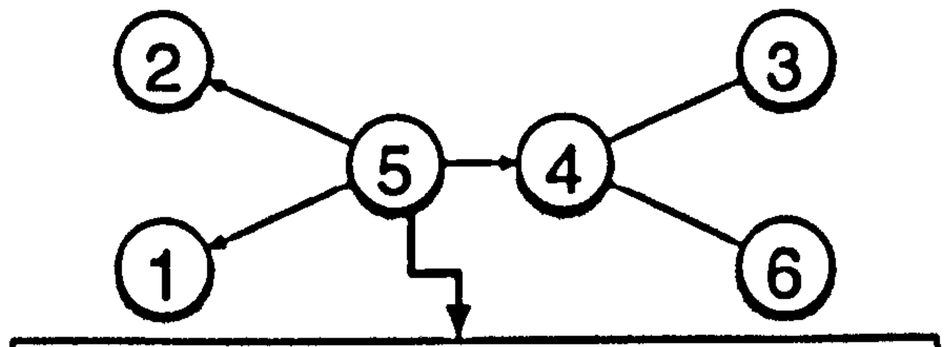
2.4.1.1 Distance Vector vs Link State

Dynamic routing protocols can be classified according to the way routing information is exchanged between routers; Distance Vector and Link State.

In Distance Vector (Figure 2.8) each router periodically sends updates to its adjacent neighbours. Each update packet contains information on all the nodes reached by that router along with their respective distances[9]. Additionally, the packet contains a piece of information regarding the direction of the path towards each route. On receipt of the updates from the adjacent nodes, the router aggregates the received table with its routing table which, enables the node to select the next hop over which traffic can be forwarded towards the appropriate destination. This method of exchanging routing information is less resource demanding since the message exchange happens only between neighbour routers rather than each router broadcasting its routing table throughout the

[9]This is not necessarily the physical distance between nodes, but any kind of cost criteria. For example, queue occupancy, delay, congestion, etc.

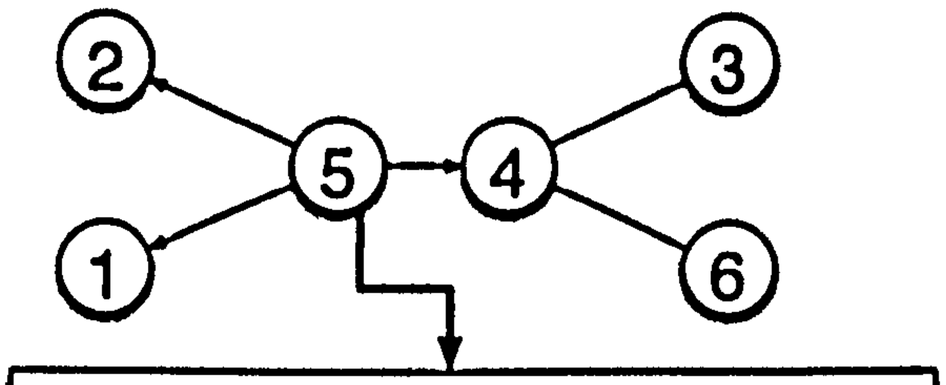
network like Link State (see below). On the other hand, the routing decision is made on per hop basis with each router having a partial view of the network and this endangers loop free routing. Loops may occur because the distances may change during the propagation of the traffic towards its final destination.



Node 5 Routing Table		
Destination	Distance	Direction
1	1	1
2	1	2
3	2	4
4	1	4
5	0	5
6	2	4

Figure 2.8: Distance Vector Routing Protocol exchange demonstration.

In Link State routing protocols (Figure 2.9) each router, using flooding, sends to all other nodes in the network, their neighbour list along with the "state" of each link (cost, which might be available capacity, delay, distance, etc.). On receipt of these broadcast tables, each router is able to form a (dynamic) representation of the complete network. In an ideal system, all nodes would have the same global picture of the network. However, in practice this is not possible simply because the transmission (and hence reception) of updates is autonomous and not coordinated. One also has to allow for the time it takes for updates to propagate throughout the network. Each node in the network is then able to compute a set of paths to all known destinations in the network. In link state routing, each link in the network is associated with a cost, and the purpose of the router is to determine which path is more cost efficient and select this path to route its traffic. In the wired domain, the weighting/cost may be based upon the hop count, available capacity, delay or any other metrics reported in the broadcast tables.



Node 5 Routing Table			
Neighbour Node	Normalised Load	Phy. Dist (Km)	Delay (ms)
1	0.1	1	0.12
2	0.6	0.8	0.20
4	0.9	5.9	0.30

Figure 2.9: Link State Routing Protocol flooding.

2.4.1.2 Common Wired Routing Protocols

This section briefly introduces routing protocols commonly used in wired networks.

- **Routing Information Protocol (RIP)** is a distance vector routing protocol and the routing metric in use is the hop count. The maximum number of hops allowed with RIP is 15. Each router using RIP transmits full updates every 30 seconds which can result in high network traffic, especially in capacity constrained networks. A mechanism called split horizon prohibits a router from advertising a route back to the interface from which it has learnt the route. Split horizon is a common technique used by distance vector routing protocols in order to avoid routing loops.

- **Open Shortest Path First (OSPF)** is a link state routing protocol which uses Dijkstra's shortest path algorithm to calculate the individual routes from a node to all other destinations. It is more sophisticated protocol compared to RIP, with the major drawback that is highly processing intensive. OSPF does not suffer from the limitations of RIP such as the limited hop count. Additionally, OSPF has better convergence time than RIP because routing changes are propagated instantaneously and not periodically. Finally, OSPF allows the creation of logical networks and the routers can be divided into areas. This limits the explosion of link state updates over the whole network; instead link state updates are propagated

	RIP	OSPF	IGRP
Type	distance-vector	link-state	distance vector
Convergence Time	slow	fast	slow
Bandwidth Consumption	high	low	high
Resource consumption	low	high	low
Multi-path Support	no	yes	yes
Scalability	poor	good	good

Table 2.4: Common routing protocols comparison [19].

only to those routers really affected.

- **Interior Gateway Routing Protocol (IGRP)** is a distance-vector routing protocol developed by Cisco Systems Inc. in the mid-1980s. IGRP developed to overcome the shortcomings of RIP namely maximum hop count, and a single routing metric. IGRP supports multiple metrics for each route, including bandwidth, load, delay, and reliability. These metrics are combined into a single metric via a user defined cost function and then compares two routers according to this single metric making the routing protocol more sophisticated and flexible. The maximum hop count of IGRP-routed packets is 255.

Table 2.4 presents a tabular comparison of these protocols.

2.4.2 Routing in Ad-hoc Networks

Routing in ad-hoc networks is subject to different constraints than those of its wired counterparts and, thus, different metrics are used when determining routes. Static routing is simply not feasible in ad-hoc networks; by definition such networks are dynamic in nature and any pre-computed routing table would either date too quickly or simply be too large to be of any use. Additionally, an ad-hoc environment demands that all the nodes in the network act as routers; not the case in wired network routing. In wired routing, there are dedicated devices that handle the routing functionality of the network. It must also pointed out that capabilities of the hosts in an ad-hoc network may be lower than those in a wired network due to the low cost implementation of the hosts; consider applications such as wireless sensor networks. In the next section, the classifi-

cation of ad-hoc routing protocols is presented and their general characteristics are discussed.

2.4.3 Classification of routing protocols

Routing protocols can be classified according to a range of criteria. One such classification is whether the protocols operate in a centralised or distributed manner. A second classification is made upon whether they operate in static or dynamic manner. The final way to classify protocols is based upon the mechanism and time when the routing information is discovered. In this final classification, routing protocols can be classified to Flooding, Proactive, Reactive and Hybrid. All three classifications can be summarised:

1. Centralised and Distributed
2. Static and Dynamic
3. Flooding , Proactive, Reactive and Hybrid

2.4.3.1 Centralised and Distributed

The efficiency of a routing protocol depends on the characteristics of the network. The deployment of ad-hoc networks is based upon the assumption that nodes are willing to contribute to the routing function of the network and they must forward traffic on behalf of others, when the routing protocol demands so.

Based on this assumption, it will be unwise to choose a centralised approach because this will reduce the reliability of the network, since partitioning might occur and the partition (which is not connected to the central point) will not be able to operate without routing functionality (single point of failure). Secondly, network nodes close to the central coordination node will be heavily congested forwarding control packets for updating topological information; required when mobility is considered. A centralised approach does not scale to large networks and distributed protocols will be more suitable since the bottleneck of the central routing administration is avoided. Taking into account these considerations, distributed routing protocols are preferred, although in principle there is nothing to prohibit development of centralised counterparts. Finally, centralised routing requires just a single highly sophisticated node, whereas distributed counterparts demand more nodes capable of handling the processing demands required by the protocol. This is acceptable in ad-hoc networks since nodes in such networks are

equipped with micro-controllers which are able to cope with the computation requirements of the routing protocols.

2.4.3.2 Static and Dynamic

A protocol is called dynamic or adaptive when its behaviour changes according to the status of the network [20]; a state change may be a corruption of a link or a node failure that results from a lack of energy or an unexpected failure. This possibility becomes more probable when mobility is introduced into the network. Static protocols on the other hand, have a pre-defined behaviour regardless of the state of the network. Therefore, the use of static routing protocols in such an unexpected environment, where nodes or links might disappear or appear in the network, is also generally not thought to be suitable, although in principle there is nothing to prohibit static protocol implementations.

2.4.3.3 Flooding, Proactive, Reactive and Hybrid

- **Flooding:** In flooding-based protocols, when wishing to transmit a packet to another node, the node transmits the packet to the next node who then forwards it to all adjacent nodes; such an operation continues until all the nodes have received the packet (broadcast approach). On receipt of the packet, nodes to whom the received packet is not intended, discard the packet while the correct final destination node processes the packet appropriately. As in the wired case, flooding is extremely inefficient and heavily uses the network capacity as residue packets continue to be carried by nodes long after the first packet was received by the destination node. Scoping [21] can be employed to limit the overhead but one should still regard any flooding based routing scheme as exerting a potentially onerous load on the network.

- **Proactive:** Routing protocols that maintain a table of routes towards all the possible destinations are referred to as global. The routes are pre-computed even if there is not a need for the route. These routing protocols perform quite well when the topology of the network is static, i.e. node mobility is not a factor or when the node mobility is very low. These protocols are also referred in the literature as Proactive [22]. Most of Distance Vector Protocols and Link State protocols use precomputed routing tables to route the traffic. Proactive routing protocols are also dynamic in nature and compute routing information in advance for all possible desti-

nations in the network. When a node in the network seeks to communicate with another node, it consults its routing table and finds the appropriate route for the intended destination. Hence, the node is not required to take any further action in finding a route to the desired destination other than searching through its routing table. In this respect, such routing protocols are called proactive, which means that they compute the paths in advance, even if most of the information they capture is not currently needed by the node. Proactive routing protocols usually have very good performance in terms of the delay involved when establishing a connection between a pair of nodes, since all the routing information has been evaluated in advance. On the other hand, such routing protocols can exert high load in the network since routing tables are updated continuously throughout the network in order to adapt to possible topological changes that might have occurred since the last update. Finally, there is a trade-off between the frequency of the updates and the routing protocol efficiency. If the frequency of updates is too high then a high routing load is applied to the network impacting the energy efficiency and capacity usage. Whereas if the frequency of updates is low, then routing tables contain dated information.

- **Reactive:** Another popular class of protocols that operates well in networks with large node population are the protocols which operate in a reactive fashion; also referred to as on-demand [22]. In reactive schemes there are no pre-computed routes for source destination pairs. However, when a node seeks to transmit information towards another node, it first sends a route request and routes are discovered "on-the-fly"[10]. The network tries then to resolve the appropriate route between the arbitrary source-destination pair. The advantage of this approach is that the node's memory can be reduced since the node does not need to store information about routes to all destinations. Furthermore, processing associated with searching a node's database in order to find the appropriate routing entry is minimised. Another advantage of on-demand routing protocol is that any new routes they discover are up-to-date, compared to the dated information cached by proactive counterparts. These protocols tend to scale well in large networks and in networks where the mobility is very high;

[10]Multipath routing protocols are capable of discovering multiple routes on-the-fly and such plurality of paths is highly likely when the source and destination devices have multiple neighbours.

the overhead packets that are sent to nodes for routing updates are also minimised as these routing updates are done on demand. Examples of reactive protocols are Dynamic Source Routing (DSR) [23], Ad-hoc On-Demand Distance Vector (AODV) [24] and Temporally-Ordered Routing Algorithm (TORA) [25]. On the other hand, on-demand protocols can not guarantee quality of service and the latency associated with establishing the connection increases because the time required to discover a route is unknown and variable. Figure 2.10 summarises and compares the proactive and reactive protocols.

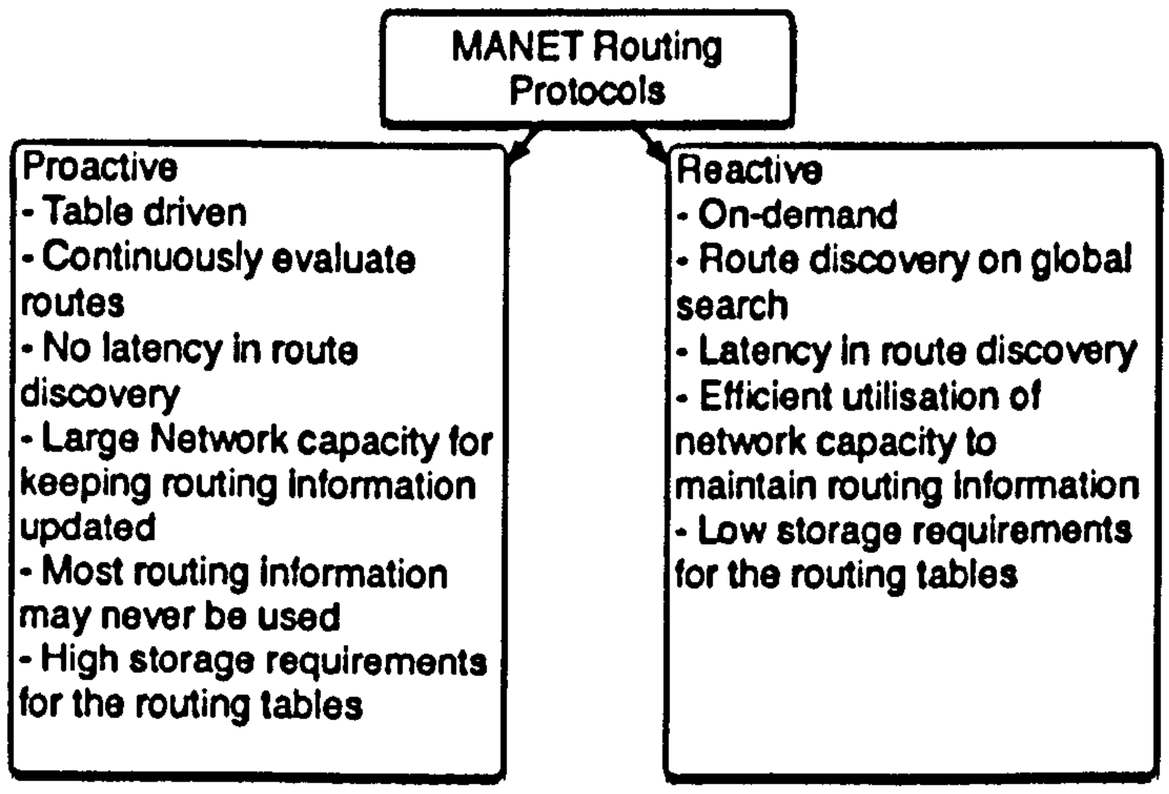


Figure 2.10: Proactive and reactive characteristics.

- **Hybrid - Hierarchical:** The final class of routing protocols considered is the hybrid case which makes use of both proactive and reactive techniques resulting in a more efficient approach. An example of hybrid routing is the zone routing [26]. In this form of hybrid routing, nodes close to the source node are grouped together to form zones and within these zones pre-computed routes are used. For nodes further away from the zone, routing information is not so precise and on-demand techniques are employed. An example of such a scheme is Fisheye State Routing[11] [28].

  Another hybrid scheme is the Hierarchical State Routing [28] (HSR), where the network is partitioned in clusters. Each cluster "elects" a cluster head and each node then reports to the cluster head. In such a way, the first

[11]Kleinrock and Stevens proposed the "Fisheye" technique to reduce size of information required to represent graphical data [27]. The eye of the fish captures with high detail the pixels near to the focal point. The detail decreases as the distance from the focal point increases.

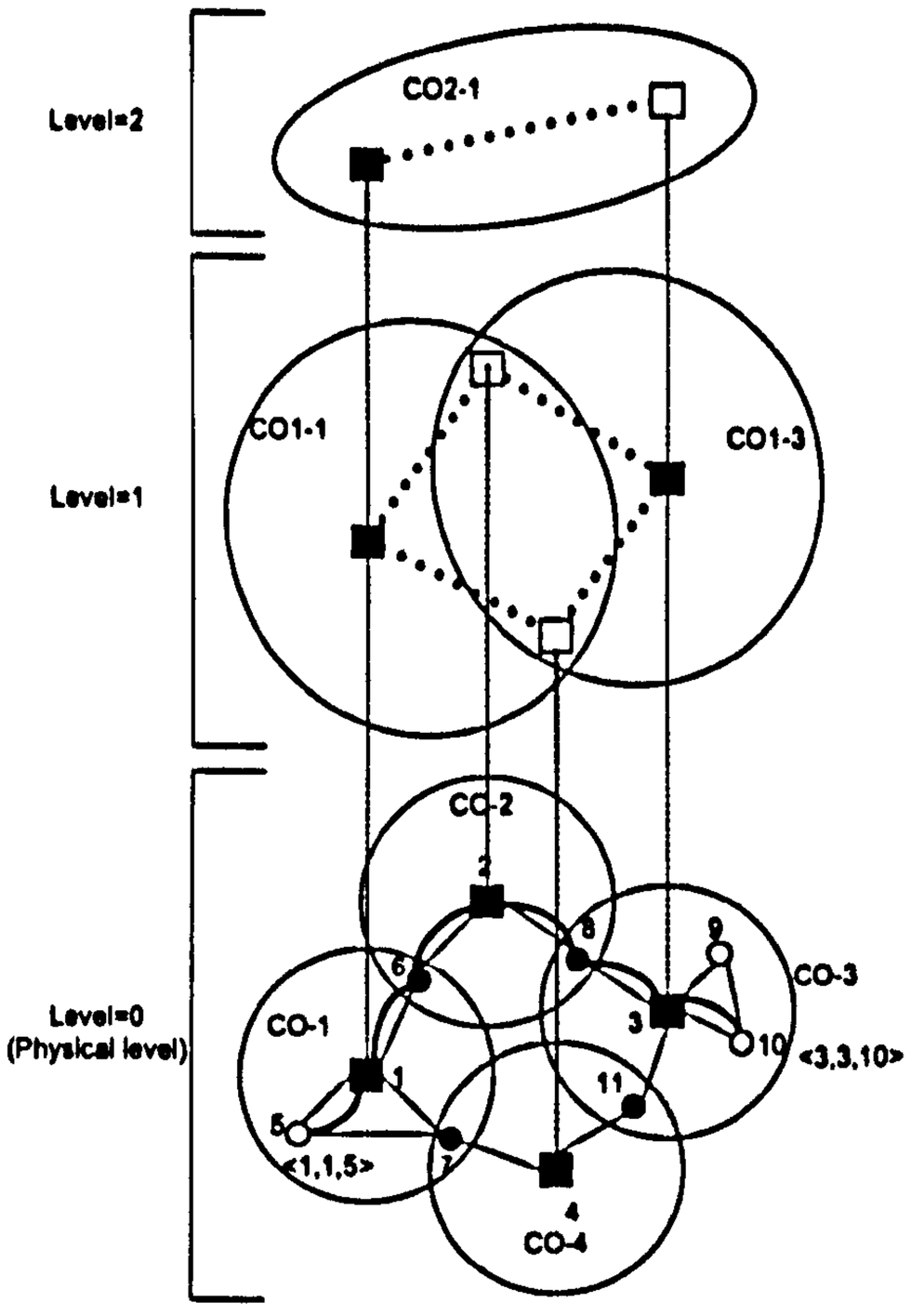


Figure 2.11: An example of physical/virtual clustering. Source: [28].

layer of cluster heads is created. The cluster heads of the first layer then form other clusters and then "elect" their own cluster head. Thereafter, when a node seeks to transmit a packet to another node, it first passes the packet to its cluster head who is then responsible for forwarding the packet to an appropriate next-layer cluster head until the packet finds the cluster head that can reach the destination node (Figure 2.11).

Such a technique is advantageous since it reduces the routing information stored in each node's database, but it can not cope particularly well with mobility because the election of cluster heads must be continuously updated in response to topological changes.

The next sections of this chapter introduce three commonly used routing protocols developed by the Internet Engineering Task Force (IETF): Dynamic Source Routing (DSR), Ad-hoc On-Demand Distance Vector (AODV) and Optimised Link State Routing (OLSR). Finally, IETF is developing another routing protocol which is a descendant of the design of previous MANET reactive routing protocols (AODV and DSR) called Dynamic MANET On-demand (DYMO) Routing [29].

2.4.4 Dynamic Source Routing (DSR)

The DSR routing protocol is a reactive routing protocol where the routes are evaluated on-demand. DSR operates in two phases. In the first phase, when a node (source) seeks to send traffic to another node (destination), it first searches through its route cache. If the node finds a route in its cache, the packet can then be transmitted in a unicast manner by sending the packet to the link layer.

The lack of a path in the node's cache may be due to the fact that previously cached routes have become invalid, i.e. mobility link breakage. A node may learn of a broken link when it receives a Route Error packet (as discussed below) or through the link-layer retransmission mechanism reporting a failure in forwarding a packet to its next-hop destination. If the source node fails to find any route for the desired destination in its cache, the routing agent of the source node enters the second state; it initiates the route discovery mechanism.

In the route discovery mechanism (Figure 2.12), the source node is now referred to as the "initiator" of the route discovery and the destination node as the "target". The initiator node creates a Route Request (RREQ) packet which, implemented as a broadcast packet, is received by all nodes within the wireless transmission range of the initiator node. When IEEE 802.11 is considered as the MAC layer protocol, the node waits until it finds the channel idle and then transmits the broadcast packet without any pre-transmission handshaking. This

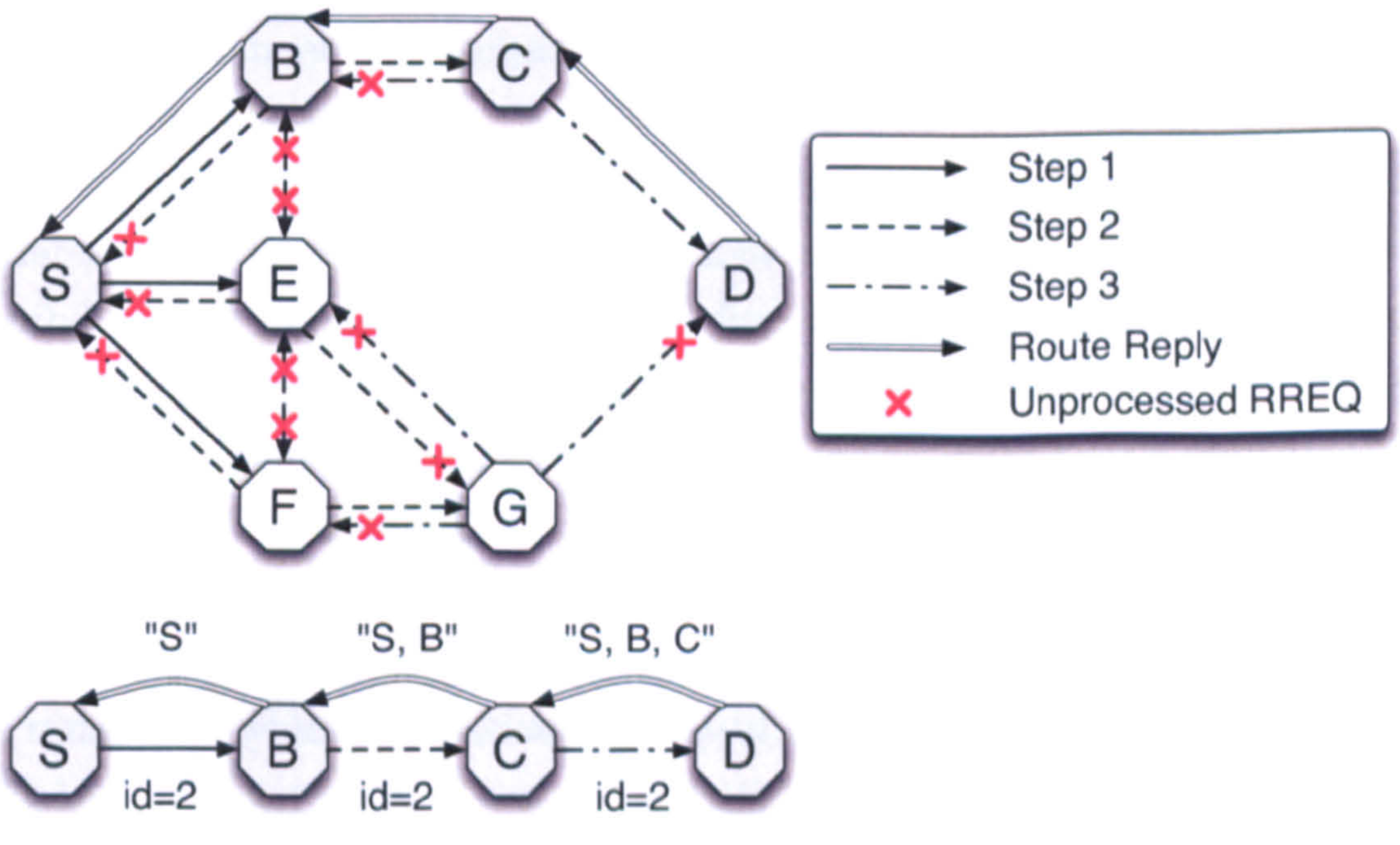


Figure 2.12: Routing discovery process of DSR.

RREQ packet contains a locally unique ID of the route discovery (which is also referred to as the request number) as well as the source of the route discovery, i.e. the node ID of the initiator node and the node ID of the target of the discovery. RREQ packets are therefore uniquely identified by the global unique couplet of the initiators node ID and the request number unique ID assigned to the packet by the route discovery instigator.

When intermediate nodes receive RREQs, examine the packets to determine the target of the discovery. If the nodes are not the target, the nodes append their node IDs to the RREQ and keep a record in their cache of the unique identifier of the RREQ. If the RREQ was previously processed, an existing entry will be found in the cache and the nodes discard the previously processed RREQ. If the RREQ was not processed before the RREQ is forwarded to adjacent nodes. In such a manner, duplicate RREQs are discarded and infinite flooding is avoided.

When the RREQ packet reaches the target of the discovery, then the target generates a Route Reply (RREPLY) packet to inform the discovery instigator. The RREPLY utilises the reverse path recorded in the received RREQ packet. The transmission of the RREPLY packet is implemented at the link layer as a unicast transmission. In the case when IEEE 802.11 is considered, the full pre-transmission handshaking of the IEEE 802.11 takes place; CSMA/CA. Upon receipt of the RREPLY, the initiator caches the route and subsequently forwards in a unicast manner any packets residing in its send buffer along the route that has just been discovered.

If the initiator has not received a RREPLY within a pre-defined period, it initiates a new discovery process with a new unique ID. However, there is the possibility that the target of the route discovery was actually unreachable as a result of network partitioning. For this reason it is necessary to specify the maximum number of attempts that the source node can make before considering the destination to be unreachable. By specifying the maximum number of retries, the non-useful traffic that is carried by the network is minimised.

The source node when it starts sending packet to the destination, appends the route over which the packet will travel to reach its final destination, in the DSR header. Intermediate nodes determine the next hop neighbours of a data packet by consulting the header. Appending the route towards the destination in every packet has a major drawback that uses significant network capacity to transmit repetitive information; i.e. route to the destination. For example, consider the scenario that a source node has 100 packets to transmit towards a

destination. Also consider the route available to the source node being 10 hops long and a node ID occupies 32 bits in the header. In this scenario, the header size of the routing information alone is $32 \cdot 10 = 320$ bits for every packet. This multiplied by 100 packets results in a routing overhead of $32 \cdot 10^3$ bits. This overhead can be reduced significantly using a DSR enhancement known as Flow State.

Flow state operates in the following manner: The source node transmits the first packet towards the destination; the packet contains in its header a path to the destination node. In addition, the node includes in the packet's header a locally unique ID known as the flow ID. The next node in the path, as soon as it receives this packet, consults the header to find the next hop neighbour towards the destination node. This information can be found in the routing header because the source node has included the full path of the connection. Once this information is extracted from the packet header, the node knows where to forward the packet; having kept track of the destination node ID and flow ID (which both of them together globally identify the flow), in addition to the next hop to which the packet should be forwarded. This process is repeated at all subsequent hops until the packet reaches its destination.

Subsequent packets sent by the source node towards the same destination do not include the whole path, but instead the source node includes the flow ID which previously had been assigned for the same connection. When intermediate nodes receive the packet and notice that the full path to the destination is not included in the packet header, they search their cache for the destination ID, flow ID pair (previously recorded) to determine the next hop neighbour. The same procedure is followed until the packet reaches the destination node. If a node can not find any entries in its records associated with this flow, it sends a flow error to the source node and the flow must be re-established. The flow state essentially is a forward pointer technique. The advantage of using such technique is that the overhead carried by the network is minimised since the packets no longer contain in their header the full path of the connection but instead store only the destination ID and the Flow ID.

If the 10 hop-100 packet scenario described previously is considered again; when flow state is enabled; the first packet carries an overhead of $32 \cdot 10$ for the full path of the connection plus 16 bits for the flow ID resulting to $32 \cdot 10 + 16 = 336$. The subsequent 99 packets contain an overhead of 32 bits for the destination node ID and 16 bits for the flow ID, giving an overall of 48 bits. For all 99 packets, the overhead carried is $48 \cdot 99 = 4752$ bits. Therefore, when transmitting 100

packets using flow state, the overall routing overhead is $336 + 4752 = 5,088$ bits which is significantly lower than the $32,000$ bits overhead carried by DSR with flow state disabled. This means that, in this scenario, DSR with flow state disable carries 6.289 times more overhead compared to the flow state enabled DSR. Of course when the path length of the connection is smaller the rating reduces.

Another way to make the routing protocol more energy efficient, is to enable an enhancement known as piggybacking [23]. In this enhancement, nodes capture routing information from forwarded or overheard packets and integrate extracted information in their cache. When the nodes forward a packet, they simply capture the information from the packet header and integrate it to their cache. In the overhearing scenario, the nodes enable their receiver and process packets which are not destined to them. For example, consider three nodes A, B and C. Consider also that node A sends traffic to node B and node C is within the transmission range of node A. There are then two possible scenarios for node C while the transmission of A to B takes place. It can either go into sleep mode to save energy for the transmission's duration, or it can listen to the packet that A transmits to extract any useful information and then discard the packet. Even if node C is not involved in such communication it can still capture useful information. However, studies have shown that a significant amount of energy can be wasted when overhearing packets [30]. Therefore, in some scenarios a piggyback approach may result in an effect that is opposite to the one desired, i.e. energy dissipation rate increases.

Another enhancement is to allow nodes to reply to any RREQ, using data stored in their cache. Here, any flooding that is created by the discovery mechanism will be limited further, since nodes do not forward the RREQ packets but reply to the initiator of the discovery. Essentially the enhancement exploits existing routes as stored in node caches and limits the number of RREQ overhead packets that must be dealt with by the network. This results in a major energy saving. However, it is likely that the nodes, as a result of say mobility, may have cached paths that no longer exist. Any invalid routing information will require the node to re-initiate the discovery mechanism and perhaps, increase the overhead (time, energy expended etc) associated with the route establishment as compared to traditional scheme.

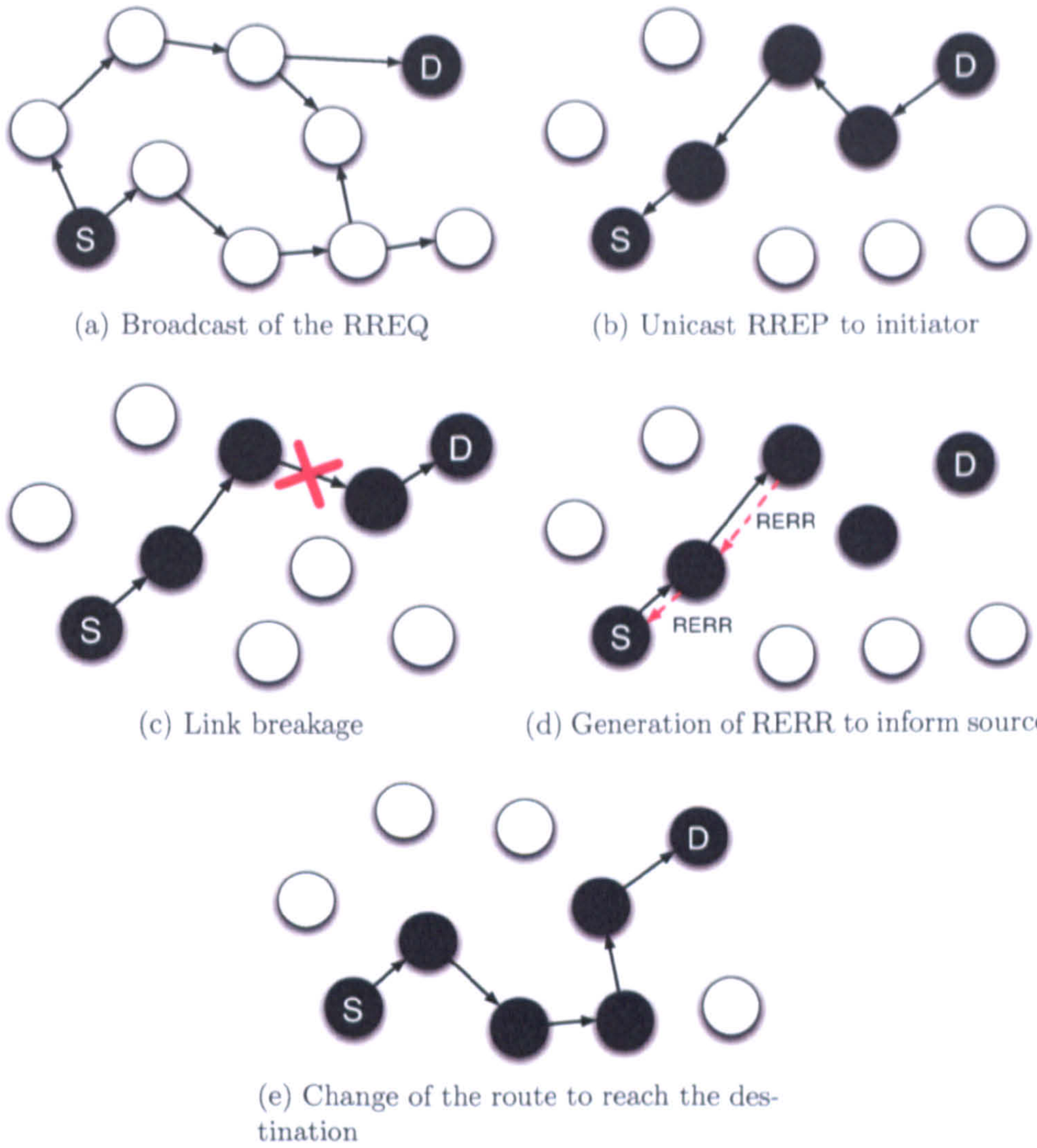
2.4.5 Ad-hoc On-Demand Distance Vector (AODV)

The Ad-hoc On-demand Distance Vector routing protocol effectively replicates the operating principles of DSR. In terms of route discovery, AODV evaluates the routes on "per-needed basis" using similar route discovery process to DSR. On the other hand, AODV differs from DSR in the way that the protocol handles the information in its cache. In an AODV capable node, the cache is organised as in a traditional table-driven routing protocol. Traditional table-driven routing protocols can result in high memory demands with large networks and therefore AODV uses timer based states for each node regarding the utilisation of individual entries in the routing tables [24, 31]. When a routing table entry is not recently used it "expires". On the other hand, DSR uses source routing, where the node organises its caches by storing a path to a destination as a linked list and therefore does not occupy memory for undiscovered destinations [32]. DSR does not use the timer based approach and keeps the routing entries for as long as it has free memory. If the memory becomes full then a mechanism known as salvaging is used to free up memory [23].

In AODV, when a source node seeks to communicate with a destination node, it first checks its cache to see if it already has a valid route to the desired destination. If not, then the data packets are queued and the route discovery mechanism is initiated. The source node broadcasts the RREQ packet to its neighbours, which they then forward to their neighbours, and so on, until the route request reaches either the destination node or a node which has a "fresh enough" route to the desired destination (Figure 2.13a). In either case, the node that will respond to the RREQ, generates a Route Reply (RREP) packet to the initiator of the discovery (Figure 2.13b). The RREP is transmitted as a unicast packet and the path that is used to reach the initiator of the route discovery is the reverse of the one used for the transmission of the route request.

Thus, when the destination receives the RREQ, it generates a RREP that uses the back pointers established by the respective RREQ. Intermediate nodes that have not received a RREP within a specified time erase the back pointer in order to allow further reverse paths for any future RREQ to be established.

AODV, in order to assure loop-free routing, uses destination sequence numbers. Each node must maintain, for every routing table entry, the sequence number of the destination. This sequence number is updated whenever the node receives new information about this destination. Such information may be obtained by an RREQ or an RREP. A node increases its own sequence numbers in two circumstances. The first case is immediately before a node initiates



(a) Broadcast of the RREQ

(b) Unicast RREP to initiator

(c) Link breakage

(d) Generation of RERR to inform source

(e) Change of the route to reach the destination

Figure 2.13: Operation of the AODV routing protocol.

a route discovery; required to avoid conflicts with previous reverse routes to the originator of a RREQ. The second situation occurs before a destination node generates a RREP in response to a RREQ.

Each node, upon reception of the RREQ packet, first checks through its records to determine whether the RREQ has been processed previously by comparing the source's sequence number with the last recorded sequence number associated with the source node. If the sequence number is equal or smaller to the one cached in its routing table, then the RREQ has been previously processed and is discarded. If the sequence number is greater, then the node must process the RREQ and forward it to its immediate neighbours. Before forwarding, the node first sets up a back pointer for this RREQ in order to allow the reverse path to be established for any respective RREPs generated by uplink

nodes.

Since every node maintains sequence numbers for every destination in its routing table, it is trivial to distinguish freshness. The node simply compares the destination sequence number of the control packet that it has just received with the number cached to its routing table. If the number cached in the table is smaller than the number obtained on reception of the control packet the node updates its cache. In the event of the node not having a corresponding entry in its routing table, a new entry is created.

The AODV supports resilience via the HELLO messages broadcasted by each node. If the node notices that a neighbour is no longer available having not received a HELLO packet within a predefined interval, then it no longer considers this "neighbouring" node as being adjacent (link breakage Figure2.13c). Consequently, the node generates a Route Error (RERR) packet to update all nodes that are using paths containing the recently expired node (Figure 2.13d). The link breakage can be also detected using link level acknowledgements, if available. Finally, a new route is used to facilitate communication between the source and destination as shown at Figure 2.13e.

2.4.6 Optimised Link State Routing (OLSR)

Optimised Link State Routing is a proactive routing protocol for mobile ad-hoc networks [33]. Its operation is based upon periodic message exchange that contain information about the topology of the network. As the name suggests, OLSR is an optimised version of the link state routing protocol used in wired networks as described at Section 2.4.1.1. In a traditional link state protocol, control messages are broadcast (flooded) to the network in order to update the topological information. Such an approach is affordable in wired networks because cable capacity is usually large, however, in the wireless domain, links tend to be capacity limited. Additionally, the topology of the network changes rapidly in wireless networks, while these changes are relatively rare in wired networks. Thus, it is necessary to optimise the way that topological updates are transmitted in a wireless network.

The OLSR protocol uses Multipoint Relays (MPRs) to optimise the inherent flooding. Each node selects a subset of its neighbours to act as relay points and the nodes selected are then required to retransmit the traffic on behalf of the node. The remaining neighbours who receive the message simply use it to update their own knowledge of the network and they are not required to broadcast

(forward) the routing packets. Therefore, routing information is flooded only through the selected relay nodes, which in turn reduces network congestion. Finally, the size of the control packets is also reduced since they contain only information regarding links that are connected to MPRs (sectors of network) rather than the full set of network links. Because the number of nodes that are involved in the broadcast are reduced, the number of collisions, and hence retransmissions, in the network are also reduced [33].

The criteria by which a candidate node can be selected to act as a multipoint relay is simply if it supports a bidirectional link. Nodes with unidirectional links are not eligible to be selected as multipoint relays. The source node selects a subset of these nodes M, excluding the set of nodes (U) that have unidirectional links.

$$M \subseteq (N \cap U)$$

The motivation behind the use of the MPRs is to minimise the transmissions within a region of the network. Thus, the set M can be equivalent to the set $(N \cap U)$ $(M \equiv (N \cap U))$ but this is essentially the same operation as the traditional link state routing. Instead, any MPR selection scheme should attempt to minimise the population of the set M as far as possible, in order to maximise the benefits of the MPRs. The set of suitable candidates for MPRs are nodes with bidirectional links and nodes must cover all the 2-hop neighbours (see Figure 2.14). This selection is autonomous for each node (algorithms for selecting MPR are described elsewhere [34, 35]). Consequently, each 1-hop node maintains a list of the "MPR selectors". This list is updated regularly by the HELLO messages sent by each node. The HELLO messages contain the information about the neighbour nodes and the status of the links with its neighbours (link state). These HELLO messages have maximum hop count equal to unity and will be only received by 1-hop neighbours.

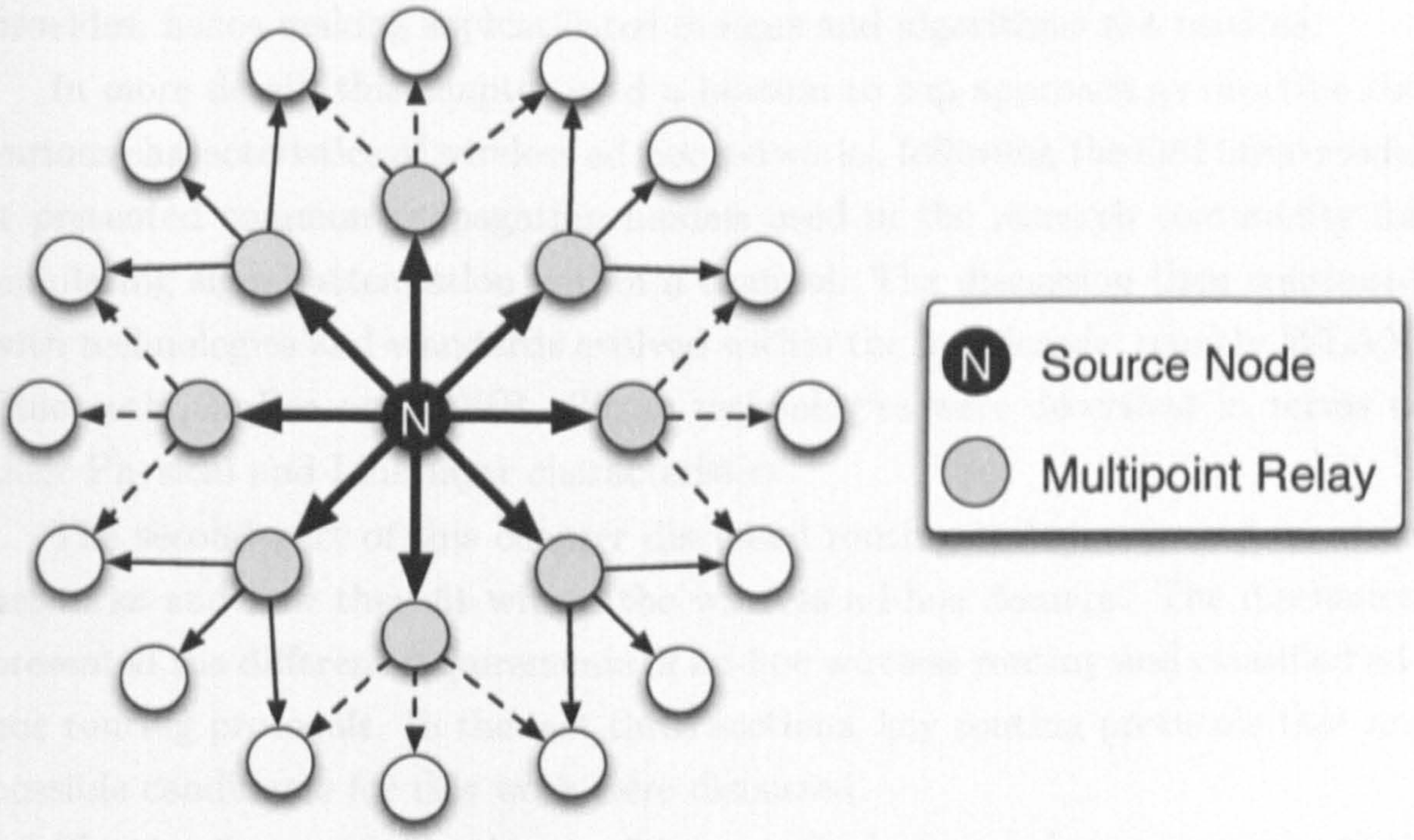


Figure 2.14: Selection of Multipoint relays from a node.

Once the MPR nodes are selected, then each node broadcasts a Topological Control (TC) message to all the network in order to inform the rest of the nodes about its neighbours. The TC messages are sent at regular intervals in order to maintain up-to-date cache information. If the MPR selectors list changes, as time progresses, then the nodes re-broadcast appropriate TC messages to inform nodes of this change. However, this is done only if the minimum interval timer between TC transmissions has expired. Otherwise, the TC update can depart immediately. If it has not observed any changes to its MPR Selector list, then the node transmits the TC messages at normal regular time intervals.

Upon reception of the TC message from a node, the information contained in the path is cached by building a descending list of connected pairs, in the form (last hop, node) [36]. "Last hop" refers to the next hop neighbour towards the destination and "node" will be the destination node.

2.5 Summary

This chapter introduced the basic concepts involved in ad-hoc networking. Wireless ad-hoc networks are substantially different when compared to wired counterparts due to the nature of the wireless medium (air). Wireless networks are also differ from ad-hoc networks as the latter lacks infrastructure, which the former

provides, hence making sophisticated designs and algorithms not feasible.

In more detail, this chapter used a bottom to top approach to describe the various characteristics of wireless ad-hoc networks; following the OSI layer model it presented common propagation models used in the research community for simulating signal attenuation within a channel. The discussion then continued with technologies and standards evolved within the last decade; namely WLAN, Bluetooth, ZigBee and UWB. These technologies were described in terms of their Physical and Link layer characteristics.

The second part of this chapter discussed routing techniques used in wired networks and how they fit within the wireless ad-hoc domain. The discussion presented the different requirements of ad-hoc wireless routing and classified ad-hoc routing protocols. In the last three sections, key routing protocols that are possible candidates for this work were discussed.

Chapter 3, introduces the candidate methods for performance evaluation. Among those candidates, a combination of simulation and analytical models are used. The chapter discusses the simulation model, the assumptions made and proposes a routing scheme that distributes the load to multiple paths. The simulation model implementation of the proposed routing scheme is validated using a low complexity scenario where analytical calculations are used to verify their validity.

Chapter 3

Simulation Modelling

3.1 Evaluation Techniques

A number of approaches can be adopted to evaluate the system performance and can be classified into the following categories:

- Measurements
- Analytical Modelling
- Simulation

The first approach would be to design and implement the complete system and evaluate its performance based on actual measurements. Such approach, however, incurs a high risk and high costs, while tuning of system parameters becomes complex in addition to long development times. For example, for evaluation of the routing protocols studied in this thesis, hardware could be deployed to measure protocol performance. A possible hardware architecture for the routing protocols evaluation could be portable notebooks equipped with wireless network cards, but the cost per node makes such approach financially prohibitive. Other hardware architectures with lower costs per node could be mirco-computers such as those produced by Gumstix Inc. [37] or low-cost wireless sensor nodes (also known as motes) produced by Crossbow Technology Inc. [38] and the formerly Moteiv Inc. and now Sentilla Corporation [39, 40]. The latter hardware architectures reduce significantly the cost per node but in order to evaluate large topologies a bulk number of devices is required and overall financial cost is still high. Finally, Harvard University offers a public Sensor Network Testbed [41, 42] deployed in the Maxwell Dworkin Laboratory called MoteLab. MoteLab has 190 TMote Sky senor motes, deployed in three floors.

While this facility has negligible costs, the network topology is fixed and the nodes are mains powered, and therefore it is not suited for studying mobile and energy aware networks.

Assuming, that there is sufficient financial budget to build such network, this approach is still prohibitive within the research environment. Multi-hop wireless networks require a significant floor space to deploy and test. For example, the thesis considers scenarios where the floor space is 500 × 200 square meters. Even if the floor space requirement is overlooked, there is still difficulty associated when measuring performance of mobile scenarios; difficulties such as "how", "when" and "how accurately" the nodes will move. In general, even if measurements seem to be the more accurate method to evaluate the system's performance, in reality the performance metrics observed are heavily dependant on the environment [43]. It is difficult to identify the environmental factors affecting the system. Systems are usually studied in isolation to understand their behaviour and then are applied to controlled environments to identify the factors affecting the performance. For such reasons, a physical network implementation is not considered due to the numerous resources required, in addition to the difficulties associate with testing and tuning the routing protocol parameters.

The second evaluation technique is analytical modelling which has the minimum cost. In general, analytical models represent the system behaviour using mathematical relations and calculation based on the system equations. Analytical models tend to require many simplifications and assumptions. Consequently, the appropriateness of the results heavily depends upon the assumption and simplifications made when the model is developed.

The final evaluation technique is computer simulation. Simulation models can be developed in relatively moderate time scales while the cost remains relatively low, i.e. a computer can be used to execute the simulation runs. The appropriateness of the results tend to be higher than those obtained by analytical models simply because simulations incorporate more details of the system's behaviour and impose fewer restrictions. Additionally, simulations can lead to more accurate conclusions than that of the measurements technique which are affected by the environment and the experimental procedure. Computer simulations provide a platform to perform consistent repeatable results.

Raj Jain [43] summarises the criteria for selecting the appropriate evaluation technique in a table shown in Table 3.1.

What it important to note is that all these techniques have pitfalls related to the appropriateness of the results and the evaluation of the system's perfor-

mance. For this reason, more than one technique should be used and combined validate each approach. If more than one technique is used and the observations do not match, then differences between the evaluation techniques must be identified and re-evaluated based on these observations.

Criterion	Analytical Modelling	Simulation	Measurement
Stage	Any	Any	Postprototype
Time required	Small	Medium	Varies
Tools	Analysts	Computer Languages	Instrumentation
Accuracy	Low	Moderate	Varies
Trade-off evaluation	Easy	Moderate	Difficult
Cost	Small	Medium	High
Saleability	Low	Medium	High

Table 3.1: Criteria for selecting an evaluation technique [43].

For the purposes of this thesis and for the reasons explained, the evaluation techniques used to carry out this work are a combination of simulation models supported by analytical models to underpin their validity.

3.1.1 Simulation Types

The simulations can be classified to four main categories; Emulations, Monte Carlo, Trace Driven and Discrete Event Simulations.

3.1.1.1 Emulations

Emulation is a type of simulation that uses system architecture to replicate the behaviour of another system. For example, a processor emulates the behaviour of another processor with a different instruction set. Within networks, one or

more computers can be used to emulate the behaviour of a network. For example, it might be difficult to change the physical topology of a real network, while emulators provide more flexibility and can be reconfigured to any logical topology. In the same way, the analyst can quickly change other network attributes such as latency, jitter, packet loss and bandwidth to behave like the real system, and enable the users to experience the performance and behaviour of different application in the emulated environment. Emulators usually involve higher cost than other simulation techniques and the emulator design issues are mostly hardware related and, hence, not the method chosen in this thesis.

3.1.1.2 Monte Carlo Simulations

The second category of simulations is Monte Carlo. Monte Carlo simulations are used to model probabilistic phenomenons that do not change characteristics with time [43]. The key steps involved in a Monte Carlo simulation are:

1. Identifying the domain of all possible inputs.
2. Generate random inputs from the identified domain and compute deterministic outputs.
3. Aggregate the results from the individual runs of step 2 to a final result using statistics.

Monte Carlo simulations are suitable in static models where the behaviour of the system does not change with time. This kind of simulation does not fit the phenomena considered in this thesis.

3.1.1.3 Trace Driven Simulations

In this category, a time-ordered list of events are recorded from a real system, usually called the trace file. This trace file is subsequently used as an input to the simulation which models different algorithms. This technique is usually used to evaluate resource management algorithms. In networks, such techniques are used to identify realistic inputs to the simulation. Finally, trace driven simulations assume that the real systems exists, which is not the case for the work carried out in this thesis. For example, mobility and traffic traces could be used as an input to the simulation model to evaluate the performance of a mobile ad-hoc networks. Such traffic traces are unavailable to carry out this work and instead, the traffic and mobility scenarios are generated as described in Sections 3.10 and 3.11.

3.1.1.4 Discrete Event Simulations

Discrete event simulations use discrete state variables to describe the state of the system at any particular time. The term discrete does not apply to the time values used in the simulation but the system state variables. For example, in a queue model, the queue length can take only discrete integer values but the length of the queue can change at any time. The most basic components in a discrete event simulation are:

- Simulation Clock: Keeps track of the current simulation time. The simulation clock is essentially a counter that is used to represent logical rather than real time intervals between events.
- Events: A structure generated by a simulation entity (object) to occur at the specified time in the future (activation time). The structure contains a function pointer (call-back function) which is known as the Event Handler.
- Event Handler: The call-back function that is called when the simulation clock is equal to the event's activation time.
- Event List: The list of events that have previously been scheduled and their activation time has not yet reached. The event list is sorted in order of activation time, i.e. the event with the earliest activation time is on the front of the list.
- Time-advancing routine: The method by which the simulation clock is increased.

There are two key approaches for the time-advancing mechanism. The first approach is to increase the time by a small time unit and then search the events list to find any events with activation time equal to the simulation clock. If such events exist, then the event handlers are executed, otherwise the simulation clock is increased by an additional small time unit, and the event list search is conducted again. The process is repeated until the simulation clock reaches the activation of the first event in the list. This approach is wasteful because resources are spent to check the first event in the list until the simulation clock reaches the activation time. The alternative approach is called event-driven and the simulation clock is immediately increased to the activation time of the next earliest event, since there are no other events scheduled between the current simulation time and the earliest event activation time.

The flowchart of Figure 3.1 illustrates the execution flow of a simulation [44]. When the simulation begins, a list of the initial events are scheduled (i.e. new

events are appended in the events list) and simulation starts. The execution enters a loop which iterates until no events remain in the events list or the stop criterion is met.

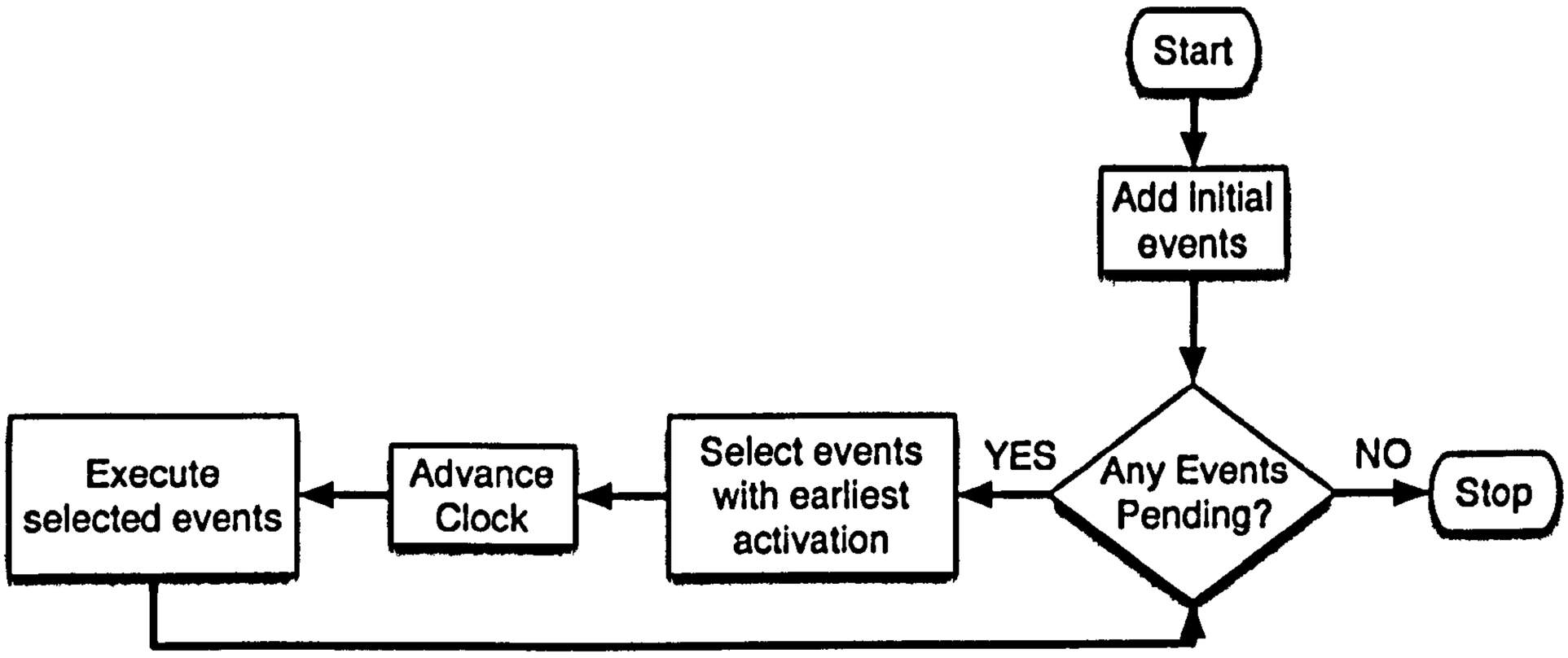


Figure 3.1: Discrete Event Simulation Execution Flow.

Within the loop, the first task is to identify the event(s)[1] with the earliest activation time. Then, increase the simulation clock to the earliest activation time of the event(s), execute the event(s) handlers, then start the process from the beginning. This process is repeated until the events list becomes empty or the stop criterion is met. It must be noted that the event handlers execute events which in return can insert, remove or modify events in the scheduler.

Finally, it should be mentioned that events are triggered by pseudo random numbers and there is a stochastic aspect to the technique. For example, random number generators are used to generate network topologies, mobility patterns and traffic profiles. Additionally, the simulator contains randomness for events, such as packet inter-arrival times and random back-off times used by the CSMA/CA protocol and the Address Resolution Protocol (ARQ). Consequently, a good understanding of probabilities is required. The remainder of this chapter discusses the aspects associated with discrete event simulation and the methodology followed in carrying out this work.

3.2 Simulation Modelling

An ideal network should be capable of providing service successfully to an infinite load, with the minimum service time. However, in real life networks,

[1]Multiple events can be identified when all of them have the same activation time.

when the load increases, the service becomes unreliable (not all requests served successfully) and the service times increase.

When the load increases gradually from zero to infinite, then the number of requests served successfully increases until it reaches a point where any further increments in the inserted network load do not significantly affect the successfully served requests. This is known as the saturation point of the network and a network can be said to perform well when the saturation point is reached only under high loads. Figure 3.2 presents the general profile of the successfully served requests versus the load graph. The position of the saturation point is affected by all the network layers, starting from the application layer down to the physical layer. Different layer configurations are shown in Table 3.2. The following sections describe the assumptions and configurations used in the simulation model associated with this study.

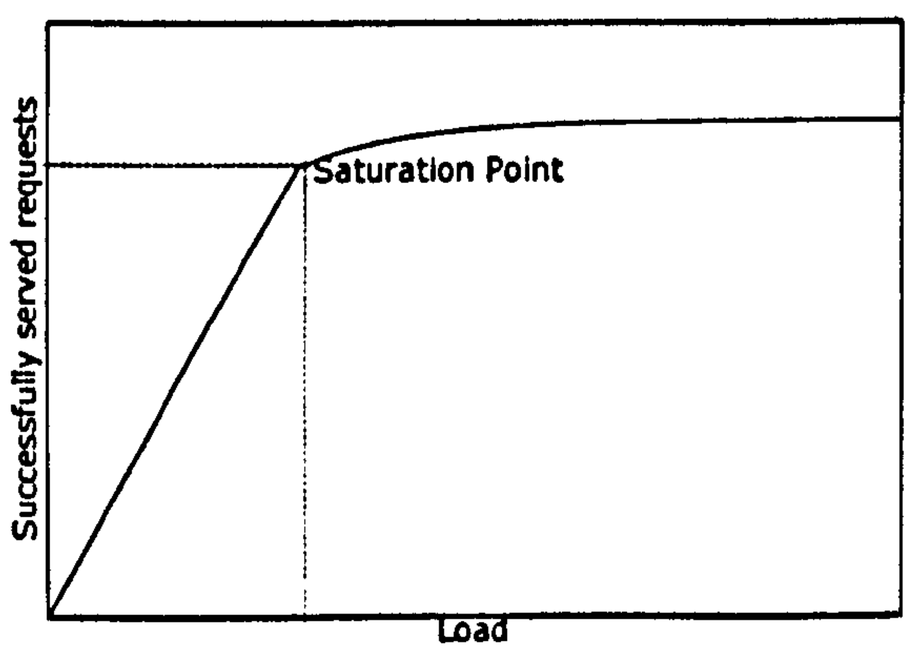


Figure 3.2: Network Saturation.

Application Layer	CBR, EXP, etc.
Transport Layer	TCP or UDP
Routing Protocol	Reactive or Proactive / Single-path or Multi-path, etc.
MAC	TDMA, CSMA/CA, etc.
Physical Layer	Frequency, Modulation Scheme, etc.

Table 3.2: Parameters affecting the performance of the network.

3.3 NS-2 - Packet Level Simulation

The tool used in this study is NS-2 [5] discrete event simulator, which is based on C++ and Tcl/Tk. Originally developed by UC Berkeley University for simulating wired networks, NS-2 has been significantly enhanced by the open source community and its current release includes wireless network support provided by the CMU Monarch extensions [45]. NS-2 uses OTcl to provide an easy and configurable interface, while computation intensive operations performed in a compiled language (C++). Hence, NS-2 combines the advantages of interpreted languages (easy configurability) and compiled languages (speed for computation intensive operations).

NS-2 is a packet level simulation which means that it does not handle events on an individual bit basis but instead in a group of bits (packets). NS-2 provides a full scale simulation platform, extending from the data link to the application layer.

The NS-2 simulation process adopted for this work is summarised in Figure 3.3. Initially, the global simulation parameters are set; propagation model, antenna characteristics, wireless interface configuration, field dimensions and simulation time.

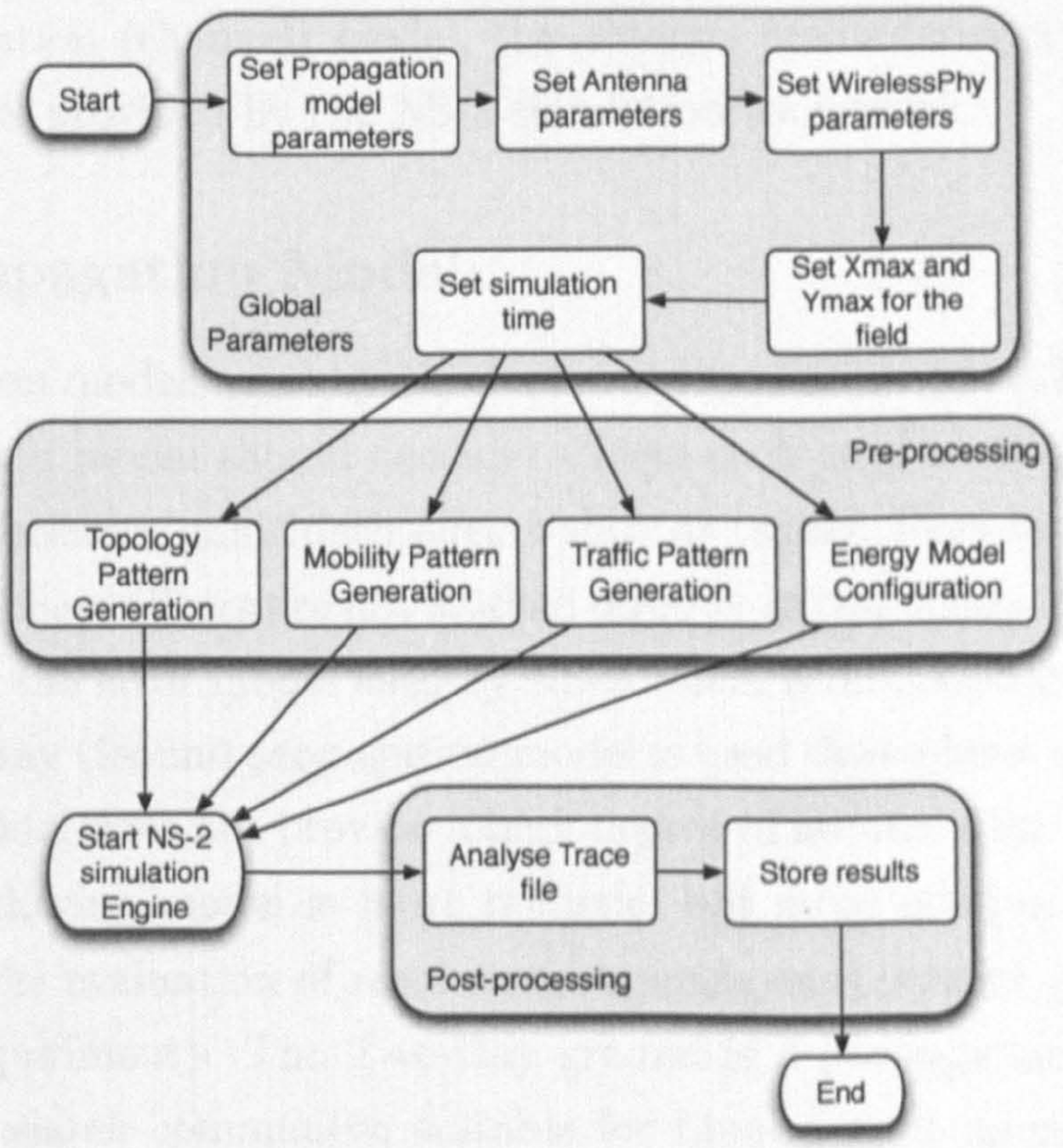


Figure 3.3: NS-2 simulation process.

The next stage on the simulation process is the pre-processing stage, where the network topology, mobility pattern, traffic pattern and energy model are configured. All these are fed to the NS-2 engine and the output is a trace file with all the events occurred during the simulation. Finally, the trace file is passed to a set of post-processing scripts to analyse and store the results. When sufficient independent repetitions of the simulation are performed, the stored results are processed to generate the desired statistics and plots.

The chapter discusses and justifies the selection of the global parameters, the pre-processing and post-processing stages. In addition, further discussion of parameters and assumptions imposed by the NS-2 framework such as the error model and the energy model, as well as the pseudo-random number generators is made. Finally, the chapter proposes a multipath routing scheme and its model is verified using a special case scenario. Subsequently, this scenario is used as benchmark to validate the methodology followed when determining system statistics.

3.4 Physical Layer Modelling

This section discusses the physical layer model used throughout this work, such as the propagation (channel) model, the antenna characteristics and describes the error model provided by the NS-2 simulation framework.

3.4.1 Propagation Model

The propagation models used by NS-2 are described in section 2.2. A more realistic propagation model should consider effects such as shadowing, joint effects of multiple interferers, multipath effects such as fading, Doppler, delay spread, etc. Although these effects are not studied directly in this thesis, their behaviour is captured by the error model used by NS-2 which is discussed in Section 3.4.2.

The Two-Ray Ground propagation model is used throughout this study. The Free Space Model does not provide a high degree of accuracy for long distances, while the shadowing model is more realistic, but more computation intensive and restricts the evaluation of results to a specific environment (such as indoor or outdoor experiment). The Two-Ray ground is a propagation model widely used in the research community, suitable for Line-of-Sight microcells in urban environments [46], and its use in mobile ad-hoc networks is justified by the environmental similarities such as low transmit power and low antenna height.

Although Two-Ray ground is not the most accurate propagation model, it provides a common "floor" for the performance comparison of routing protocols. The effect of an inaccurate propagation model will affect evenly all routing protocols under comparison; therefore the relative metrics will remain the same. Finally, the Two-Ray ground model was selected in order to make the results obtained from this study comparable with the work of other researchers who have adopted the same approach.

The antenna type used in each node is an omni-directional antenna with coverage area equivalent to a perfect circle. The transmitter antenna gain and the receiver antenna gain are set equal to unity and the system loss coefficient is also unity. The antennas of the transmitter and the receiver are placed at a height of 1.5 metres above the ground.

Finally, NS-2 has the ability to set the antenna position based on an offset of the node's actual position. In this work, the X and Y offsets are set to zero, i.e. the antenna is positioned exactly where the node is located assuming that the node does not have a surface and can be represented as a point in space. The simulation parameters are summarised in Table 3.3.

Parameter Type	Parameter Name	Value
Antenna/OmniAntenna	X_	0 metres
Antenna/OmniAntenna	Y_	0 metres
Antenna/OmniAntenna	Z_	1.5 metres
Antenna/OmniAntenna	Gt_	1
Antenna/OmniAntenna	Gr_	1

Table 3.3: Antenna parameters.

The attenuation of the transmitted signal can be calculated for every distance. Figure 3.4 presents the signal attenuation (in dB) versus the distance (in metres) graph, for frequencies 914 MHz and 2,450 MHz. Two propagation models considered; the Free Space (Friis) and Two-Ray ground. From the graph of Figure 3.4, it can be easily seen that when the frequency increases, the attenuation increases. Moreover, it is noted that the reception power for the Two-Ray ground model calculated using the free space Equation 2.1 for small distances; when the distance d increases more than the cross over distance d_c the Equation 2.2 is used. The switch between equations is realised at points A and B for the frequencies of 914 MHz and 2,450 MHz respectively.

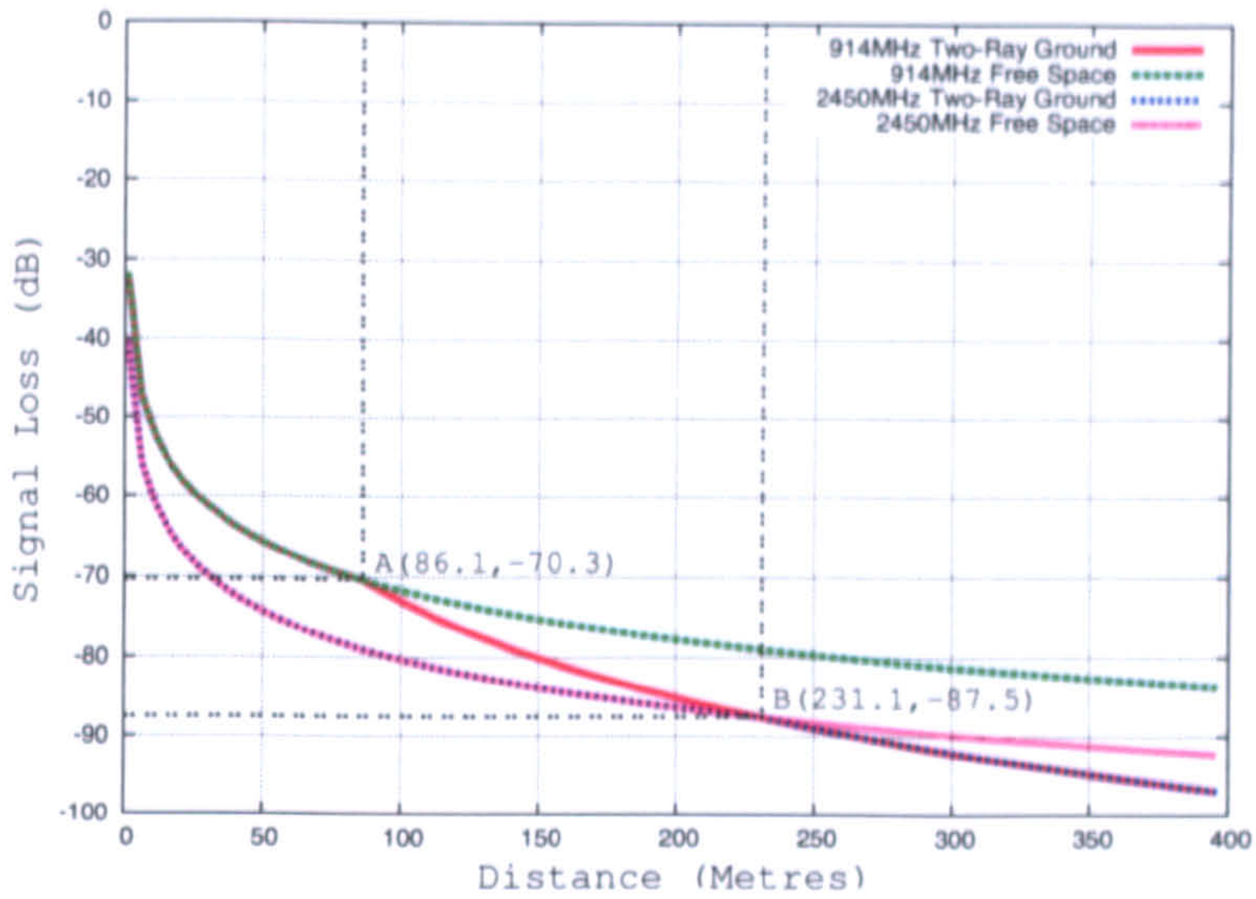


Figure 3.4: Signal attenuation for Free Space and Two-Ray ground propagation models for frequencies of 914 MHz and 2,450 MHz.

3.4.2 Error Model

In real transmissions there is always a chance that the receiver will receive the transmitted information in error. In digital communication systems, errors result from an inability of the receiver to distinguish symbols. For example, in a binary system; the receiver detects zeros as ones and the other way around. This happens due to the attenuation and distortion that the signal suffers during its propagation through the channel. The metric used in communication systems to describe the quality of the signal is the Signal-to-Noise Ratio (SNR) or its variants such as Signal-to-Noise and Interference Ratio (SNIR). SNR is the ratio of the received signal strength over the noise strength, while SNIR is the ratio of the signal strength over the summation of the noise and interference strength.

The reliability of a communication channel is often expressed in terms of the Bit Error Rate (BER), which is directly associated with the SNR, data rate, modulation scheme in use, etc. The BER performance is crucial since it is a good metric that describes the probability of a frame being received in error. Therefore, if a frame is long, it has higher probability of being received in error than a shorter frame. Consequently, the Frame Error Rate (FER) is directly affected by the frame length. Of course, there are commonly used techniques which are capable of recovering from a specified number of errors; namely Forward Error Correction (FEC). FEC recovery is achieved by introducing some redundancy in the frame. Such techniques can be preferable even if the number of bits transmitted is increased because a complete frame re-transmission is

avoided.

Transmission errors and how these are handled are key components of a network simulation. The ability to detect errors and the reaction to such errors has a critical impact upon the observed network performance. The mechanism which introduces transmission errors is referred to as the Error Model. Simulation packages used in the research community typically adopt two error models; BER based or SNR Threshold (SNRT) based. The former makes a decision on whether a frame is received in error based upon the BER (which is calculated from the SNR and the modulation scheme in use) and the length of the frame. This model is realistic but it is not perfect, because the noise introduced by the channel is a true random process and the selection of pseudo random generator is crucial. Of course, a true random number generator (RNG) does not exist; that is the reason that such generators are usually called pseudo-random number generators (see Section 3.6). In order to save computation power, the BER models generally used in the simulation packages, calculate the BER every time the SNR changes. Each frame then is divided into segments and, only if all the segments of the frame are received error free, the whole frame is assumed to be error free.

In the SNR threshold model, the SNR of the received frame is compared directly to a threshold value (for the whole duration of the transmission) and the receiver base on this comparison decides on whether the frame is received error free.

Of the two models, the BER is considered to be more realistic and more accurate. However, the SNR threshold model requires less computation and can be a good abstraction when the frame length is long. In NS-2, the SNRT error model is used and the rest of this section describes the configurations used for the radio interface and thresholds within the SNRT error model.

The communication frequency in the model is selected to match the operation of IEEE 802.11 (as provided by NS-2); essentially 914 MHz. When a frame is received, the receiver compares the received signal strength with the Receiver Threshold (RxThreshold) and, if greater than this threshold, the frame reception is successful (Figure 3.5(a)). If the received signal strength is below RxThreshold and above Carrier Sense Threshold (CSThreshold), the transmission can be detected but the frame is received in error and hence discarded (ignored) (Figure 3.5(b)). If the received signal strength is below CSThreshold then the receiver does not detect the transmission at all (Figure 3.5(c)).

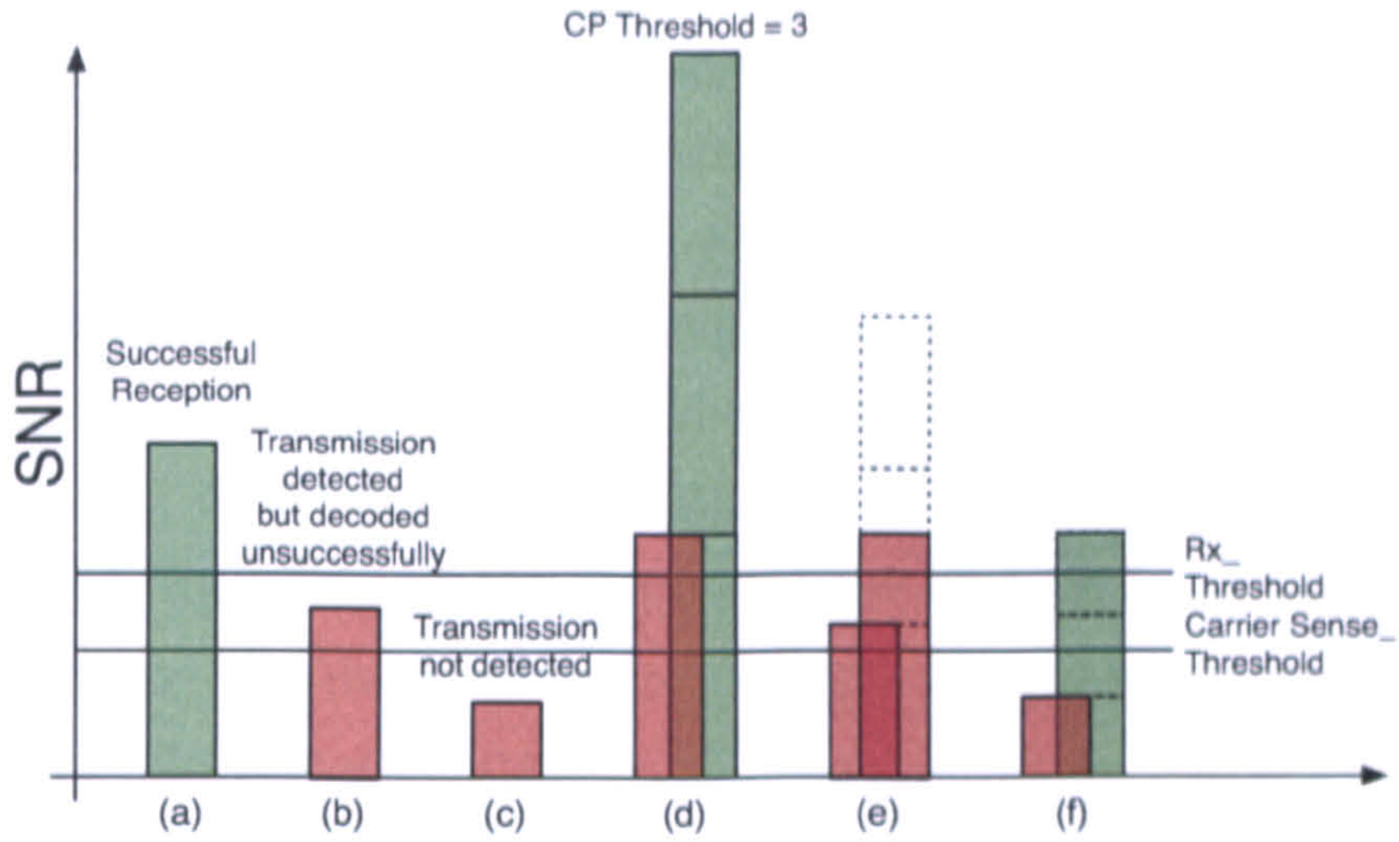


Figure 3.5: Demonstration of the SNR Threshold model operation.

Finally, when multiple concurrent transmissions occur, the receiver will decode the strongest signal only if its strength is greater by Capture Phenomenon Threshold dbs (CPThreshold) stronger than the weak signal. Let a frame "A" being received by a node with duration $[0, T]$ seconds. Also consider that another frame "B" is transmitted during the $[0, T]$ interval. If frame "B" reaches the receiver with power that is CPThreshold dBs below the received power of the first frame "A"; frame "A" will be decoded successfully and frame "B" will be ignored. Otherwise both frames will collide and, consequently, any attempt to decode these signals will be unsuccessful (Figure 3.5(d)).

3.5 Energy Model

Unless otherwise stated, the energy model used is the one provided by NS-2 simulator. There are four radio states considered by NS-2: transmit, receive, idle and sleep. The energy consumed in each state is given by the equation

$$E' = E - P_s \cdot t \tag{3.1}$$

Where E' is the node's remaining energy after a frame is transmitted, E is the current node energy, P_s the power consumption at the specified state (transmit, receive, idle, sleep) and t is the duration that the device remains in that state.

Such a linear energy model suits this study because the key focus point is the performance of the network lifetime as whole rather than from the prospective of

an individual device. Other battery models which model more closely physical reality are non-linear. When studying the performance of an individual device, battery discharge should follow non-linear curve and consider effects such as Rate Capacity Effect and Recovery Effect [47, 48, 49, 50].

The idle power consumption is assumed to be 0 and does not affect the conclusions drawn in this thesis since the remaining energy is subtracted equally for all nodes when they are in an idle state. The sleeping state is also disabled since this work is focussing on the network layer and exploitation of sleep state is an issue affecting the data link layer. All the compared routing schemes are affected equally by these assumptions.

Table 3.4 summarises the NS-2 energy model configuration.

State		Consumption (Watts)
Transmit	(P_t)	0.660
Receive	(P_r)	0.395
Idle	(P_i)	0
Sleep	(P_s)	N/A

Table 3.4: Energy Model parameters.

3.5.1 Initial Node Energy

Typically the battery capacity is measured in milliAmpere-hour (mAh). However, mAh is not a direct measure of energy like Joule (J) and Watt-hour (Wh)[2]. In fact mAh is a unit of electrical charge and 1 mAh is equal to electrical change of 3,600 Coulombs (C). A typical AA battery has 2,000 mAh and provides 1.5 Volts (V). The battery capacity in Wh is:[3]

$$2\ Ah \cdot 1.5\ V = 3\ Wh = 10,800\ J \tag{3.2}$$

There are two different initial node energy configuration scenarios considered in the experiments. In the first scenario the initial node energies are uniformly distributed in the interval $[E_{min}, E_{min} + E_0]$ while in the second scenario the

[2] 1 Wh=3,600 J

[3] Note that simulation runs with lower initial energy values otherwise the length of the simulations will be very high. Discussion on appropriate initial energy can be found in Section 3.8.3.

initial node energies are all equal. The flowchart of Figure 3.6 illustrates the process of setting up the initial node energy. For fixed initial energy $E_0 = 0$.

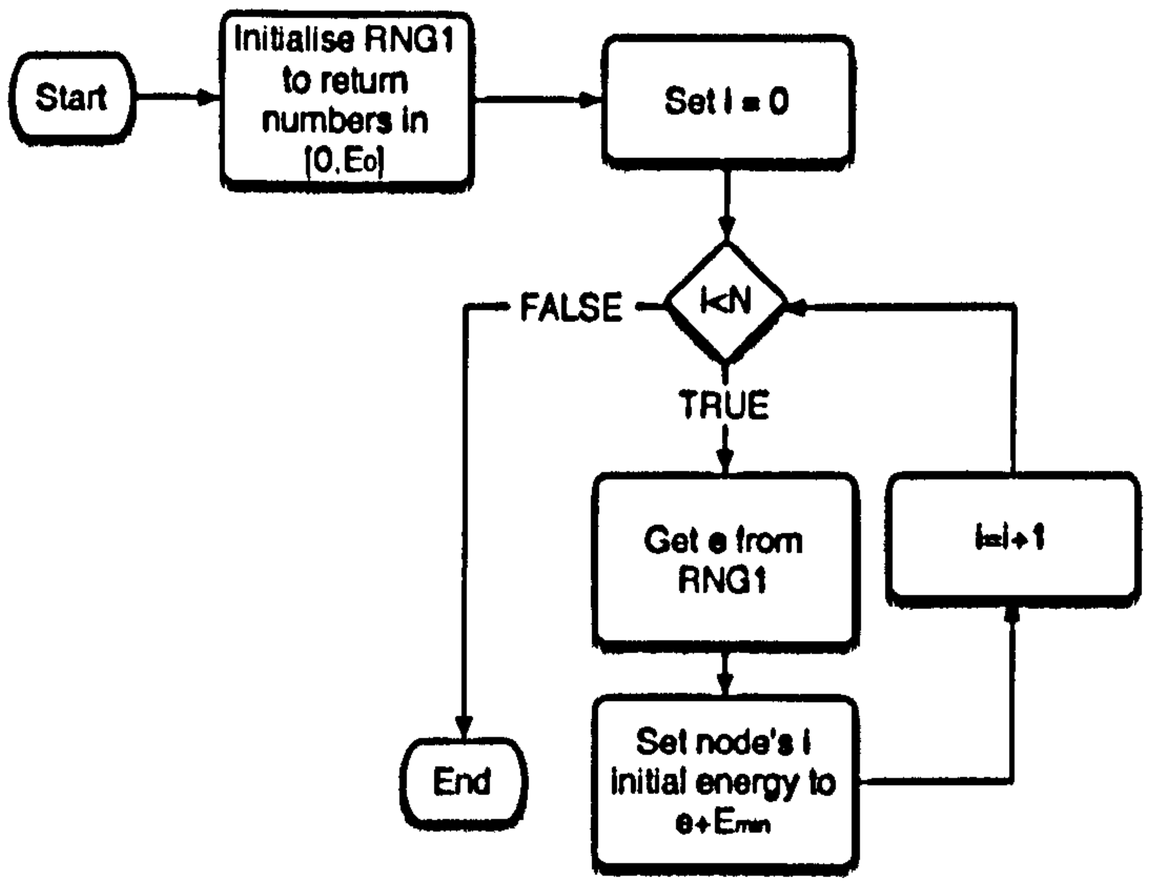


Figure 3.6: Energy initialisation flowchart.

3.5.2 Pre-transmission handshaking energy consumption

This section calculates the energy consumed by a node involved in a successful pre-transmission handshaking of IEEE 802.11 as described in Section 2.3.7.

The transmitting node is involved in the following events when a packet is transmitted to the next hop.

- Transmit RTS
- Receive CTS
- Transmit DATA
- Receive ACK

The RTS, CTS and ACK lengths, excluding the Physical Layer Convergence Protocol (PLCP) header, are 24, 20 and 20 bytes respectively according to the IEEE 802.11 specification [9]. The PLCP header is 24 bytes long and transmitted at basic rate. Then the total control frame lengths and their notations are $L_{RTS} = 44$ bytes for RTS frames and $L_{CTS} = L_{ACK} = 38$ bytes for CTS and ACK frames.

When a packet is created in the application layer, it is subsequently passed to lower layers of the OSI stack (Section 2.3) and additional headers are added to the packet which effectively increases packet length. The process is known as packet encapsulation because headers are added in front of the packet as the

packet travels the OSI stack downwards and the headers are removed as the packet travels upwards the stack. Therefore, the length of packet transmitted by the wireless interface is longer and its length can be calculated as described below.

Since the key area of the study is the routing layer, it is assumed that the length of the generated packets (L_{app}) include the User Datagram Protocol (UDP) header. Additionally, it is assumed that the packet is routed using DSR with flowstate disabled. The DSR packet header is variable and it depends upon the path length over which the packet is to be routed. If the path length is H hops long, then the total length of the packet including IP and IEEE 802.11 headers is:

$$L = L_{app} + L_{IP} + L_{STD_{DSR}} + 4 \cdot (H - 1) + L_{802.11} \tag{3.3}$$

The $L_{IP} = 20$ bytes is the length of the IP header and $L_{STD_{DSR}} = 4$ bytes is the length of the standard (constant) DSR header. The term $4 \cdot (H - 1)$ signifies the bytes required for appending the path; 4 bytes for each node address. $L_{802.11} = 52$ bytes is the IEEE 802.11 header including the PLCP header length[4]. For example, when a packet of 500 bytes (including the UDP header) is routed over a path of 5 hops, the total packet size transmitted by the wireless interface is 592 bytes long.

Assuming that the power consumptions when the network interface are at receive and transmit states are P_{CR} and P_{CT} respectively. Then the energy consumed $E_{T_{RTS/CTS/DATA/ACK}}$ is:

$$E_{T_{RTS/CTS/DATA/ACK}} = \frac{(L_{RTS} + L) \cdot 8 \cdot P_{CT} + (L_{CTS} + L_{ACK}) \cdot 8 \cdot P_{CR}}{1,000,000} \tag{3.4}$$

At the receiving node, the following events occur:

- Receive RTS
- Transmit CTS
- Receive DATA
- Transmit ACK

Following the same method as before, the energy consumption of the receiving

[4]Note that the PLCP header length is 24 bytes and it is transmitted at the basic rate while the rest of the packet can be transmitted in higher data rate. These calculations assume that the data rate is equal to the basic rate as explained in Section 3.7.

node $E_{R_{RTS/CTS/DATA/ACK}}$ is:

$$E_{R_{RTS/CTS/DATA/ACK}} = \frac{(L_{CTS} + L_{ACK}) \cdot 8 \cdot P_{CT} + (L_{RTS} + L) \cdot 8 \cdot P_{CR}}{1,000,000} \quad (3.5)$$

For the 500 bytes packet and for $P_{CT} = 0.660$ Watts and $P_{CR} = 0.395$ Watts the energy consumed for transmitting and receiving is $3,598.24 \times 10^{-6}$ J and $2,411.04 \times 10^{-6}$ J respectively.

3.6 Random Number Generators (RNGs)

Random number generators are used extensively in simulation systems and therefore it is important that the quality of the Random Number Generator (RNG) is sufficient. Poor random number generators can lead a simulation study to generate bad results and invalid conclusions [43, 44, 51]. This section describes the random number generators used throughout this work.

The first stage in ensuring that the random number generators have sufficient performance is to test their independence and uniformity. The uniform random number generators, subsequently, can be used to generate numbers of any required distribution.

The random number generators should really be called pseudo-random number generators.

> "The term pseudo implies that the very act of generating random numbers by a known method removes the potential of randomness" [51].

This quotation refers to the fact that, in the strict sense, the sequence of numbers produced by the generator are repeatable and predictable for the same input seed values, but the numbers inside the sequence have the appearance (from a statistical perspective) of being random. In other words, if the pseudo-RNG is "seeded" with the same seed an identical sequence is generated, but the numbers of the sequence themselves exhibit the statistical properties of random numbers.

A pseudo-random number generator must satisfy the following properties [44, 51] in order to consider it as usable.

1. Generated numbers must be uniformly distributed.
2. Generated numbers should be ideally continuous-valued instead of discrete-valued. A good RNG shall be able to provide numbers from as large set of values as possible.

3. The mean of the generated numbers should not be too high or too low; for a uniform random number generator [0,1] the mean should be 0.5.
4. The variance of the generated numbers should not be too high or too low.
5. The generated numbers must be independent. This means:

   (a) Numbers should not be correlated but independent.
   (b) Numbers should not be successively higher or lower than adjacent numbers.
   (c) The pseudo-RNG should not generate several numbers above the mean followed by several numbers below the mean.

The random number generator used in NS-2 simulator is MRG32ka proposed by L'Ecuyer [52]. This random number generator is extensively used by the research community. L'Ecuyer's RNG provides 1.8×10^{19} independent streams and 2.3×10^{15} sub-streams. Each sub-stream has a period of 7.6×10^{22}. The period of the entire generator is 3.1×10^{57} and adheres the above criteria.

The second RNG used outside the NS-2 framework is Mersenne Twister (MT) proposed by Matsumoto [53]. Theoretical tests on randomness are presented in [54, 55, 56]. Mersenne Twister is also widely used in the research community and it has period of $2^{19,937} - 1$. In this thesis, the MT is used extensively to generate random topologies, communication patterns and mobility scenarios and adheres all the five criteria mentioned above.

Finally, random numbers uniformly distributed between 0 and 1 can be used to generate random number of any desired distribution. The most common method to achieve this is the inverse transform sampling method (also known as the probability integral transform) [44]. To generate numbers of any distribution the following steps are taken.

1. Generate a random number $Y \in [0, 1]$ using the standard RNG.

2. Select the Cumulative Distribution Function (CDF) of the desired distribution. For example, for exponential distribution this function is: $F(x) = 1 - e^{-\lambda x}$, where $1/\lambda$ is the mean of the exponential distribution.

3. Solve the equation with unknown x, i.e. $x = -\frac{ln(1-F(x))}{\lambda}$, to find the Inverse Cumulative Distribution Function ($F^{-1}(x)$).

4. Substitute $F(x) = Y$, where Y is the random number obtained in step 1. x comes from the desired distribution.

5. Repeat the above steps to generate more numbers from that distribution

This general approach can be used to generate numbers of any distribution such as Weibull, Pareto, Normal, etc. Figure 3.7 illustrates the process of generating exponentially distributed random numbers. The mean of the exponential distribution is unity.

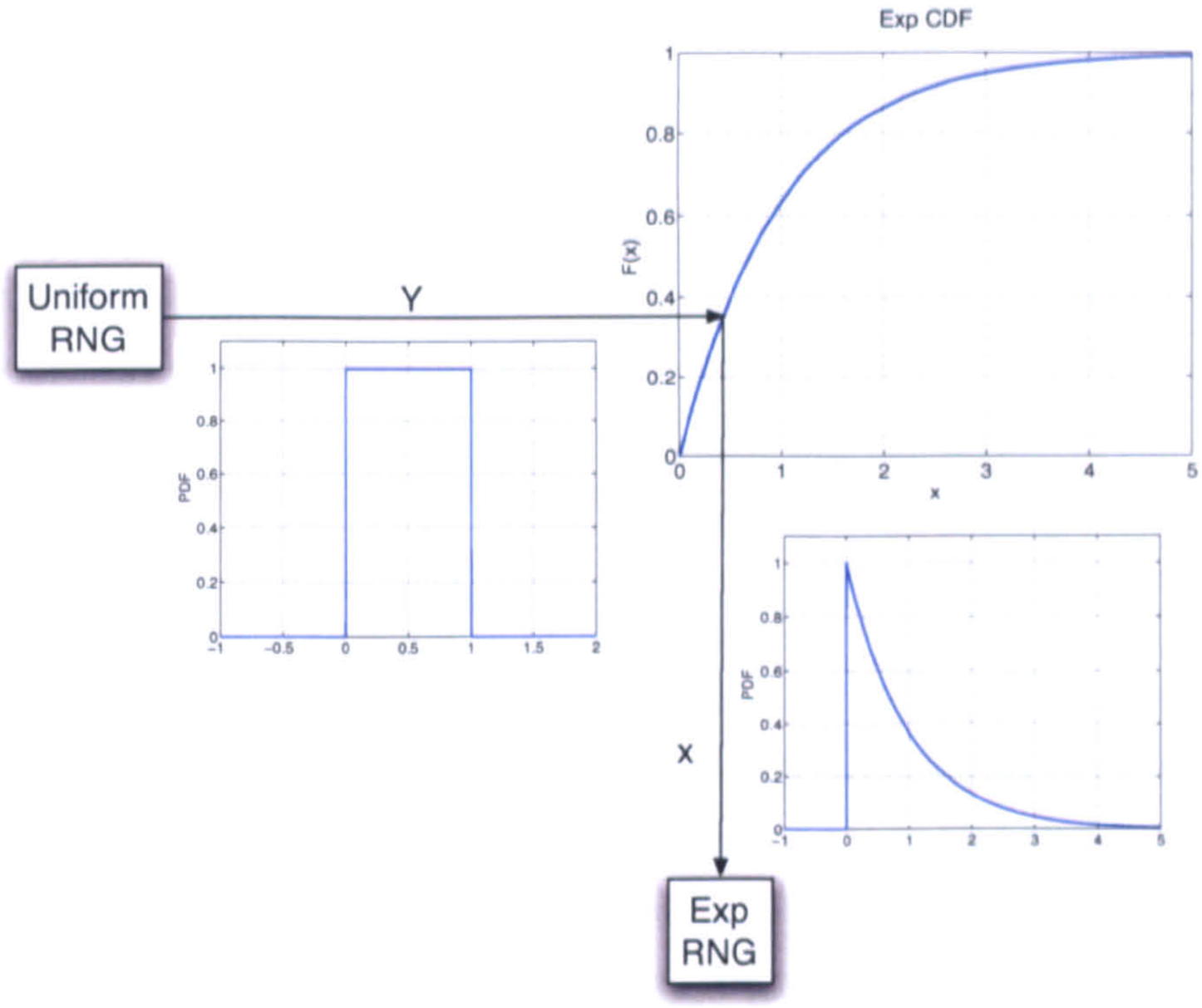


Figure 3.7: Generating random numbers with inverse transform sampling.

3.7 Wireless Network Interface

The Medium Access control protocol used in this study is the IEEE 802.11 provided with the NS-2 distribution version 2.28. The IEEE 802.11 standard defines the "basic rate" of 1 Mbps for the transmission of control frames such as RTS, CTS and ACK, while data frames can be transmitted at higher rates. For example, IEEE 802.11b defines data rates of 1, 2, 5.5, 11 Mbps with the actual rate depending upon the quality of the link. In order to avoid unnecessary complex calculations in this work, the data rate is fixed to 1 Mbps which minimises complexity associated with differentiation between control and data frames. The modulation scheme used is BPSK in 914 MHz with DSSS and CCK of 11 chips. For more details regarding the physical specification of the IEEE 802.11, refer to section 2.3.2.

The configuration parameters used for the wireless interface mimic the Lu-

cent WaveLAN DSSS network card (Table 3.5).

Parameter Type	Parameter Name	Value
Phy/WirelessPhy	freq_	914e+6 Hz
Phy/WirelessPhy	Pt_	0.281838 Watts
Phy/WirelessPhy	RXThresh_	3.41828e-8 Watts
Phy/WirelessPhy	CSThresh_	1.559e-11 Watts
Phy/WirelessPhy	CPThresh_	10
Phy/WirelessPhy	L_	1

Table 3.5: Wireless interface parameters.

3.8 Routing Protocol

The main aspects studied in this thesis is energy awareness and load distribution issues in MANETs. The techniques developed in the thesis are based on Dynamic Source Routing (DSR) described in section 2.4.4. DSR was selected from among other protocols in the research arena because it is a reactive routing protocol and reactive routing protocols tend to perform better in dynamic environments compared to proactive counterparts [57, 58]. The NS-2 simulator provides a DSR implementation and this allows a more focussed study on the key areas of the thesis; load distribution and energy awareness. Among the other routing protocols implemented in NS-2, DSR was selected because when flow-state is disabled, the end-to-end path is appended to the packet's header. This DSR feature permits more flexible monitoring of the packet and more flexible tracing of the path, when it is propagated from its source to destination. The techniques developed are not DSR specific and can be applied to any routing protocol with the appropriate adjustments.

Reactive routing protocols such as DSR, find optimal routes based on the shortest path algorithm. The shortest path found is not necessarily the shortest in terms of hop count, but the shortest in terms of the delay incurred for the route discovery packets to reach the final destination. For example, consider the network of Figure 3.8. If node 1 seeks to communicate with node 5, it initiates the route discovery process, by broadcasting RREQ packets as described in section 2.4.4. The shortest path in terms of hop count is the path $A(1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5)$. However the broadcasted packets might reach the destination node

5 through path $B(1 \rightarrow 15 \rightarrow 17 \rightarrow 18 \rightarrow 19 \rightarrow 20 \rightarrow 21 \rightarrow 16 \rightarrow 5)$ first. Such scenario might occur if one or more collisions occur when propagating RREQs over path A or because the congestion along path A is higher than the congestion in path B.

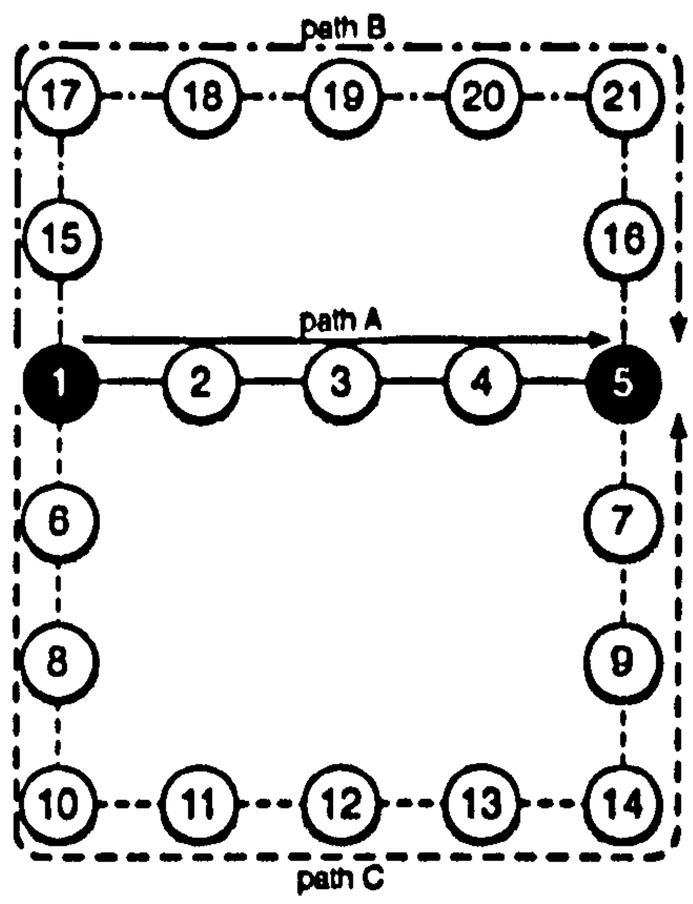


Figure 3.8: Sample three branch network.

Assuming that the RREQ packets reach the destination node through path A first (shortest path in terms of hop count), the destination node generates a RREPLY which is destined to the initiator of the route discovery using the reverse path found within the RREQ. As a consequence, a source-destination pair will continuously use an arbitrary path for their communication, until the path is no longer valid or new shorter path (in terms of hop count) is discovered. In the example described above, node 1 will use path A for all its communications with node 5. Invalidation of paths might occur due to mobility which result in topology changes or because nodes along path A consume all their energy reserves and "die".

When a source-destination pair uses continuously an arbitrary path, the load or energy depletion will occur only to those nodes of the network being members of the arbitrary path, leaving the network with a wide disparity of load and node energy levels. From an energy perspective, this will result in network partitioning and eventually, disconnection of the network. If the nodes in the network "burn" energy more equitably, then the nodes in the "centre" of the network will continue to provide connectivity. In the example described above (Figure 3.8), DSR will result in: 1) exhaustion of the nodes along path A and 2) increased congestion in path A.

In the remaining sections of this chapter, a simple DSR modification is proposed which enables DSR to operate using multiple routes. Multipath routing is discussed extensively in Chapter 4. The remaining sections serve as an example to discuss the evaluation methods used in this thesis.

3.8.1 Load Distribution Algorithm

In order to enable load distribution in the network, the nodes should be aware of multiple routes for every destination. Nodes gain knowledge of the network topology through the route discovery process. However, DSR does not allow the destination node to respond to multiple paths. For this reason it is necessary to modify DSR in the following manner.

Consider the example of Figure 3.8. When the node 5 receives the RREQ packets from path A it will respond to the received RREQ using the reverse path but will also reply to subsequent RREQ received from other paths: B and C($1 \rightarrow 6 \rightarrow 8 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 9 \rightarrow 7 \rightarrow 5$). The destination node (node 5) responds to RREQs immediately after receiving them in order to minimise the delay of the route discovery process. An alternative solution would be to respond to RREQs after aggregating the paths from all route requests, but the delay in discovering a route will increase and is undesirable. As soon as node 1 (source) receives the response RREPLY packets from paths A, B and C it will save these paths into its cache. From now on, when node 1 seeks to transmit a packet to node 5, node 1 will search its cache and it will find 3 routes. The structure and the maintenance of the caches are as follows:

Each node has 3 caches. Upon receiving the RREPY the node caches up to 3 routes pertaining to the same destination. The number of cached paths was limited to 3 in order reduce the memory footprint of the caches and because more paths do not offer additional benefit [59]. This opinion was also validated independently in this thesis and the additional benefit was limited, if at all. Consequently, it was decided to maintain the number of cached paths to less or equal to three. Each cache contains, at most, one entry for every destination. All routes have an associated cost which is proportional to the path length. When all the caches have one route to a destination, and a new route to this destination is discovered, this route replaces the most expensive route among the routes found in all caches. When a node seeks to send a packet across the network, these caches are searched and one route from each cache becomes a candidate. Then, a random selection is performed and a route is chosen according to the

probabilities of each route. The probability of a route i when n routes are candidates is given by:

$$P_i = \frac{1}{C_i} \cdot \frac{1}{\sum_{k=1}^{n} \frac{1}{C_k}}, \quad \text{where } n \text{ number of route candidates} \tag{3.6}$$

The cost of each route C_i is equal to the number of hops and therefore paths with smaller number of hops are more probable than paths with greater number of hops.

Applying this technique to the example in Figure 3.8 and assuming that the node 1 seeks to communicate with node 5, three paths A, B and C are discovered with the cost of each path being 5, 9 and 11 respectively. The probability of node 1 selecting path A, B or C for sending a packet to node 5 is 0.4974, 0.2763 and 0.2261 respectively.

The probability function of equation 3.6 was selected because it is favourable to shortest paths and consequently will maintain lower mean delay and jitter, while the alternative paths will be used to aid the load distribution. Additionally, the probability function is based on the path length and the source node explicitly knows the cost of each path without any additional signalling (overhead) which may be required by other schemes. This is especially advantageous in the ad hoc networks environment where the network capacity is more precious.

3.8.2 Simulation Parameters

The simulations are performed on the static[5] network shown in Figure 3.8 using the NS-2 simulator [5], described in Section 3.3. A Constant Bit Rate (CBR) source is attached on node 1 generating packets for destination node 5. The CBR source creates packets every 0.05 seconds. The packet size is set to 500 bytes. With real CBR services there is a requirement that packets follow the same route and are delivered in order. In this thesis, such a constraint is relaxed as the purpose is to load the network and examine the effects of using each scheme.

The transport protocol used for connecting the source and destination is the User Datagram Protocol (UDP) which does not require transport layer acknowledgements to be send. It has been shown that Transport Control Protocol (TCP) significantly reduces the throughput of the system when it operates in a wireless multihop environments and can be worse when multipath routing

[5]Mobile scenarios are discussed in Chapter 4.

is introduced [60, 61, 62].

The aim of this experiment is to illustrate that the implemented model distributes the load according to the mechanism described in Section 3.8.1. The probabilities of sending a packet over the paths A, B or C are 0.4974, 0.2763 and 0.2261 respectively according to the designed specification. The simulation model should match these probabilities when an infinite number of packets simulated. However, such a simulation is impossible and a more realistic simulation with a results approaching the theoretical calculations will be given if the simulation lasts long enough[6].

The required time to generate 50,000 packets when a packet is generated every 0.05 seconds is 2,500 seconds.

The energy model parameters are set according to Table 3.4 and the following calculations performed to establish a reasonable initial energy value.

Every time node 1 sends a packet to the destination node, node 1 consumes energy for the following events: RTS transmission, CTS reception, DATA transmission, ACK reception. When the packet is forwarded to the next hop node 1 also consumes energy for the following events: RTS reception and DATA[7] reception. Performing the calculation in the same manner as the calculation of Section 3.5.2 the total minimum[8] energy consumed by the node 1 for a single packet is $5,608 \times 10^{-6}$ J. Therefore, to accommodate 50,000 packets the initial node energy should be at least 280.4 J. However, this initial energy calculation does not take into consideration the route discovery process and collisions which might occur and these require additional energy. To be on the safe side, the initial energy value is set to 400 J which is sufficient to accommodate more than 70,000 packets. Additionally, in order to ensure that all the energy of the node 1 is indeed consumed, the simulation time is set to 4,000 seconds (enough to generate 80,000 packets).

3.8.3 Results

A single simulation run is performed to establish if the algorithm distributes the load to the three paths of the network of Figure 3.8 while the original DSR implementation should only use a single path until the nodes along the primary

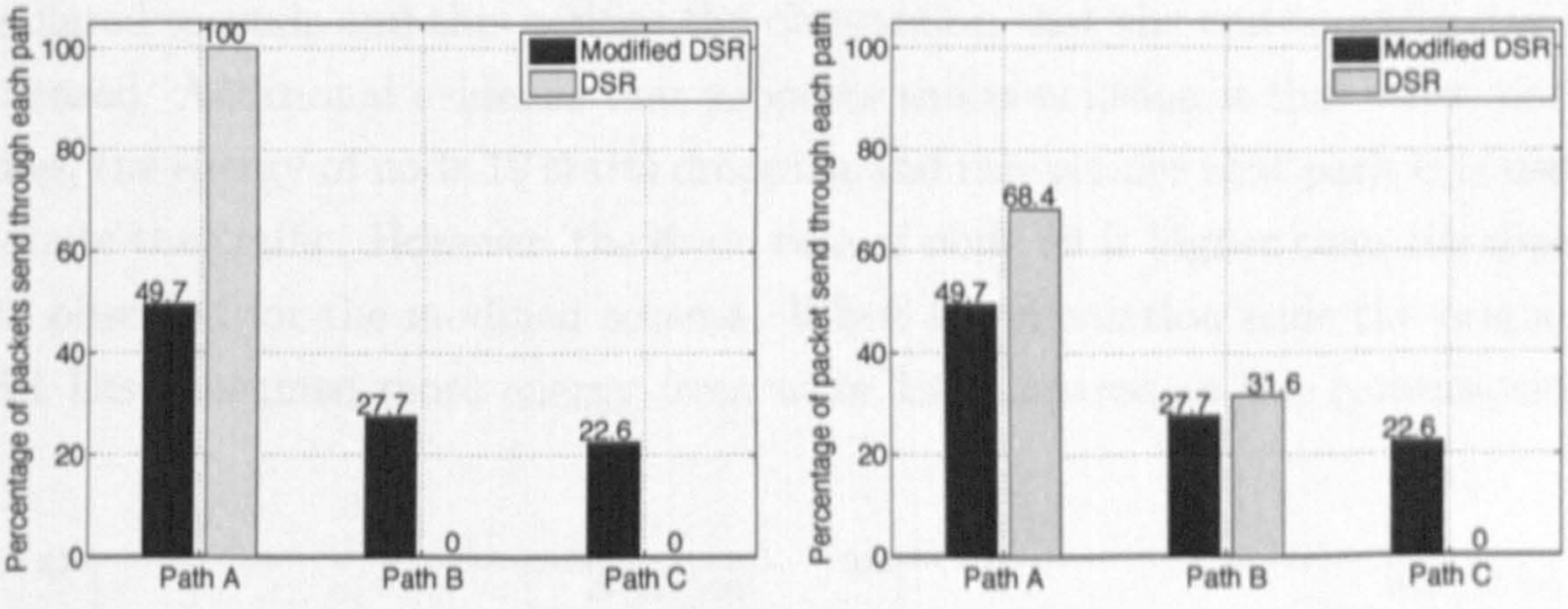
[6]The simulation time was extended until resultant probabilities from the simulation evaluation converge to the design probabilities within 3 decimal points.

[7]Node 1 consumes energy only at these events because the nodes are arranged in a chain topology.

[8]This calculation assumes that all packets are routed through the shortest path which is 5 hops long.

path are exhausted.

The reader is reminded that nodes 3, 19 and 12 are located on paths A, B and C respectively. Probes on these nodes are used to indicate when the paths are invalidated. Figure 3.9 shows how the traffic is distributed when the Modified and Original DSR implementations are in use. Figure 3.9a shows the traffic distribution from the beginning of the simulation until one node is exhausted. Indeed the modified DSR distributes the traffic according to the designed probabilities. Path A serves 49.7% of the packets, path B serves 27.7% and paths C serves 22.6%. On the other hand, the original DSR implementation is using exclusively the shortest path until the path becomes invalid. Figure 3.9b shows the distribution of traffic from the beginning of the simulation until the source node (node 1) is exhausted. At this point, no more traffic is injected to the network. The modified DSR implementation maintains the same load distribution while the original implementation initially uses path A to serve all traffic and when path A is invalidated, it starts using path B, giving an aggregate load distribution of 68.4% for path A and 31.6% for path B, while path C is not used to route any traffic.



(a) First Node exhausted

(b) Source Node exhausted

Figure 3.9: Load distribution for both schemes.

Figure 3.10 shows the end-to-end delay each packet suffers for both schemes. The delay is measured from the time the packet is generated until the packet is successfully delivered to its destination. When node 1 seeks to communicate with node 5; it initiates a route discovery and packets are buffered until the route discovery completes. As soon as a route becomes available, packets are dispatched to their destination. The lack of a path to the destination results in an increased delay for both schemes (Point I in Figure 3.10a and 3.10b). After

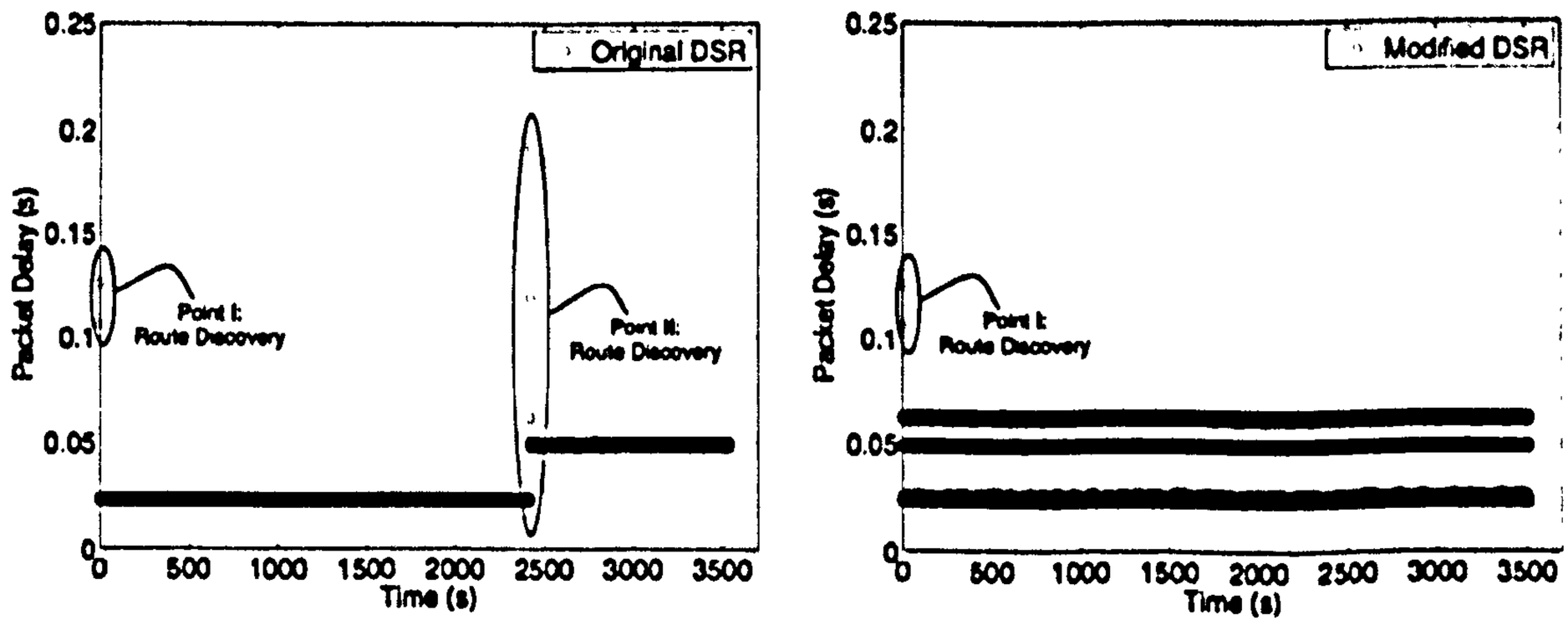
the route discovery is completed, the delay drops because node 1 has cached the route to destination node 5 and the packets are immediately dispatched to their destination.

The original DSR uses exclusively the shortest path until the path is invalidated. Path A is invalidated for the original implementation before 2,500 simulated seconds (Point II in Figure 3.10a). At this point, a new discovery process is initiated and the end-to-end delay is momentarily increased. After the second route discovery is completed the delay falls again but this time the average delay is higher than before, because the path B is longer compared to the primary path A.

For the modified scheme the end-to-end delay varies around 3 values. A single discovery process can be depicted at the beginning of the simulation. After the route discovery process is completed, the end-to-end delay varies around 3 values because packets are routed over paths with different lengths.

Figure 3.11 shows the energy profile of nodes 1, 3, 12 and 19 for both schemes. For the original DSR implementation, the following observations should be noted.

Figure 3.11b shows that path A breaks for the original DSR before 2,500 simulated seconds and this verifies the observation that the end-to-end delay is increased. Additional evidence that supports this conclusion is that, after node 3 dies, the energy of node 19 starts dropping and this verifies that path B is used to route the traffic. However, the drain rate of node 19 is higher than the drain rate observed for the modified scheme. When the simulation ends the original DSR has consumed more energy from node 19 compared to the consumption



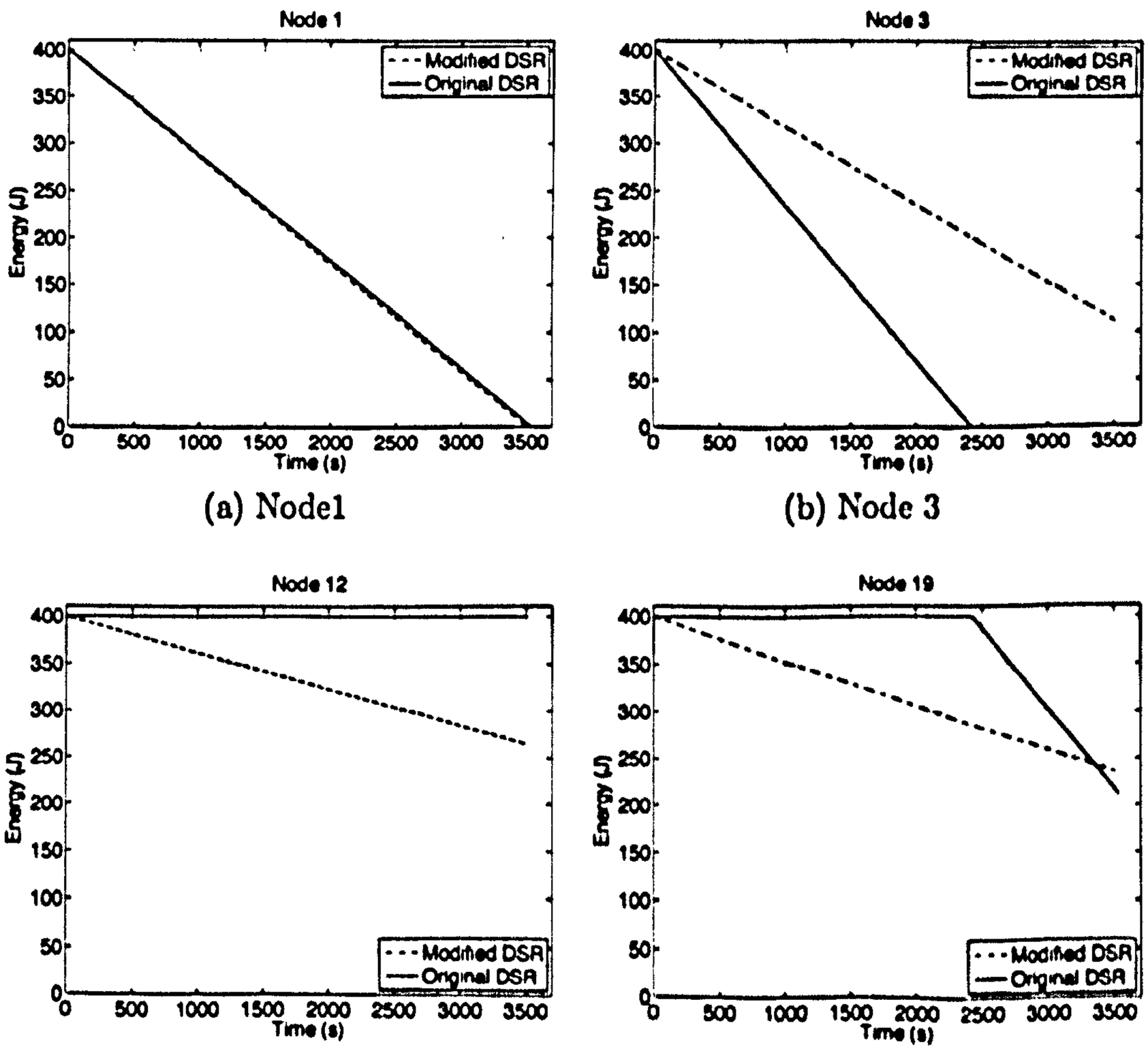
(a) Original DSR

(b) Modified DSR

Figure 3.10: Instantaneous end-to-end delay for both schemes.

made by the modified counterpart. This is because original DSR is aggressively using path B while the modified scheme distributes the load to the alternative paths. Finally node 12 is not used to route any traffic and therefore does not consume any energy. The only energy consumed by node 12 is caused by the route discovery process.

For the modified DSR scheme all paths are used according to the designed probabilities and this can be observed from the different energy drain rates. For example, node 3 has energy drain rate of -82.423×10^{-3}, node 19 of -46.840×10^{-3} and finally the drain rate of node 12 is -38.561×10^{-3}. The ratio of packets route over paths A and B is $\frac{49.7}{27.7} = 1.794$ and the ratio of drain rates for nodes 3 and 19 is $\frac{-82.423 \times 10^{-3}}{-46.840 \times 10^{-3}} = 1.759$. In a similar fashion, the ratio of packets routed over paths A and C is $\frac{49.7}{22.6} = 2.199$ and the respective drain ratios is $\frac{-82.423 \times 10^{-3}}{-38.561 \times 10^{-3}} = 2.137$. Note there is a divergence in drain ratios and load distribution ratios. This divergence is caused by the fact that packets routed over different paths have different sizes due to the variable DSR header caused by



(a) Node1

(b) Node 3

(c) Node 12

(d) Node 19

Figure 3.11: Energy Consumption versus Time graphs.

the variable path lengths. Packets routed over longer paths are longer because they contain more addresses in the DSR header as explained in Section 3.5.2; Equation (3.3). The variable packet length is also evident in the energy drain graph of Figure 3.11a. The modified DSR scheme exhausts node 1 slightly faster than the original DSR implementation at slightly over 3,500 seconds. The increased energy consumption of the modified scheme is also evident to the number of packets node 1 managed to inject into the network. When the original DSR was in use, node 1 injected 70,784 packets and the modified injected 70,283 packets (a ratio of $\frac{70,283-70,784}{70,283} = -0.00712$).

In summary, the modified scheme distributes the load across three available paths and it is more graceful to relay nodes as it does not consume all their energy resources. In a different topology with more connections and higher number of nodes the scheme will reduce path congestion since the load is distributed. The disadvantage evident in this specific topology is that the source node consumes slightly more energy because it uses sub-optimal longer paths. However, when larger topologies with higher traffic loads are considered, the modified scheme reduces congestion and hence collisions. Collisions tend to consume more energy resources because whole packets must be retransmitted and this trades off with the higher header length. These claims are discussed in more detail in Chapter 4.

3.9 Topology Generation

This section describes the topology generator used to create random topologies. The process is illustrated in Figure 3.12. N denotes the number of nodes in the simulated scenario and X_{max}, Y_{max} the dimensions of the simulated field. Two independent pseudo-random number generators were used to pick the co-ordinates of each node. The two random number generators were configured to return values in the intervals $[0, X_{max}]$ and $[0, Y_{max}]$ for the X and Y coordinates respectively.

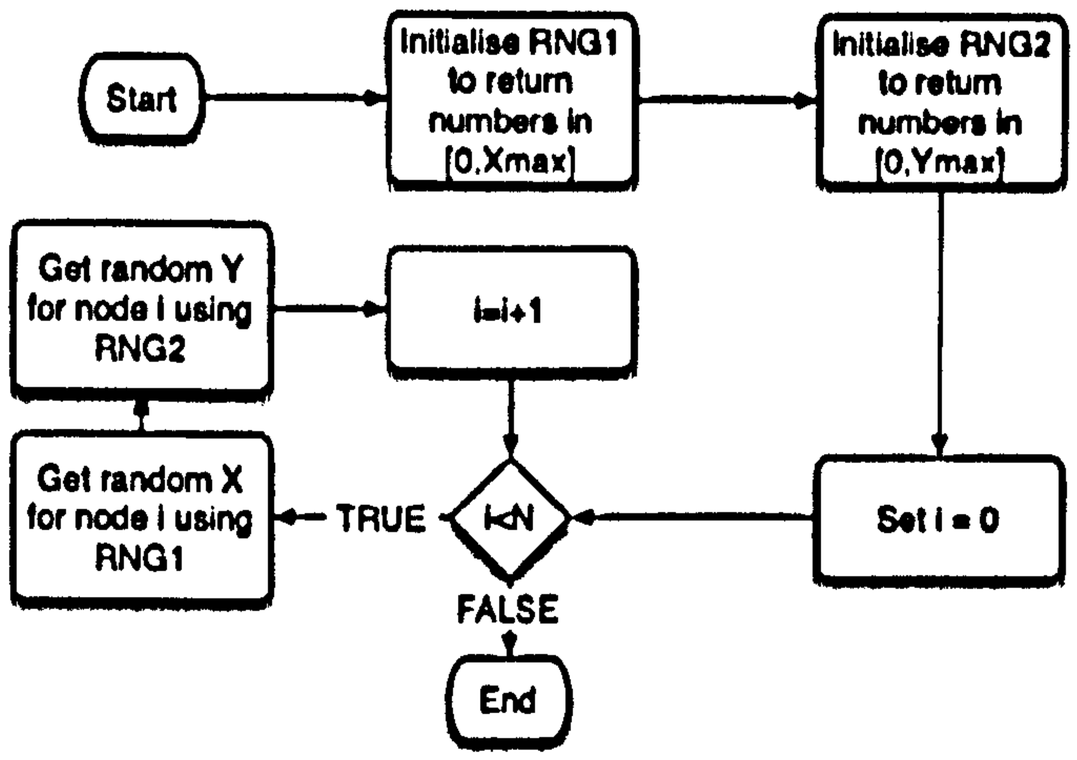


Figure 3.12: Topology generator flowchart.

3.10 Traffic Generator

When large topologies are simulated, it is necessary to create random traffic patterns which will be applied on the given scenarios. The flow diagram illustrating the operation of the traffic generator is shown in Figure 3.13. The traffic generator requires as an input the simulation time (T_s), the number of nodes (N), the number of connections ($NCon$) and the average connection duration (D).

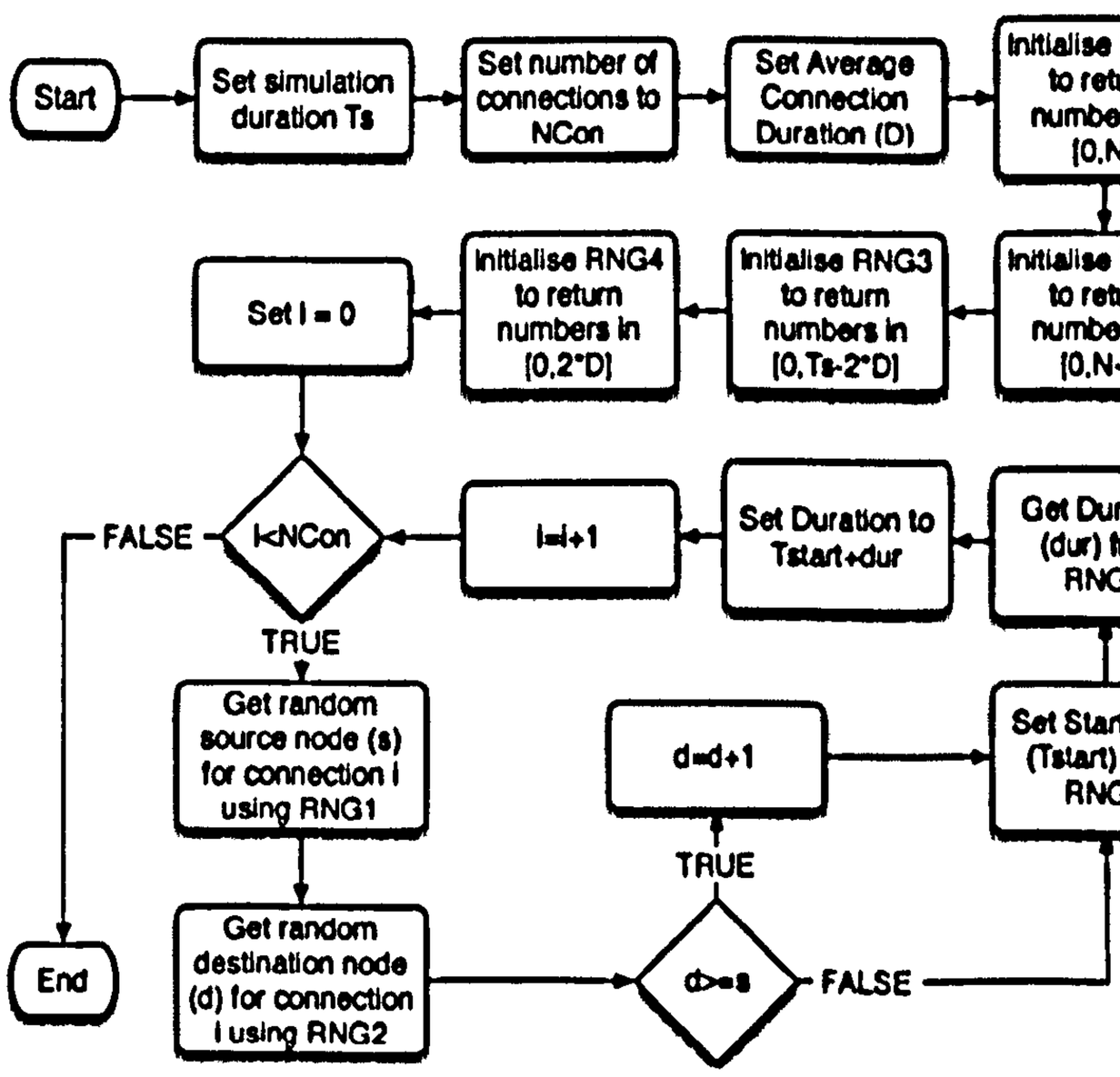


Figure 3.13: Traffic generator flowchart.

The source and destination nodes are selected at random from two independent uniform distributions. The duration of the connections is selected from a uniform distribution $[0, 2 \cdot D]$. Finally, each connection has a start time (T_{start}) randomly selected within the length of the simulation ($(0, T_s - 2 \cdot D)$). A connection can not start after $T_s - 2 \cdot D$ because the connection stop events will exceed the simulated time.

The traffic generator has two transient phases. Transient removal is discussed in Section 3.12.1. The first transient phase is at the beginning of the simulation. For the first transient phase, the initial number of connections is 0 and it ramps up to the average number of connections, where stability is reached. For the duration of the stable region the active connections fluctuate until the second transient phase is reached. The second transitional region is located at the end of the simulation because the all active connections must stop before the end of the simulation is reached. The second transient phase is introduced because the simulation framework provides sanity checks which ensure that there no events scheduled outside the duration of the simulation. More discussions on transients and model stability can be found in Section 3.12.1.

3.11 Mobility

Mobility is a crucial factor in modelling mobile ad-hoc network routing schemes. Random Waypoint mobility model (RWP) [63, 57] is the most commonly used model when evaluating MANETs. In ACM MobiHoc 2002 [64] ten papers considered mobility and nine of them used Random Waypoint mobility.

In the RWP model, nodes are uniformly distributed in a field. Nodes choose a random destination coordinate and move to it with a randomly chosen speed uniformly distributed in $(0, V_{max}]$, where V_{max} is the maximum velocity. When the nodes reach their destination, they pause for pre-defined time and subsequently select a new random destination with a new velocity.

Someone might mistakenly assume that the average velocity when the simulation reaches steady-state is $\frac{V_{max}}{2}$, however this not the case. When the simulation starts the nodes select a random speed from a uniform distribution and the average speed is indeed $\frac{V_{max}}{2}$. However, some nodes select a very low speed towards their destination. These nodes become trapped because of the low speed and it can take very long time to reach their destination. As more and more nodes with moderate or high speed reach their destination they have a chance to select a low speed too. As the simulation progresses the number of trapped

nodes is increasing and the average node speed drops and eventually the nodes become static.

A solution is provided [64] which avoids the speed decay issue. The reason behind the decay issue is that nodes are "trapped" in very long journeys to their destinations and, in order to avoid this, nodes must select their velocity from a uniform distribution in the interval $(1, V_{max}]$. The mean of this uniform distribution is $1 + \frac{V_{max}-1}{2}$, however the average node speed does not stabilise around the mean of the uniform distribution but at a significantly lower speed. However, the average node speed is indeed stabilised.

The mobility scene generator is illustrated in Figure 3.14. In the first part of the mobility scene generator initialisations are performed using the RNGs. After initialisations are complete, Phase A starts. In this phase, the nodes are given a random initial location in the field, a random destination and a random velocity.

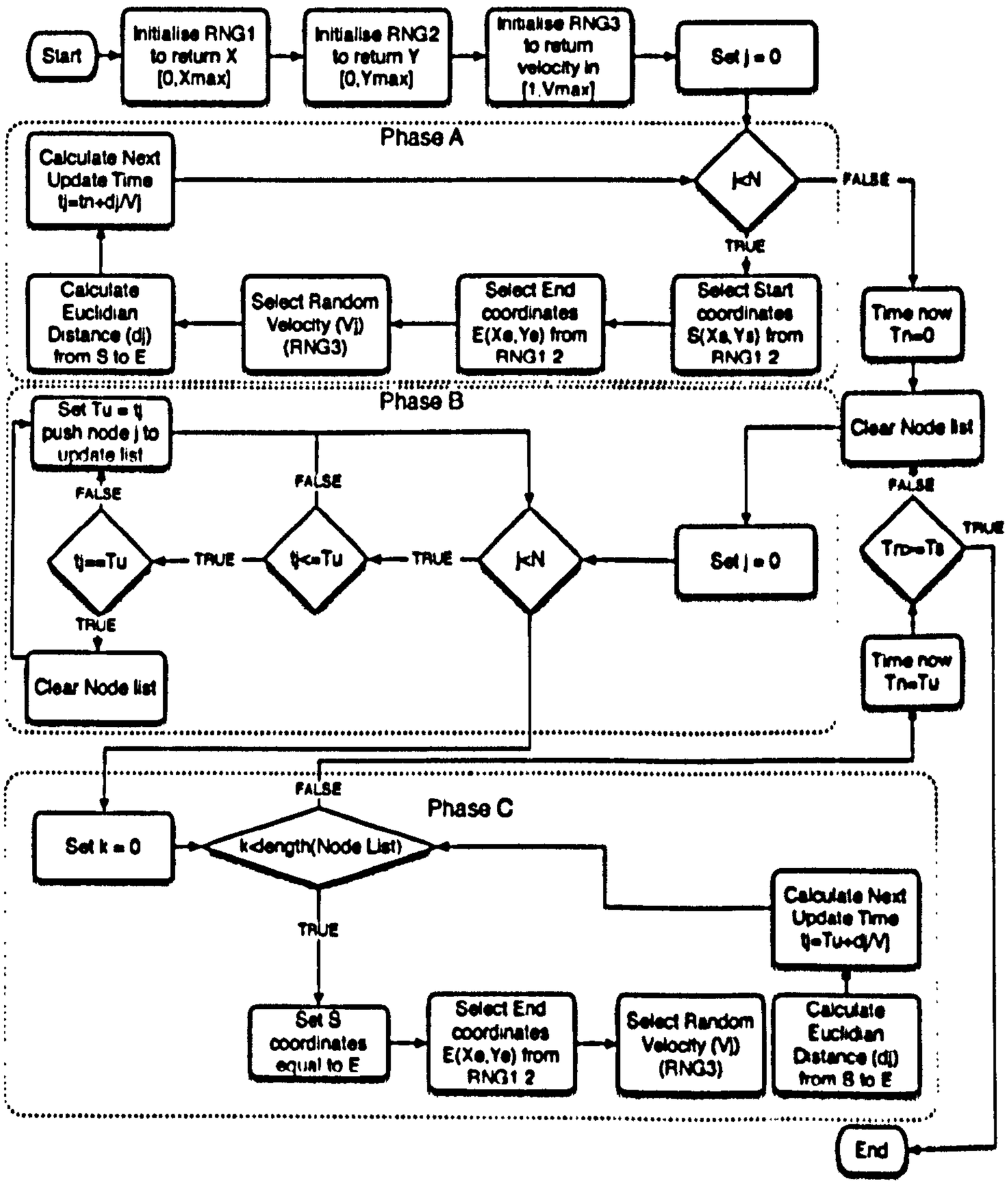


Figure 3.14: Mobility generator flowchart.

Once all these are set in a node's attributes, the final piece of information is the time it will take the node to reach its destination. In phase B, a loop checks which updates must be performed; i.e. which node reached their destination. All the nodes reaching their destination are recorded in a list and get a new velocity and destination in phase C. Finally the generator continues phases B and C until the end of the simulation is reached and the generator exits.

In order to determine if the generator stabilises the nodes' average speed, 50 different scenarios were created. For the modified random waypoint mobility model (M-RWP) nodes select their speed from uniform distribution $(V_{min}, V_{max}]$. V_{min} is set to 1 metres/second and V_{max} to 10, 20 and 30 metres/second. The same set of experiments is performed for the random waypoint mobility model (RWP). RWP selects speeds from a uniform distribution of $(0, V_{max}]$ and V_{max} is set to 10, 20 and 30 metres/second. The pause time for both schemes was set to zero. The simulation duration was 2,000 seconds. In Figure 3.15, the red lines represent the traditional RWP model and the black lines the M-RWP. It is clear that the average node speed decays for the traditional RWP model while the modified counterpart reaches steady-state relatively fast[9]. When the maximum velocity is increased, M-RWP stabilises on a higher average speed as expected. However, the overall average speed is not equal to the mean of the uniform distribution for the reasons explained previously.

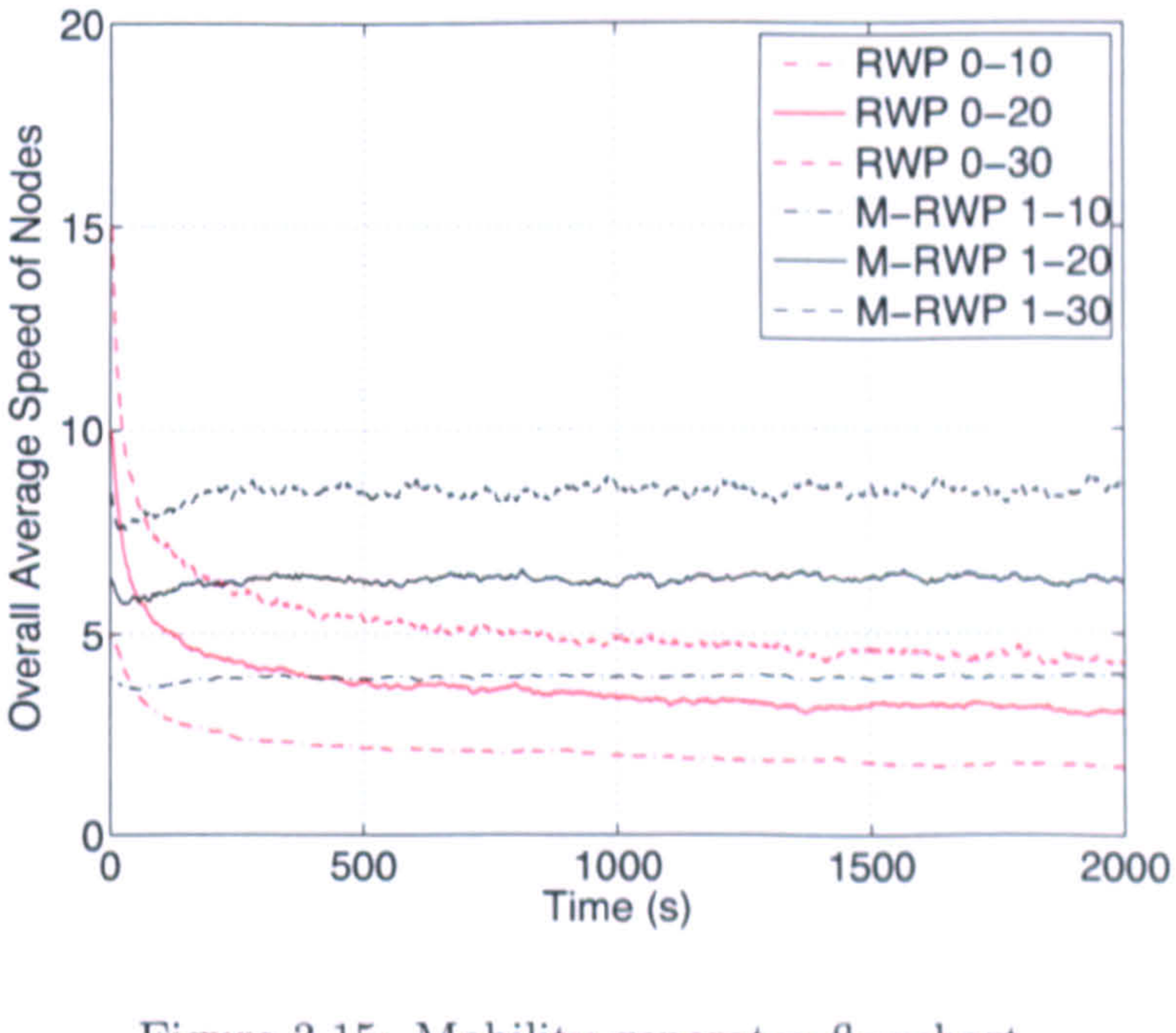


Figure 3.15: Mobility generator flowchart.

[9]The stability is discussed in detail in Section 3.12.1.

3.12 Output Analysis

This section deals with the methods used during the analysis of the generated results in order to guarantee their accuracy and repeatability. The behaviour of a system at different stages can be categorised in three classes; transient behaviour, steady-state behaviour and regeneration behaviour. In the former, the average output of the system fluctuates and does not settle to a single value. On the other hand, a system is said to have reached a steady-state behaviour when the average fluctuation is limited. Finally, the regenerative systems tend to cycle continuously between transient and steady-state behaviour [44].

3.12.1 Transient and Stability

For most systems the steady-state behaviour is of interest as the systems usually operate on steady-state after the initial period has elapsed. Therefore, it is necessary to exclude the transient observations from the system evaluation. Such a process is called transient removal and can be achieved using one of the following methods [43].

1. Long Runs
2. Proper Initialisation
3. Truncation
4. Initial Data Deletion
5. Batch Means
6. Moving Average of Independent Replications

A combination of these methods is used to determine the region from where the results should be obtained and it is important to discuss these methods in more detail.

The first method (long runs) is a brute force technique and the simulation is simply run for a very large number of events; long enough to ensure that the initial condition will not affect the result. This method is expensive in terms of the computation resources and should be avoided if possible. In addition, it is not always clear how long the length should be in order to eliminate any effect of the transient region.

For the second method, the system starts at a state close to the expected steady-state which is determined from previous test runs. This method reduces the transient period length which subsequently minimises the transient effect or eliminates it completely.

The first and second methods are relatively unsophisticated, but sometimes unavoidable, particularly in the initial stages of the modelling. Other methods (3-6) are generally more appropriate and are now discussed in detail.

It must be emphasised that methods 3-6 can only be applied when the variability during the steady-state is less than the variability during the transient phase.

The truncation method can be summarised in the following steps (n is the total number of observations).

1. Set $l = 1$.
2. Remove l observation from n.
3. Find maximum and minimum among the $n - l$ observations.
4. If the observation $l+1$ is equal to the maximum or the minimum obtained in step 3; repeat steps 2-4 for $l = l + 1$.
5. The transient period is l observations long.

This method gives incorrect results if there is a transient overshoot or undershoot and therefore is not often used.

In initial data removal, initial observations are removed and the average of the remaining observations is computed. The initial observations are removed in iterative manner until any additional removals do not significantly affect the average of the remaining observations. However, the average will fluctuate slightly even when the system reaches steady-state. For this reason, it is necessary to perform multiple replications and average the observations across replications before applying this technique. This method is not used to determine system stability and hence it is not discussed any further.

The method of batch means requires running very long simulations. The simulation is then divided into multiple parts n (batches). In the first step, the mean ($\overline{x}_i$) of each batch (i) is computed. In the second step, the variance ($Var(\overline{x})$) of the means is computed. Finally, the same process is repeated for increasing batch numbers (n) and the variance is plotted against the number of batches. The transient interval is the value of n for which the variance starts decreasing.

The last method for identifying the transient interval is the Moving Average of Independent Replications. This is similar to the Initial Data Deletion method but the average is calculated over a moving interval window. In order to avoid this slight variation in the mean of the moving window, multiple independent replications are performed which are subsequently averaged. Each

of the independent replications is performed without changing any of the input parameters except for the seeds of the random number generators. Averaging across replications results in a smoother trajectory mean.

Table 3.6 illustrates the steps taken to identify the transient period using the moving average of the independent replication technique. The notation used in the following table is: m denotes the number of independent replications, n the number of observations in each replication and x_{ij} denotes the j^{th} observation of the i^{th} replication. $\overline{x}_j$ is the trajectory mean across m replications and $\overline{\overline{x}}_j$ is the mean of observations within the window of length k. The window k is increased until the plot is sufficiently smooth.

Step	Description	Equation
1	Compute the trajectory mean by averaging across replications.	$\overline{x}_j = \frac{1}{m}\sum_{i=1}^{m} x_{ij}$, for $j = 1, 2, ..., n$
2	Set $k = 1$ and plot a trajectory of the moving average of successive $2k+1$ values	$\overline{\overline{x}}_j = \frac{1}{2k+1}\sum_{l=-k}^{k} \overline{x}_{j+l}$, for $j = k+1, k+2, ..., n-k$
3	Repeat step 2, with $k = 2, 3, ...$ until the plot is sufficiently smooth	
4	Find the knee of the plot, the value j at the knee gives the length of the transient phase.	

Table 3.6: Steady-State Detection using Moving Average of Independent replications.

Although most systems reach steady-state at some point, this is not the case in ad-hoc networks. The reason is that the system is continuously changing and the output performance depends on a wide variety of conditions. As a case study, the end-to-end delay will be examined. The instantaneous end-to-end delay suffered by each packet, is shown in Figure 3.16.

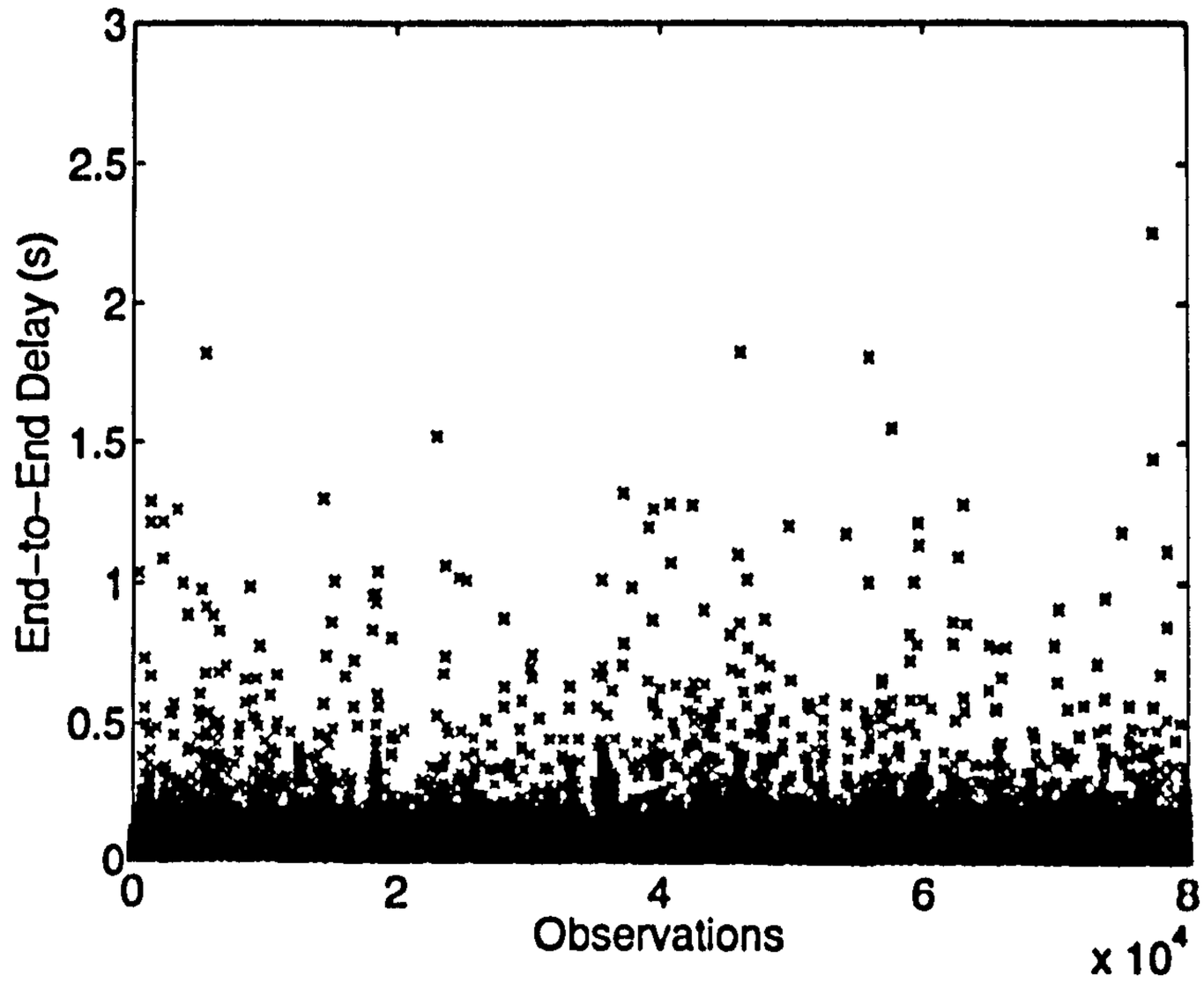
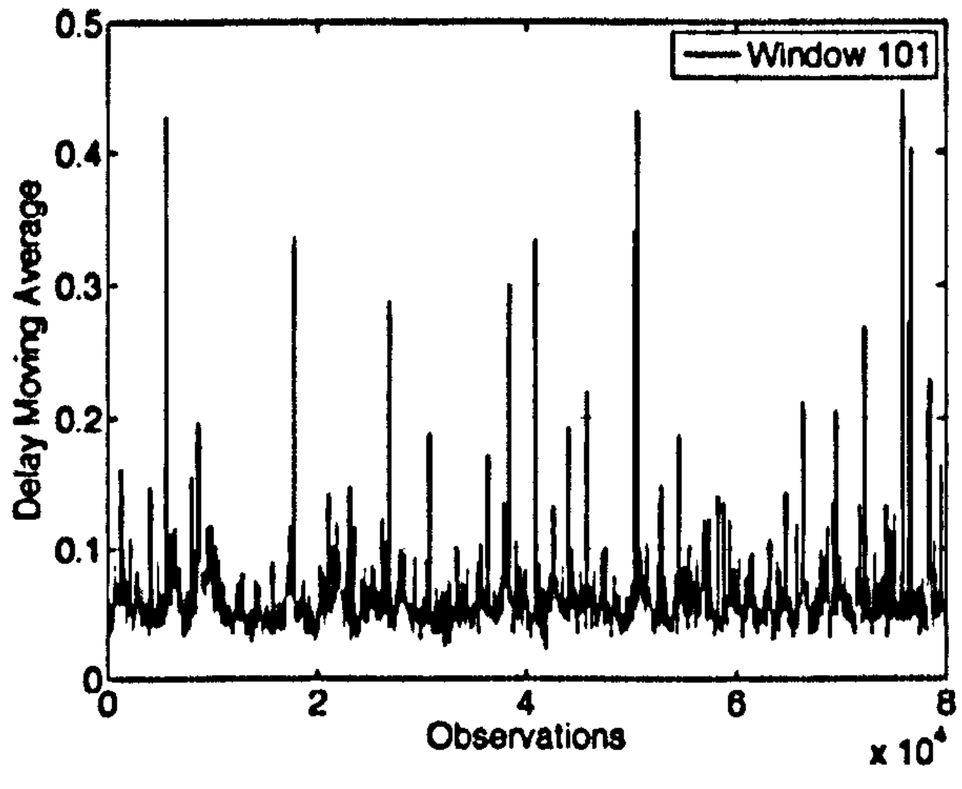


Figure 3.16: Instantaneous End-to-End delay.

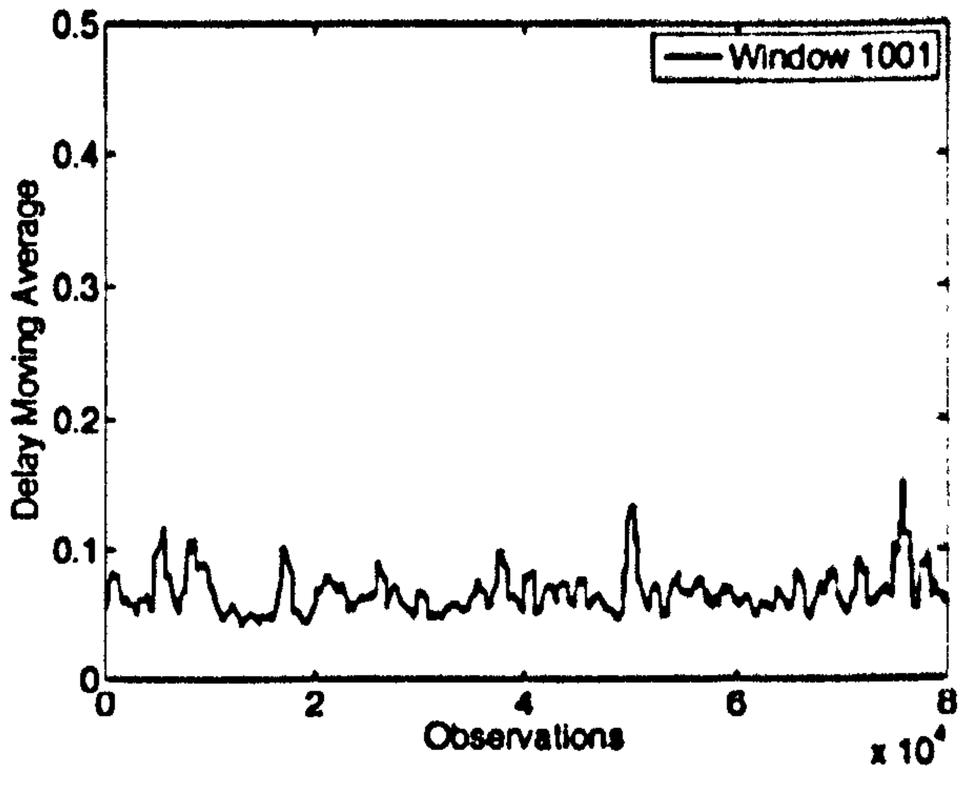
From the graph of Figure 3.16, it is clear that the end-to-end delay never stabilises. Such systems are referred as non-terminating systems [44]. Non-terminating systems continue to oscillate indefinitely and the system appears to be in a permanent transient state and hence short term mean values do not converge. However, long-term behaviour is predictable and converges. The key reasons contributing to the variability of the end-to-end delay are: i) IEEE 802.11 is a CSMA/CA protocol and when congestion is increased, the back-off mechanism is triggered with an exponentially increasing back-off window which results in an increased delay. When the network congestion decreases, the back-off window decreases and the end-to-end delay drops. ii) Packets are propagated along paths with variable lengths which subsequently contribute the variable end-to-end delay iii) the DSR routing protocol is a reactive routing protocol and when the nodes do not have a route to the desired destination or when routes break, the nodes trigger route discoveries which further disturb the output. This is especially true in mobile and battery powered scenarios where the topology is continuously changing and routing information becomes obsolete.

To backup the claim that the mean value converges and the system behaviour becomes repeatable, the Moving Average of Independent Replications technique was applied to the end-to-end delay. Even if the technique is not ideal for detecting steady-state due to variability as explained in the beginning of the section, it will show that the statistics converge over extended runs. To compute

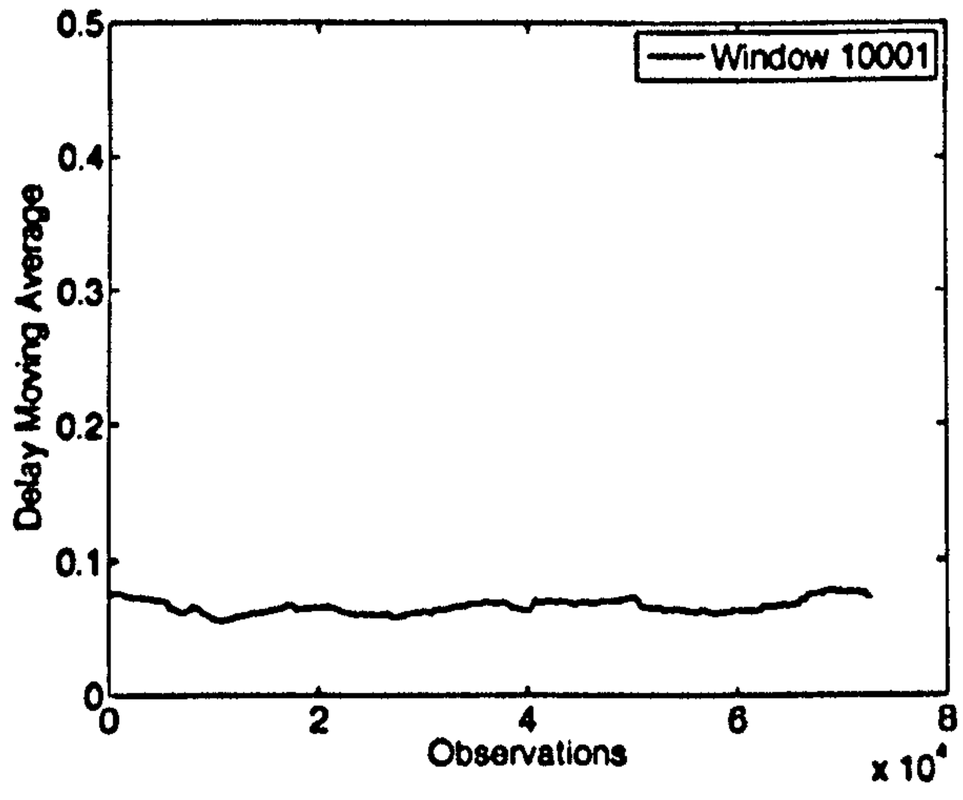
the Moving Average, 5 traffic profiles were generated with 20,000 connections. The average connection duration was selected to be 10 seconds. The simulation time was set to 2,000 seconds. Figure 3.17 plots the Moving Average for different window sizes. For a window size of 101 observations the plot remains variable and this indicates that if the statistics obtained based upon such a small window size, the results will be variable, inaccurate and unrepeatable. As the window size increases, the plot becomes smoother and this indicates that the behaviour is predictable and the mean values converge.



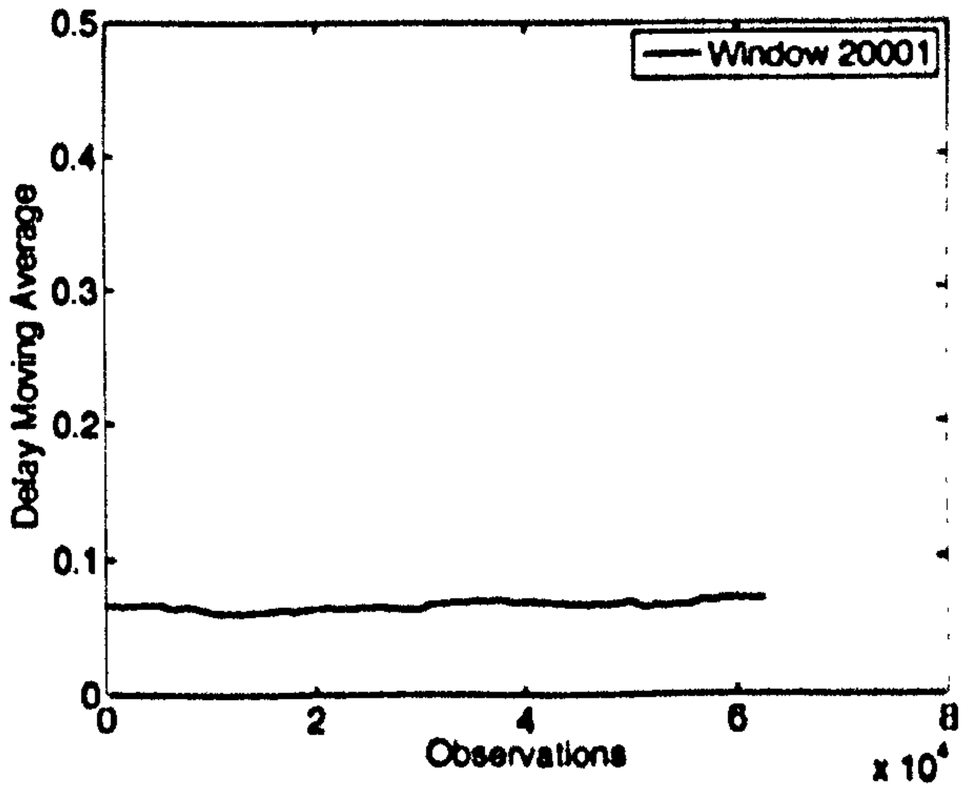
(a) Window Size 101



(b) Window Size 1,001



(c) Window Size 10,001



(d) Window Size 20,001

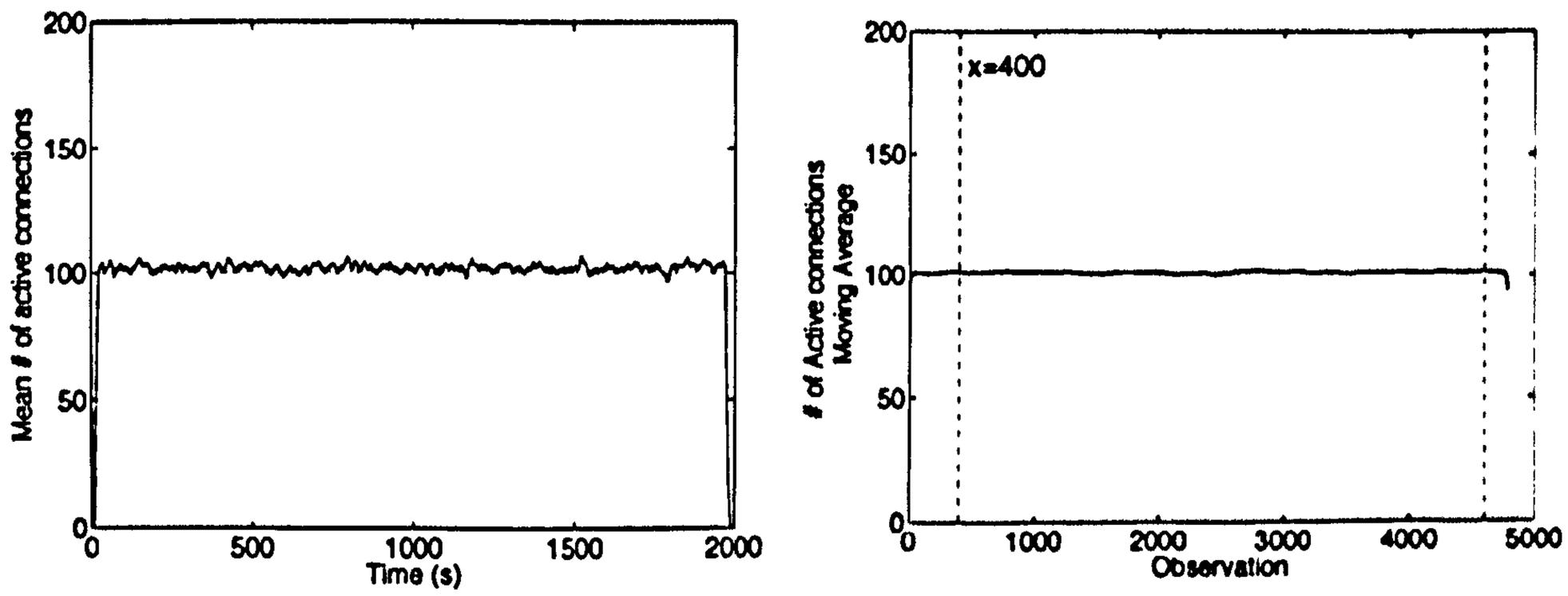
Figure 3.17: End-to-End delay moving average for variable window sizes.

In order to ensure that any transient effects are completely removed the number of active connections and the average node speed is observed[10]. Applying

[10]It might seem unnecessary to remove more observations since the simulation runs are very long and any transient effects will be eliminated (as discussed in the method of "Long runs"). However, as an extra measure, the system is allowed to warmup in terms of active connections

the moving average technique to both the output of the traffic and mobility generators give indications of where warmup phase ends. Subsequently these periods are removed.

Applying the Moving Average technique on the output of the traffic generator, the graph of Figure 3.18 is obtained. In order to determine when the generated traffic stabilises, the number of active connections is sampled every 1 second. 5 connection patterns are generated with 20,000 connections over the period of 2,000 seconds. The average connection duration is 10 seconds. In Figure 3.18a the trajectory mean across replications is plotted against time. Figure 3.18b shows the moving average for window k=101. The knee of the plot occurs at approximately 50 seconds. In order to ensure that the transient was completely removed the first 400 seconds are discarded.



(a) Trajectory Mean across Replications. (b) Moving Average for window k=101

Figure 3.18: Transient Removal using Moving Average Technique.

Another effect to note is that the average number of connections drop near the end of the simulation. This is known as a border effect and it results because NS-2 does not allow events to be scheduled outside the simulated time. Hence, the traffic generator does not initiate any connections after time $T_s - 2 \cdot D$ (where T_s is the simulation time and D the average connection duration), as there is a chance that the stop events fall outside the simulated time. In order to eliminate the border effect, the last 400 simulated seconds are also discarded.

The same procedure must be performed to determine the warmup time of the mobility scenario. Section 3.11 showed that the average node speed for the traditional RWP mobility model decays and an alternative model was chosen.

and average node speed.

When the Moving Average technique is applied to the average node speed of the modified RWP model, the graph of Figure 3.19 is obtained. The knee of the plot is at 100 seconds but, in order to ensure that the transient is completely removed, the first 300 seconds of the simulation are ignored. The mobility generator does not have a border effect near the end of the simulation because it does not require the scheduling of stop events.

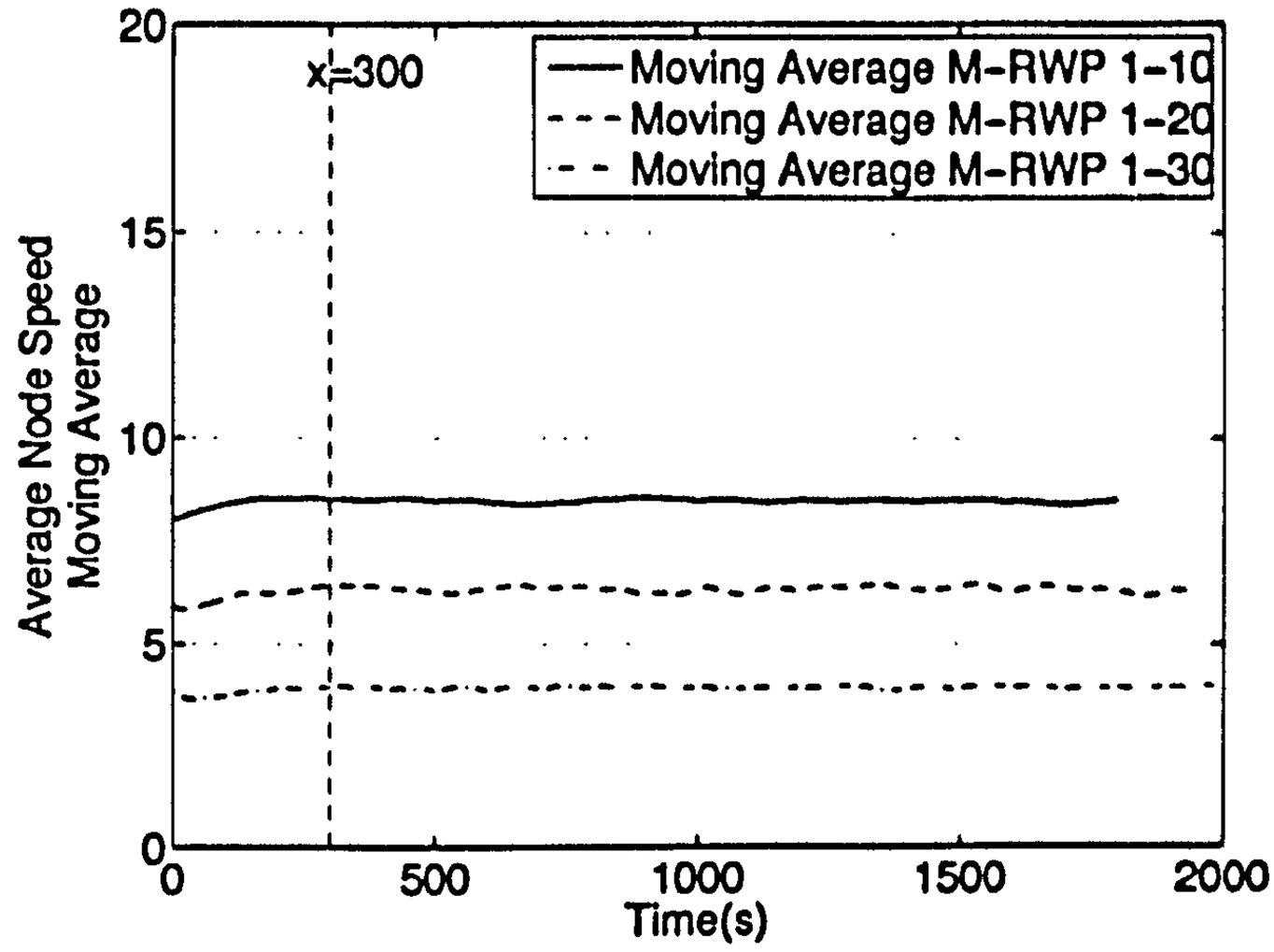


Figure 3.19: Moving Average applied to the Average Node Speed.

From the two parameters of average node speed and number of active connections, two different transient periods where obtained. In order to ensure that both aspects have sufficient warmup, the aggregate lengths of the warmup periods are discarded. The aggregate discard period is 400 seconds from the beginning of the simulation and 400 seconds from the end.

However, not all system metrics are to be studied under steady-state conditions as they never reach a steady-state. For example, in MANET environments when battery powered nodes[11] are deployed, it is important to include the transient period in order to predict the future status of the network. If the warm-up period is excluded when predicting the network lifetime, it will lead to inaccurate conclusions. At the beginning of any deployment, the routing knowledge of the nodes is non-existent and the nodes consume a significant amount of energy to collect routing information. The energy consumption is due mainly to the fact that the nodes flood the network to obtain routing knowledge and this flooding

[11]It is assumed that nodes do not have scavenging capabilities.

incurs high overheads in all nodes participating in the discovery process. Hence, it will be unwise to exclude the transient region when considering metrics such as network lifetime.

On the other hand, when considering energy drain rate the warmup period must not be removed, as the point of interest is the rate at which energy is decaying when the network operates in the stable region. In this thesis, when results are quoted the warmup period is discarded unless otherwise stated and when the results include data within the warmup period justifications are provided.

3.12.2 Stopping Criterion & Confidence Intervals

The results presented in section 3.8 were based only on a single simulation run. The purpose of that section was to verify that the model designed correctly and distributes traffic across multiple paths.

The simulation length is important because long simulations waste computer resources and short simulations result in variable and inaccurate conclusions. The accuracy of the final results is measured with a certain degree of confidence. A confidence interval is inversely proportional to $\sqrt{mn}$, where m is the number of independent replications and n the number of observations in each repetition [43]. Therefore, a confidence interval can be narrowed in two ways; either by increasing the number of independent replications or by increasing the number of samples in each repetition. Increasing the number of repetitions (m) is not particularly efficient because the number of observation that would be ignored for transient removal will increase. Hence, it is more preferable to increase the length (n) of each repetition. However, the bottleneck of this approach is that longer simulations require more memory to run, since NS-2 requires all events to be scheduled in advance. Consider a connection traffic scenario which consists of 100,000 connections then 200,000 events must be loaded into memory (100,000 events to initiate connections and 100,000 to terminate them) before the NS-2 engine starts simulating events. Therefore, a combination of repetition (m) and length (n) is used to narrow the confidence intervals. An additional challenge when simulating ad-hoc networks is that the system behaves in a non-terminating manner and therefore long runs aid the convergence of the results.

In order to ensure that independent replications were conducted m traffic, topology and mobility scenarios where generated and then loaded into the NS-2 simulator. It should be noted that all replications are of the same length in terms of simulated time but the number of the output observations is variable. For

example, when end-to-end delay is considered, the number of the observations is different for each replication, because an unpredictable number of packets are successfully delivered to their destination.

For all repetitions, the initial warmup period n_{init} is removed in addition to the final portion of the system's response n_{final} as described in Section 3.12.1. Note that $n_{init} = n_{final} = max(n_{init}, n_{final}) = n_0$ and the total number of observation in repetition i is denoted by n_i. Initially, the mean value of each repetition is computed. In order to calculate the confidence interval of the means, the mean of the means is required in addition the variance of the means. Once these two values are calculated the confidence interval can be found using the Student's t-distribution, as the number of replications is less than thirty [43]. If the number of repetitions is higher than thirty, student's t-distribution approximates normal distribution and the normal distribution should be used.

The confidence level used throughout the thesis is 0.05 (the confidence level is also referred to as confidence interval of 95%) and the confidence intervals are denoted with vertical bars around the mean values. Table 3.7 shows the steps taken to compute the confidence intervals.

Step	Description	Equation
1	Compute the mean value for each replication.	$\overline{x}_i = \frac{1}{n_i - 2n_0} \sum_{i=n_0+1}^{n_i - n_0} x_{ij}$, for $i = 1, 2, ..., m$
2	Compute the overall mean across the replications	$\overline{\overline{x}} = \frac{1}{m} \sum_{i=1}^{m} \overline{x}_i$
3	Compute variance of the replicate means	$Var(\overline{x}) = \frac{1}{m-1} \sum_{i=1}^{m} (\overline{x}_i - \overline{\overline{x}})^2$
4	The confidence interval of the mean response is:	$C = \overline{\overline{x}} \pm t_{[1-\infty/m-1]} \sqrt{Var(\overline{x})}$

Table 3.7: Independent Replications.

3.13 Computation Resources

The simulation results presented in this thesis were performed in a range of hardware architectures, and this section aims to provide an indication to the

Description	Processor Time
Single simulation run (Simulated Time 2,000 s)	10 *minutes*
Single post-processing run	5 *minutes*
Single simulation run with post-processing	15 *minutes*
10 repetitions for each point in graph	$15 \cdot 10 = 150$ *minutes*
30 points per graph	$150 \cdot 28 =$ 4200 *minutes*
2 routing schemes	$2 \cdot 4200 =$ 8400 *minutes* = 140 *hours* = 5.83 *days*
4 sets with variable number of connections	$5.83 \cdot 4 = 23.33$ *days*

Table 3.8: Computation resources consumed for the results of Section 4.3.1.

reader of the computation time consumed. Table 3.8 presents computation time consumed to complete the results of Section 4.3.1. The computation times presented show approximate timings for a single-core "Intel(R) Xeon(TM) CPU 2.40GHz" processor.

Latterly, an eight-core processor server was used and parallel simulation runs were executed, but not all eight-cores could be utilised in parallel. As a result of the throughput of Storage Input/Output (hard drives), only four cores could be utilised.

3.14 Summary

This chapter introduced and justified the configuration parameters and methodology used for the evaluation of routing schemes. It was shown that the system is non-terminating. However, the long term prediction gives repeatable and accurate results. The results are taken over an extended period of time, in addition to the warm-up period removal, to ensure high accuracy.

Secondly, the chapter proposed a modification to traditional DSR which enables the protocol to operate using multiple routes. The modified protocol was verified using a special case study. The proposed protocol distributes the load across different paths according to its designed probabilities. It was also noted that the overall delay increased when the multipath DSR was used; a consequence of variable path length. However, under the presence of more com-

prehensive traffic profiles, the system reduces the delay as will be shown in the next chapter, because network congestion is reduced. The number of packets sent by the source node in this scenario is higher in traditional DSR. However, the final energy map of the network was heavily unbalanced. The modified DSR did not exhaust any of the relay nodes and this is preferable behaviour for more elaborate scenarios.

In the next chapter, an extensive evaluation of the proposed scheme is performed to determine if the scheme offers merits compared to its unipath counterpart. The evaluation will be done using an enhanced NS-2 simulation model that is underpinned by the concepts described in the current chapter. The aim of chapter 3 was to provide confidence for the reader of the evaluation methodology adopted and to simplify the discussion of the results. In subsequent chapters, it is not necessary to repeat the common aspects of validation and verification.

Chapter 4

Multipath Routing

Multipath routing has the potential to significantly improve wireless or wired network performance. In chapter 3, preliminary results have shown that the proposed scheme distributes the load to multiple paths. However, the scheme was only examined in a special case scenario. This chapter examines the proposed scheme in larger static and mobile topologies supporting multiple simultaneous connections. The work described in this chapter represents an extended version of the concepts presented at the International Symposium on Performance Evaluation of Computer and Telecommunication Systems [65].

In the first part of this chapter a full description is given of the implementation issues associated with the development and design of a multipath routing protocol. The second part of the chapter discusses related published work and finally the chapter concludes with the results, discussion and contributions.

4.1 Multipath Routing Components

The ad-hoc network environment is continuously changing and devices are actively involved in routing operations. Reactive routing protocols such as DSR and AODV compute routes to the destination in an on-demand fashion. The protocol's operation can be organised to three key phases (Figure 4.1); the route discovery, route selection and traffic allocation. The route discovery process is a rather expensive operation because nodes flood the network with control packets whose aim is to find relaying information to the destination. This phase consists of two sub-phases; the route request propagation and the route reply propagation.

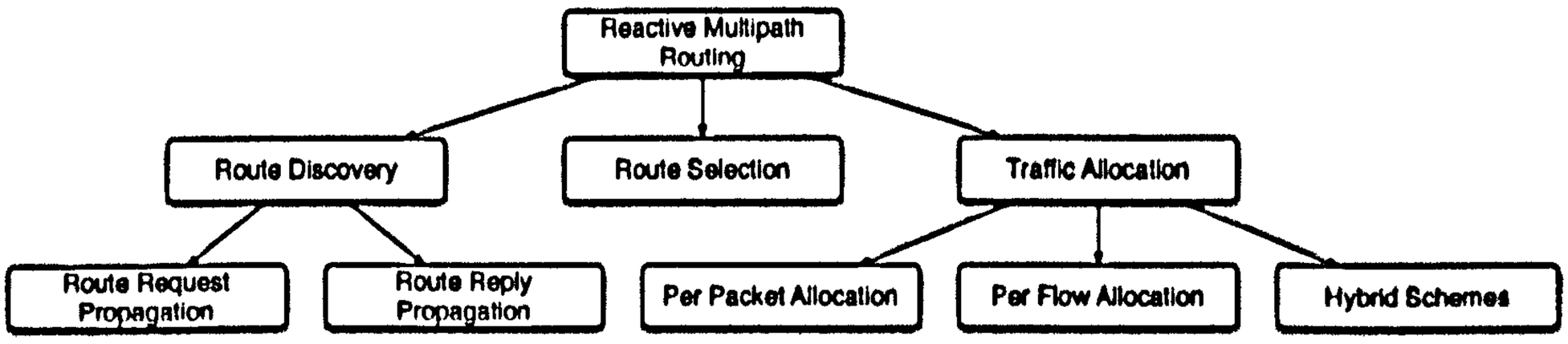


Figure 4.1: Reactive Multipath routing main phases.

In the second phase, the routing protocol searches its cache to find an optimal path-set for packets queued in its buffers. Optimal paths are subjective and their optimality depends upon the targeted objective. The route selection can be optimised based on a number of criteria such as hop count, end-to-end delay, path reliability, energy efficiency etc. In chapters 5 and 6, a route selection mechanism with the aim to extend network lifetime and increase energy map homogeneity is studied.

When a candidate path-set is selected, the routing protocol must decide the amount of traffic to send over the selected path. This is the third phase of the protocol operation and there are three available schemes to choose from. The first scheme is the per-packet allocation where the choice of the path is made on per-packet basis. The decision making mechanism is triggered individually for every packet. The drawback of this scheme is the increased out-of-order delivery to the destination. This is particularly problematic when TCP is in use because the delay jitter is higher and TCP stalls connections, wrongly assuming that packets were lost. On the other hand, per-packet allocation offers finer control over the performance of network resources.

The second scheme triggers the decision making process on a per-flow basis. The reordering problem is avoided in this scheme but the allocation granularity is coarser and it is then difficult to ensure that the traffic is distributed evenly over multiple paths. Of course, the allocation granularity depends highly upon the average connection duration and rate. If the connection duration is low then the per-flow allocation converges to per-packet allocation.

Hybrid schemes combine per-flow and per-packet allocation. In such schemes the decision making process is triggered at regular intervals. For example, when a connection is initiated, the decision process is initially triggered. When the connection duration exceeds a pre-defined window, then another decision process is initiated and so on. This method avoids an increase in out-of-order packet arrivals at the destination while at the same time, offers higher control on the

load distribution.

In this thesis, per-packet allocation was chosen as the allocation method because it gives higher control in a highly dynamic environment; it is more difficult to obtain network conditions such as available bandwidth to apply sophisticated per-flow and hybrid schemes.

4.1.1 Route Discovery Propagation

In the first sub-phase of the route discovery process, nodes broadcast a route request packet to their neighbours, specifying the destination for which a route is requested. The neighbours subsequently forward request packets to their neighbours until the destination is reached. Subsequent route requests received from intermediate nodes are ignored. The second sub-phase of the route discovery is the route reply propagation. In this phase, the destination node is aware that another peer attempts to communicate with it and responds to the route request instigator using the knowledge obtain from the received route requests (i.e. the reverse path).

Multipath routing protocols tend to modify both sub-phases of the discovery procedure when attempting to find multiple routes. Each multipath routing protocol aims to discover paths which fall within one of the following categories [66, 67, 68, 69, 70]:

- Node Disjoint (also known as Totally Disjoint)
- Link Disjoint
- Maximally Disjoint (also known as Best Partially Disjoint)
- Non-Disjoint (also known as Partially Disjoint)

Node disjoint routes have no common nodes other than the source and destination node. Link Disjoint routes have no common links but they may have common nodes. Maximally disjoint paths are those paths that have the minimum number of common nodes. Non-disjoint paths are those with any common nodes. Maximally disjoint paths are a special case of non-disjoint paths.

Well known algorithms exist in the literature for obtaining all these types of paths when the full network graph $G(V, E)$ is known. These algorithms usually employ a shortest path algorithm such as the modified Dijkstra's shortest path [66, 71] and some graph modification techniques to obtain paths obeying the search criteria. In order to apply one of these search algorithms, the devices have either to gain complete network knowledge (which is rather expensive and

perhaps superfluous because network topologies change particularly rapidly) or to engineer a route discovery process which finds sub-optimal paths in a distributed manner.

Non-disjoint routes are more prone to faults because a single node or link failure can result in multiple route failures. In node disjoint routes, a single link or node failure will only cause a single route failure. Therefore, node disjoint routes are more preferable compared to non-disjoint counterparts because of the higher resilience provided. However, there are scenarios where node disjoint routes do not exist. A routing protocol which discovers only node disjoint paths is restricted by the connectivity of bridges connecting the source and the destination. A well known algorithm for computing k-node disjoint paths[1] is specified by Bhandari [66]. Consider the network of Figure 4.2; disjoint paths between S and D do not exist because there is only one connection (bridge) between sub-graphs II and III through nodes E and F. Assuming that more bridges exist between sub-graphs II and III then the bottleneck for disjoint routes becomes the connectivity between sub-graphs I and II, which have only two bridges and no more than two disjoint paths can exist. In such scenarios, full disjointedness is restrictive but is still more advantageous than unipath counterparts.

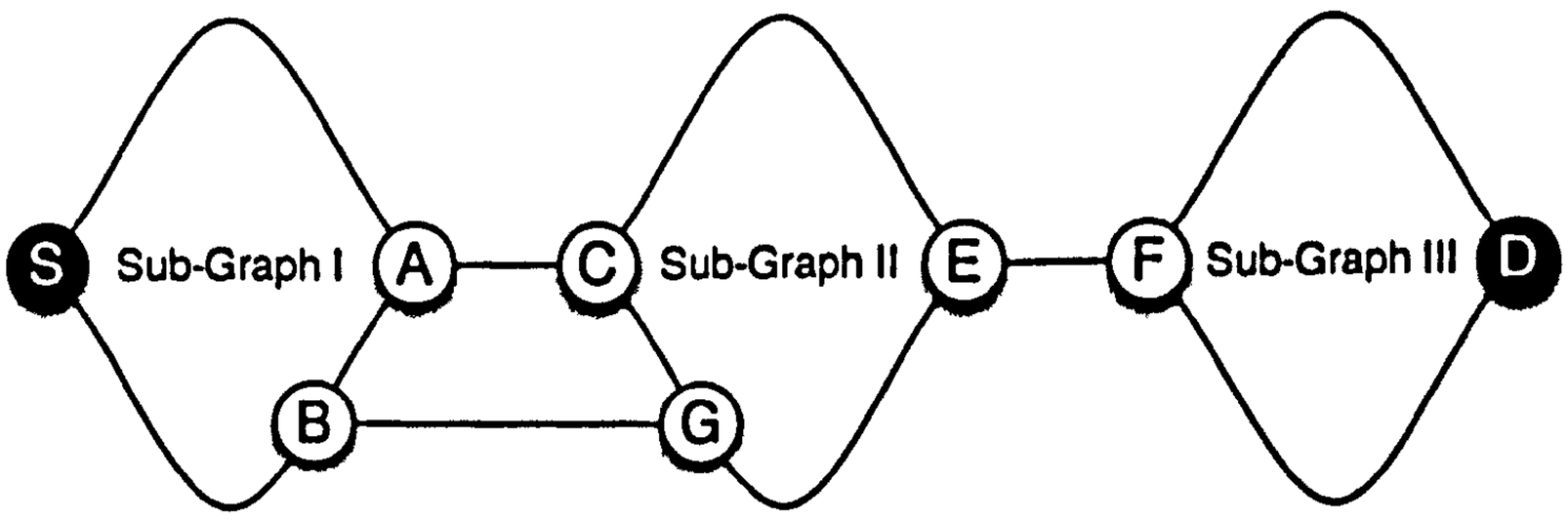
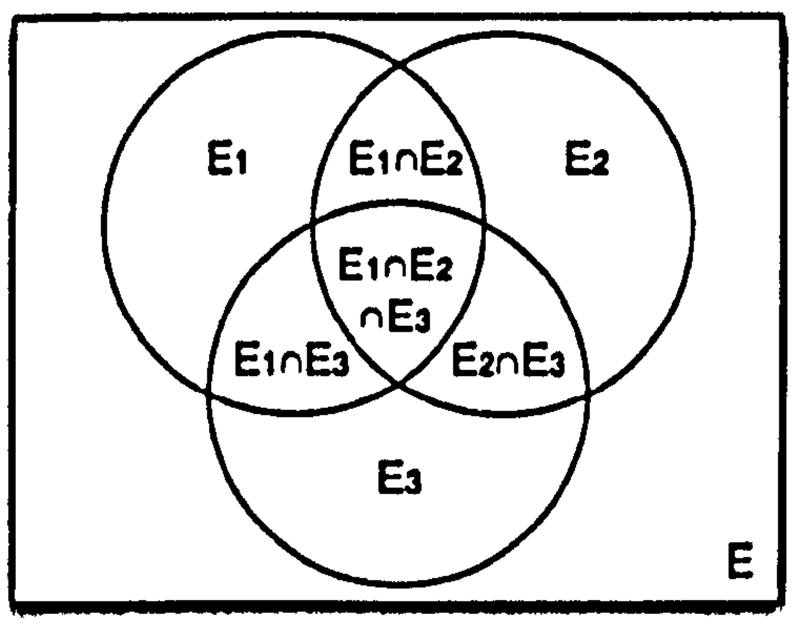


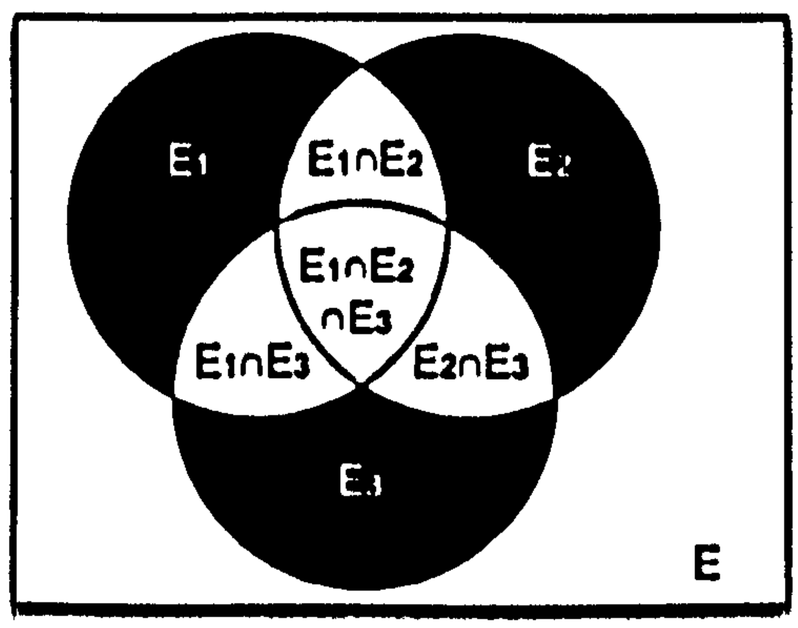
Figure 4.2: Sample network with non-disjoint routes.

To illustrate the reduced resilience of two non-disjoint paths consider a network $G(V, E)$ and source and destination pair (S, D) which is connected with paths $J_1, J_2, \ldots J_r$. The set of links in each path is represented with $E_1, E_2 \ldots E_r$. The cardinality of set A is denoted as $|A|$. Figure 4.3a shows an example of three paths using a Venn diagram.

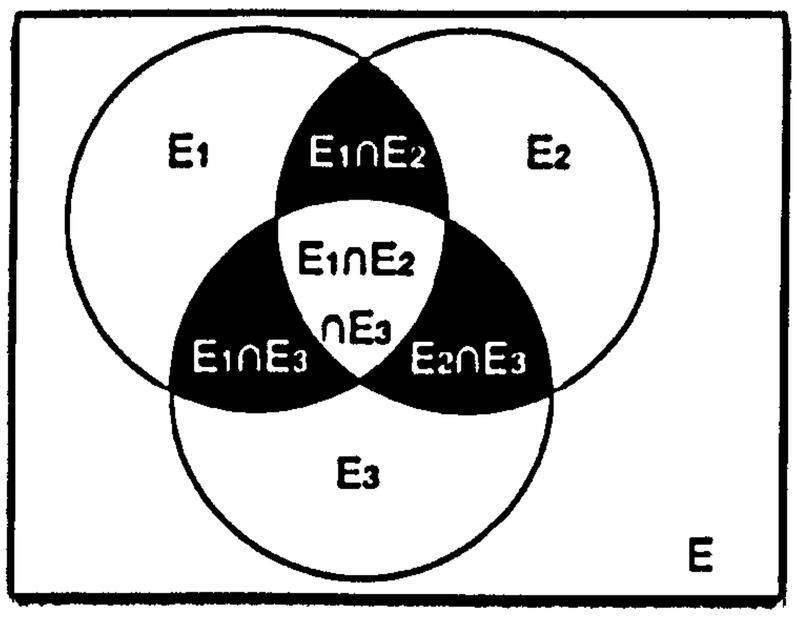
[1]Note that the algorithm requires full knowledge of the network topology and is therefore not suited for distributed approaches where devices only have partial network knowledge.



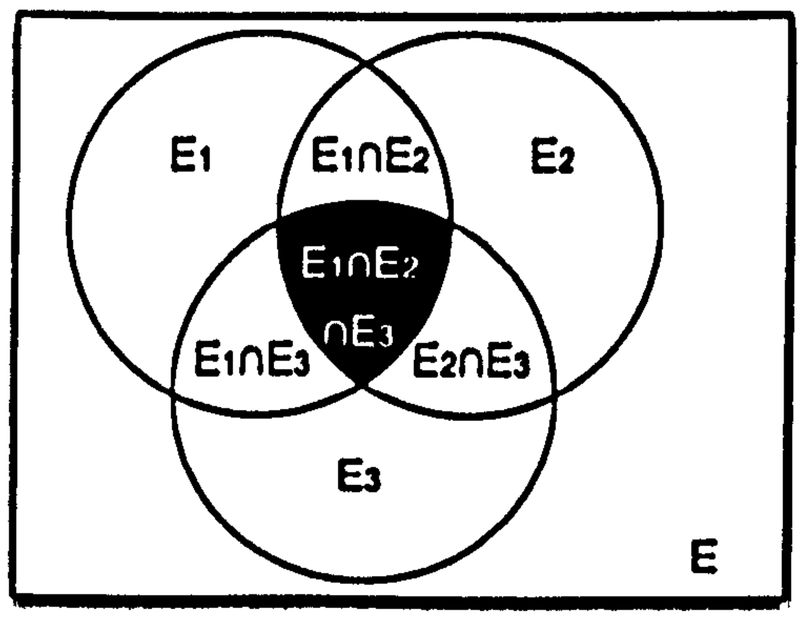
(a) General Venn diagram



(b) Exactly one path to fail



(c) Exactly two paths to fail



(d) Exactly three paths to fail

Figure 4.3: Venn diagrams for the different failure combinations.

The r paths represents a sub-graph of G denoted as $G'(V', E')$. The total number of links ($|E'|$) in the sub-graph G' can be found using the inclusion-exclusion principle. Equation 4.1 shows this calculation:

$$\begin{aligned} |E'| = \left|\bigcup_{i=1}^{r} E_i\right| = \sum_{i=1}^{r} |E_i| - \sum_{i,j:1\le i<j\le r} |E_i \cap E_j| \\ + \sum_{i,j,k:1\le i<j<k\le r} |E_i \cap E_j \cap E_k| - \\ \cdots \\ + (-1)^{r-1} |E_1 \cap E_2 \cap \ldots \cap E_r| \end{aligned} \tag{4.1}$$

Given that a single link failure had occurred in sub-graph G', and assuming that the path failure probabilities are independent, the probability of only one path failing can be calculated as follows[2]: Only path J_1 fails when the prob-

[2]The calculated probabilities are not the path failure probabilities, but the probabilities of 1, 2 or 3 paths to fail given that a single link has failed.

lematic link belongs to E_1 and not in $E_2, E_3, \ldots E_r$. The number of links that belong to that category is the cardinality of the set $E_1 \cap \overline{E_2} \cap \overline{E_3} \cap \ldots \cap \overline{E_r}$, which is equivalent to $E_1 \cap \overline{\bigcap_{i=2}^{r} E_i} = E_1 \cap \overline{\bigcup_{i=2}^{r} E_i}$ (de Morgan's theorem). Similarly, only path J_{k_1} fails when the problematic link belongs in the set $E_{k_1} \cap \overline{\bigcup_{i=1, i \neq k_1}^{r} E_i}$.

From these observations the probability of only path J_{k_1} failing can be found from the formula.

$$P(J_{k_1}) = \frac{\left| E_{k_1} \cap \overline{\bigcup_{i=1, i \neq k_1}^{r} E_i} \right|}{|E'|}. \tag{4.2}$$

where $|E'|$ can be calculated using Equation 4.1.

The event of a single path failure occurs when only one of the paths fails. This scenario is illustrated in Figure 4.3b for the case of three paths. Therefore, the probability of a single path failure is given by:

$$P(x = 1) = P(J_1) + P(J_2) + \ldots + P(J_r) = \sum_{l=1}^{r} P(J_l) \tag{4.3}$$

The probability of two paths J_{k_1} and J_{k_2} failing simultaneously due to a single link failure is given by the equation:

$$P(J_{k_1}, J_{k_2}) = \frac{\left| E_{k_1} \cap E_{k_2} \cap \overline{\bigcup_{i=1, i \neq \{k_1, k_2\}}^{r} E_i} \right|}{|E'|} \tag{4.4}$$

and the probability of exactly two paths failing (Figure 4.3c) is then the sum of all independent events.

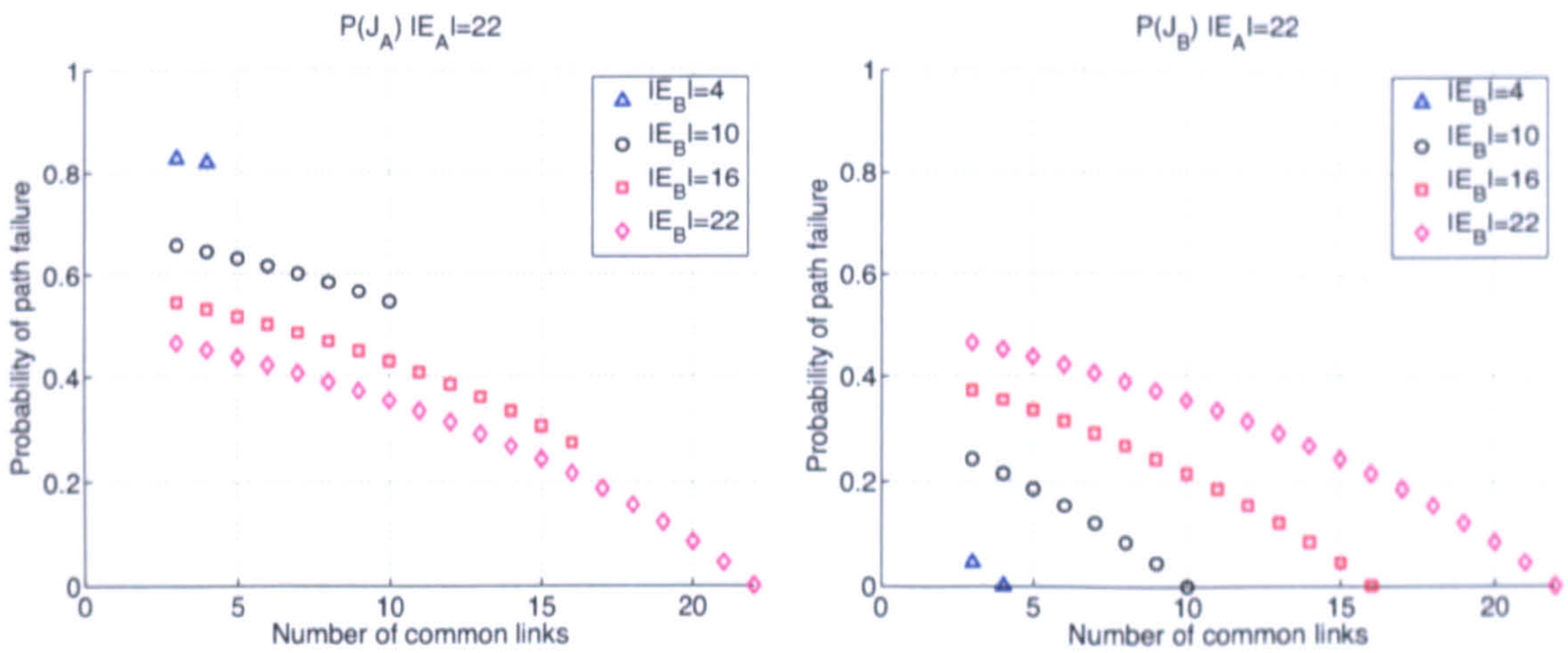
$$P(x = 2) = \sum_{l,m:1 \leq l < m \leq r}^{r} P(J_l, J_m) \tag{4.5}$$

Now these equations can be generalised for exactly k paths $(J_{k_1}, J_{k_2}, \ldots, J_{k_k})$ to fail.

$$P(J_{k_1}, J_{k_2}, \ldots, J_{k_k}) = \frac{\left| \left(\bigcap_{j=k_1,k_2,\ldots,k_k}^{r} E_j \right) \cap \left(\overline{\bigcup_{i=1,i\neq\{k_1,k_2,\ldots,k_k\}}^{r} E_i} \right) \right|}{|E'|} \tag{4.6}$$

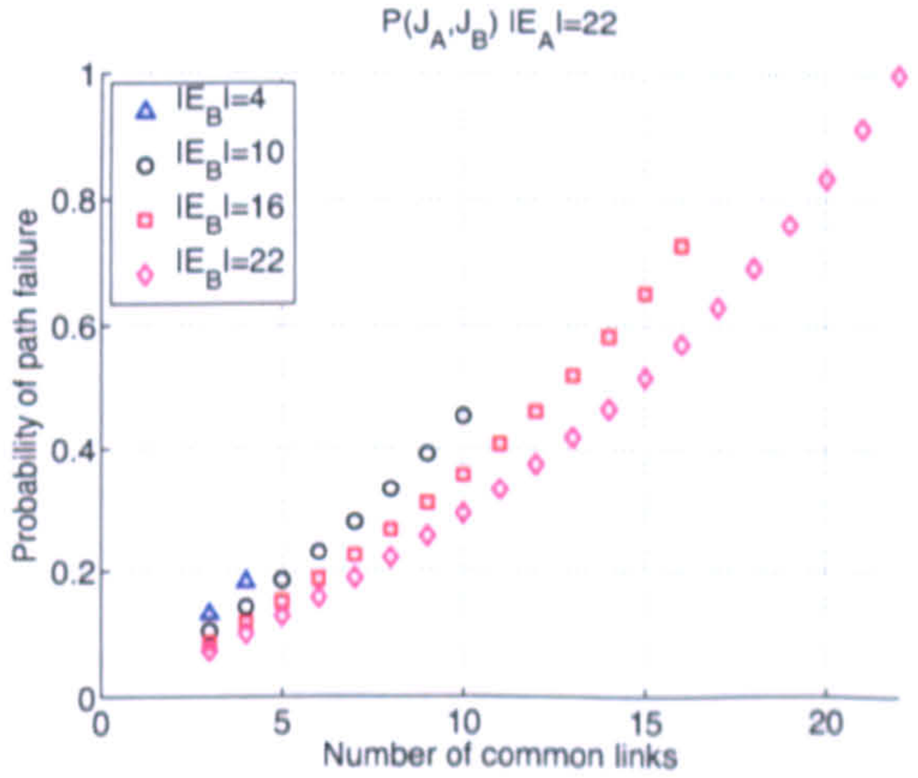
$$P(x = k) = \sum_{k_1,k_2,\ldots,k_k : 1 \leq k_1 < k_2 < \ldots < k_k \leq r}^{r} P(J_{k_1}, J_{k_2}, \ldots, J_{k_k}) \tag{4.7}$$

Figure 4.4 shows the path failure probability for two non-disjoint paths versus the number of common links. In all figures the length of path J_A is maintained constant and equal to 22 while the length of path J_B is varied between 4, 10, 16,



(a) Probability of path J_A to fail

(b) Probability of path J_B to fail



(c) Probability of path both paths J_A and J_B to fail.

Figure 4.4: Path failure probabilities for two partially disjoint paths versus number of common links, given that a single link failure has occurred.

22 which are shown with blue triangles, the black circles, the red squares and the magenta diamonds respectively. The Figure 4.4a plots the probability for only path J_A to fail given that a single link failure occurred, versus the number of common links, while Figure 4.4b shows the probability of only path J_B failing versus the number of common links. Finally, Figure 4.4c shows the probability of both paths failing simultaneously versus the number of common links.

4.2 Related Work

There are ambiguous opinions in the literature on whether multipath routing offers significant merit in terms of the network performance. One camp contends the belief that multipath routing is indeed advantageous compared to unipath counterparts; the other contends that the improvement is not significant if any at all, because of the increased overhead incurred to discover multiple paths or because the paths are closely coupled with each other. The most noteworthy multipath protocols proposed in the literature are able to find link and node disjoint paths [72, 73, 74, 75, 76, 77].

Nasipuri and Das [72] propose and evaluate two DSR extensions which enable DSR to discover multiple routes. No modification was made to the route request propagation but modifications were made to the route reply process. The destination node will be reached by multiple route requests transversing through different sets of links. The destination node upon reception of the first RREQ builds the primary path. Subsequent RREQs only get attention if the route they contain is link disjoint with the primary path. In such a way, failure of one link does not impact the remaining routes while the reply storms are avoided.

In the first proposed extension, the destination node equips the source node with alternative routes and an intermediate link failure initiates a route error report to the source node. The source node, upon reception of the error report, switches all traffic to one of the alternative paths but all data packets currently in transit over the primary path are lost.

The second proposed extension operates in a slightly different manner. The destination node does not equip only the source node with alternative paths but all intermediate nodes, if possible. The source node is considered as an intermediate node in this scheme and it is equipped with a single alternative path. Upon a link failure, intermediate nodes do not send route error reports to the source node but they modify the packet's header with the alternative path.

In such a way, traffic already in transit is not lost. However, if the node does not have an alternative path to the destination the scheme defaults to the first proposal where a route error packet is sent to the source.

The disadvantage of these schemes is that the alternative paths are not used simultaneously, but only when a route failure occurs. Hence, the scheme does not attempt to distribute the load across multiple paths (as in the scheme proposed in Chapter 3), but is used for fast route failure recovery. Similar approaches are proposed elsewhere [78, 79, 80, 75]. Additionally, some authors [72] claim that caching more than one alternative path has a minimal additional advantage.

Lee and Gerla proposed a protocol called Split Multipath Routing (SMR) in [73]. SMR is an on-demand routing protocol based on DSR. Unlike DSR intermediate nodes are not allowed to respond to RREQs from their cache. This way multiple RREQs reach the target of the discovery and the target selects maximally disjoint routes. In order to maximise the number of RREQs reaching the destination, the route request propagation process is also modified. Intermediate nodes do not necessarily discard any duplicate RREQ packets received, but instead intermediate nodes forward RREQs that are received from different incoming links and whose hop count is lower than the previously received RREQs. The destination node, upon reception of the first RREQ, responds immediately to the instigator of the route request and then waits to receive additional RREQs. A timer is used to determine for how long the destination shall wait for additional RREQs. From the additional RREQs received the destination node is now able to determine a second maximally disjoint path[3]. When it does, the destination node sends a route reply packet for the selected RREQ. Once the instigator receives the first RREQ, it starts sending traffic to the destination to avoid additional delay to the already buffered packets. When the second RREPLY is received the source node allocates traffic using per-packet allocation scheme [81] in a round-robin fashion.

The main drawback of the scheme is increased overhead introduced by relaxing the criteria of the route request propagation process. Additionally, packets are allocated to the two paths with equal probability, which results in the shortest path being used equally along with the sub-optimal maximally disjoint path.

Wang et al. proposed a Multipath Source Routing (MSR) protocol [82, 83]. The proposed scheme was one of the first to consider load balancing. MSR retains the same route discovery procedure as DSR and alternative paths are

[3]If more than one maximally disjoint path exists the shortest is selected. If more than one shortest paths exist the destination selects the one whose RREQ was received first (lowest delay).

discovered through overhearing when the nodes operate in promiscuous mode. The source node upon receiving a route reply maintains additional information about this route. This information is the estimated end-to-end delay a packet will suffer if it is sent over this specific route and a cost weight calculated based on the end-to-end delay of all paths to the same destination (higher latency paths are used less than lower latency paths). Additional work regarding the selection of the weight heuristic equation is presented [84]. The source node, in order to maintain end-to-end delay estimates uses a Round-Trip Time (RTT) tool called SRPing [82]. The SRPing tool, however, introduces additional overhead to the network to perform the round trip measurements. Too frequent pings incur additional protocol overhead, while infrequent pings may result in outdated route cost information.

A similar approach is followed [76] which focuses on node disjoint paths and the weighting function uses an additional consideration; link reliability. The cached cost information is maintained in intervals specified by the source node using a probe technique. Finally, many proposals exist in the literature which attempt to optimise parameters such as energy depletion, end-to-end delay, reduce congestion and combination of those [85, 86, 87, 88, 89, 90].

The other camp in the research community contends that multipath routing does not offer significant merit. Ganjali and Keshavarzian [91] suggests that, in order for multipath routing to offer advantages over unipath counterparts, a huge number of paths must be discovered to improve the load balancing. Essentially, the authors claim that, if a small number of paths is considered, then the alternative paths tend to be physically close to the shortest path. Since devices use a shared medium the congestion is not improved. In order to make improvements routing paths must be such that they push the traffic further from the centre of the network. Subsequently, techniques are proposed which take into consideration path diversity [92, 93]. This problem is also tackled by proposals to use directed antennas to de-couple the paths further [94, 95]. In addition, Pham and Perreau [59] support that multipath routing incurs high overheads and careful consideration must be paid when designing multipath routing protocols. When the protocol attempts to find more than three paths then the overhead is increased significantly. Additionally, the authors calculate the upper bound of the path length in order for multipath routing to offer merit; such a bound depends upon traffic intensity.

The remainder of the chapter evaluates the mechanism proposed in Chapter 3 to see whether multipath routing offers any advantages over unipath protocols

and, if it does, under what conditions. The advantage of the proposed scheme, compared with the previous schemes, is its simplicity. The protocol discovers node disjoint paths to avoid path coupling and the source node selects one of the alternative paths for each packet based on path length. The route request propagation process remains unchanged to avoid significant increase in overhead. The destination node, upon reception of the first RREQ, responds immediately while subsequent RREQs examined to determine if they are node disjoint. If this criterion is satisfied, the destination node equips the instigator with the respective additional node disjoint paths. The proposed mechanism is referred as Multipath Dynamic Source Routing with Node Disjoint Routes (MDSR-NDR).

4.3 Results

Two key sets of experiments were conducted to evaluate the improvements offered by the proposed scheme. The first set applies to static scenarios, while the second considers mobile scenarios for variable numbers of mobile hosts and variable maximum speed.

4.3.1 Static Scenarios

The authors of [91] claim that in networks with high nodal density, the shortest path(s) tend to be very close to the line segments connecting the source and destination pairs. Multipath routing behaves like unipath routing and does not offer any performance improvements. Instead they argue that one has to find paths which push the traffic further from the centre of the network. For this reason this work considers scenarios where the nodal density is high and proves that even in dense scenarios multipath routing offers performance improvements.

The first set is conducted over static topologies of 100 nodes uniformly distributed in an area of 500×200 square metres and each node has range 75 metres. 5,000 connections are scheduled in each simulation run and the duration of each run is 2,000 seconds. The connection duration is selected from a uniform distribution of $0-10$ seconds. Each individual connection is represented by a Constant Bit Rate (CBR) source transmitting packets of 512 bytes each, at an appropriate rate with fixed inter-packet gaps.

Simulations are repeated for 10 random traffic connection patterns and topologies and the load applied to the network (traffic intensity) is varied by changing the rate of the connections. Figure 4.5 shows the goodput (the ratio of data

packets received over the total data packet transmitted, also known as packet delivery ratio) versus the load applied to the network for DSR and MDSR-NDR. The points on the graph represent the mean goodput of the 10 traffic profiles for the corresponding rate, and the error bars represent the corresponding confidence interval of 95%, calculated as described in Section 3.12.2. It can be clearly seen from Figure 4.5, that MDSR-NDR performs better than traditional DSR. Initially (at low loads), the goodput of DSR and MDSR-NDR is almost identical, with MDSR-NDR performing slightly better. As the rate of the connections increases to intermediate loads, the unipath counterpart suffers from congestion and, as a result, the successfully delivered traffic drops, resulting in reduced goodput. On the other hand, MDSR-NDR, distributes the load over multiple paths and suffers less congestion. The performance improvement offered by MDSR-NDR is significant, and in some cases the improvement is almost double. As the rate increases further to high loads, the performance improvement reduces, because the whole network is saturated and any further increase to the injected traffic is dropped immediately.

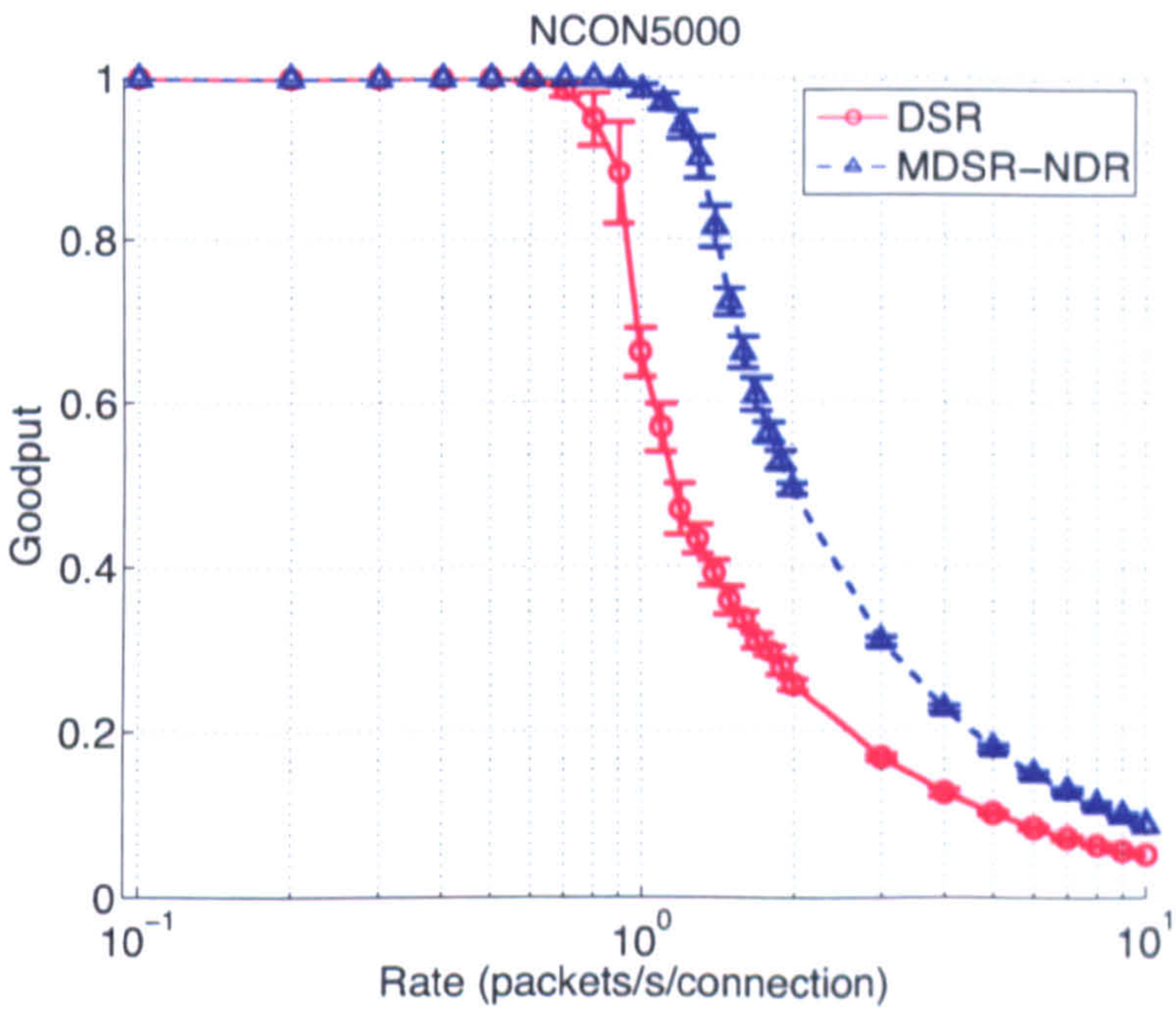


Figure 4.5: Goodput versus Rate for 5,000 scheduled connections.

Figure 4.6 shows the mean path length against the connection rate. The mean path length of MDSR-NDR is equal or higher than that of its counterpart, for all rates. This behaviour was expected, because MDSR-NDR discovers and exploits multiple paths which can be longer than the shortest path, and this justifies the increased mean path length. What might seem irregular is that, at

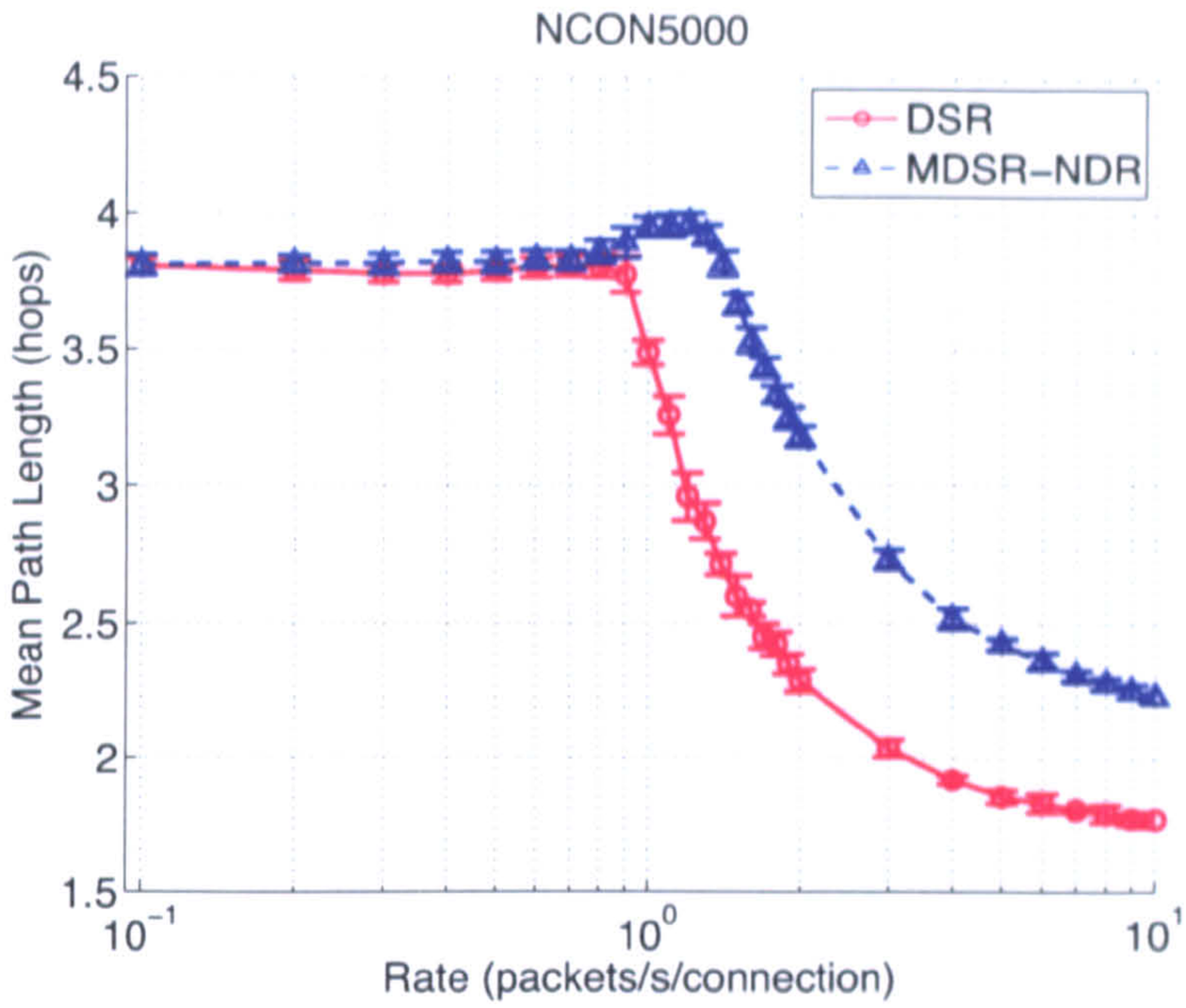
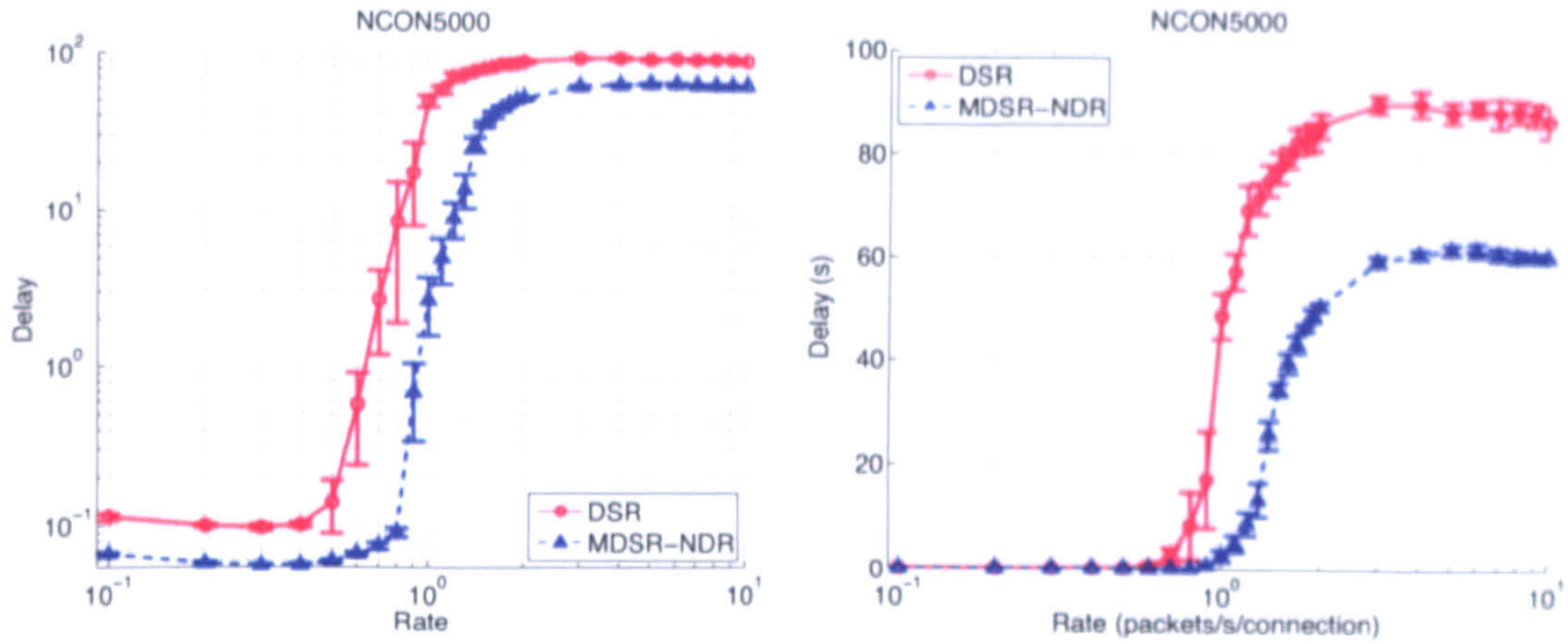


Figure 4.6: Mean path length versus Rate for 5,000 scheduled connections.

low loads, both counterparts have almost identical mean path lengths, and as the rate increases, the two competitors diverge with MDSR-NDR overshooting. However, such behaviour is far from irregular. When the connections rate is low (below 1 packet/second), only a small number of packets is generated for each connection and more traffic is not available to exploit the multiple paths discovered by MDSR-NDR. Consider a connection with duration of 5 seconds and rate of 1 packet/second. A connection with such characteristics will inject into the network a total of 5 packets. Going at even lower rate of 0.2 packets/second, results in a connection which injects only one packet to the network. Therefore, the lower the number of packets transmitted per connection, the lower the chances for MDSR-NDR to select a path longer than the shortest path. These chances are further reduced when MDSR-NDR applies the selection probability which is inversely proportional to path length. As the rate of the connections increases to intermediate rates, MDSR-NDR has the chance to exploit the additional discovered paths and this results in an increase of the mean path length. However, when the rates are increased even further, network congestion is increased, packet loss starts to occur. Packet loss is more likely to occur in packets propagated over long paths, leading to decreased mean path length.

Another metric of interest is the average end-to-end delay suffered by the traffic in the network. Figure 4.7a and 4.7b shows how average end-to-end delay varies against the connection rate for logarithmic and linear y-axes respectively.



(a) Logarithmic delay axis (b) Linear delay scale

Figure 4.7: Average end-to-end delay versus Rate for 5,000 scheduled connections.

MDSR-NDR offers improvement compared to its unipath counterpart for all rates. Initially, the improvement of MDSR-NDR is almost double. For example, for rate 0.2 packets/s per connection, the average end to end delay is 56.92 ms and 99.63 ms for the multipath and unipath schemes respectively (Figure 4.7a). Once the rate is further increased, the end-to-end delay is increased for both schemes because congestion increases and the data packets are queued in the node buffers, until they get served. When the rate increases and exceeds the 2 packets/second mark (Figure 4.7b), the delay remains almost constant and this is expected because the network reaches saturation and any more injected traffic is lost immediately due to the finite buffer length of the nodes.

Figure 4.8 shows the normalised routing load against the connection rate. Normalised routing load is the ratio of routing overhead packets inserted to the network over the number of successfully delivered data traffic. The routing overhead packets consists of Route Request (RREQ), Route Reply (RREPLY) and Route Error (RERR) packets. This metric is an indicator of protocol efficiency and a relative measure of control packets (routing overhead) inserted by both schemes for a given traffic intensity (number of connections and rate). MDSR-NDR offers higher efficiency (lower normalised routing load) through-out the graph.

Initially, both schemes have high normalised routing load and the efficiency of both protocols is low. This is because both schemes perform route discoveries to facilitate connection requests and when routes are found to serve these request, only a small number of packets are routed per connection. As the connection

rate increases, both counterparts become more efficient because the delivered data traffic per connection is increased, while the number of route discoveries remain constant. DSR and MDSR-NDR reach the maximum efficiency at rates of 0.7 and 1 packets/second respectively.

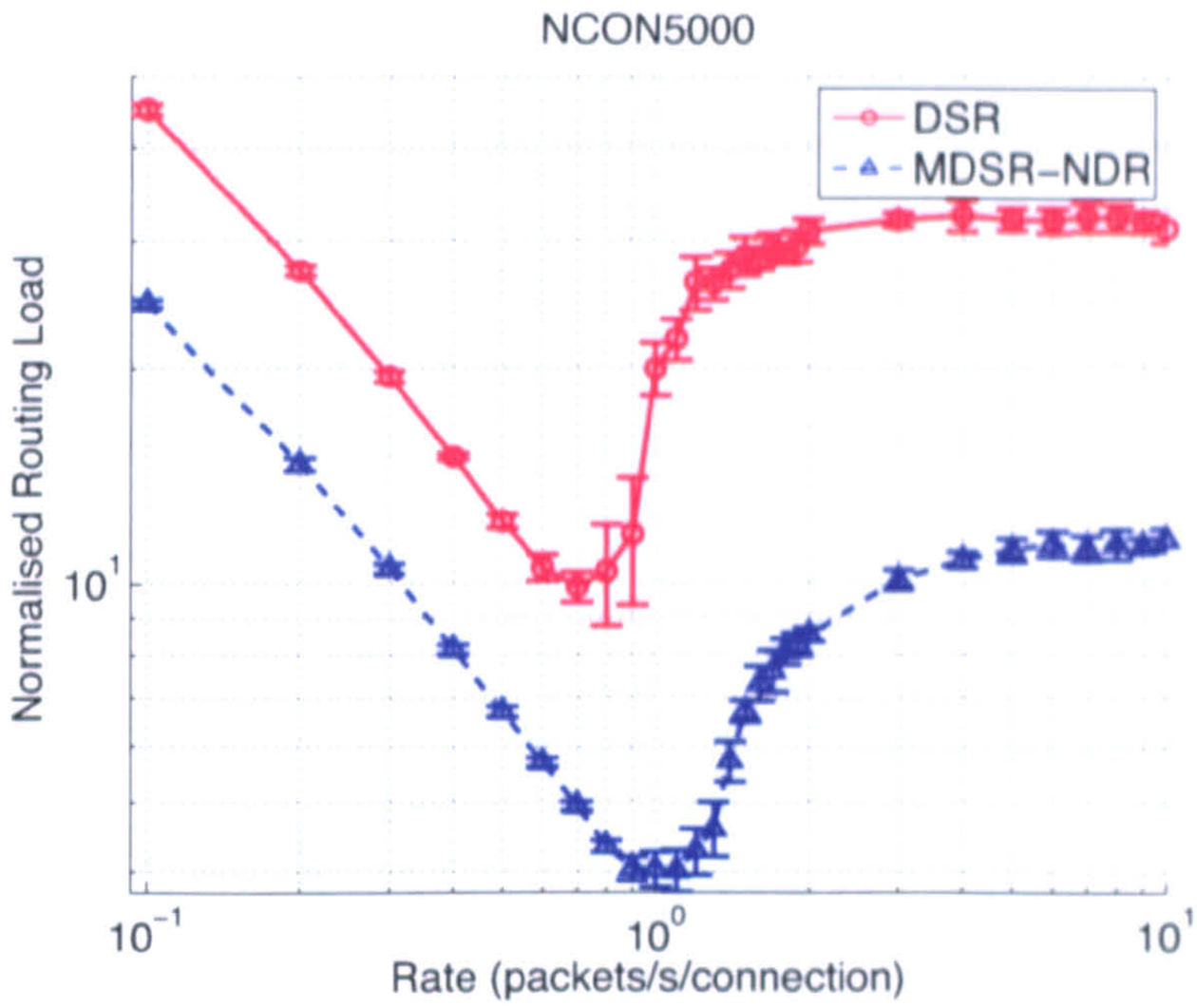


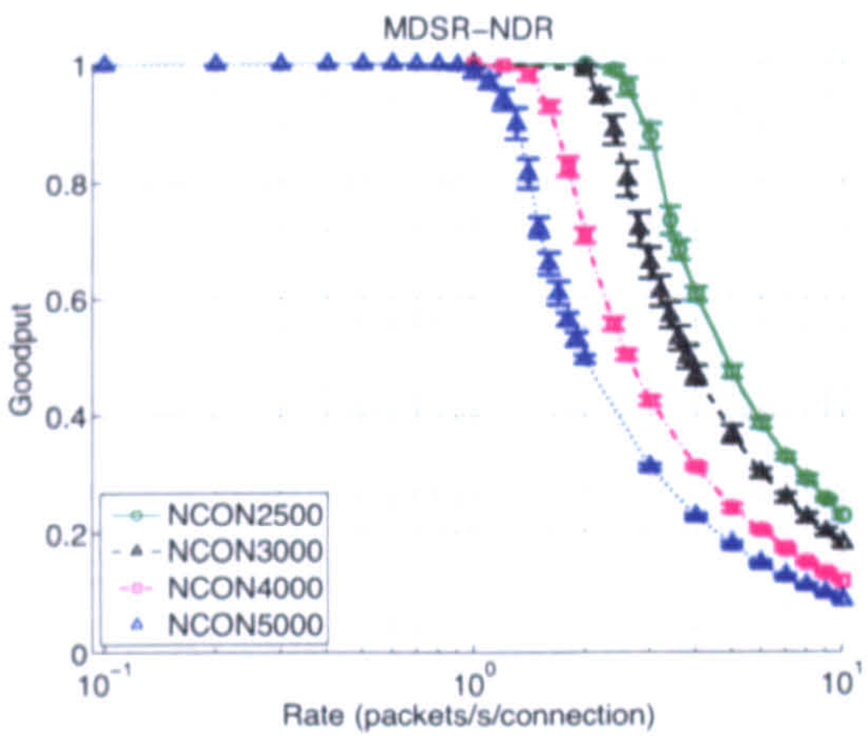
Figure 4.8: Normalised Routing Load versus Rate for 5,000 scheduled connections.

Further increases in the connection rate results in congestion and traffic loss starts to occur. When the maximum number of retransmits is reached, the MAC layer notifies the routing layer that it was unable to deliver the traffic to the next hop, and the routing protocol generates a Route Error (RERR) packet to notify the source of the connection that the path is broken. As a result, the source node searches its cache for alternative paths to route its traffic and if none is found, a new route discovery procedure is triggered. However, DSR does not have any alternative paths cached and triggers route discoveries, which increase the routing overhead and consequently,the normalised routing load is increased significantly. On the other hand, MDSR-NDR has alternative routes cached and the affected traffic is switched to one of the alternative paths. MDSR-NDR triggers new route discoveries only when all the cached paths are broken, resulting in lower normalised routing load compared to its unipath competitor.

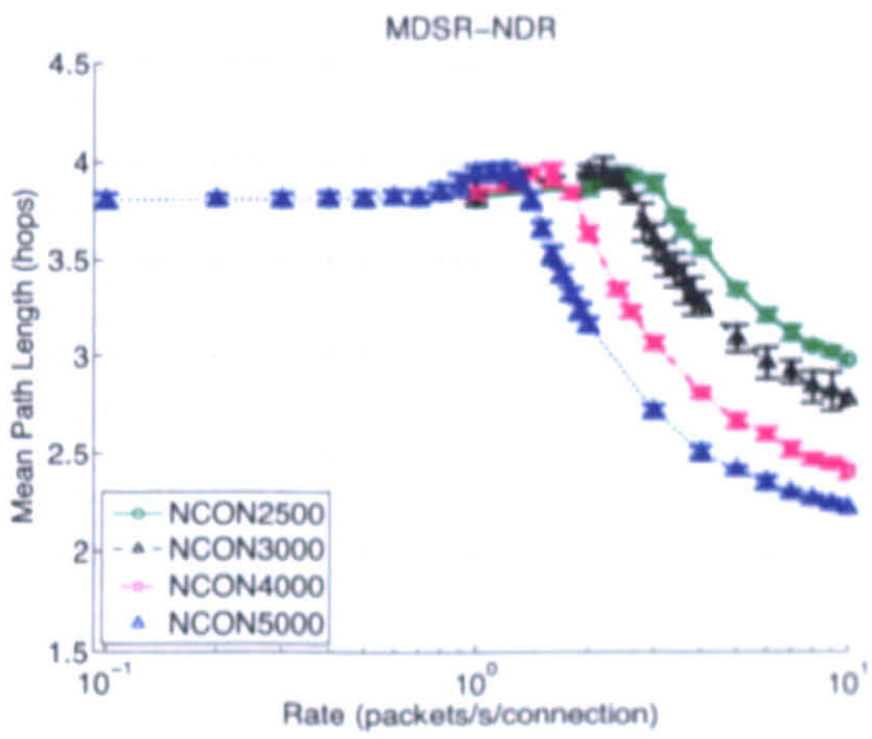
Observing the results for 5,000 number of connections, it is clear that MDSR-NDR offers significant benefits over DSR for static topologies, and this is especially true at intermediate to high loads. For low rates, it might seem that

both schemes had almost identical performance in terms of goodput, mean path length and delay, however, MDSR-NDR achieves this performance with lower routing overhead and this parameter is of crucial importance, when the network nodes are battery powered and energy consumption becomes an issue. Higher protocol efficiency (lower normalised routing load) results in lower rate of energy depletion. Energy constraint routing is studied in later chapters 5 and 6.

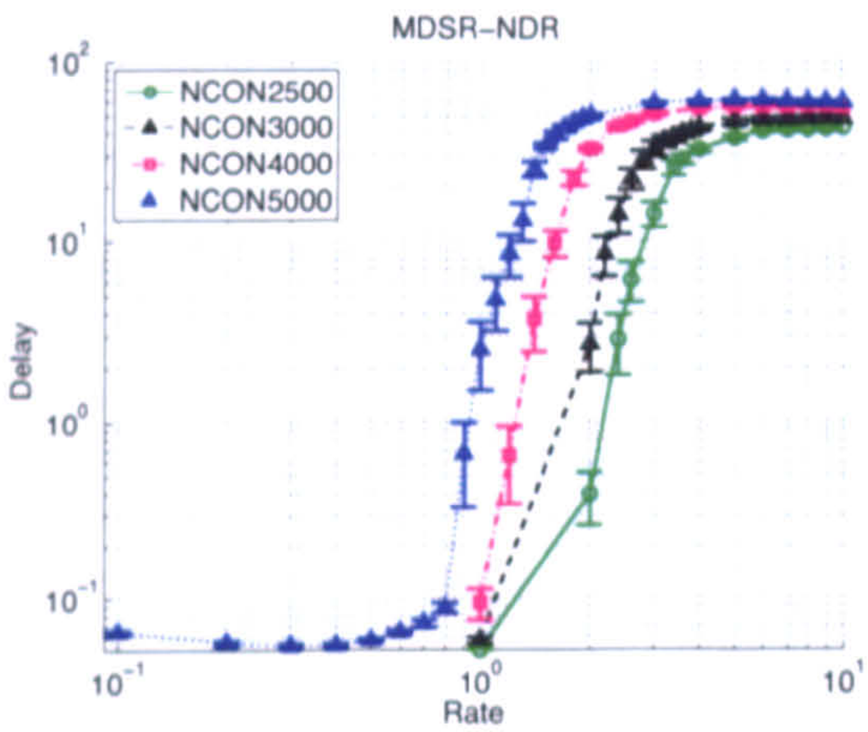
The same set of experiments were carried out for variable number of connections, keeping all the remaining parameters the same. In all cases, the comparative graphs between DSR and MDSR-NDR led to the same conclusion, but the purpose of these graphs was to quantify how the varying number of connections affects the network performance when MDSR-NDR is used.



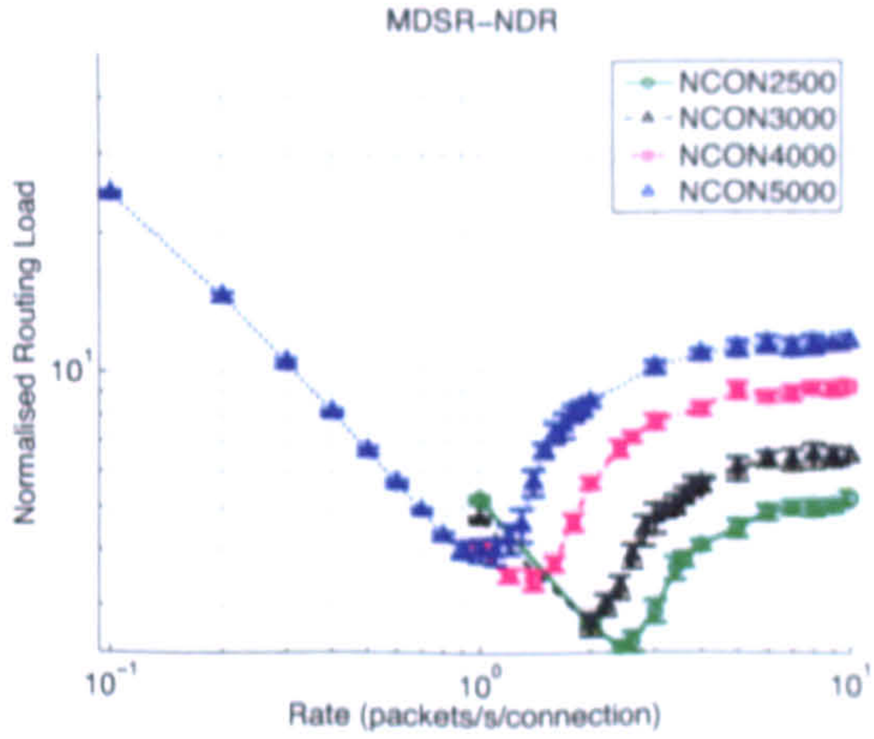
(a) Goodput.



(b) Mean Path Length.



(c) Average end-to-end delay.



(d) Normalised Routing Load.

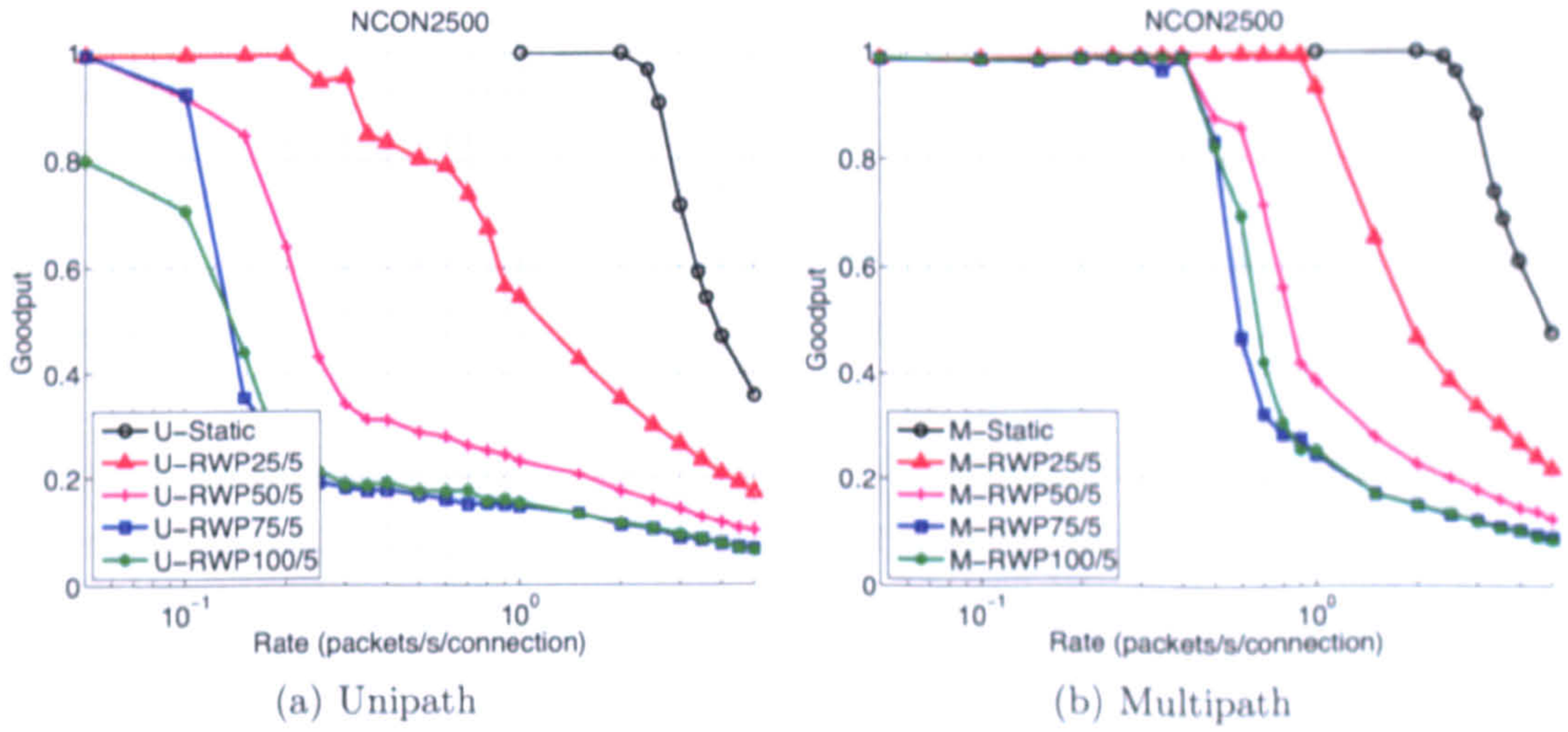
Figure 4.9: MDSR-NDR performance metrics for various number of connections.

4.3.2 Mobile Scenarios

Clearly, modelling routing protocols for mobile ad-hoc networks in a static environment has limited value. The second part of the experimental work considers scenarios where nodes are able to move. To maintain consistency, the number of nodes, transmission range and simulated area remain the same as in the static case (100 nodes, 75 metres and $500 \times 200\ m^2$). However, nodes now have mobility capabilities following the modified random waypoint mobility model with zero pause time as described in Section 3.11. The experiments were performed for 2,500 connections and the duration of each connection is still selected from a uniform distribution of 0-10 seconds. The experiments are repeated for 10 random traffic and movement scenarios.

4.3.2.1 Variable percentage of mobile hosts, constant maximum speed

In the experiments of this set, the mobile intensity changes by varying the percentage of mobile nodes, while all mobile nodes select their speed from a uniform distribution between $[1, V_{max}]$, where V_{max} is 5 metres/second (18 km/h). The experiments are conducted for 25%, 50%, 75% and of nodes in the network being mobile and marked in the graphs as RWP25, RWP50, RWP75 and RWP100 respectively. The prefix U (unipath) is used in the labels to signify DSR and M for MDSR-NDR. Figure 4.10 shows the goodput versus rate when the mobile intensity is changing by varying the number of mobile hosts in the network. As the percentage of mobile nodes increases, the goodput drops for both protocols,



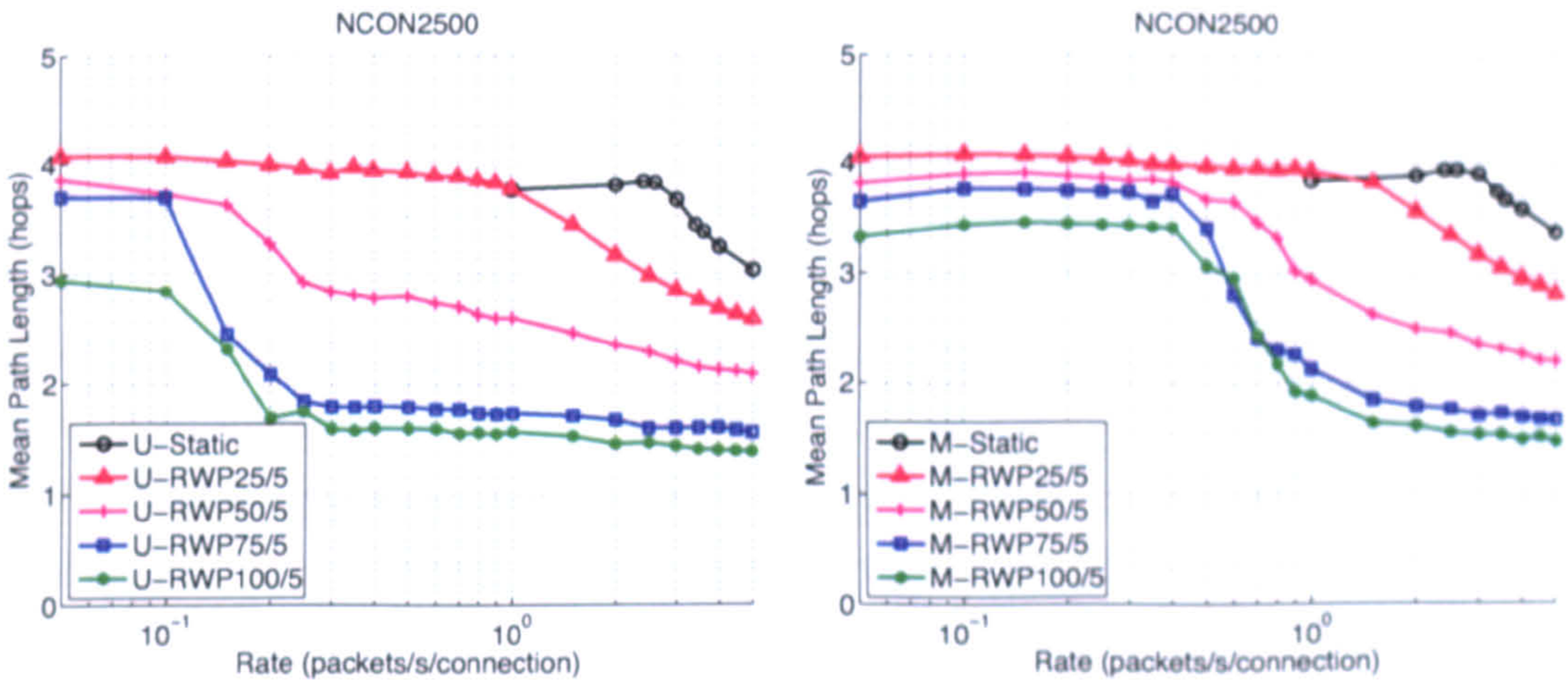
(a) Unipath (b) Multipath

Figure 4.10: Goodput versus Rate for speeds selected between [1-5] m/s with 0%, 25%, 50%, 75% and 100% of mobile hosts.

but, in all cases, the performance of MDSR-NDR is higher compared to the unipath counterpart.

It is worth noting, that when the percentage of mobile nodes increase from 75% to 100%, the goodput versus rate performance for both counterparts remains the same. This is because the mobility intensity of 75% and 100% results in a similar average number of path failures and any difference in goodput occurs due solely to the random nature of the traffic profiles and the mobility scenarios. However, MDSR-NDR, in both cases, achieves higher packet delivery ratio than DSR.

Figure 4.11 shows the mean path length versus the rate. When the mobile intensity increases, the mean path length drops. Increasing the percentage of mobile nodes, the probability of path failures increases, because nodes move and the network topology is changing. Long paths have even higher probability of failure and this can be observed by the reduced average packet length achieved by the highly mobile scenarios; as it previously explained in Section 4.3.1.

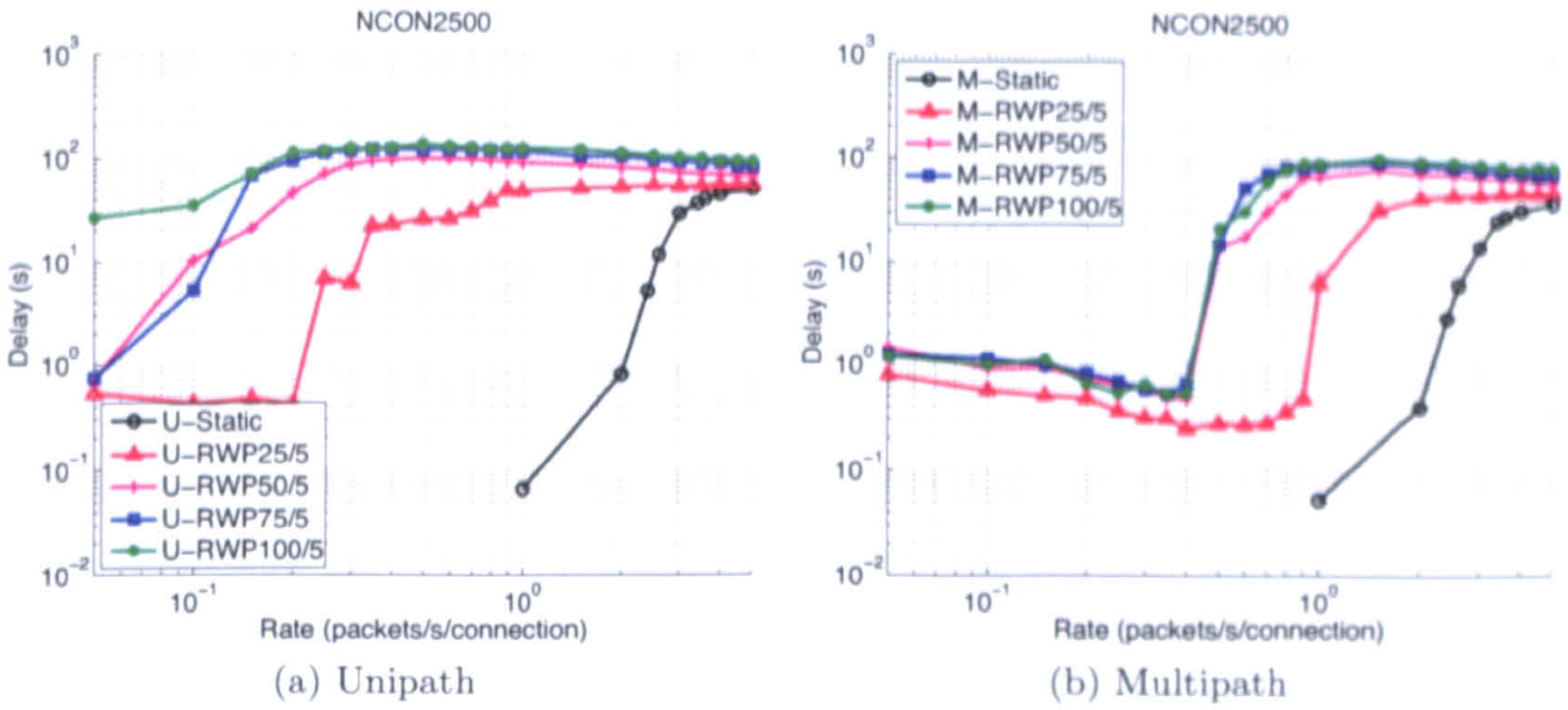


(a) Unipath (b) Multipath

Figure 4.11: Mean Path Length versus Rate for speeds selected between [1-5] m/s with 0%, 25%, 50%, 75% and 100% of mobile hosts.

Figure 4.12 shows that average end-to-end delay versus rate for different mobile intensities. Observing the end-to-end delay in pairs between the unipath and the multipath counterparts, it is clear that MDSR-NDR offers significant improvement especially at low mobile intensities.

Initially, at low rates, the delay of MDSR-NDR for all mobile intensities remains low while the respective delay for the unipath counterpart is higher. Another effect to note is that as the load increases the average end-to-end delay



(a) Unipath (b) Multipath

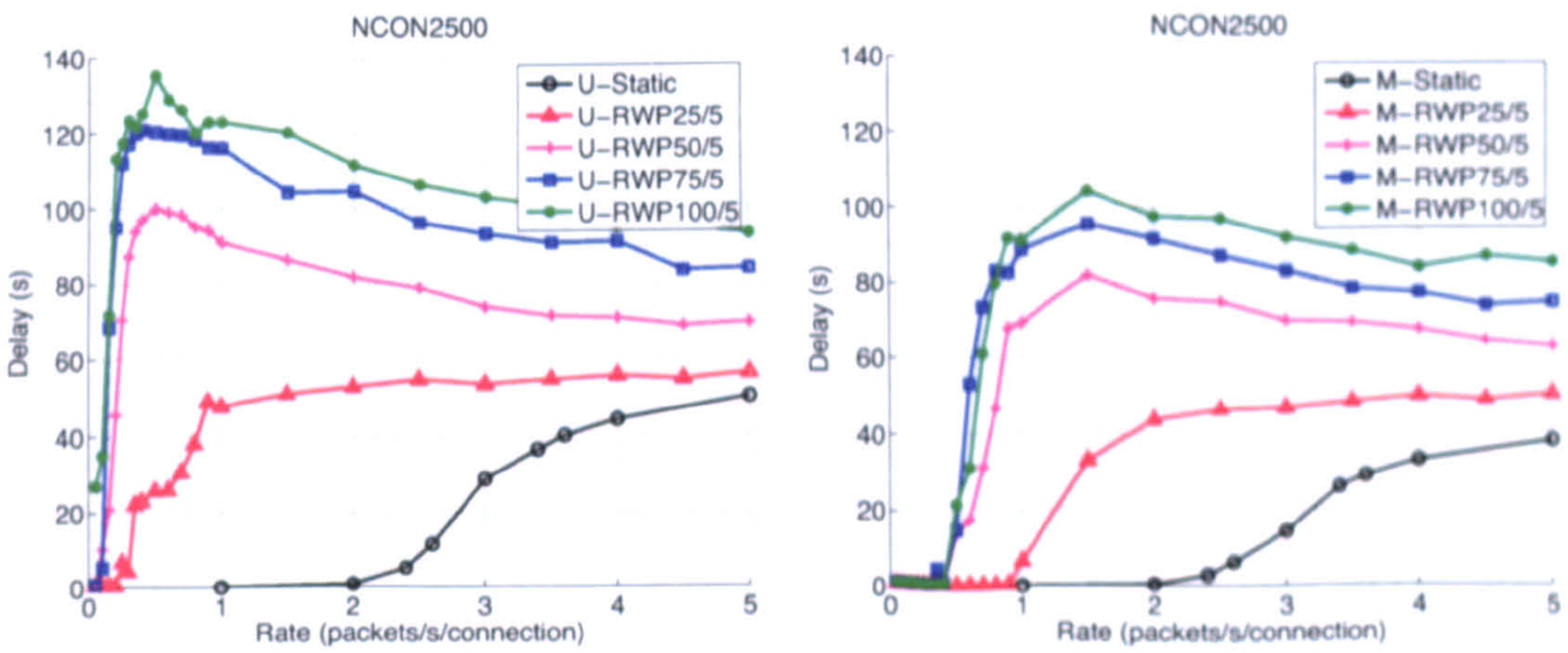
Figure 4.12: Average End-to-End Delay versus Rate for speeds selected between [1-5] m/s with 0%, 25%, 50%, 75% and 100% of mobile hosts (Logarithmic scale).

drops. To explain this effect consider the following: When traffic is generated for a new destination the routing protocol initiates route discoveries and the packets are buffered until a path becomes available. Subsequent packets have a path readily available and are dispatched immediately. At low rates, only a few packets are sent per connection and the ratio of number of packets buffered waiting for a path to become available over the number of packets which are dispatched immediately is high, resulting to a high average end-to-end delay. As the load increases, this ratio decreases because the packets that have a path readily available increase resulting in a reduced average end-to-end delay. When the load increases further, the end-to-end delays start to increase an impact of buffered packets in the intermediate nodes due to the increased congestion. At high loads, the network reaches saturation and the end-to-end delay can appear to be reduced.

The reader is reminded that DSR does not exploit the route propagation procedure to discover additional paths and consequently, when a path failure occurs a new route discovery procedure is triggered. MDSR-NDR, on the other hand, mitigates the route discovery "storms" by discovering multiple paths. Since DSR no longer has available routes to the destination, the data packets are delayed in the node buffers, waiting for a path to become available and this explains the increased delay. As a side effect, the additional RREQ packets in the network increase congestion and delay increases even further. Because the overall route discovery procedure is delayed, the nodes time-out and assume that

the discovery failed and trigger new route discoveries. The effect is a "vicious circle"; route discoveries fail because of the increased delay and additional route discoveries increase the delay further. Additionally, goodput drops because the buffers become full of queued packets waiting for paths to be discovered and this is evident in goodput graph of Figure 4.10. Finally, the impact of the route discovery "storms" can be seen in the normalised routing load graph, shown in Figure 4.14. In fact, for RWP100, DSR injects into the network 120 packets of routing overhead for every successfully delivered data packet.

Before discussing the normalised routing load metric, it is appropriate to first consider why end-to-end delay drops as the rate approaches relatively high values (Figure 4.13), which contradicts the constant delay behaviour experienced under static scenarios. At high loads, congestion is increased and traffic propagated through long paths has a higher probability of being dropped and only traffic propagated through short paths is delivered successfully. Since the average end-to-end delay is measured only for packets which are successfully delivered, this results in reduced end-to-end delay. This can also be verified by the mean path length and goodput achieved at these rates. On the other hand, in static scenarios, the delay is constant when the network reaches saturation; delay is only caused by network congestion of data packets and not due to path failures of long paths and hence it is expected to be constant.

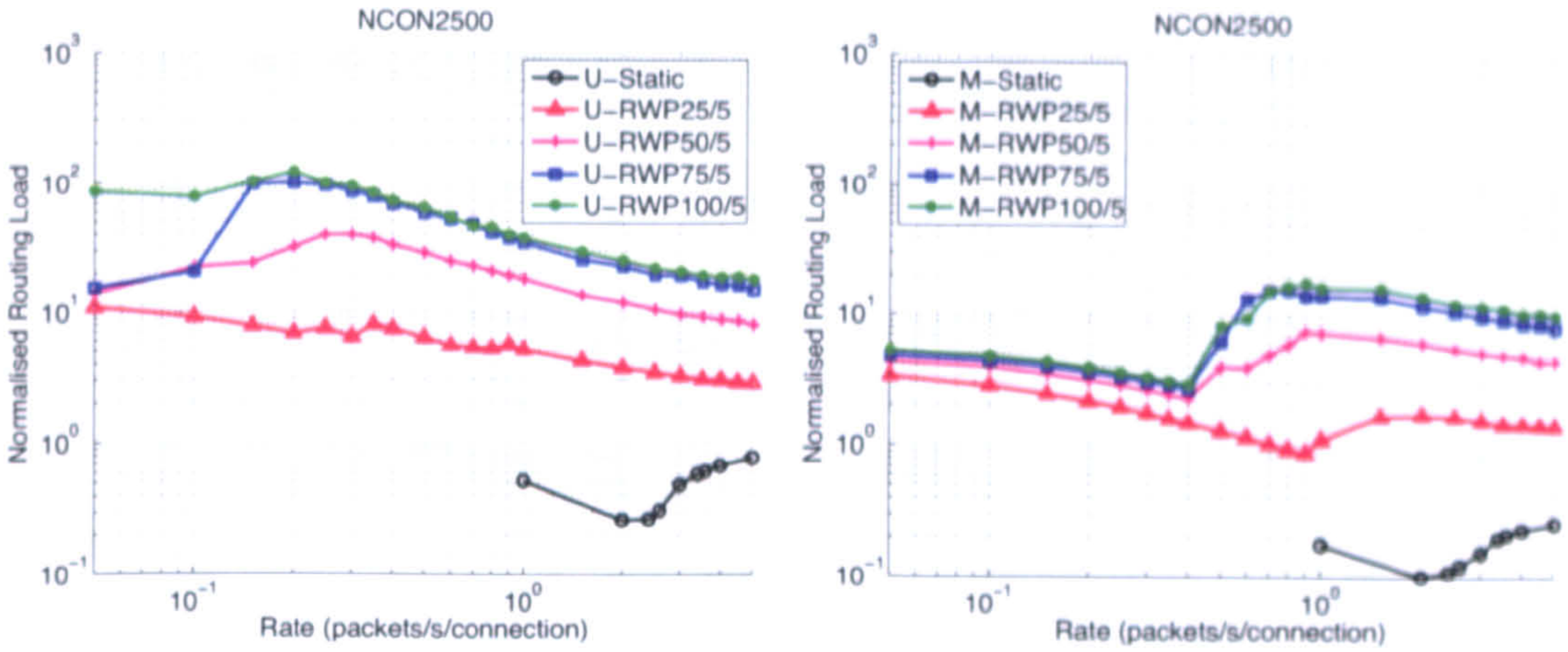


(a) Unipath (Linear scale)

(b) Multipath (Linear scale)

Figure 4.13: Average End-to-End Delay versus Rate for speeds selected between [1-5] m/s with 0%, 25%, 50%, 75% and 100% of mobile hosts (Linear scale).

Figure 4.14 shows the normalised routing load. As expected, high mobile intensities result in more frequent route failures which, in turn, trigger route discoveries and the protocol efficiency drops (higher normalised routing load).

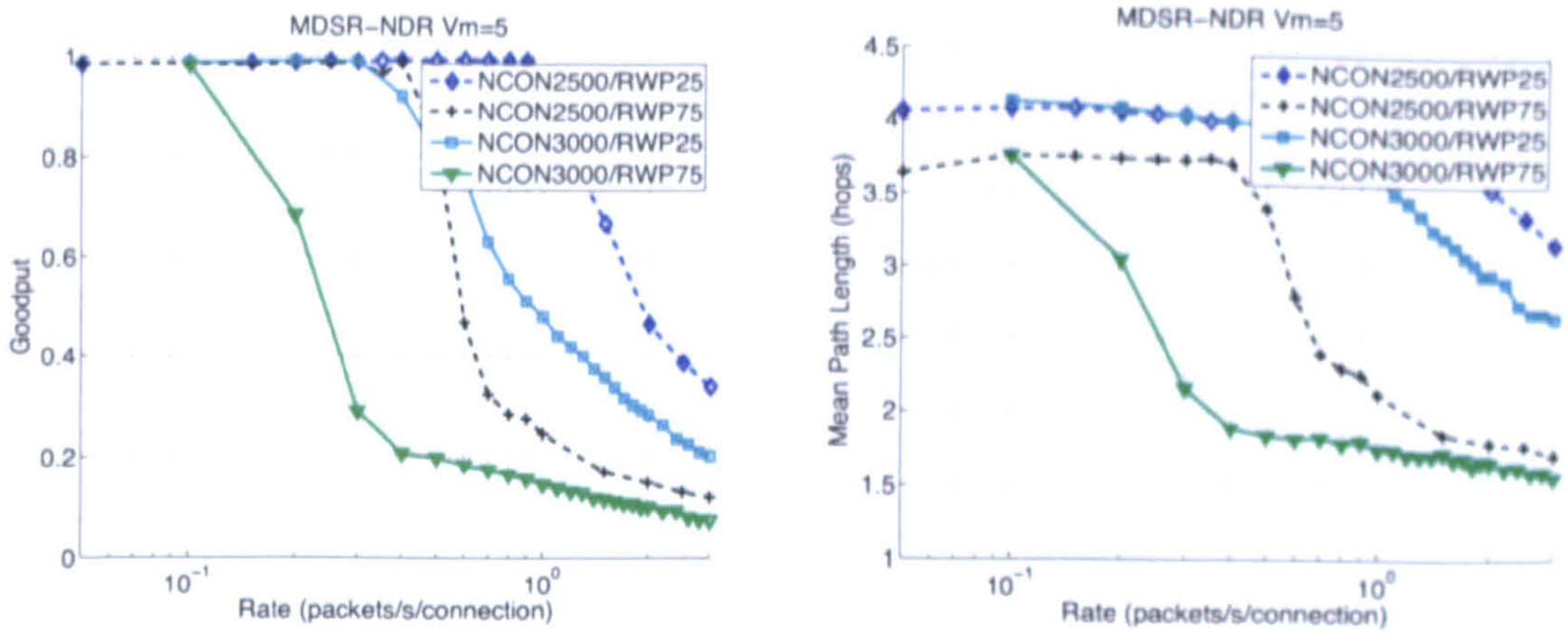


(a) Unipath (b) Multipath

Figure 4.14: Normalised Routing Load versus Rate for speeds selected between [1-5] m/s with 0%, 25%, 50%, 75% and 100% of mobile hosts.

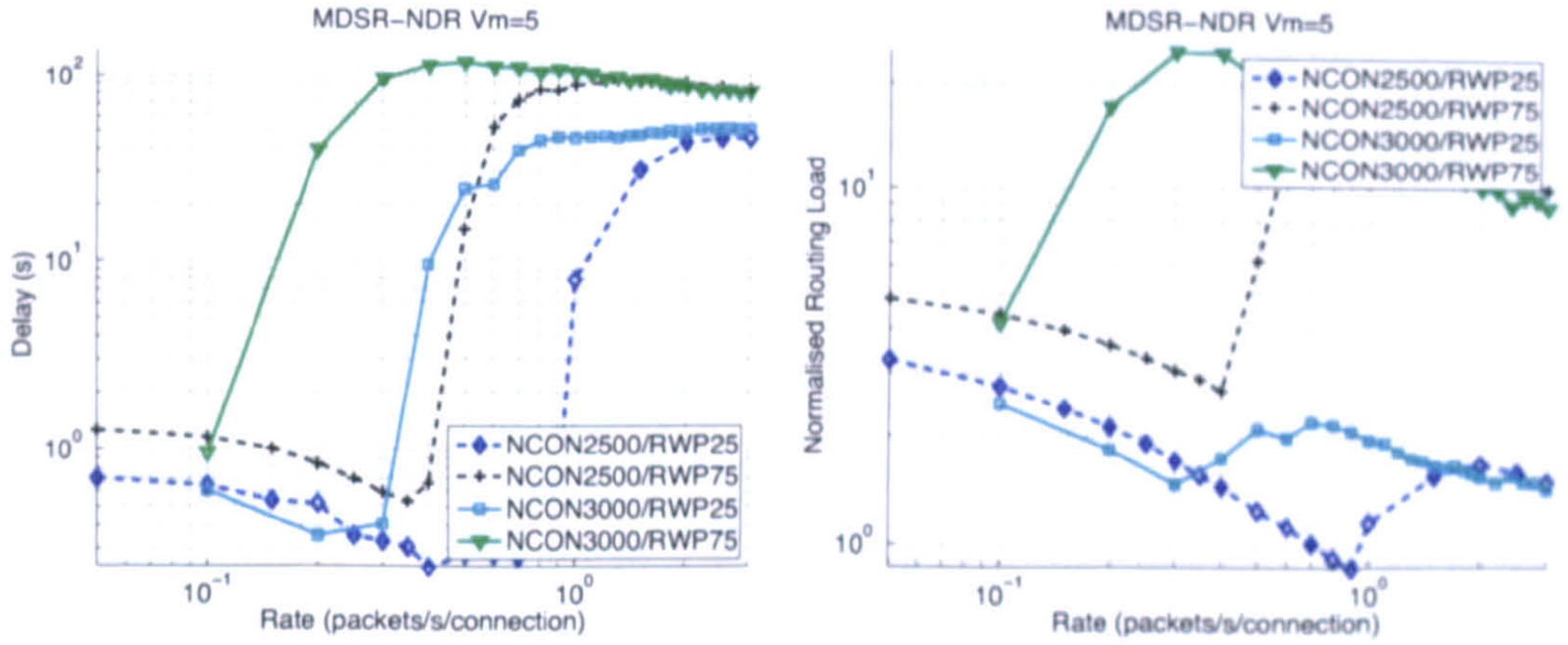
Again, what might seem irregular is the normalised routing load for U-RWP50, U-RWP75 and U-RWP100 overshoots and then starts to drop, indicating that the protocol efficiency is increased for higher rates. The increased protocol efficiency is an illusion, but this behaviour is far from irregular. At low rates, the packet interarrival time is very long and hence long lasting routes (usually 1 or 2 hops long) serve very little traffic. As the connection rate increases, these long lasting routes are exploited, which results in an increase in the successfully delivered data packets; hence the routing load decreases. In this connection rate region, the normalised routing load is very sensitive and doubling the successfully delivered traffic results in a big drop in routing load. For example, the routing load for RWP75 at 0.2 packet/second is 99.23. For double the connection rate (0.4 packets/second), the routing load becomes 68.27, while the load becomes 41.48 for 0.8 packets/second. However, MDSR-NDR is more efficient throughout the range and, for 5 packets/second, the normalised routing loads for DSR and MDSR-NDR are 14.99 and 8.16 respectively.

The same experiments were repeated for 3,000 connections and all the other parameters remained the same. The results for RWP25 and RWP75 are shown in Figures 4.15 and 4.16 and lead to the same conclusion as previously.



(a) Goodput. (b) Mean Path Length.

Figure 4.15: Goodput and Path Length for variable percentage of mobile hosts; maximum speed 5 metres/second.

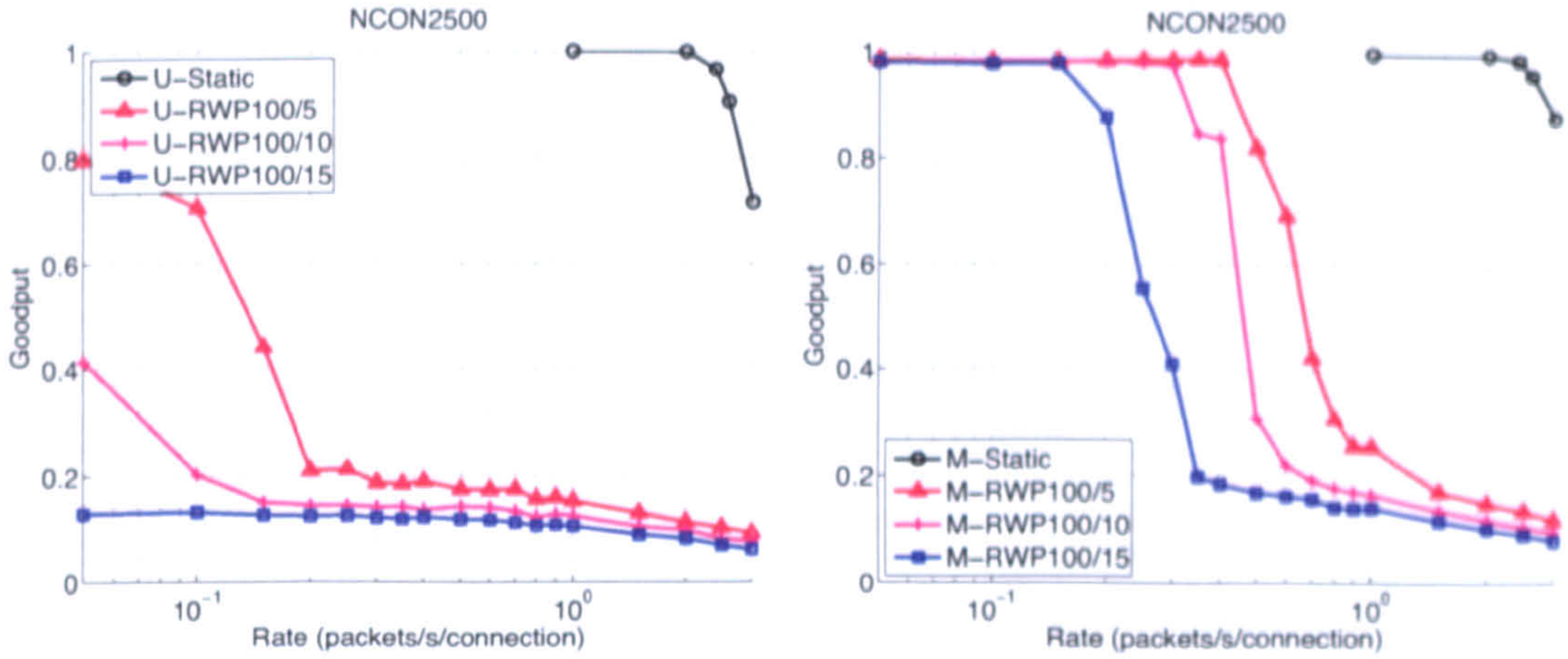


(a) Average end-to-end delay. (b) Normalised Routing Load.

Figure 4.16: Delay and Normalised Routing Load for variable percentage of mobile hosts; maximum speed 5 metres/second.

4.3.2.2 All hosts mobile, variable maximum speed

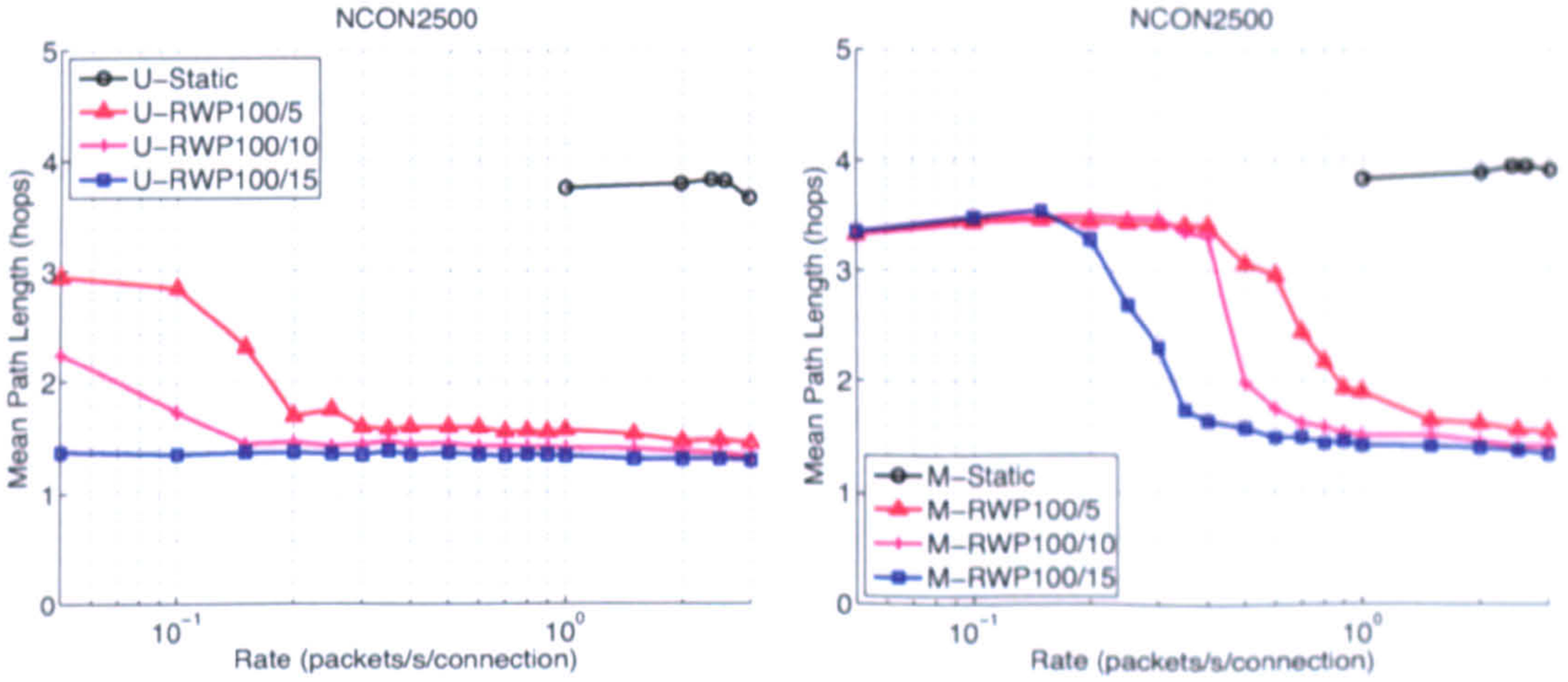
In this set of experiments, all nodes are mobile (RWP100) and select their speed from a uniform distribution between $[1, V_{max}]$, where V_{max} is 5, 10 and 15 metres/second (18, 36, 54 km/h respectively). The resultant goodput versus load for MDSR-NDR and traditional DSR are presented in Figure 4.17. As the mobility intensity increases, the goodputs of both MDSR-NDR and DSR decrease. For all mobility intensities and packet rates, the multipath protocol outperforms its unipath counterpart.



(a) Unipath (b) Multipath

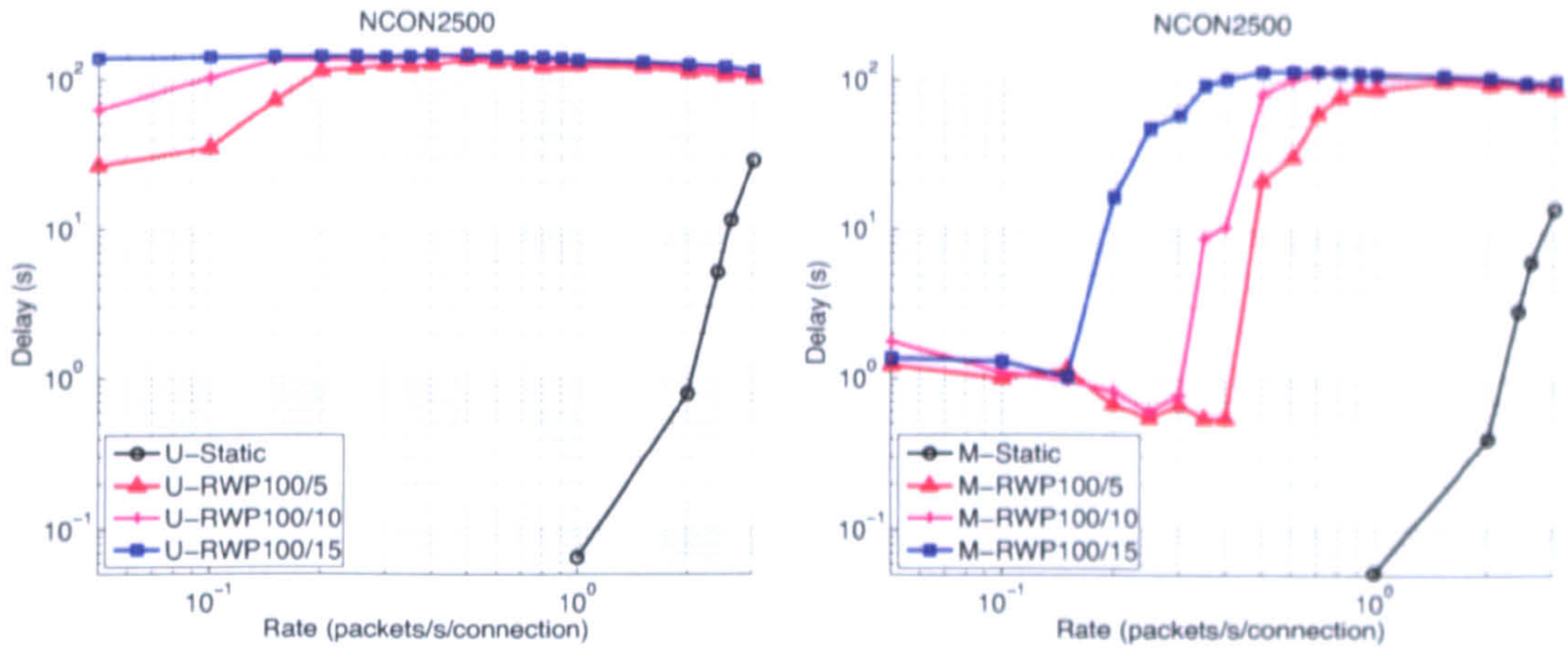
Figure 4.17: Goodput versus Rate for all nodes mobile with variable maximum speed.

Figure 4.18 shows the mean path length versus rate. In all cases, the multipath protocol has, as expected, higher average path length. Increasing the connection rate, congestion appears and traffic propagated over long paths is dropped, resulting in a decrease in mean path length. The end-to-end delay is shown in Figure 4.19.



(a) Unipath (b) Multipath

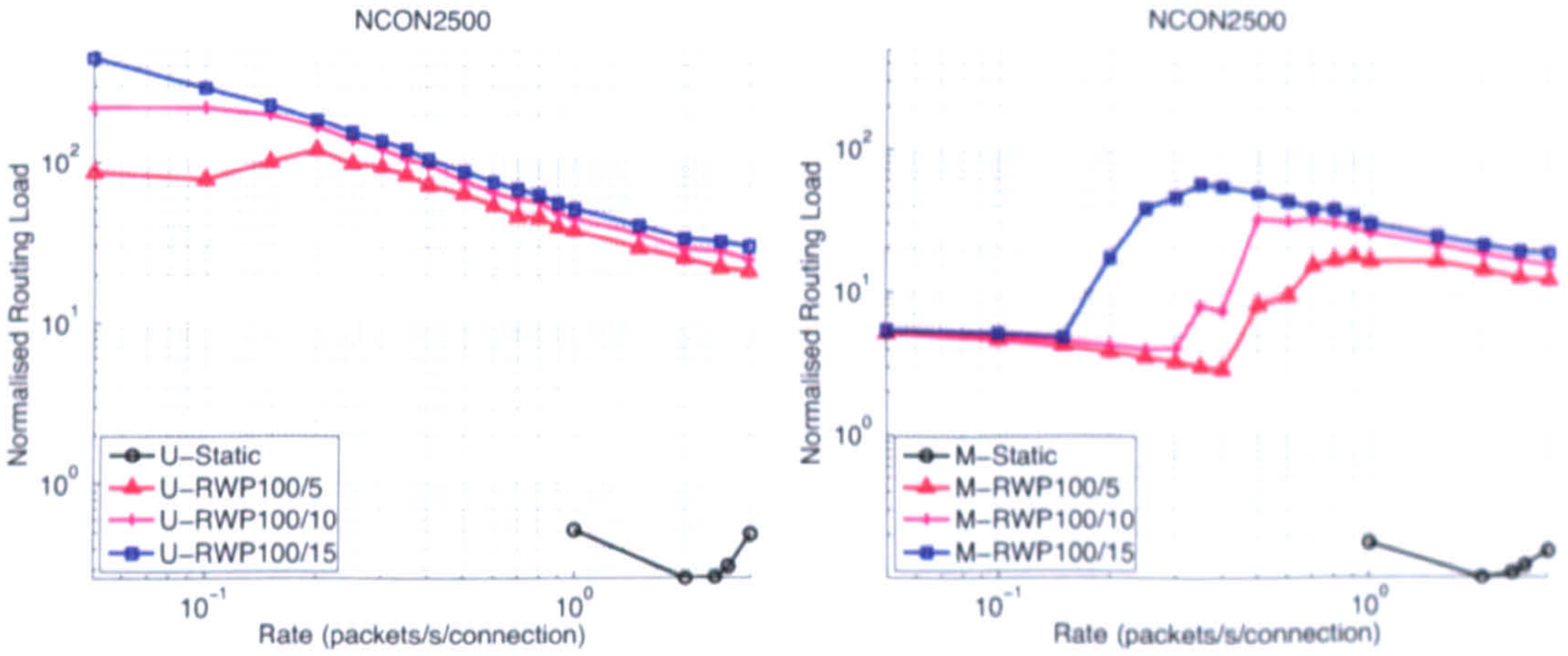
Figure 4.18: Mean Path Length versus Rate for all nodes mobile with variable maximum speed.



(a) Unipath (b) Multipath

Figure 4.19: Average End-to-End Delay versus Rate for all nodes mobile with variable maximum speed.

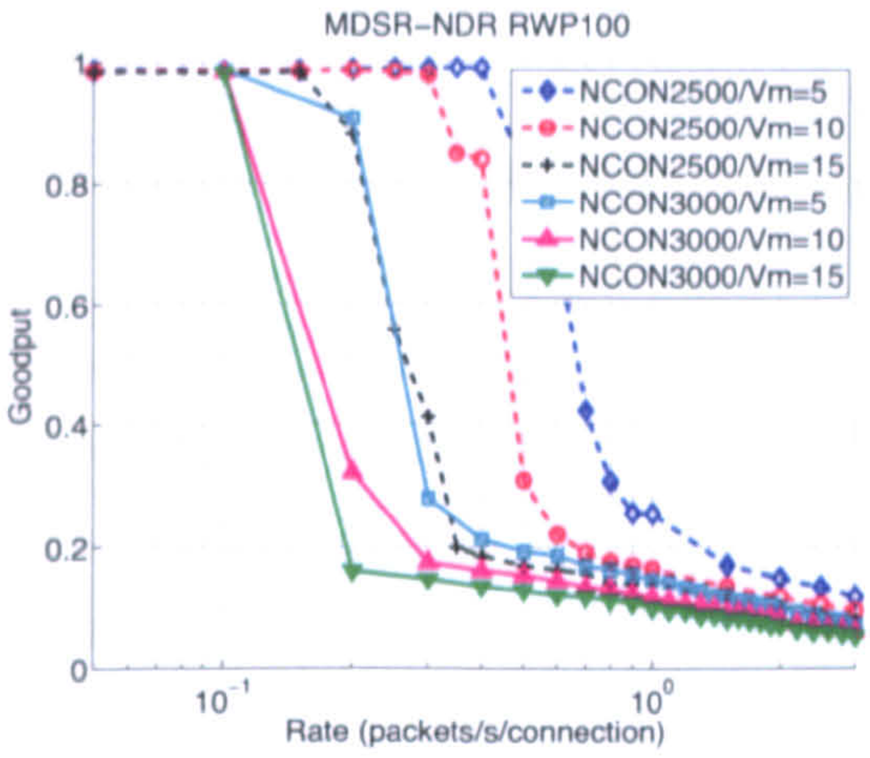
Finally, Figure 4.20 shows the normalised routing load versus rate. For a decreasing mobile intensity, the efficiency of the routing protocols increases as expected. The pattern followed is identical to the one observed in Section 4.3.2.1.



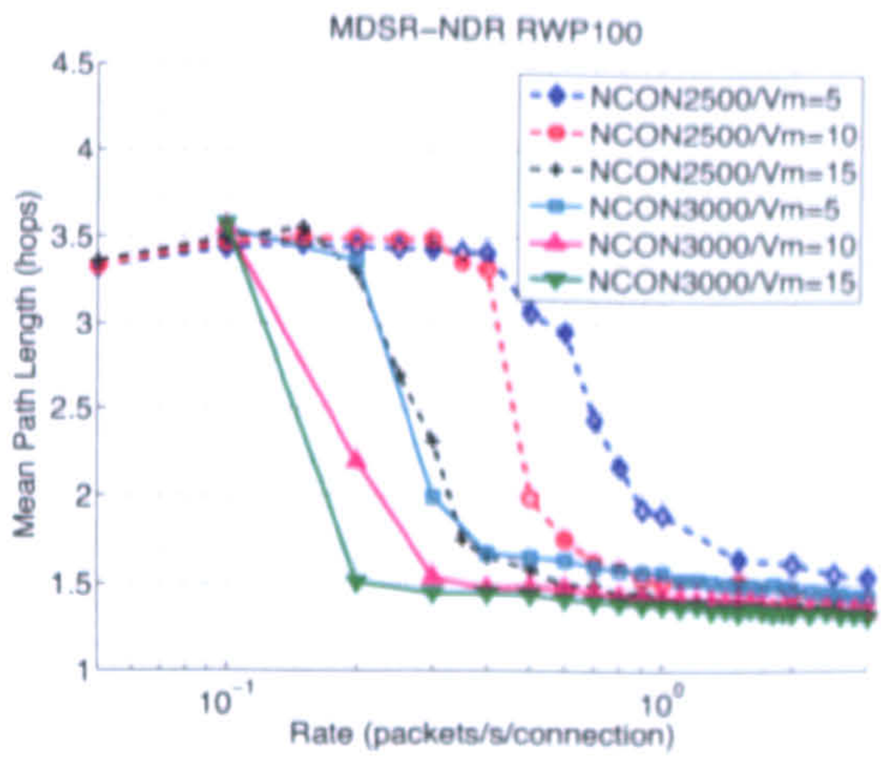
(a) Unipath (b) Multipath

Figure 4.20: Normalised Routing Load versus Rate for all nodes mobile with variable maximum speed.

For completeness, the Figures 4.21 and 4.22 show the goodput, mean path length, delay and normalised routing load for mobile intensities for different maximum speeds.

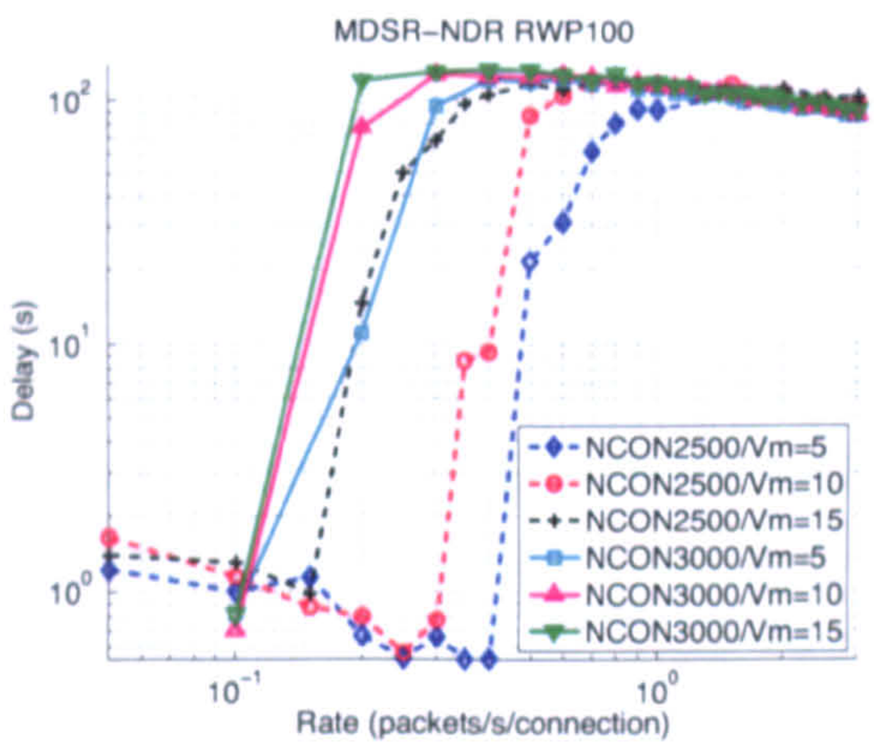


(a) Goodput.

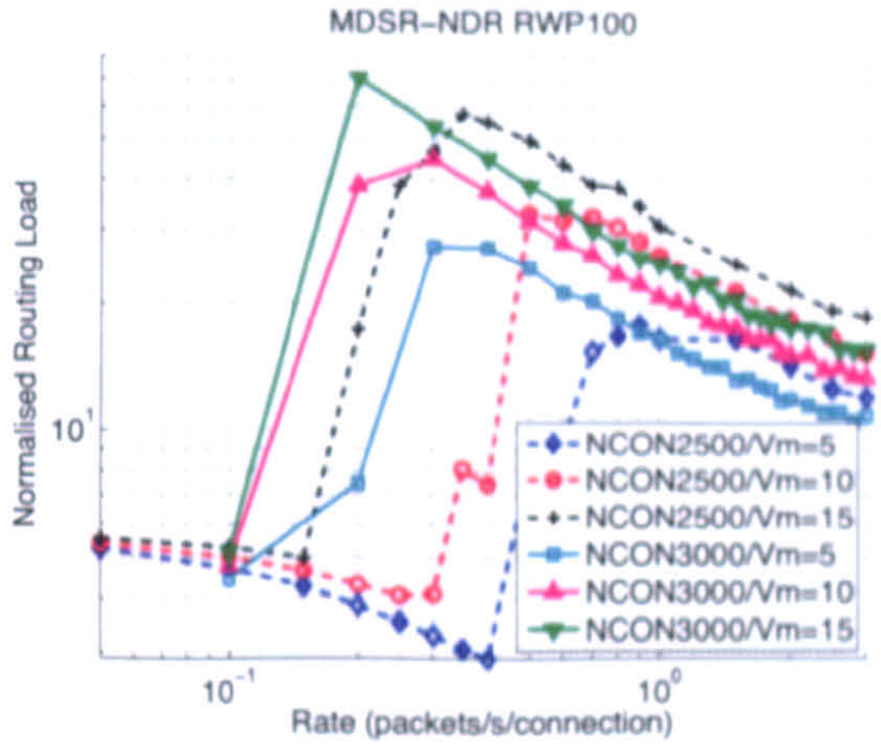


(b) Mean Path Length.

Figure 4.21: Goodput and Path length for all hosts mobile and variable maximum speed.



(a) Average end-to-end delay.



(b) Normalised Routing Load.

Figure 4.22: Delay and Normalised Routing Load for all hosts mobile and variable maximum speed.

4.4 Summary

The proposed routing scheme, Multipath Dynamic Source Routing with Node Disjoint Routes (MDSR-NDR) offers improved performance in terms of goodput, end-to-end delay and higher protocol efficiency by exploiting multiple node disjoint routes. Network congestion is reduced because the proposed protocol distributes the load over multiple paths. The merits of the proposed scheme are quantified for variable number of connections and different rates. MDSR-NDR

offers clear improvement over its unipath competitor in static dense topologies, with significant benefits in goodput under intermediate and high loads while the benefit in delay and protocol efficiency is clear under all loads.

The performance of MDSR-NDR under mobile scenarios also demonstrates the merits of the algorithm. Two sets of experiments were conducted to quantify the benefits under mobility; for variable number of mobile nodes and for variable maximum speed. In both cases, MDSR-NDR outperformed DSR and the necessity of the multipath routing schemes becomes more apparent in mobile scenarios.

This is in contrast with the views expressed by others [91], where it is claimed that multipath approaches do not offer benefits in highly dense networks unless a large number of paths is employed. This chapter considered multipath routing in dense networks and employed three paths [59] to examine that very scenario and it has shown that multipath routing offers benefits compared to its unipath counterpart. Additionally, the scheme requires simple modifications to the existing DSR protocol which makes the proposed scheme (MDSR-NDR) attractive for existing network deployments. Finally, the probability function used to allocate traffic to the alternative paths is favourable for the shortest path to reduce the end-to-end delay.

Multipath routing is indeed advantageous and the next chapter considers multipath routing from the energy perspective. Two novel route selection schemes are proposed, which extend the network lifetime through energy homogenisation. The analytical model of the two route selection schemes is developed and a C++ simulator is used to demonstrate and quantify their benefits over traditional/benchmark energy aware routing protocols. It is also demonstrated that selecting routes, which homogenise the energy map of the network, can prolong the network lifetime and reduce network partitioning.

Chapter 5

Network Lifetime and Energy Map Homogeneity

The previous chapter discussed multipath routing and it was shown that the proposed multipath routing scheme is indeed advantageous. In this chapter, multipath routing is studied from the perspective of node. Unipath reactive routing protocols generally discover a single path from source to destination and this shortest path often goes through the centre of the network, creating hot-spot regions and, as a consequence, network partitioning becomes more probable.

This chapter considers network lifetime as a key parameter and proposes two novel routing schemes which attempt to increase the energy map homogenisation. For reasons of simplicity, the proposed schemes are evaluated using a static grid topology. The aim is to demonstrate the operation of the schemes and show that for the simple topology they have merit. More extensive evaluation of the schemes, along with practical implementations are presented in Chapter 6. The two mechanisms initially assume that each node has complete knowledge of the network's energy map; an assumption which may be unrealistic in a real-life implementation. However, the aim is to investigate if energy homogenisation is a suitable approach for extending the network lifetime and maintain higher connectivity (for longer). The work described in this chapter represents an extended version of the concepts presented at the International Telecommunication Congress [96] undertaken under the auspices of EPSRC project GR/R25026/02.

The remainder of the chapter is organised as follows. Initially, an overview of energy efficient routing protocols proposed in the literature is presented, followed by a description of the two proposed schemes. Finally, results for the two proposed schemes are presented which demonstrate that energy map homogenisation is a suitable approach for extending network lifetime.

5.1 Related Work

Progressive technology advances have made it possible to build ad-hoc, low power, wireless sensor networks [97] of hundreds and even thousands of devices of low computation and communication capabilities. Typically, devices are battery powered and, since every message transmitted and received and the associated computation performed drains the battery, care is required in the utilisation of power.

An energy efficient architecture must be optimised over a wide range of aspects. The key aspects in an energy efficient architecture are Hardware Design, Operating System, Medium Access Control, Routing Protocol. These key aspects are summarised in Figure 5.1.

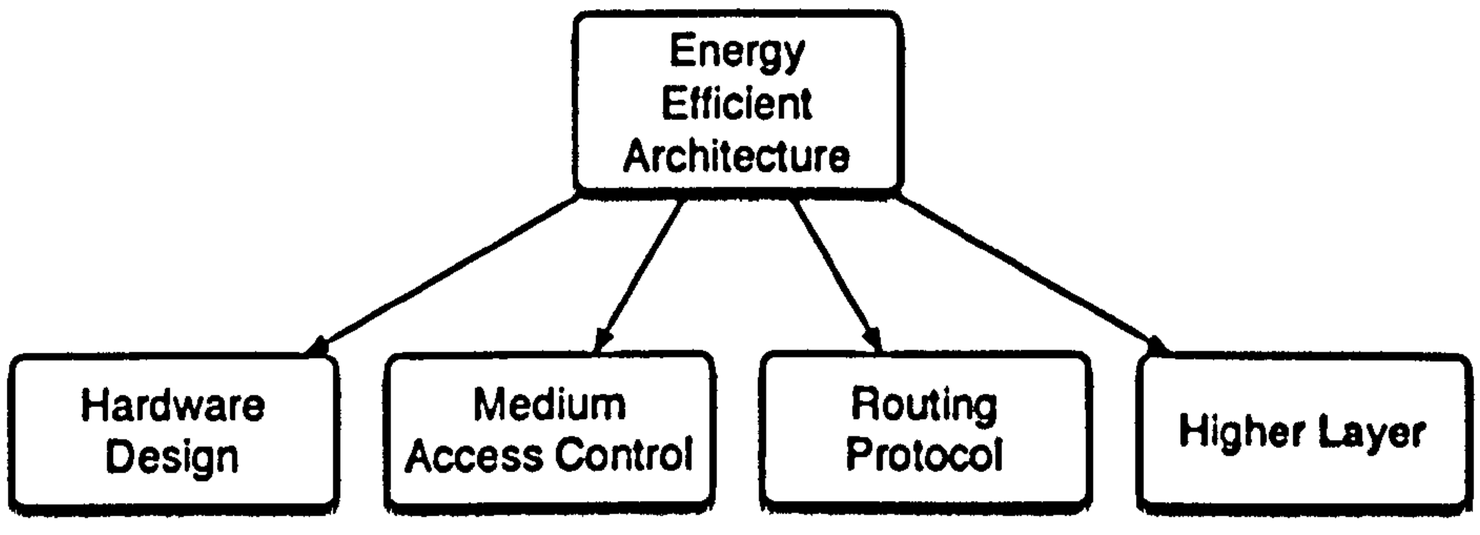


Figure 5.1: Key aspects of an energy efficient architecture.

The first aspect of an energy efficient architecture is low-power hardware design. Good hardware designs employ a range of techniques to improve the energy efficiency of a device such as varying CPU clock speed, disk spin down, flash memory [98]. Device power management is also important; various hardware units can be switched off when they are not in use and a significant energy saving can be achieved. Power aware CPUs employ methods such as Dynamic Voltage Scaling (DVS) [99] and Pipeline Stage Unification (PSU) [100] which varies the clock frequency to achieve lower energy consumption. Although interesting in their own right, these techniques are out of the scope of this thesis and hence are not discussed further.

Most energy in mobile devices is usually consumed by the radio interface [101, 102] and displays. Displays are not within scope of this thesis hence are not discussed any further. The MAC protocol is directly associated with the radio interface and thus impacts directly the device's efficiency. The key areas of MAC energy optimisation are power and topology control, rate adaptation and overhearing avoidance. In the first aspect, the data link layer attempts to select

the optimum transmission power which the device must transmit in order to reach its destination. If the transmission power is high, the energy consumption is higher while if the power is low the communication between peers will fail. The transmission power is also associated with the transmission rate; higher transmission rates require higher transmission power. On the other hand, high transmission power contributes to increased interference. Therefore, an efficient MAC will select the optimum power and rate to mitigate interference and achieve the communication at hand. Such MACs use implicit or explicit loop signalling mechanisms to determine the optimal power and rate. A variety of power-aware MAC protocols have been proposed [103, 104].

Overhearing is a significant cause of energy consumption. Devices, when they are not involved in communication but are within range of two communicating peers, consume energy attempting to decode the transmitted message. Additionally, idle devices also consume energy for little purpose. A variety of energy efficient MACs has been proposed in the literature which control the awake/sleep power circles of the device using sophisticated scheduling techniques. The most notable protocol is Sensor-MAC (S-MAC) [30] which reduces idle times and overhearing by introducing listen and sleep periods. Other protocols in that category are Berkley-MAC (B-MAC) [105] and Z-MAC [106].

The third component in an energy efficient architecture is the routing protocol. Routing protocols in such architectures can be classified in three key categories.

- Power-Aware
- Energy-Aware
- Hybrid

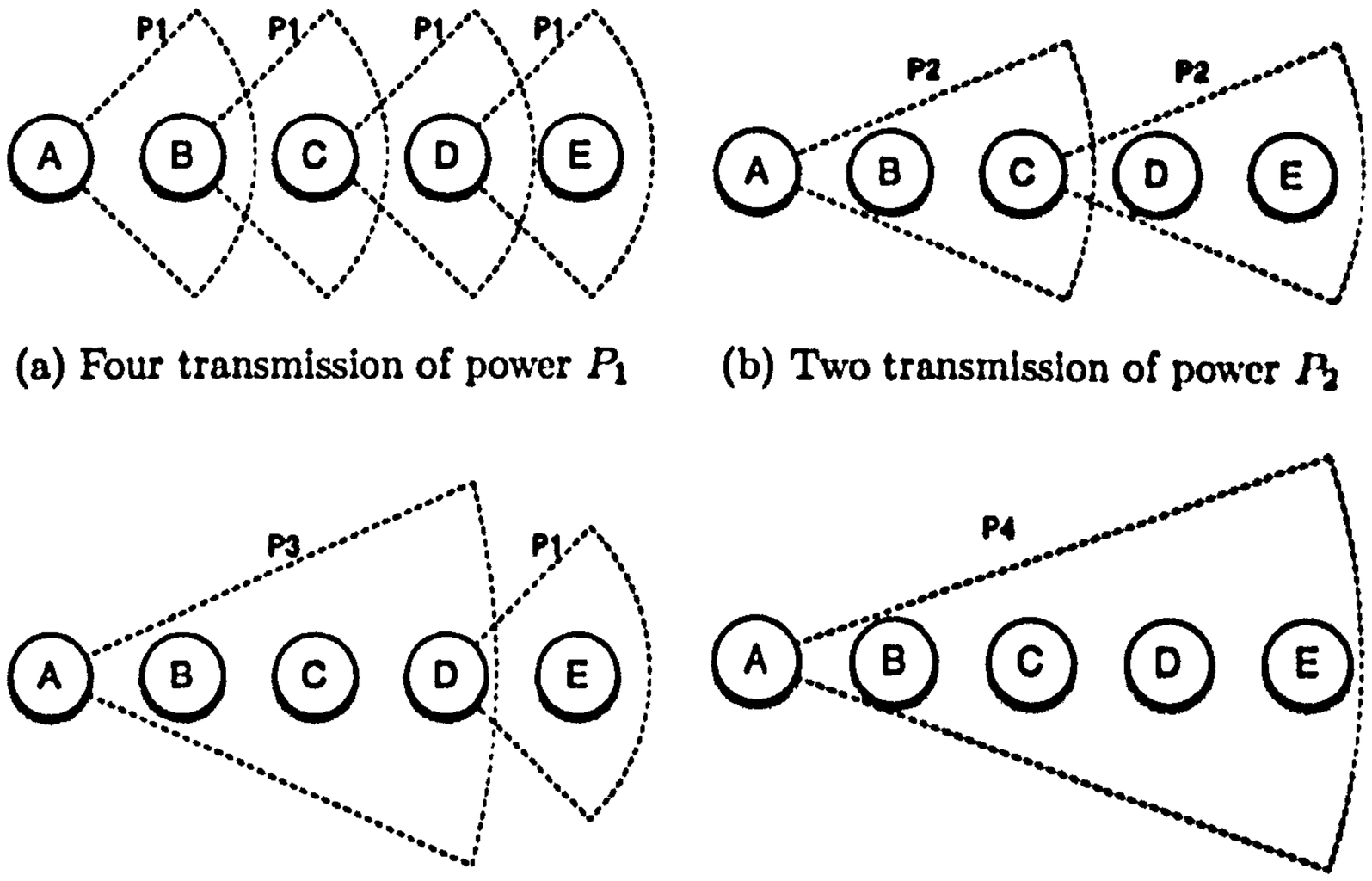
Power-Aware routing protocols select the best path to minimise the total power required to route the packets to their destination. Such schemes reduce the per packet energy consumption. However, such protocols do not guarantee to extend the lifetime of each node because they do not take into consideration the residual energy of the node. The most notable routing protocol in that category is the Minimum Total Transmission Power Routing (MTPR) [107, 108]. MTPR operates in the following manner: Consider a path J_k with nodes $N_0, N_1, \ldots N_d$ and the power required to transmit a message from node N_i to N_j is equal to $T(N_i, N_j)$ then the total transmission power for path J_k is $C(J_k) = \sum_{i=0}^{d-1} T(N_i, N_{i+1})$. Then the optimal route (J_0) with the minimum total

transmission power is

$$J_0 = \min_{J_k \in J_*} C(J_k) \tag{5.1}$$

where J_* is the set of all possible routes.

This class of routing protocols has meaning only when the data link layer supports transmissions with multiple transmission power levels. Figure 5.2 illustrates how a packet can be transmitted over one path with various transmission power combinations. MTPR will assign the cost of $4 \cdot P_1$ to the route of Figure 5.2a, $2 \cdot P_2$ to the route of Figure 5.2b and so on. Subsequently, MTPR selects the path with the minimum cost ($min(4 \cdot P_1,\ 2 \cdot P_2,\ P3 + P_1,\ P_4)$). When the data link layer supports only a single transmission power level then MTPR selects paths based on their hop count (shortest in number of hops).



(a) Four transmission of power P_1

(b) Two transmission of power P_2

(c) One transmission of power P_3 and one of P_1

(d) One transmission of power P_4

Figure 5.2: Four different ways to route a packet to its destination over the same path with different transmission power combinations.

Energy-aware routing protocols take into consideration a node's residual energy. Such protocols avoid the use of nodes with low residual energy at the expense of higher total power consumption. Because such protocols avoid nodes with low energy reserves, they tend to extend the network lifetime by reducing network partitioning and maintaining higher connectivity for longer. It is generally appreciated that network partitioning occurs due to the overuse of the nodes in the centre of the network because they are involved in more routing

operations. An energy-aware routing protocol will push the traffic towards the edge of the network, when the nodes in the centre become overused; total energy consumption may be increased but central nodes are kept alive for longer. Power-aware routing schemes optimise the total power consumption without taking into consideration the residual node energy. The two classes of protocols can have contradictory objectives.

The most notable energy-aware routing protocol is the Minimum Battery Cost Routing (MBCR) [108, 109]. If the energy of node N_i at time t is denoted by $N_i(t)$, the objective is to define a function whereby nodes with low residual energy are more reluctant to forward packets. An example of a non-linear function is shown below and others do exist.

$$f_i(N_i(t)) = \frac{1}{N_i(t)} \tag{5.2}$$

Here, when a node has low residual energy, the cost of using this node is high while nodes with more energy reserves have lower usage cost. For a route J_k with nodes $N_0, N_1, \ldots N_d$ the path cost $C(J_k)$ is:

$$C(J_k) = \sum_{j=0}^{d} f_j(N_j(t)) \tag{5.3}$$

Once these costs are calculated for all paths, the optimum (MBCR) route J_0 is:

$$J_0 = \min_{J_k \in J_*} C(J_k) \tag{5.4}$$

again J_* is the set of all possible paths.

At first glance this mechanism might seem appropriate for extending the energy lifetime. However, consider the scenario of Figure 5.3. Source Node S has two candidate paths $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow T$ ($\frac{1}{8} + \frac{1}{5} + \frac{1}{6} + \frac{1}{5} = 0.692$) and $S \rightarrow E \rightarrow F \rightarrow G \rightarrow T$ ($\frac{1}{10} + \frac{1}{2} + \frac{1}{12} = 0.683$). Using the Minimum Battery Cost Routing, the path which will be selected is the second ($0.683 < 0.692$).

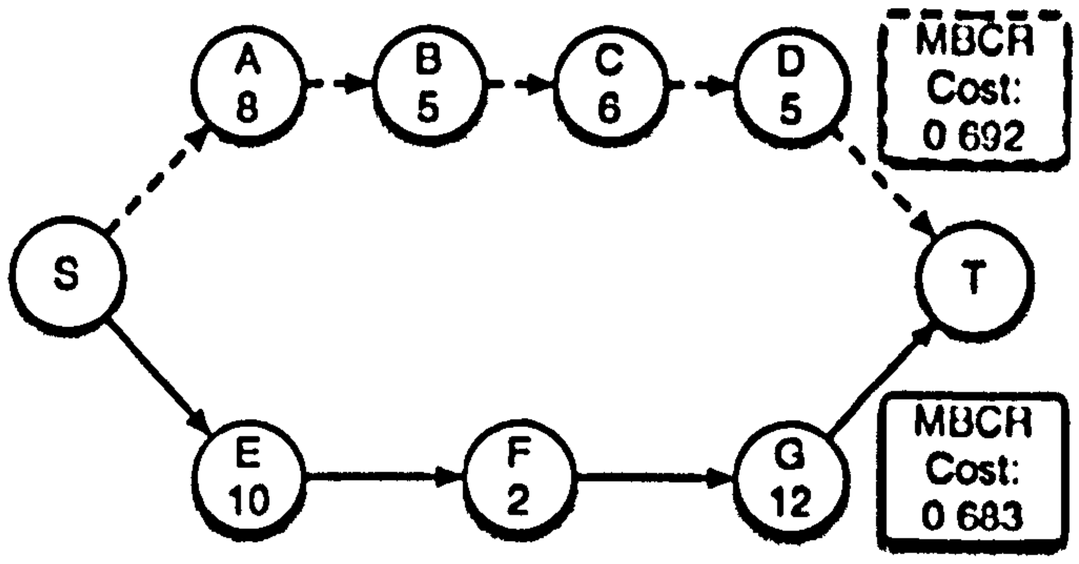


Figure 5.3: An example where MBCR selects a path with lower residual energy.

Such a scenario illustrates that MBCR considers the summation of weights rather than the weights of the individual devices and consequently, the protocol might select paths which contain nodes with little residual energy and minimum aggregate cost (maximum aggregate energy reserves) and such selections will lead to a reduced lifetime.

To avoid such scenarios a modification proposed [108] which modifies the cost function of equation 5.3. In this scheme, the path cost is defined as the maximum of the individual weights:

$$C(J_k) = \max_{j=0...d} f_j(N_j(t)) \tag{5.5}$$

And then a path selected according to the function:

$$J_0 = \min_{J_k \in J_*} C(J_k) \tag{5.6}$$

This routing discipline is known as Min-Max Battery Cost Routing (MMBCR). In MMBCR, the cost assigned to each route is the maximum of weights (Minimum Residual Energy) and subsequently the selection function chooses the path with the minimum cost (higher minimum residual energy). In such a manner routes with "weak" nodes are avoided and the network lifetime can be extended. In the example shown in Figure 5.3, MMBCR will select the first path $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow T$ which is more preferable in the given scenario. On the other hand, MMBCR does not take into consideration the aggregate energies of the routes and consequently might select a route with more hops or with lower total residual energies only because the route contains the strongest-weakest device.

Another notable energy aware routing protocol is the Minimum Drain Rate (MDR) [110] which considers as cost function the rate at which the energy is

consumed and hence the protocol attempts to avoid routes with overused nodes. Energy drain rate is closely related to the traffic load passing through the node of the devices and, based on this rate, the protocol can predict when the nodes will get exhausted based on the current traffic conditions.

The final class of protocols are hybrid and attempt to combine power and energy awareness. Extending the network lifetime and reducing the total energy consumption per packet is not a trivial task. For example, MTPR does not guarantee that the network lifetime will be extended while, on the other hand, MMBCR does not guarantee that the total power consumption per packet is minimised. Toh et al. [108] propose Conditional Max-Min Battery Routing (CMMBR) which combines MTPR and MMBCR by considering both the total power consumption and residual node energy. When all nodes have residual energy above a threshold γ, a route is selected based on the total power consumption (MTPR) while if all routes have nodes with residual energy below the threshold γ the routing scheme selects the routes based on the residual energy (MMBCR). A similar hybrid scheme is applied by Kim et al. [110] that uses the energy drain rate as a metric.

The work presented in the remainder of the chapter focusses upon novel energy aware routing protocols, which can be used either individually or as part of a hybrid scheme. On one hand, Minimum Battery Cost routing attempts to minimise the overall path cost while, on the other hand, Min-Max Battery Cost Routing attempts to maximise the life of the individual nodes. The proposed mechanisms endeavour to maintain a homogenised network energy map.

5.2 Model Description

Consider a planar network G composed of $M = N^2$ nodes with coordinates $(X_i, Y_i) \in \{1, 2 \ldots, n\} \times \{1, 2 \ldots, n\}$, arranged in a grid topology (Figure 5.4). The graph $G(V, E) = G(M, [\frac{4 \cdot 2 + 4 \cdot (N-2) \cdot 3 + (M-4 \cdot N+4) \cdot 4}{2}])$ is a bidirectional graph with edge weights equal to unity. The energy level of a node $N_i \in G$ at time t is a whole number $E(N_i(t)) \geq 0$, the node is 'alive' at time t if $E(N_i(t)) > 0$ and 'dead', otherwise. The energies are not restorable, so once a node 'dies' it stays inactive forever. Each node has a radio interface, capable of communicating with all neighbouring nodes; the nodes within distance 1 (i.e. only with adjacent devices located North, South, West and East as shown in Figure 5.4 for node S). At every discrete point of time t, $t = 1, 2, \ldots$, a node $S = S(t) = (X_s, Y_s)$ (source) sends a packet to a destination node $D = D(t) = (X_d, Y_d)$ which is

instantly relayed through a sequence

$$J(t) = [S(t) = N_{i_0}(t) \rightarrow N_{i_1}(t) \rightarrow \ldots, \rightarrow D(t) = N_{i_l}(t)] \qquad (5.7)$$

of neighbouring nodes alive at time t. The energies of all the nodes on the path are then reduced by 1 unit.

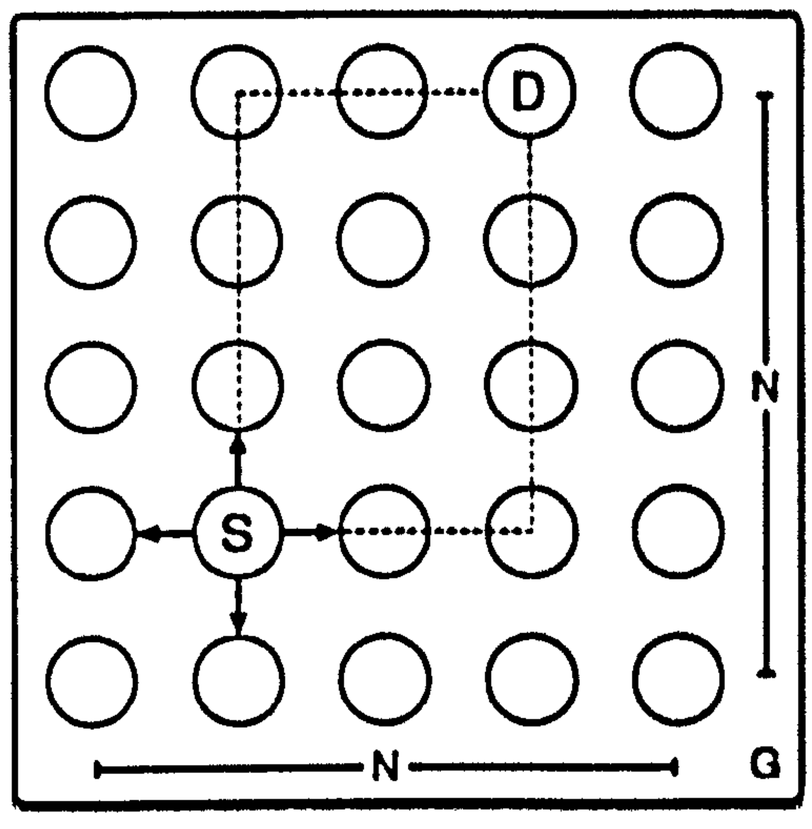


Figure 5.4: An example of a 5×5 grid network.

The path taken by the packet is determined by the routing protocol used among the set $J_*(S(t), D(t))$ of feasible paths. It is assumed that the feasible paths are those that contain only active nodes which only move packets in the direction of the destination. In the grid topology, this means that all loop-free feasible paths lie inside the rectangle formed by S and D in the corners (Figure 5.4). The number of nodes in the path is then equals its Euclidean length $l = |X_s - X_d| + |Y_s - Y_d|$. If all the nodes in the rectangle are active, the total number of possible paths from S to D then equals the Binomial coefficient $\binom{l}{|X_s - X_d|}$.

Additionally, it is implicitly assumed that the routing algorithms require complete knowledge of the topology and energy levels of the nodes in the network at all times. However, no knowledge of future transmissions is assumed. In practical network scenarios, the full knowledge of the energy distribution for all nodes and by all nodes is generally not achievable. However, knowing an optimal strategy for the whole system permits distributed variants of the protocol to be applied to configuration sets known to the sending nodes.

Having defined this model, two new route selection schemes which extend the lifetime and quality of the networks can be defined: Point-To-Diagonal (PTD) and Point-To-Zero (PTZ).

The state of the network at any time t can be described by a $E(t)$ of the space $\mathbb{R}_+^M$, whose coordinates represent the remaining energies of the nodes at time t:

$$E(t) = (E_1(t), E_2(t) \ldots, E_M(t)),$$

where $E_i(t) = E(N_i(t))$, $i = 1 \ldots, M$. Transmitting a packet over a path $J(t)$ (as in Equation 5.7), will result in decreasing by one the energies of the corresponding transmitting nodes. If $\pi(t)$ is the M-dimensional vector having 1 at the coordinates i_k such that $N_{i_k}(t) \in J(t)$, $k = 0 \ldots, l$ and 0 in all other coordinates, then the resulting energies at time $t+1$ will be $E(t+1) = E(t) - \pi(t)$. Therefore, evolution of the system is completely described by the initial energy distribution plus one of either a sequence of vectors $\pi(t)$ or a sequence of the remaining energies $E(t)$ for all t.

5.3 Proposed Energy Homogenising Algorithms

When the energy of a node drops to zero, the vector of remaining energies lies on a lower dimensional subspace of the quadrant $\mathbb{R}_+^M$ thus significantly reducing the choice of paths for subsequent transmissions. A good routing protocol should prevent this from happening for as long as possible in order to extend network lifetime.

The proposed Point-To-Diagonal (PTD) algorithm aims to keep the energy vector $E(t)$ at each instant t as close as possible to the diagonal line $\Delta = (C, C, \ldots, C)$ of $\mathbb{R}^M$ thus maintaining the homogeneity of battery reserve distribution in the network. The proposed Point-To-Zero (PTZ) algorithm tries to maintain the same relative proportion of the remaining energies between the nodes, anticipating the situation when overall re-balancing of energies may be impractical.

5.3.1 Point to Diagonal Algorithm (PTD)

The proposed PTD algorithm attempts to maintain even energy consumption throughout the network. This will avoid the unwise overuse of nodes.

Every time node N_i transmits a message, its battery reserve decreases; therefore overused nodes can quickly lose their energy. Geometrically, this means that the energy vectors $E(t)$ approach the hyper-plane $E_i = 0$. PTD endeavours to avoid this situation and prefers networks containing nodes with homogeneous energy levels rather than networks whose nodes exhibit large fluctuations in en-

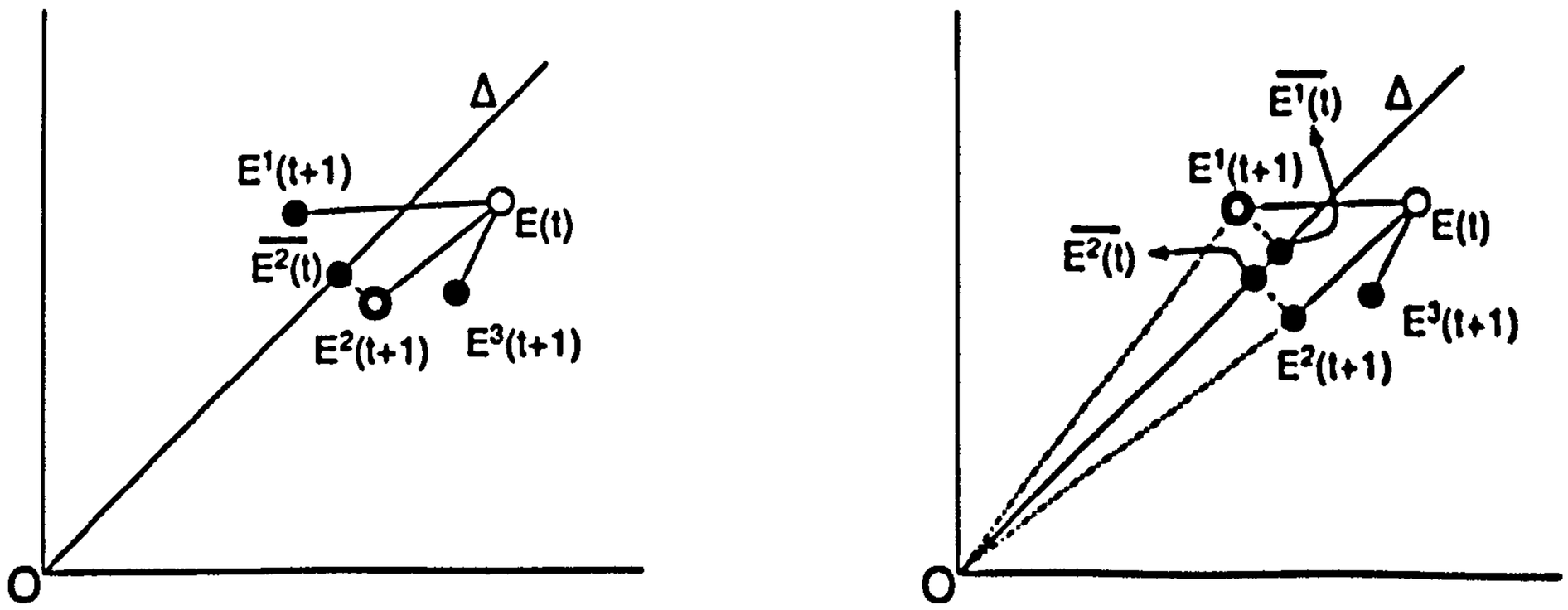
ergy levels. This translates into forcing the multidimensional point $E(t)$ to be as close as possible to the diagonal Δ. Recalling that the diagonal is the set of points farthest away from the multidimensional spaces defined by network status containing dead nodes. Consequently, the proposed algorithm at time t, selects the routing path $\pi^j(t)$ which minimises the distance from $E^j(t+1)$ to the projection point onto Δ ($\overline{E^j(t+1)}$).

The algorithm works in the following manner. Given a communication request from the source node $S(t)$ to destination $D(t)$ at time t, let $R(t) = J_*(S(t), D(t))$ denote the set of vectors π corresponding to all feasible paths from $S(t)$ to $D(t)$. Thus the set $E(t) - R(t)$ represents the range for $E(t+1)$. Elementary geometrical calculations show that the projection of a vector $E^j = (E_1^j(t) \ldots, E_M^j(t))$ onto Δ is the point at which $\overline{E^j(t)} = M^{-1} \sum_{i=1}^{M} E_i^j(t)$ - the mean residual energy of the nodes in the network. The PTD algorithm then chooses the path j corresponding to $\pi^j(t)$ which minimises the following:

$$\sum_{i=1}^{M} \left(E_i^j(t) - \pi_i^j(t) - \overline{E_i^j(t+1)}\right)^2 = \sum_{i=1}^{M} \left(E_i^j(t+1) - \overline{E_i^j(t+1)}\right)^2. \qquad (5.8)$$

Therefore, effectively, PTD minimises the variance of the energy distribution in the network.

For instance, on Figure 5.5a the set of possible values for $E(t+1)$ are three points: $E^1(t+1), E^2(t+1)$ and $E^3(t+1)$. The PTD algorithm will then choose the path corresponding to $E(t) - E^2(t+1)$ (Figure 5.5a).



(a) Unique minimum distance to Δ (b) Multiple minimum distances to Δ

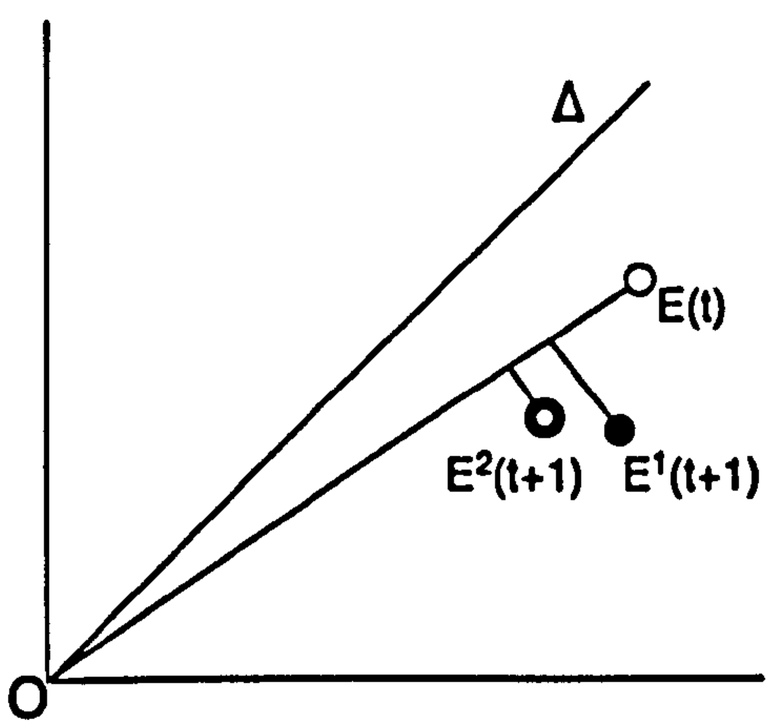
Figure 5.5: Path Selection for PTD.

If there are multiple energy vectors in the set $E(t)-R(t)$ realising the minimal distance to Δ (Figure 5.5b), then the path selected is that one that has maximum

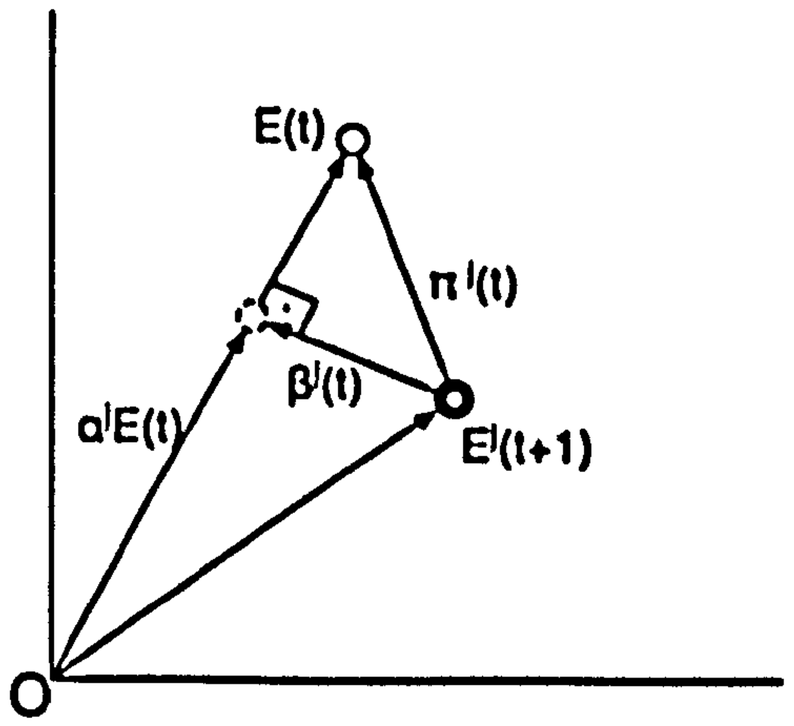
distance from the origin of the $\mathbb{R}^M$ space. If more than one path has equal distance from the space origin, then a path is selected at random.

5.3.2 Point to Zero Routing Algorithm (PTZ)

The idea behind the proposed PTZ algorithm, as with PTD, is to proportionally decrease the energy states of the network while avoiding converging too quickly towards states containing one or more zero energy components. This is done in a less constrained way than PTD. In each energy state $E(t)$, the algorithm tries to select a path j leading to the new state $E^j(t+1)$ minimising the distance to $(OE(t))$ among all feasible energy states $E(t) - R(t)$, $(OE(t))$ being the line passing through the origin O and $E(t)$, see Figure 5.6a).



(a) Path Selection for PTZ.



(b) Geometry for calculating the projection length.

Figure 5.6: Path Selection for PTZ.

It is easy to obtain the projection of a vector $E^j(t+1) = E(t) - \pi^j(t)$ onto the line $(0, E(t))$ is the point $\alpha^j E(t)$ (Figure 5.6b). The vectors $E(t)$ and $\beta^j(t+1)$ are perpendicular and their inner product is equal to zero.

$$E(t) \cdot \beta^j(t+1) = 0 \tag{5.9}$$

From Figure 5.6b the vector $\beta^j(t+1)$ is equal to $\beta^j(t+1) = E^j(t+1) - \alpha^j E(t) = E(t) - \pi^j(t) - \alpha^j E(t) = (1-\alpha^j)E(t) - \pi^j(t)$ hence equation 5.9 becomes:

$$E(t) \cdot [(1-\alpha^j)E(t) - \pi^j(t)] = 0$$

Solving this equation with the only unknown being α^j yields:

$$\alpha^j = 1 - \frac{E(t) \cdot \pi^j(t)}{\|E(t)\|^2} = 1 - \|E(t)\|^{-2} \sum_{i \in J(t)} E_i^j ,$$

where $a^j \cdot b^j$ is the scalar product of vectors a^j and b^j and $\|E\|^2 = \sum_{i=1}^{M} (E_i^j)^2$. Knowing the α^j coefficient, the length of vector $\beta^j(t+1)$ can be determined:

$$\sum_{i=1}^{M} \left[\frac{E(t) \cdot \pi(t)}{\|E(t)\|^2} E_i(t) - \pi_i(t) \right]^2 \tag{5.10}$$

Thus, the PTZ algorithm chooses a feasible $\pi^j(t)$ minimising the length of $\beta^j(t+1)$ of equation 5.10, as shown in Figure 5.6a.

If there are multiple $\pi^j(t)$ providing such a minimum, the priority is given to the one for which $E(t) - \pi^j(t)$ is closest to the diagonal Δ. If there are multiple points with the same distance from the diagonal, a point is chosen in the same manner as in PTD.

5.4 Reference Routing Algorithms

In order to assess the quality of the proposed algorithms, each is compared with two reference algorithms: Strongest Path Routing (SPR) and Min-Max Battery Cost Routing (MMBCR) algorithm. Power-aware and hybrid algorithms are not comparable with the proposed routing scheme since all devices are equipped with the same transceivers using constant transmission power. The use of any adaptive power device in a comparison would not really be appropriate as to ensure a fair comparison, the nodes will not be able to fully exploit the ability to vary energy consumption. Also, the discussion in this Section is intended to, in the first instance, demonstrate the benefits of the proposed algorithms against standard schemes. Both reference algorithms associate a certain cost with each feasible route and choose the one which has the smallest/largest cost. If there are multiple equal cost routes, then one is chosen randomly.

SPR, on the other hand, chooses the path with the largest sum of energies of the participating nodes. Since there may be more than one feasible path, the maximal energy path is detected by running a simple path search algorithm. As the algorithms are not anticipating any future demands, choosing the maximal energy path every time is, perhaps, the most natural strategy. However, running SPR every time is computationally consuming and may not be practical, but it

does represent a suitable benchmark in terms of performance comparison. SPR is similar, though not equivalent, to Minimum Battery Cost Routing (MBCR); see Section 5.1.

5.5 Results

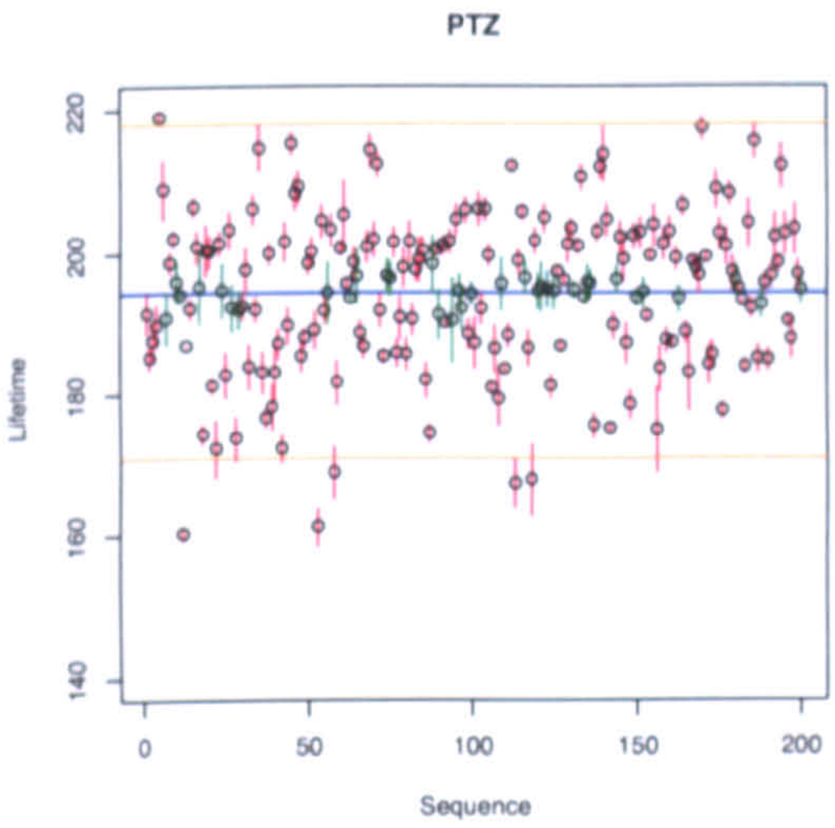
The results presented in this chapter were obtained using a C++ iterative simulator and plotted and analysed using R statistical package [111]. In all the experiments, the network is a 10×10 grid with $M = 100$ nodes. Two configurations of initial node energies are considered: either all set to 20 (denoted by IC20 in the legend) or independently uniformly distributed in the range from 10 to 20 (notation: IC10-20). Randomly charged nodes represent a situation that exists after a prolonged period of system operation (to study relaxation phenomena of different algorithms) or when the nodes were in different external environments for some time prior to the start of the service or when 'falling asleep–awakening' protocols were used. The use of randomly assigned initial energies also enables differentiation between PTD and PTZ algorithms as the starting line $(0, E(0))$ in PTZ coincides with Δ in the case when the initial energies are equal.

Two hundred different sufficiently long sequences[1] of the communication requests $(S(t), D(t))$, $t = 0, 1, \ldots$ were generated to test all four considered routing algorithms: PTD, PTZ, SPR and MMBCR. Moreover, since the algorithms sometimes choose paths randomly, 50 repetitions were carried out for each of these sequences of requests, to avoid any "rogue" results. In making the comparisons between algorithms, a range of derived statistics is used. One area of interest is the point at which full connectivity is lost; this can either described by the point at which the first node (out of the grid) becomes permanently inactive as a result of energy reserves being fully depleted, or the point at which the first call request is rejected because no suitable path can be found. There is a significant difference between these two events. The former is the point when 100% connectivity can no longer be guaranteed and thus there is the potential for calls to be rejected. The latter is the first point at which this potential is realised. Also recorded are the points at which further nodes $(10, 20, 30 \ldots)$ become permanently inactive, along with the associated running total of supported and rejected call requests.

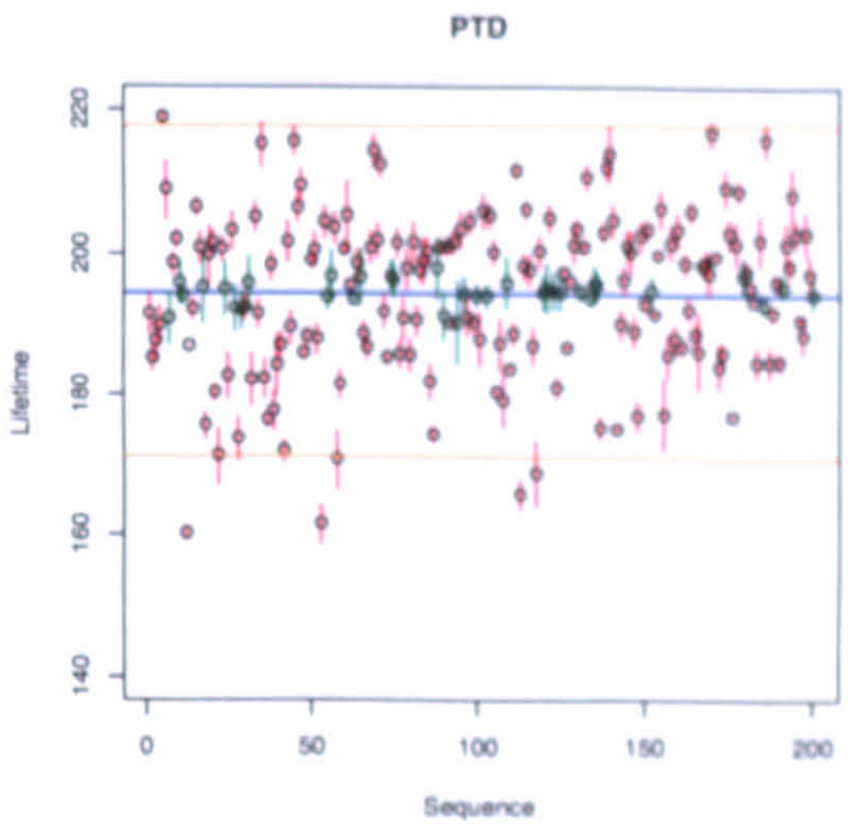
The first observation is the relatively large variability in the network lifetime

[1]The length of communication request was extended until the confidence intervals reached variability below 5% of the mean lifetime.

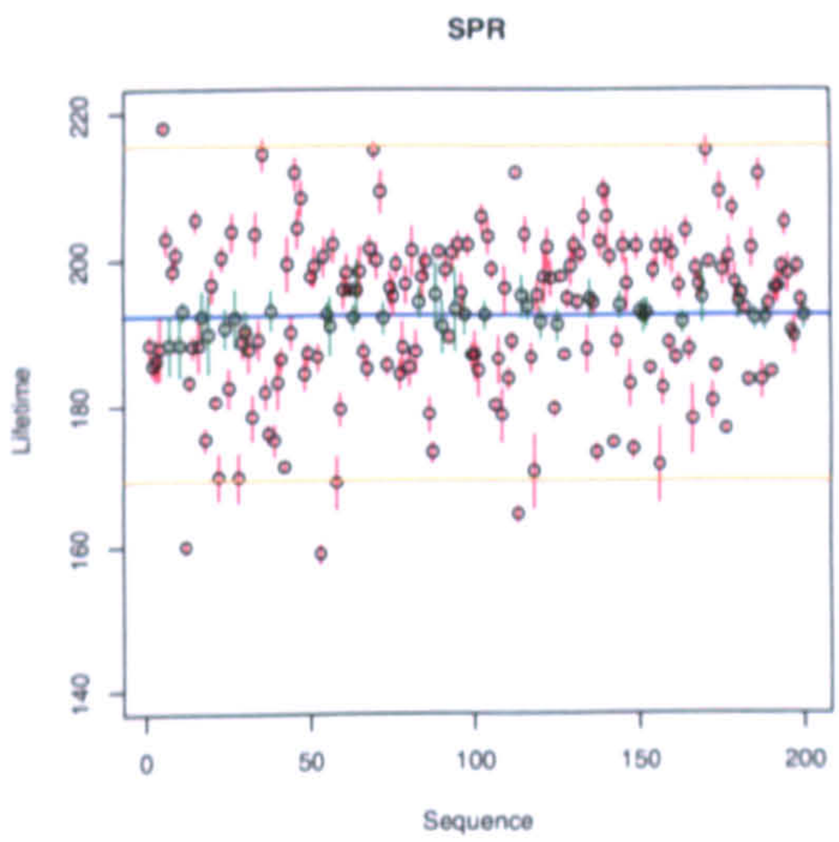
as a function of the particular simulated sequence of the communicating pairs $(S(t), D(t))$, $t = 0, 1, \ldots$. The lifetime in these graphs is measured until the first node in the network is dead and full connectivity can no longer be guaranteed. On the other hand, any variation in the lifetime that results from random choice of equivalent paths is usually just a few units. Figure 5.7a shows lifetimes achieved by running PTZ algorithms on 200 sequences with the initial nodal energies of 20. For a given sequence number, the dot shows the lifetime averaged over 50 realisations of the same sequence of transmissions together with the segment representing 95% confidence interval for the lifetime estimated from



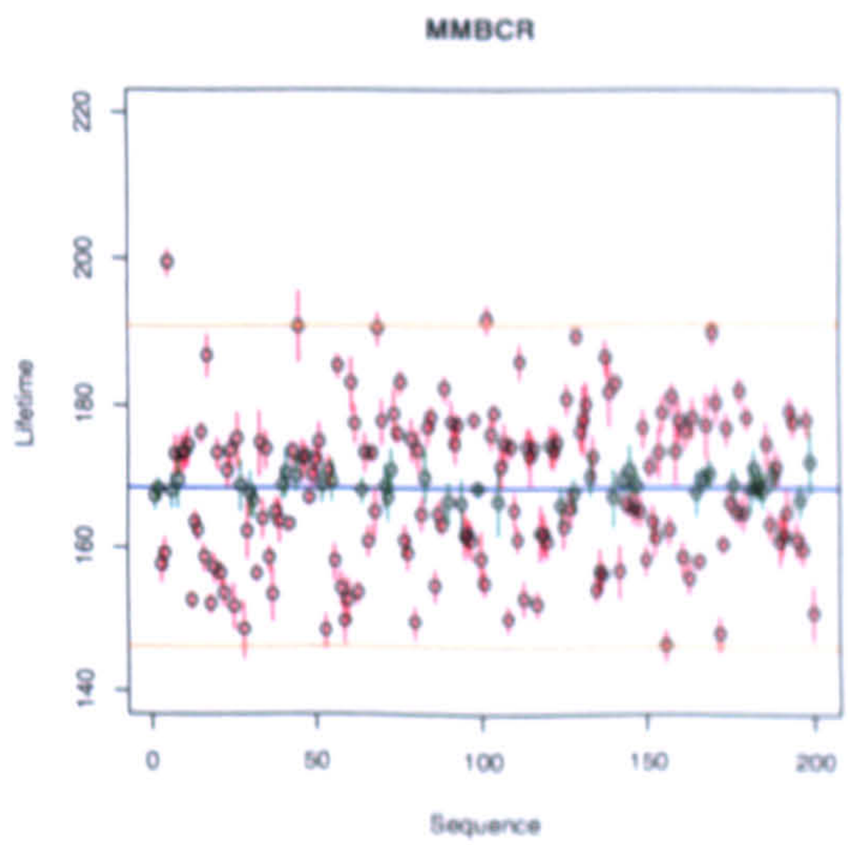
(a) PTZ IC20



(b) PTD IC20



(c) SPR IC20



(d) MMBCR IC20

Figure 5.7: Average lifetime (first node dead) for 200 realisations of communicating pairs sequences for IC20.

these 50 iterations. Variations here are due only to a randomised choice of an optimal path if there are multiple equivalent paths. The blue horizontal line represents the overall mean lifetime (194.5 here) and the orange horizontal lines represent the overall 95% confidence bounds.

A similar picture is observed for all other algorithms: for IC20 the overall mean lifetimes are 194.4 for PTD (Figure 5.7b), 192.5 for SPR (Figure 5.7c) and 168.6 for MMBCR (Figure 5.7d), therefore a comparison of algorithms solely on the basis of the overall mean lifetimes achieved is not really possible. A more appropriate comparison of two algorithms is to consider the sample lifetimes as paired observations as they share the same sequences of communicating pairs.

The most common test for comparing paired observations is the Student's $t-test$ [112]. This test is a parametric test and possesses strict assumptions. The two main assumptions imposed by the $t-test$ are:

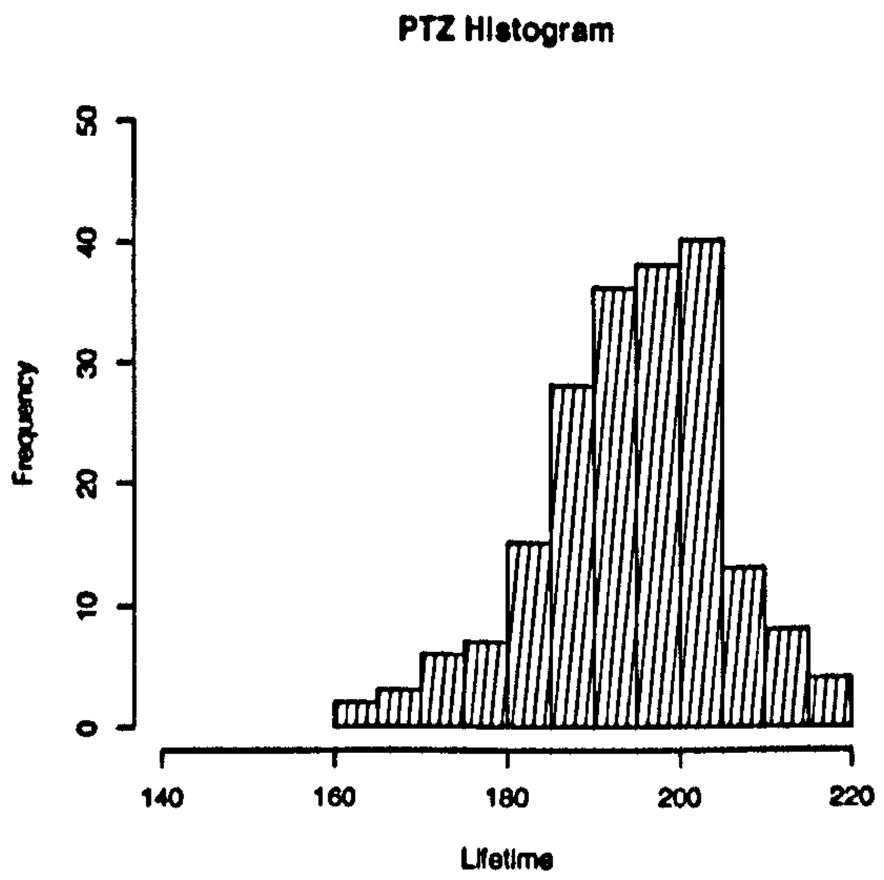
1. The observations are drawn randomly from a normal distribution.
2. The two groups of observations have the same variance.

Initially, the frequency histograms of the observations are plotted to determine if their probability density function diverges from the Normal. Figure 5.8 shows the frequency histogram for the lifetimes of all compared routing protocols.

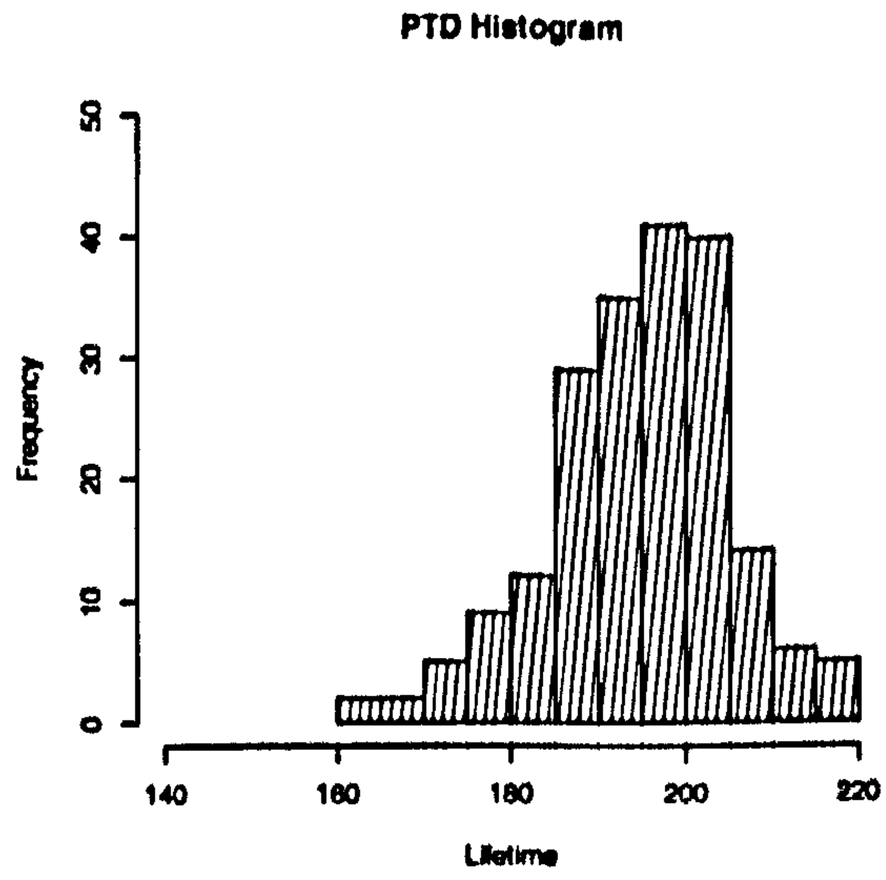
There is a relative skewness in the histograms of all routing protocols. Therefore the $t-test$ can not be used because the non-normality of the observations in all cases. In order to quantify this conclusion further, two normality tests were employed, the Shapiro-Wilk nomality test and Kolmogorov-Smirnov test [113]. Both tests examine the null hypothesis (that data comes from a normal distribution) and reject it with a given confidence. In the observations collected for all routing protocols, both tests rejected the null hypothesis in all cases with confidence level higher than 0.05. Finally, additional evidence of non-normality is that the lifetime is a positive random variable, while a true normal distribution takes values in $\mathbb{R}$.

Since the measured sample failed to comply with the first assumption of the $t-test$, the second assumption imposed by the $t-test$ is not examined further[2], but instead a non-parametric Wilcoxon signed rank test [113] was employed. The test examined if the lifetimes have the same or different means. Non-parametric tests do not make any assumptions about the frequency distributions of the assessed observations but impose the following assumption instead:

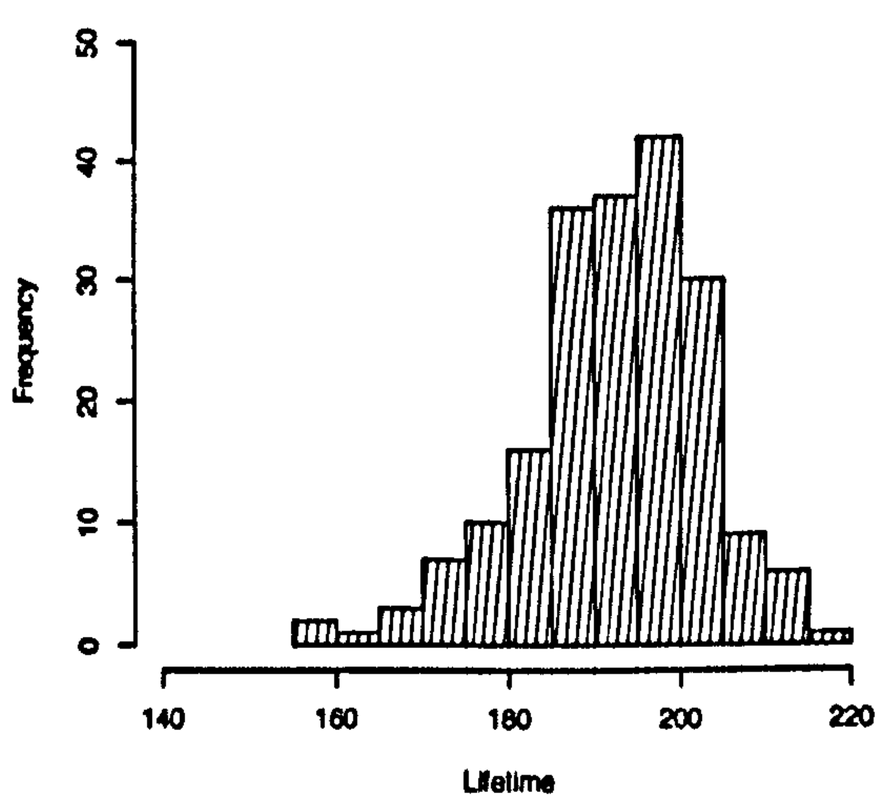
[2]Examine the variance, tests such as the F-test could be used to test the equality of variance.



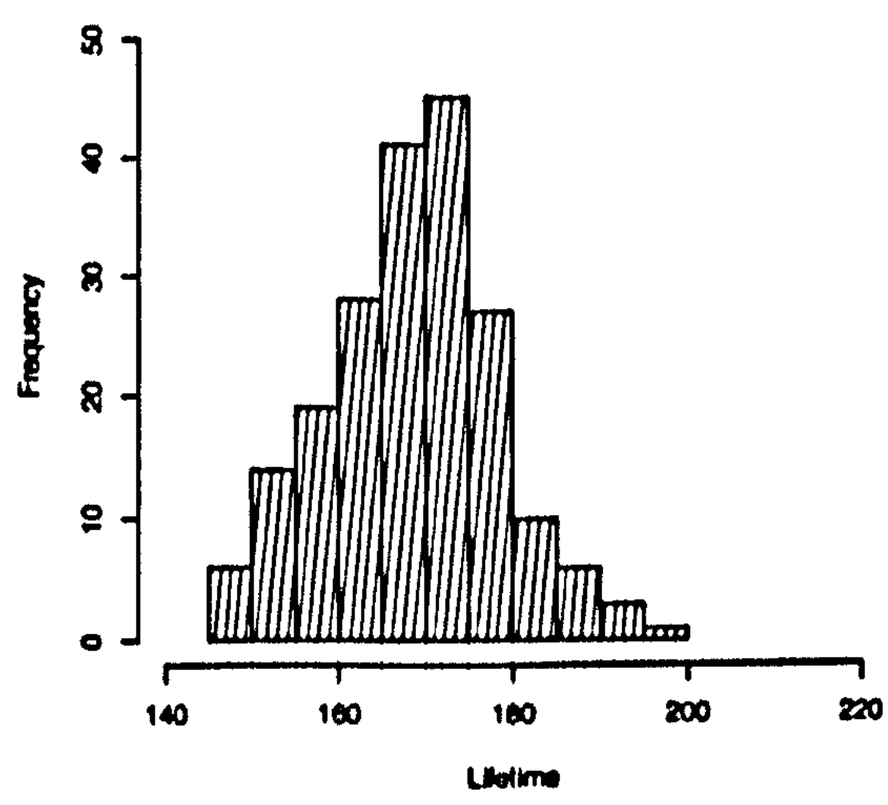
(a) PTZ IC20



(b) PTD IC20



(c) SPR IC20



(d) MMBCR IC20

Figure 5.8: Histogram of the average lifetimes for all routing protocols.

1. The paired values are randomly and independent drawn.
2. The observations are continuous.

Both of these assumptions are valid in the examined scenario. The paired values are indeed randomly drawn since the communication sequence pairs are random and they are independent as each observation is taken from a fresh run. The Wilcoxon Signed ranked test is applied to the lifetime over 50 repetitions and hence making the means not discrete. Therefore, the test is suitable for comparing the examined routing protocols.

5.5.1 Lifetime Analysis

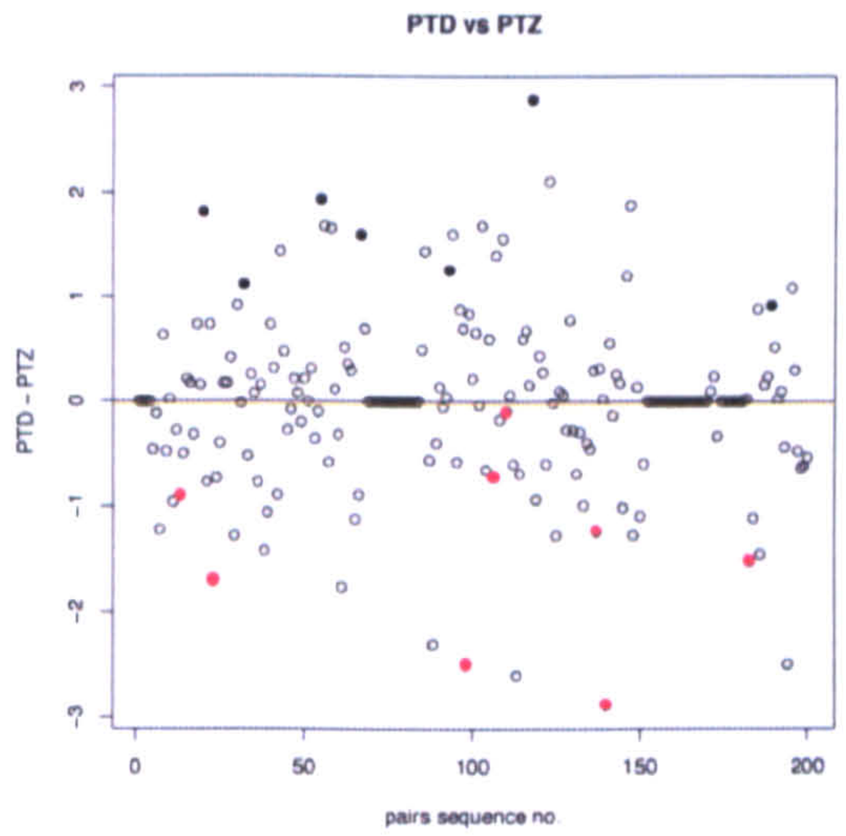
The simulation results show that the two algorithms: PTD and PTZ give the same lifetime and any difference of the corresponding means for given sequence of communicating pairs is not significant. For instance, in case of IC20, PTD performed significantly (with 95% confidence level) better than PTZ for seven sequences (solid dots on Figure 5.9a) and for 8 sequences PTZ outperformed PTD (red dots). In another 191 cases, the Wilcoxon test shows no significant difference in means (empty dots) and a few zero points can be observed. With the initial energy point $E(0)$ lying on the diagonal Δ, PTD and PTZ actually coincide, because their starting points coincide.

A similar picture emerges for IC10-20: in 12 cases PTD was better than PTZ while the reverse was true in 15 cases. In the remaining 173 cases there were no significant differences; see Figure 5.9b.

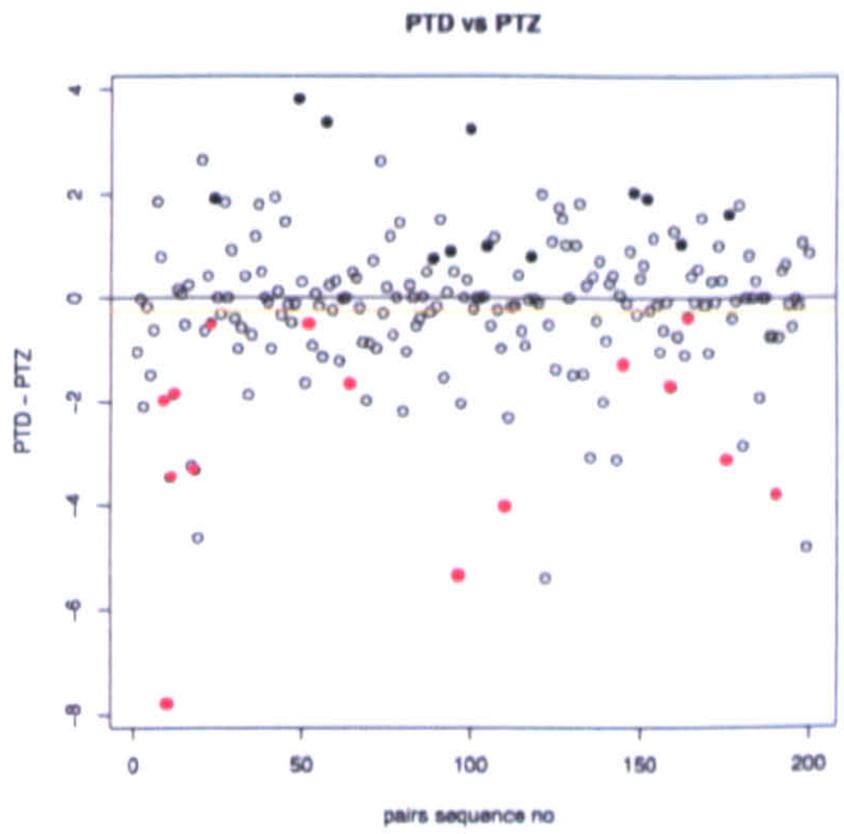
The next best algorithm is SPR. Although the overall average lifetimes (192.49 for IC20 and 122.1 for IC10-20) are very close to PTD, for IC20, in 97 communicating pairs sequences PTD outperformed SPR compared to only 2 sequences when SPR was better. For IC10-20 the figures are 105 and 4, respectively. It is easily seen in Figures 5.9c and 5.9d that, in many cases, PTD carried more than 10 (and sometimes more than 20) additional requests than SPR until the first transmission failure due to the absence of any available path.

The worst in terms of achieved lifetime is MMBCR routing. The average number of successful transmission are 168.6 for IC20 and 106.6 for IC10-20. PTD/PTZ performed significantly better than MMBCR in all cases for IC20 and in 184 cases for IC10-20.

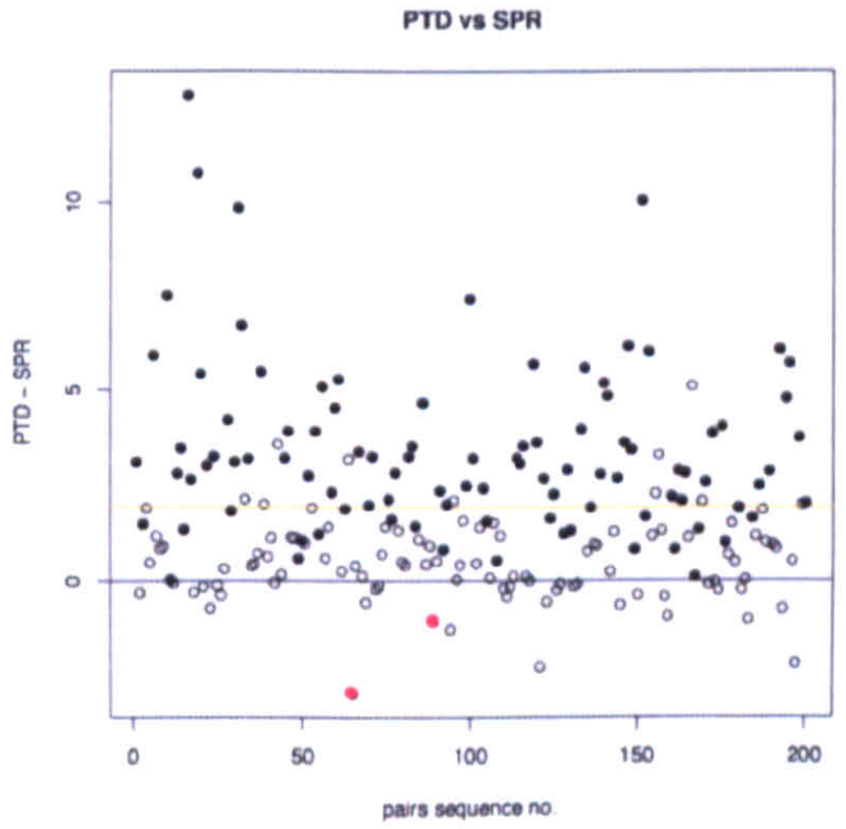
Comparisons between the remaining algorithm pairs lead to the following lifetime performance ordering relationship: PTD=PTZ>SPR>>MMBCR. Additional evaluation of the proposed schemes is performed to quantify their merits in terms of the residual energy map and request rejection ratio when one or more nodes become permanently inactive due to lack of residual energy.



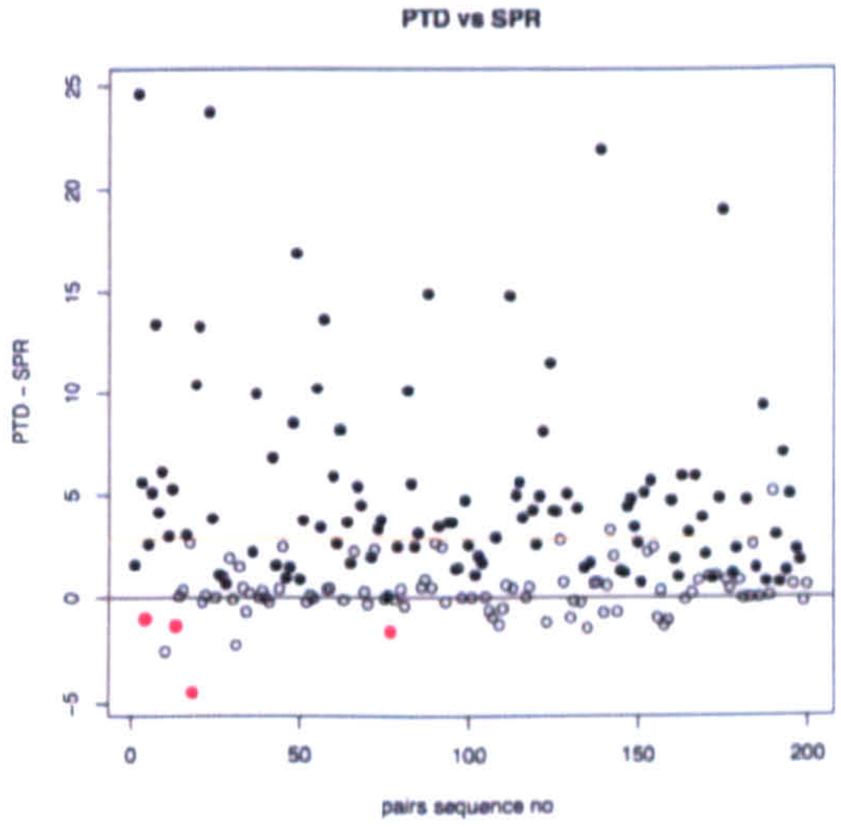
(a) IC20



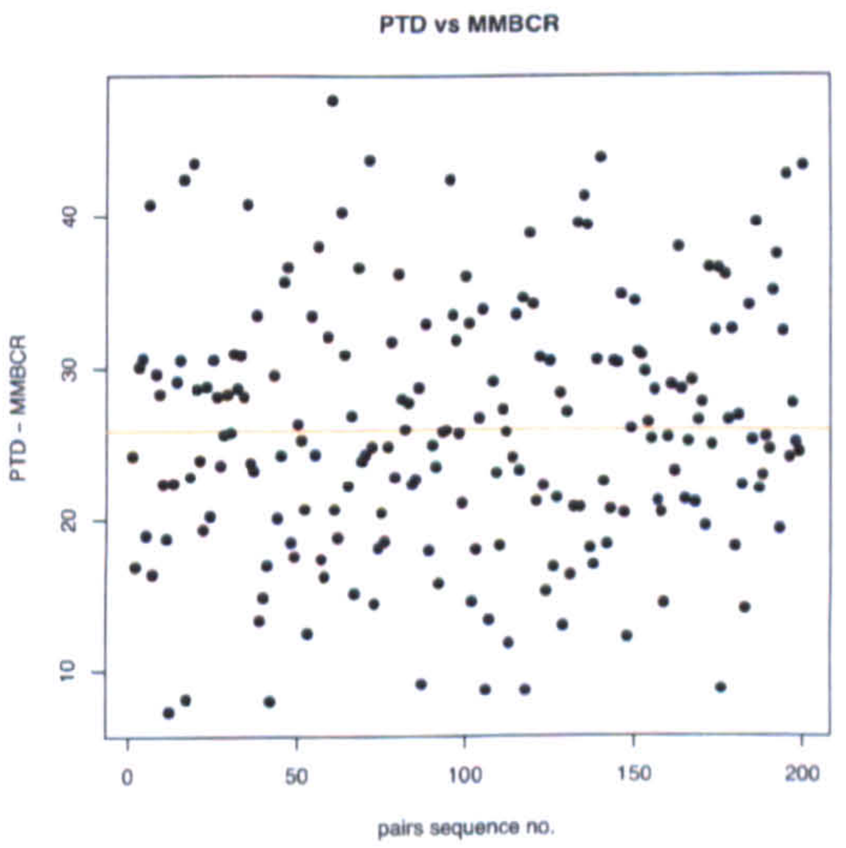
(b) IC10-20



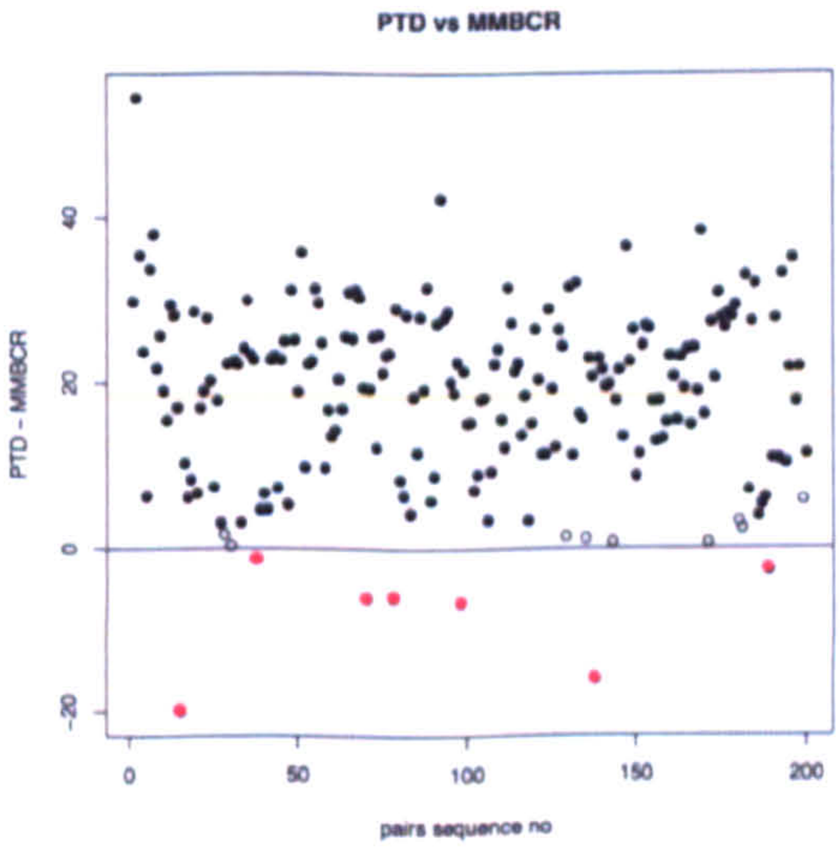
(c) IC20



(d) IC10-20



(e) IC20



(f) IC10-20

Figure 5.9: Number of successful connections vs. communicating pairs sequence number.

5.5.2 Request and Node Loss

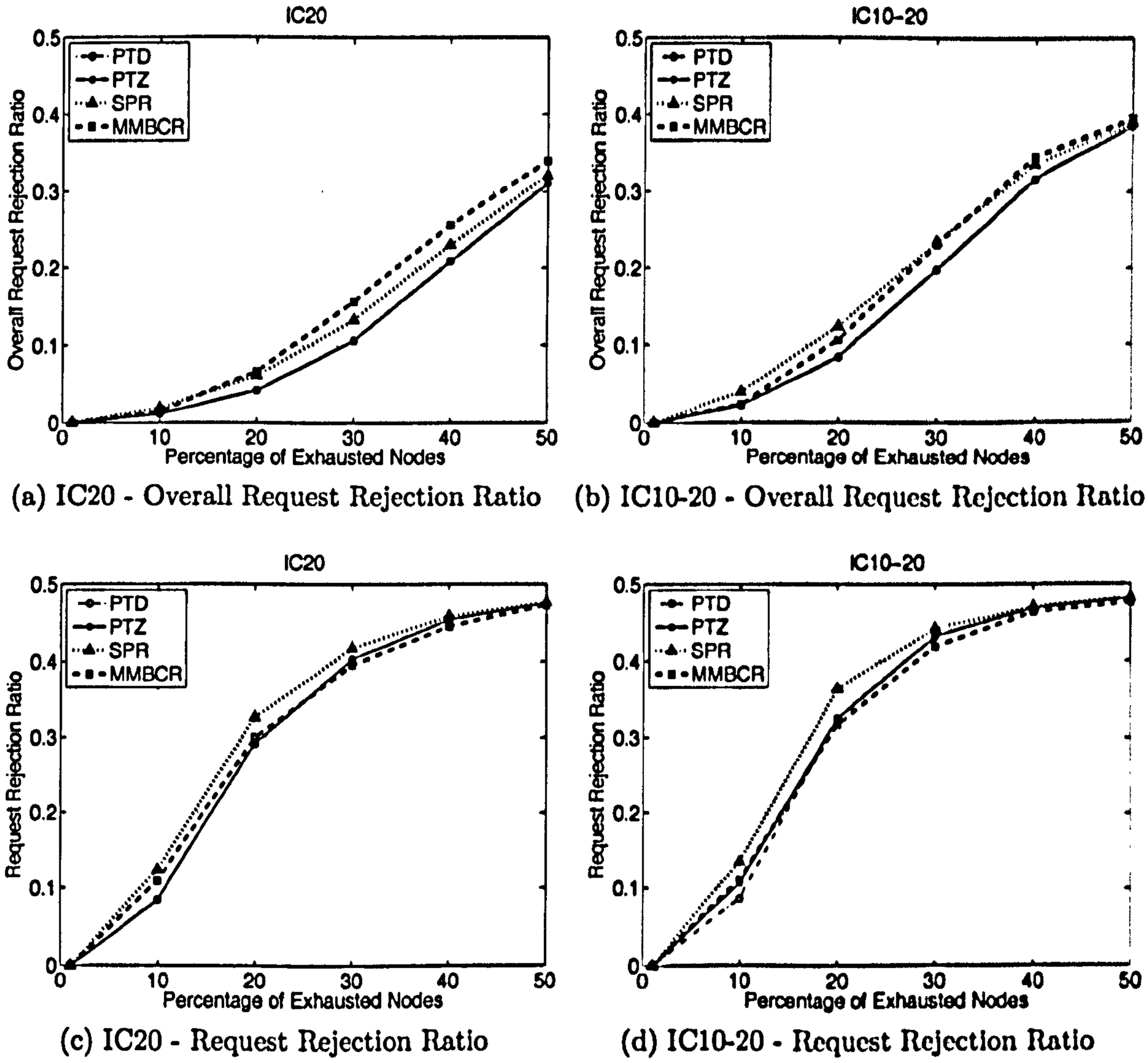
The results discussed in the previous section clearly indicate that the two proposed algorithms (PTZ and PTD) offer potentially improved performance when compared to the two benchmark algorithms (SPR and MMBCR) with respect to the point when full connectivity can no longer be guaranteed. Further experiments were undertaken to determine the relative merits of each algorithm as the network energy levels were allowed to further decay and nodes progressively became inactive. Results were obtained for both homogeneous and random initial energy levels when the networks were allowed to degrade to 90%, 80%, 70%, 60% and 50% of initial capacity in terms of active nodes. Figure 5.10a and Figure 5.10b show the overall mean call request rejection ratio as a function of the number of remaining active nodes for both initial energy distributions[3].

The overall mean ratio is measured from the beginning of the simulation until the percentage of exhausted nodes reaches 1%, 10%, 20%, 30%, 40% and 50%. Both PTZ and PTD algorithms outperform SPR in terms of their ability to support requests in situations when full network connectivity cannot be guaranteed. The request rejection ratios for PTD and PTZ are indistinguishable and both are lower than that obtained for SPR for the same number of active/inactive nodes. Indeed, it appears that the use of PTD/PTZ algorithms gives a rejection performance benefit equivalent to having (at least) 3% of extra active nodes. MMBCR also offers inferior performance when compared to the proposed algorithms.

On the other hand, Figures 5.10c and 5.10d show the instantaneous mean ratio of rejected calls for the period between the probes. For example, the ratio of dropped packets for 20% of exhausted nodes is equal to the ratio of dropped events which have occurred since the network reached the 10% of exhausted nodes over the number of packets offered to the network over the same period. From these graphs it is noted that PTD and PTZ compare favourably to SPR. The same picture is obtained for MMBCR until the network of active nodes becomes rather sparse. When the number of inactive nodes increase beyond 20% the situation changes and MMBCR incurs lower instantaneous rejection ratio. However, even in this case, the number of overall dropped packets is significantly lower for the proposed algorithms (Figures 5.10a, 5.10b).

The superior performance of PTD and PTZ can be explained by the fact that the proposed schemes attain more homogeneous energy configuration. Fig-

[3]Note that PTD and PTZ are almost indistinguishable.

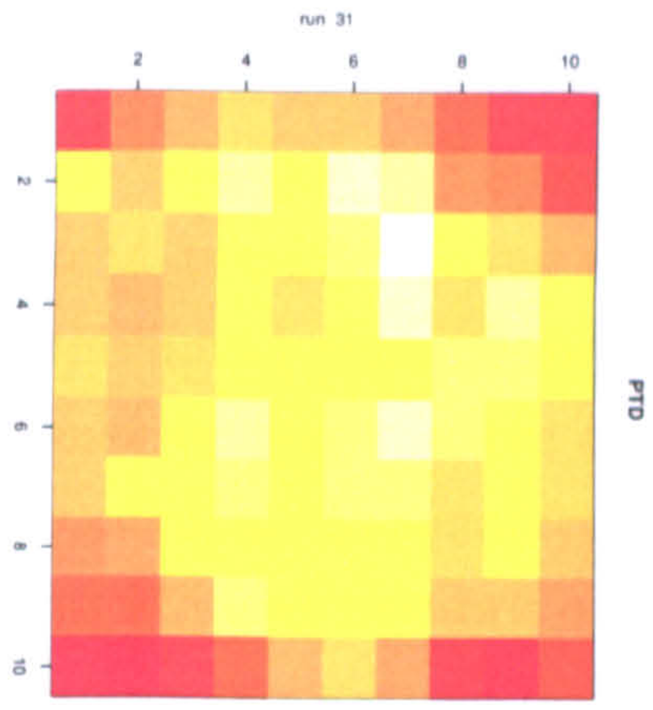


(a) IC20 - Overall Request Rejection Ratio (b) IC10-20 - Overall Request Rejection Ratio

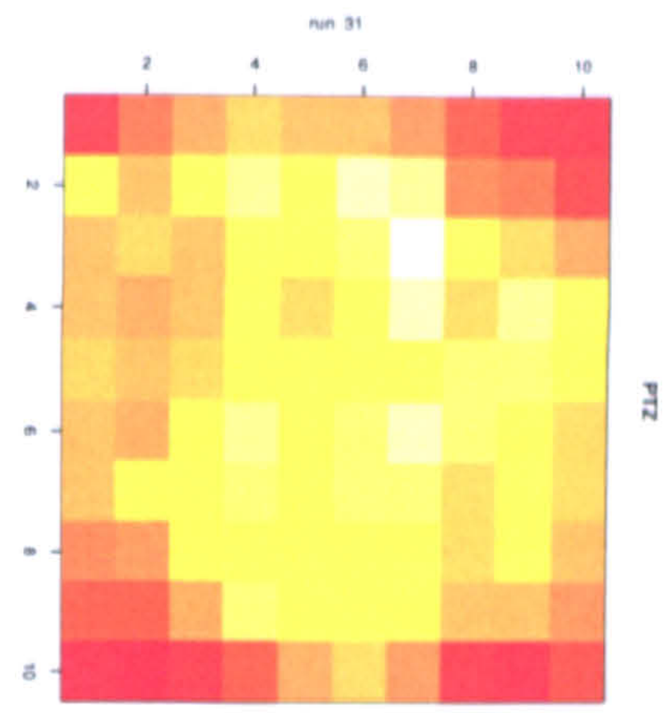
(c) IC20 - Request Rejection Ratio (d) IC10-20 - Request Rejection Ratio

Figure 5.10: Request rejection ratio as a function of the percentage of exhausted nodes.

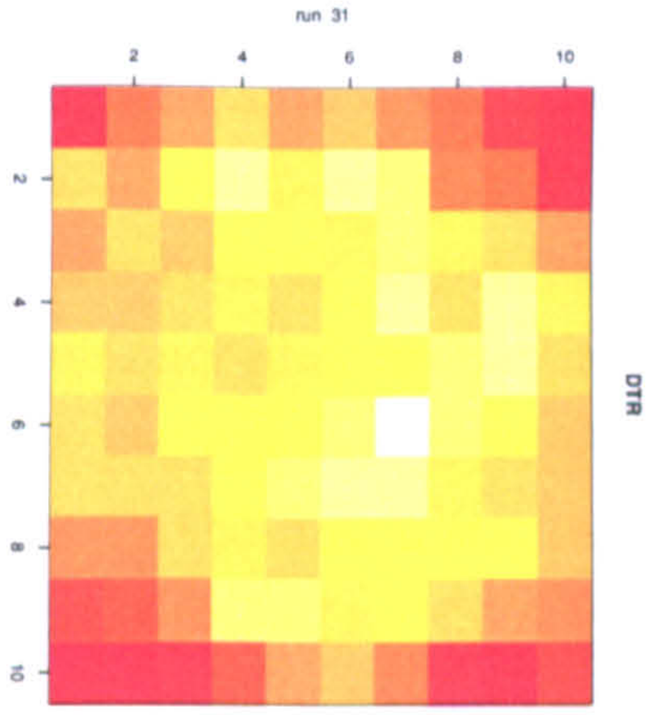
ure 5.11 shows a typical distribution of the node residual energy at the time (specific for each algorithm) when the first node dies. The colour coding of contour plots use red for nodes with high residual energy and as the residual energy drops the colour goes towards yellow with zero being white. It can be seen that MMBCR under-utilises the boundary nodes while overloading the central ones. SPR produces more spiky configuration than the smoother PTZ and PTD. The two new proposed algorithms seem to use more evenly the network resources and attain almost the same picture of residual energy when the first node is exhausted.



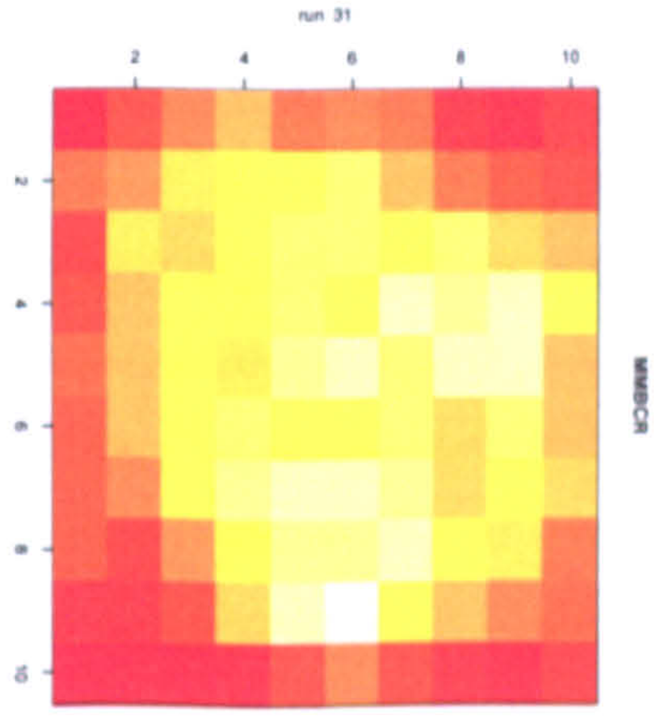
(a) PTD



(b) PTZ



(c) SPR



(d) MMBCR

Figure 5.11: A typical energy configuration when the first node dies (IC20).

5.6 Summary

Two new routing algorithms were proposed and showed potential in extending the lifetime of a sensor network. The premise behind these algorithms is that homogenisation of the energy distribution is a key factor in providing loss-free operation for as long as possible. Furthermore, PTD and PTZ offer lower overall request rejection ratios compared to previously proposed schemes. The algorithms tend towards forcing a more homogeneous energy distribution, regardless of the initial state of the network. Finally, the use of the PTD/PTZ algorithms ensures that the loss of nodes as a result of energy exhaustion will not be as catastrophic as connectivity (between node pairs) is more likely to be preserved and will give network operators more time to take remedial action in the form of sensor replacement or battery recharge and with reduced disruption.

For the two new routing algorithms, global knowledge is required and this is not feasible for real life applications. In order to maintain global knowledge of the system, background-overhead traffic must be carried by the energy constrained network. However, by the time this information is gathered the status of the network energy map changes and becomes outdated. Additionally, there is the danger that more energy will be expended for collecting this global knowledge compared to the merits gained and thus result in a reduction of the network lifetime. Finally, the protocol will have poor scalability if a signalling protocol is used to build complete energy maps for each node. Consequently, modified versions of the routing algorithms will be necessary which will operate in a distributed manner and use partial knowledge of the network state.

The next chapter considers the proposed routing schemes under realistic assumptions. In the first instance, the protocols still assume global knowledge but the model is ported to NS-2. Subsequently, the chapter studies the effects of varying the precision of known residual energies and draws conclusions as to the relative merits of changing the precision of reporting energy levels. Practical implementations of PTD and PTZ are developed which operate using partial knowledge (each node knows the residual energy of a subset of the network nodes rather than the full energy map). The partial knowledge information is collected and maintained with the aid of an in-band signalling scheme. It is shown that there is a trade-off between obtaining precise information and the signalling overhead associated with its collection and maintenance. Finally, it is shown that the proposed routing schemes outperform SPR and are excellent candidates for energy constrained ad-hoc networks.

Chapter 6

Practical Implementation of Point-To-Diagonal and Point-To-Zero

Chapter 5 introduced two novel path selection disciplines; Point-To-Diagonal (PTD) and Point-To-Zero (PTZ), which have both shown potential to improve network lifetime. In this chapter, the analytical model is ported into NS-2 in order to perform a full varied simulation. The analytical approach by necessity is restricted in that it considers only one specific case. The simulation approach allows for more variety. The model developed in NS-2 is based on the DSR protocol, but can be applied to any routing protocol with the appropriate modifications.

The remainder of this chapter describes the steps taken to port the model from the C++ iterative simulator to NS-2. The first stage of modelling assumes that the nodes have knowledge of the residual energy of all other nodes. This implementation stage will be referred as Global Knowledge. In the next stage, this assumption is removed and the model describes a routing protocol which can be applied to networks to maximise the network lifetime. The second stage of the implementation is referred to as the Partial Knowledge stage.

6.1 Global Knowledge Implementation in NS-2

The model presented in Chapter 5 imposes the following assumptions:

1. Packets are relayed instantly over the selected path.
2. All packets have the same length.

3. A node's residual energy decreases by a unit (1) every time a packet is relayed and, subsequently, the residual energy belongs to the whole number set ($\mathbb{N}$).
4. The residual energy decreases affect only nodes that belong to the relaying path and not the neighbours who would spend energy overhearing or receiving transmissions.
5. All candidate paths lie within the rectangle formed by the source and destination nodes in the corners, which, in terms of hop count, leads to paths of equal lengths.

The Global Knowledge implementation in NS-2 removes these assumptions and employs the Global Operations Director (GOD) to provide knowledge of topology and the residual energy to the nodes. The assumptions in the Global Knowledge implementation are:

- Any network node can obtain global topological information from GOD.
- Any network node can obtain knowledge about the residual energies of all other nodes in the network through GOD.

6.1.1 Global Knowledge Strongest Path Routing

The method used to find the strongest path in Section 5.4 can be described as: iterate through all paths in the rectangle and calculate the cost for each path by adding the residual energies of nodes belonging to the path. This approach is not computationally possible when paths are no longer restricted within the rectangle; a consequence of the number of possible paths.

For the global knowledge implementation, the nodes construct a directed graph $G(V, E)$ and the weight $(W(e))$ of an edge $(e = (u, v))$ with starting vertex u and ending vertex v, is assigned to $W(e) = E(N_v(t))$, $\forall e \in E$. When this graph is constructed, the problem of finding the strongest path becomes equivalent to the problem of finding the longest loop-free path. On the other hand, if the edge weights are assigned with negative residual energy; $W(e) = -E(N_v(t))$, $\forall e \in E$, then the problem becomes the well known shortest path problem. Algorithms such as Bellman-Ford [66] are general enough to operate on graphs with negative link weights. However, all weights in graph G will be negative and hence the graph will contain negative cycles. Since Bellman-Ford attempts to find the path with the lowest possible distance (sum of weights), the algorithm will get trapped in negative cycles and loop indefinitely. This will

occur because, through every negative loop, the distance will further decrease. In order to avoid this indefinite loop, the edge weights must be modified to eliminate negative weights (and subsequently negative cycles). This can be achieved by adding a positive constant number, sufficiently large in magnitude to eliminate negative weights. If the positive sufficiently large constant is C_0, the weights will become $W(e) = C_0 - E(N_v(t)), \forall e \in E$. Once this constant C_0 is computed, Bellman-Ford operates on a digraph with non-negative weights but the computation complexity is $O(N^3)$. Hence, it would be more preferable to use Dijkstra's shortest path algorithm [66, 71] which computes the shortest path more efficiently; $O(N^2)$.

The value of C_0 has to be significantly large to ensure non-negative weights. In order to satisfy this requirement, C_0 is selected to be equal to the maximum residual energy of the initial state vector E(0) plus ϵ.

$$C_0 = \epsilon + \max_{N_i \in G'} \left(E(N_i(0)) \right) \tag{6.1}$$

where $\epsilon > 0$ being a small arbitrary value (i.e. 0.0001). The addition of this small value is necessary to ensure non-zero weights at the beginning of the network's operation. For the remainder of this chapter, this method with edge weights equal to $W(e) = C_0 - E(N_v(t)), \forall e \in E'$ will be referred as SPRv1.

An alternative solution of finding the Strongest Path from the graph G, would be to assign the weights to $W(e) = \frac{1}{E(N_v(t))}$. This method is called Minimum Battery Cost Routing [108, 109]. For the remainder of this thesis, this second routing scheme will be referred as SPRv2.

6.1.2 Global Knowledge PTD & PTZ

Initial NS-2 implementations of global knowledge Point-To-Diagonal and Point-to-Zero attempted to consider all possible paths from source to destination as candidates for selection. However, finding all possible paths for a grid with dimensions greater than 5×5 is a very computation intensive task. Additionally, there is minimal benefit in finding all routes. Consider a 5×5 grid network with source and destination nodes located on the corners of the grid as shown in Figure 6.1. One of the possible paths would be the "snake-like" path that passes through all the nodes in the network. The minimum number of hops between S and D is 8, but if traffic routed over the "snake-like" path the number of relays will be 25. Because PTD and PTZ essentially attempt to minimise the variance of the future states, these algorithms will select the "snake-like" paths because

energy is reduced from all nodes in the network and the variance remains almost constant[1]. This behaviour is not desirable because such paths contain virtual loops. To appreciate the term "virtual loop" consider the first transmission from node S. The transmission is overheard from the node A, but it is not processed because the next hop of the packet is node B. Subsequently, the packet is relayed over the nodes of the first and second row until the packet reaches for a second time node A. This is referred to as a virtual loop and paths containing such loops are clearly not desirable.

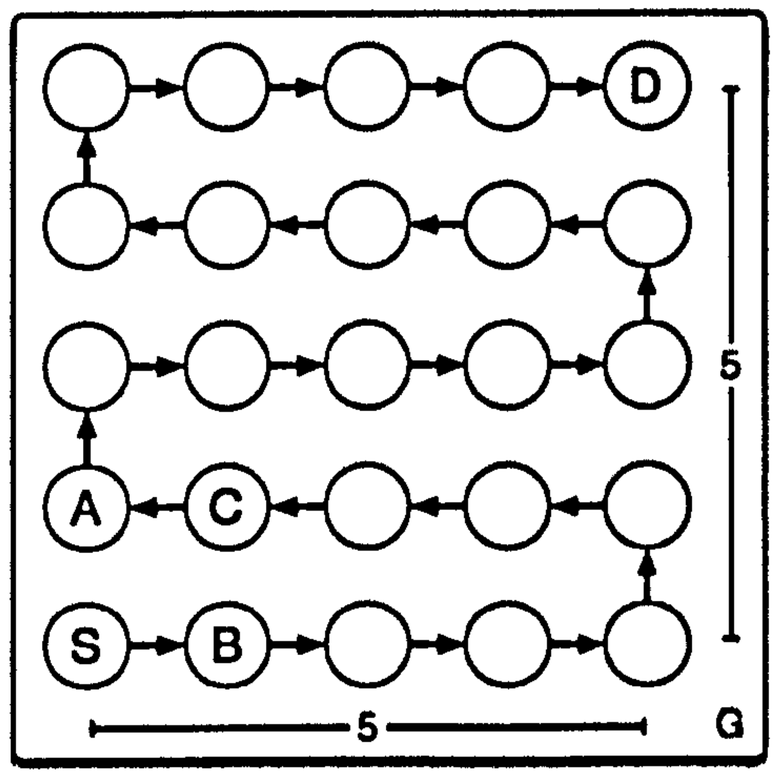


Figure 6.1: Illustration of a "snake-like" path.

In order to reduce the computation overhead of PTD and PTZ and avoid virtual loops, both algorithms construct a connectivity graph $G(V, E)$ with edge weights equal to unity for every link in the network. Then, a single run of Dijkstra's Shortest Path algorithm is performed. The algorithm returns a predecessor map and the distance of each node from the source node. Using the predecessor map and the distance of each node, the flowchart of Figure 6.2 is used to determine all shortest paths.

[1]The variance does not remain constant because not all nodes perform the same number of transmit/receive operations. For example, nodes located on the border, perform one fewer receive operation than those located in the centre of the network. In the same manner, nodes located on the corners of the network perform two fewer reception operations compared to nodes located in the centre of the network.

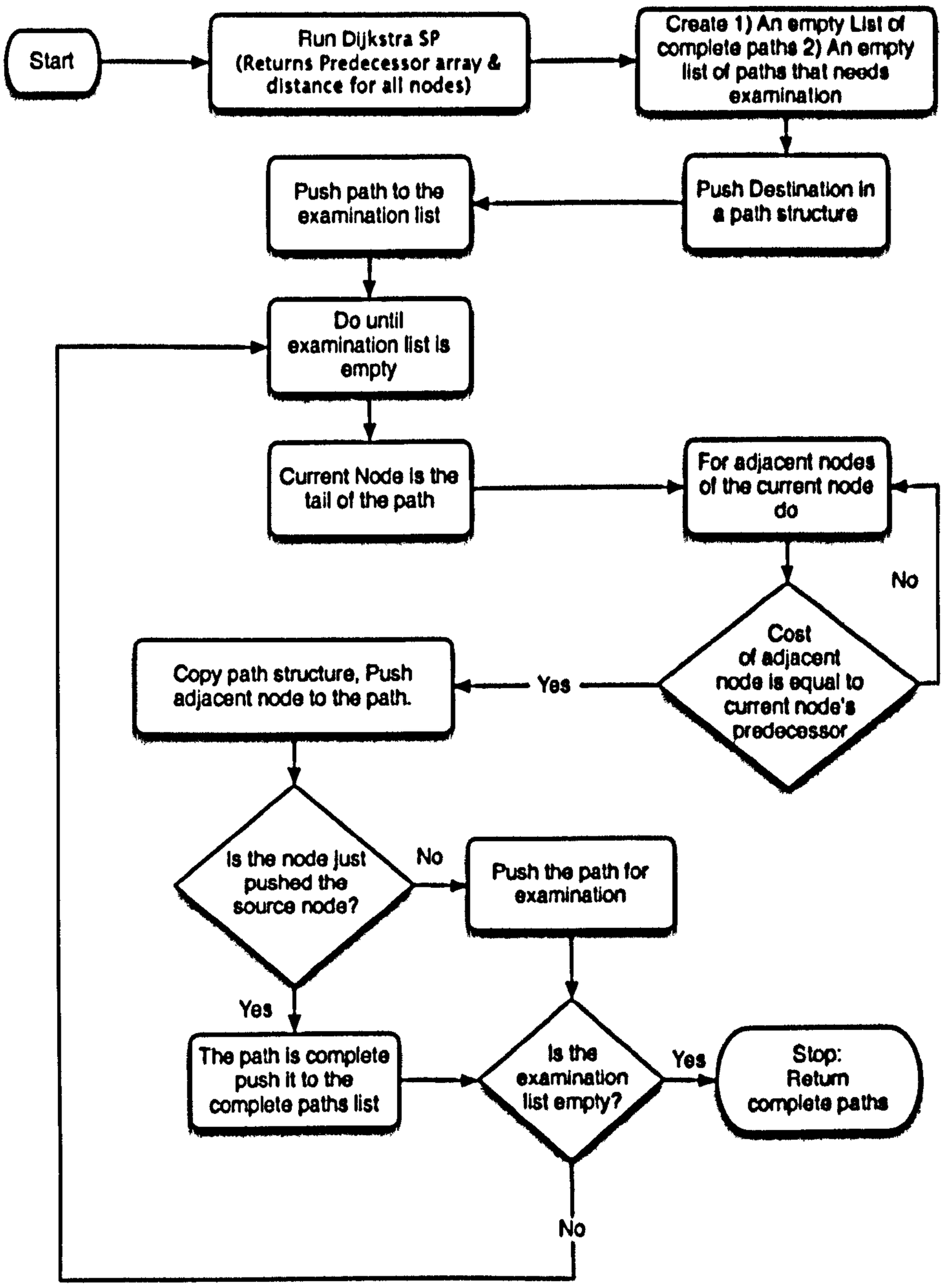
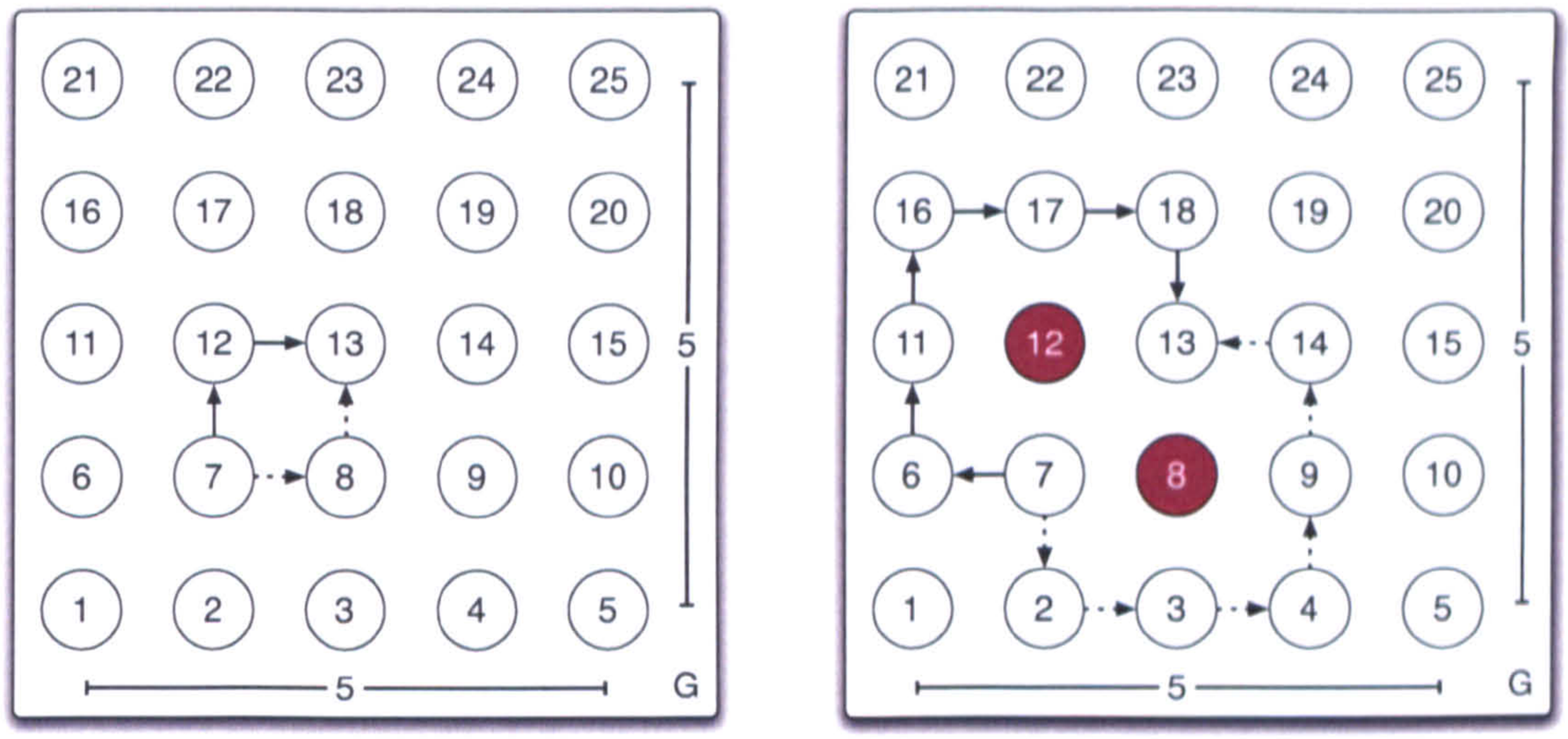


Figure 6.2: Flowchart to illustrate the all shortest paths search algorithm.

At first glance, this might seem similar to the model developed in Chapter 5. However, the approaches differ significantly when connectivity is lost. For example, consider the network of Figure 6.3a. For the node pair $(7, 13)$ there are 2 candidate paths $7 \rightarrow 12 \rightarrow 13$ and $7 \rightarrow 8 \rightarrow 13$. When nodes 8 and 12 exhaust all their energy reserves, then the model of Chapter 5 wrongly assumes that connectivity between 7 and 13 is lost. However, both new shortest path algorithms find new paths as indicated in Figure 6.3b.



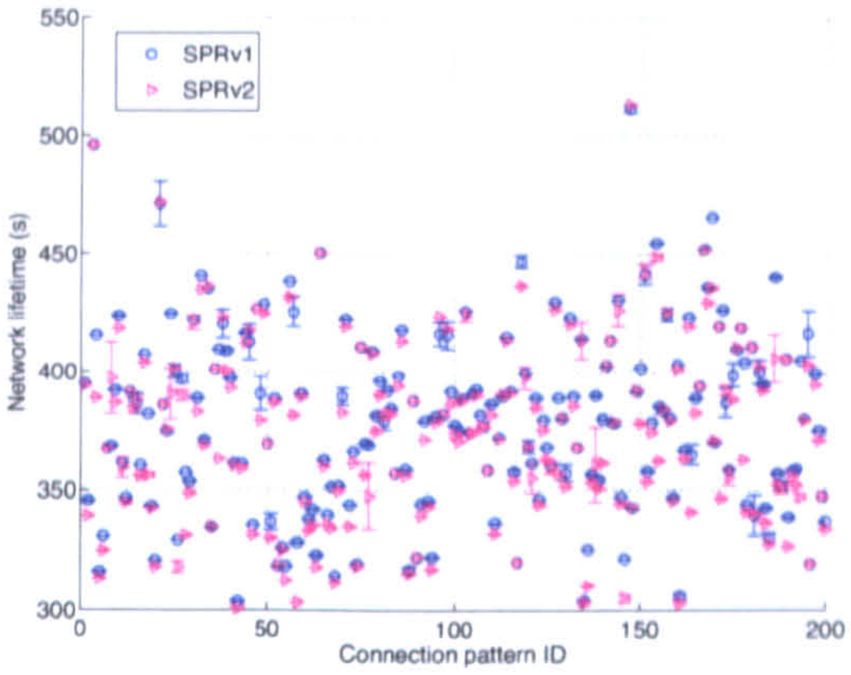
(a) All nodes alive. (b) Nodes 8, 12 exhausted.

Figure 6.3: Losing connectivity within the rectangle finds alternative routes.

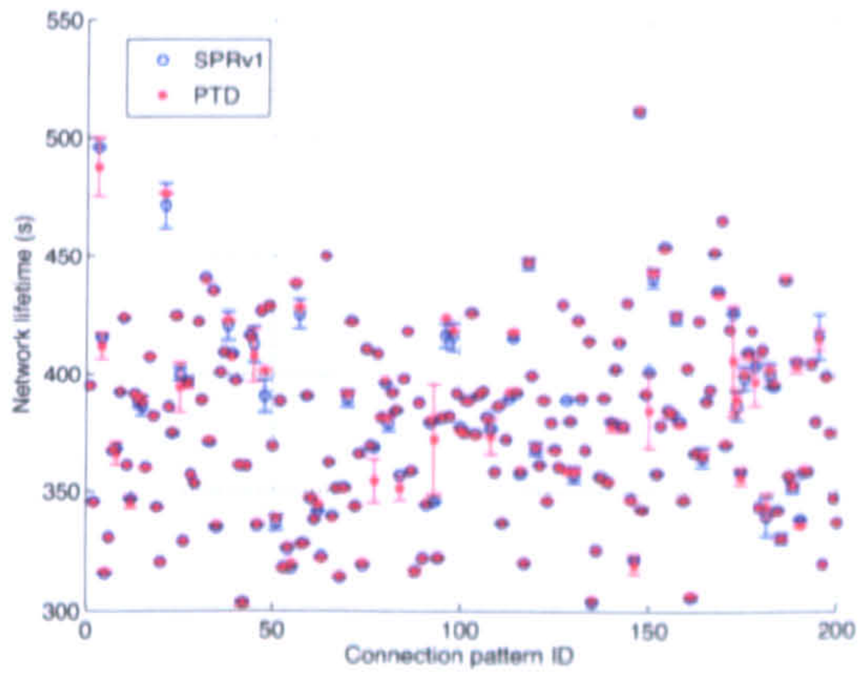
6.1.3 Simulation Setup and Results

The simulations are carried out on a grid static network with dimensions 10×10. The nodes are located in distance of 10 metres apart and the transmission power selected, is such that the communication range is 15 metres. The initial nodal energy is set to 50 J. The simulation time is set to 3,000 seconds. 200 random connections patterns are used to load the network, with 1,000 connections each with an average duration set to 30 seconds. The data rate is 1 packet per second.

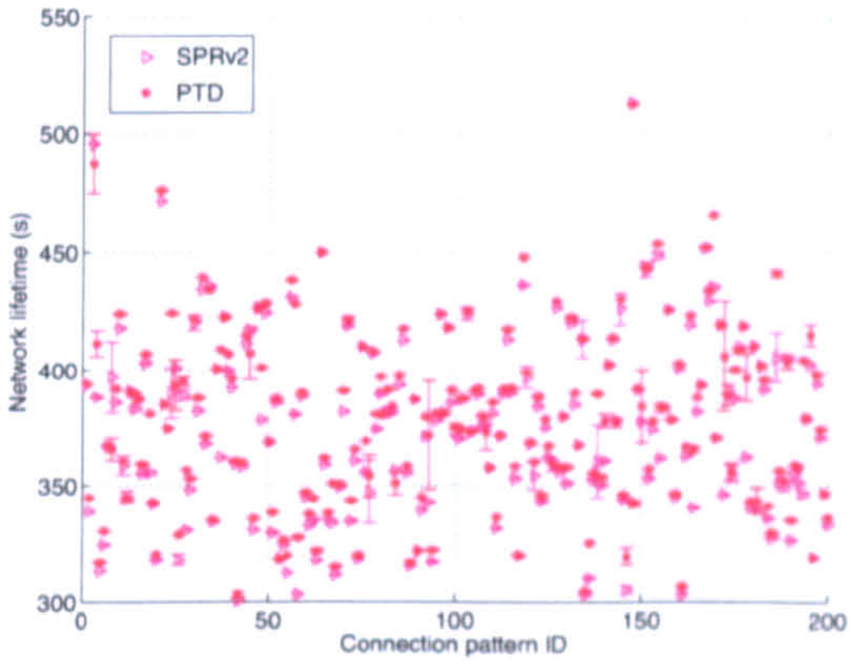
Initially, a comparison between SPRv1, SPRv2 and PTD is performed and the following benchmark results are obtained. Figure 6.4 shows the achieved lifetime for the 200 connection patterns for all three algorithms; SPRv1, SPRv2 and PTD. The reader is reminded that the lifetime definition used is described in Section 5.5, and it is the point where one of the network nodes has exhausted all its energy reserves and thus becomes permanently inactive. From Figure 6.4, it is first unclear if the new algorithms offer improvement. For this reason, a histogram of their difference is drawn to give an alternative view of how the algorithms compare.



(a) SPRv1 and SPRv2.



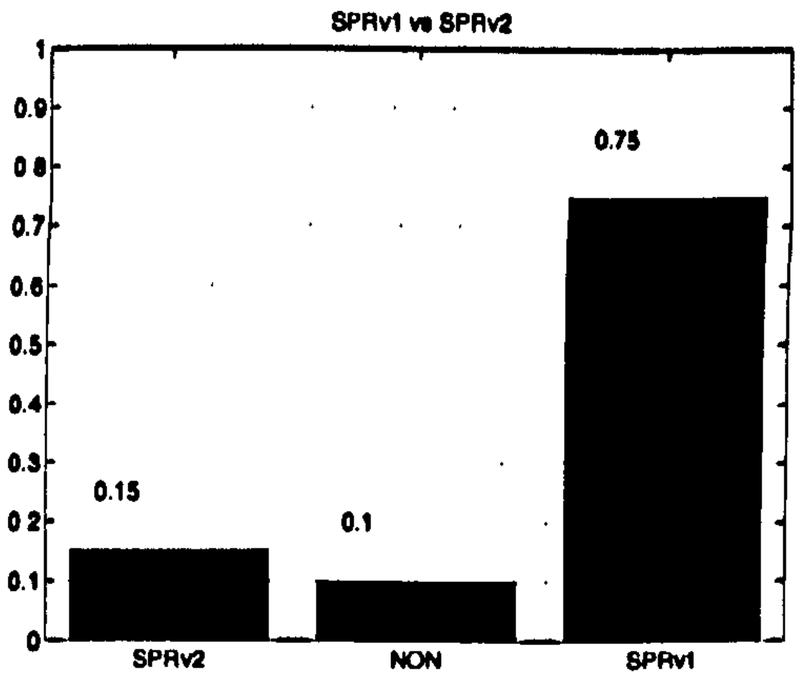
(b) SPRv1 and PTD.



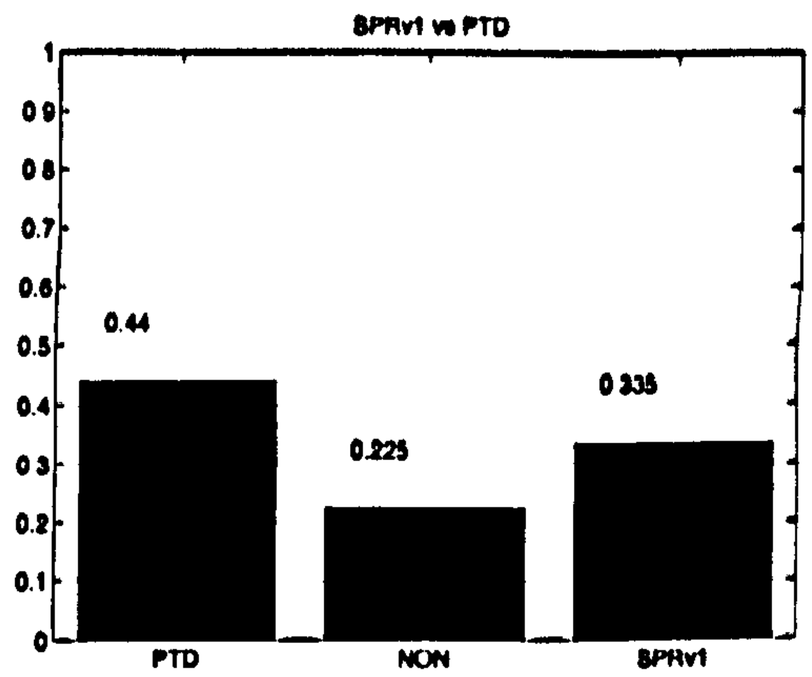
(c) SPRv2 and PTD.

Figure 6.4: Network lifetime in seconds for 200 connection patterns.

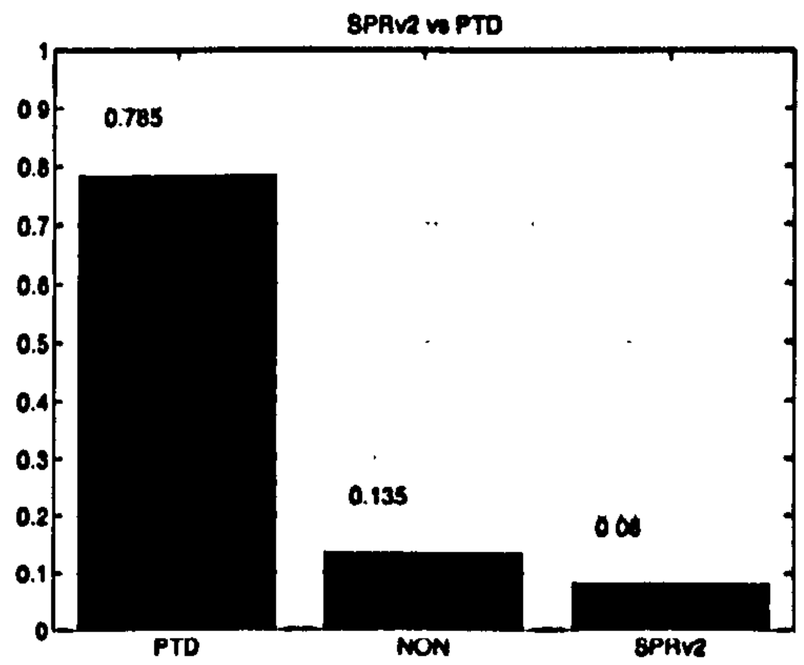
Figure 6.5 shows a histogram of the difference between lifetimes. More specifically, Figure 6.5a shows that 75% of the time SPRv1 outperformed SPRv2, 10% of the time both algorithms achieve the same lifetime and 15% of the time SPRv2 outperformed SPRv1. Figure 6.5b shows that 44% of the times PTD outperformed SPRv1, 22.5% of the time the two algorithms performed the same and 33.5% SPRv1 outperformed PTD. Finally, the numbers for SPRv2 and PTD are in favour of PTD with 78.5%, 8% in favour of SPRv2 while for 13.5% it is a draw.



(a) SPRv1 and SPRv2.



(b) SPRv1 and PTD.



(c) SPRv2 and PTD.

Figure 6.5: Network lifetime in seconds for 200 connection patterns.

The call rejection ratio is another factor of interest. Before going into detail regarding the call rejection ratio, the reader is reminded of the series of events as shown in Figure 6.6.

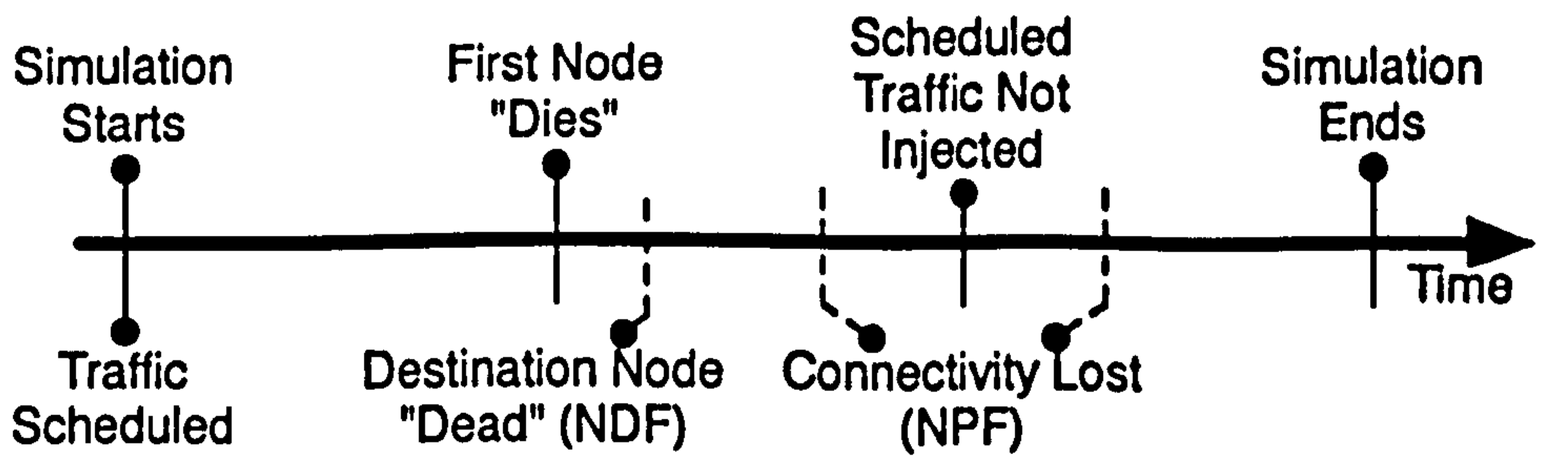


Figure 6.6: Sample simulation timeline.

Before the simulation starts, the traffic generator schedules all traffic that the nodes will send during the simulation. When it starts, nodes communicate with each other and consume energy. At a later point in time, the first node "dies" and the network has the potential to reject communication requests which have

the exhausted node as their destination. This potential is realised at a later point; marked in Figure 6.6 with the label Destination Node "Dead" (NDF). Subsequently, more nodes might "die" and more requests may get rejected because the destination node is exhausted (NDF). Additionally, rejection of more requests can occur because the algorithm can not find a suitable path to serve the request and this indicates that the network has become partitioned. This is marked in Figure 6.6 as Connectivity Lost - No Path Found (NPF).

Figure 6.7 shows the call rejection ratio versus the percentage of exhausted nodes. The definition of call rejection ratio is the total number of rejected connections over the total number of connections attempted. The total number of connections rejected are decomposed into 1) connections which were rejected because the destination node was exhausted and 2) connections that were dropped because the destination node was unreachable.

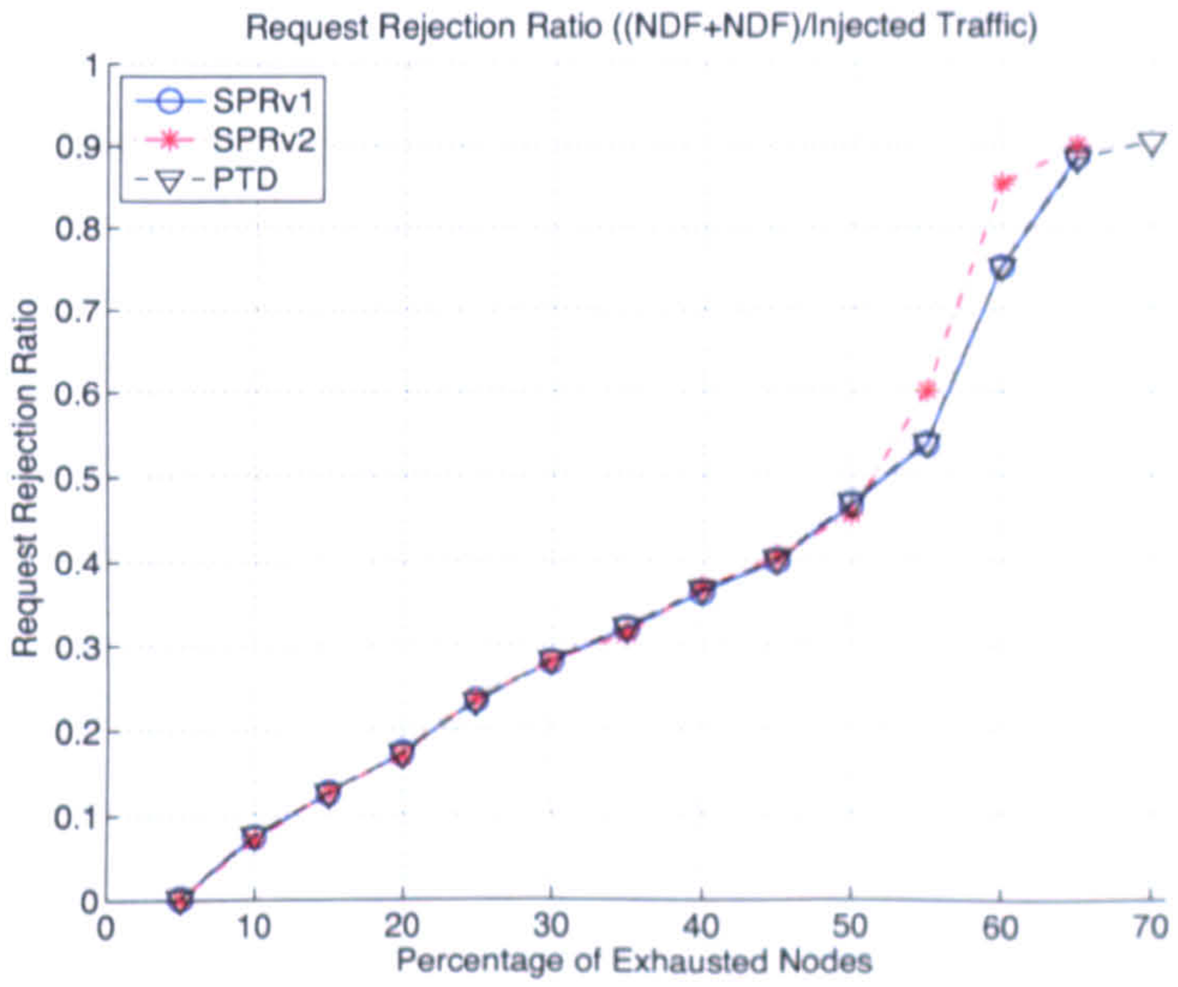
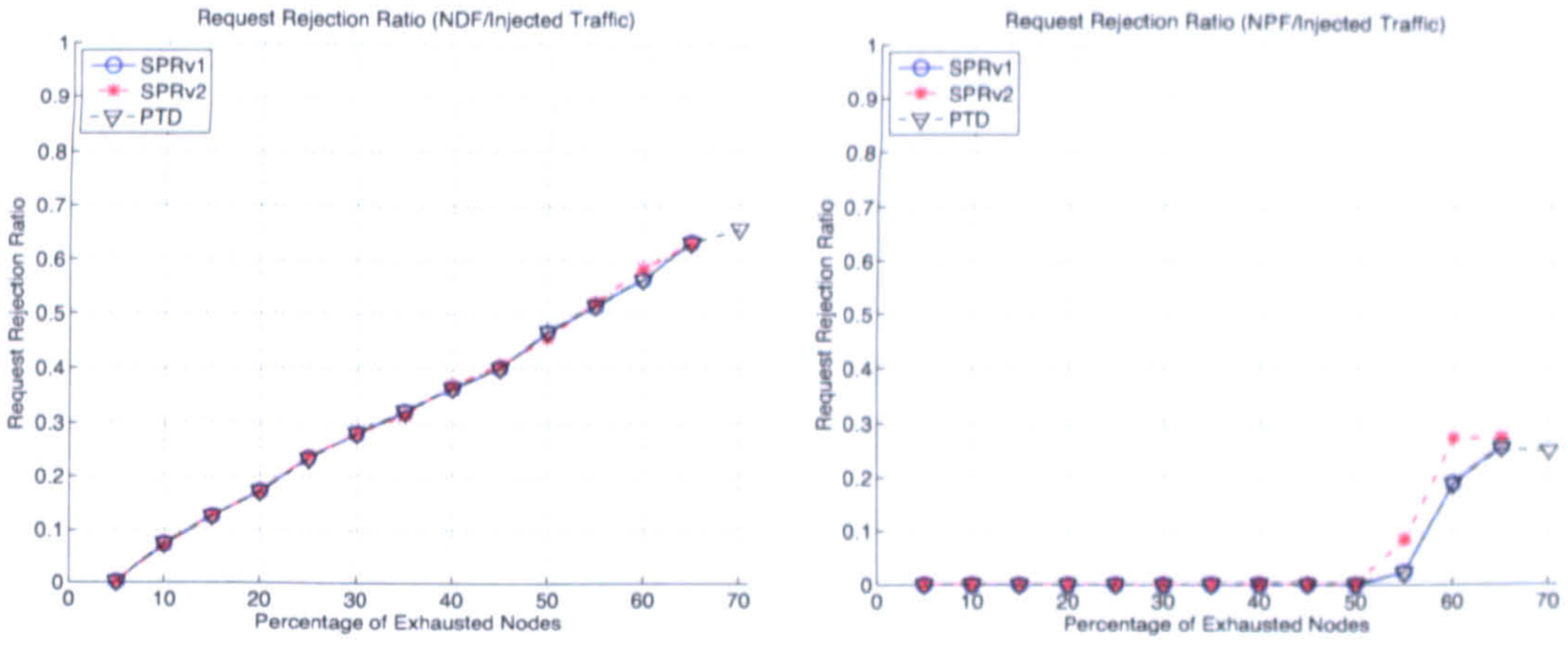


Figure 6.7: Request Rejection Ratio for injected traffic versus percentage of exhausted nodes.

In order to gain better understanding of the call rejection ratio graph, the decomposed rejection ratios are plotted in Figure 6.8. The graph of Figure 6.8a shows the ratio of rejected calls due to the unavailability of the destination node. As it is expected, the number of call rejections due to unavailability of the destination node, relates linearly with the percentage of exhausted nodes with slope almost unity. This is the expected behaviour because, for each connection, the destination node is selected uniformly among the nodes of the network. For example, it is expected that when 50% of nodes are exhausted half of the injected

traffic will get NDF dropped.



(a) No Destination Found (NDF). (b) No Path Found (NPF).

Figure 6.8: Request rejection ratio behaviour.

Figure 6.8b shows the NPF call rejection versus the percentage of exhausted nodes. It must be noted that, initially, no calls are rejected due to path unavailability until the percentage of exhausted nodes reaches 50%. After the rejected calls reaches the 50% mark, the graph is convex until it reaches 60% of exhausted nodes where the concavity changes and the function becomes concave.

The first derivative of a function represents the rate of change of the request rejection ratio ("speed"), and a positive first derivative represents a monotonically increasing graph[2], while a negative first derivative a monotonically decreasing graph. Clearly, the request rejection ratio is expected to be a monotonically increasing function since the number of exhausted nodes is increasing in the network and therefore, it is expected that the number of rejected calls will increase.

The second derivative represents how fast the rate is changing ("acceleration"). A positive second derivative represents a monotonically increasing rate of change (first derivative) and therefore an upwards cavity. A negative second derivative indicates a monotonically decreasing rate of change (first derivative) and therefore a downwards cavity. An instantaneous zero second deviate indicates a point of cavity change, such as the one observed at 60% of exhausted nodes in Figure 6.8b.

[2]This is true when the function is continuous within the interval of study. In this case, the graph is not a function but a sequence and its domain is $\mathbb{N}$. For this reason, the discussion is made on the extended function of the sequence, which connects the points of the sequence with lines. Since the discussion applies to the extended function which is a superset of the sequence, it will also apply to the sequence which is a subset of the extended function.

One might expect that the request rejection ratio graph will be convex function even when the percentage of exhausted nodes exceeds 60%. As the number of exhausted nodes increase, the rate at which calls are rejected will continue to increase and therefore expect an upwards cavity. However, this is not the case in the graph of Figure 6.8b.

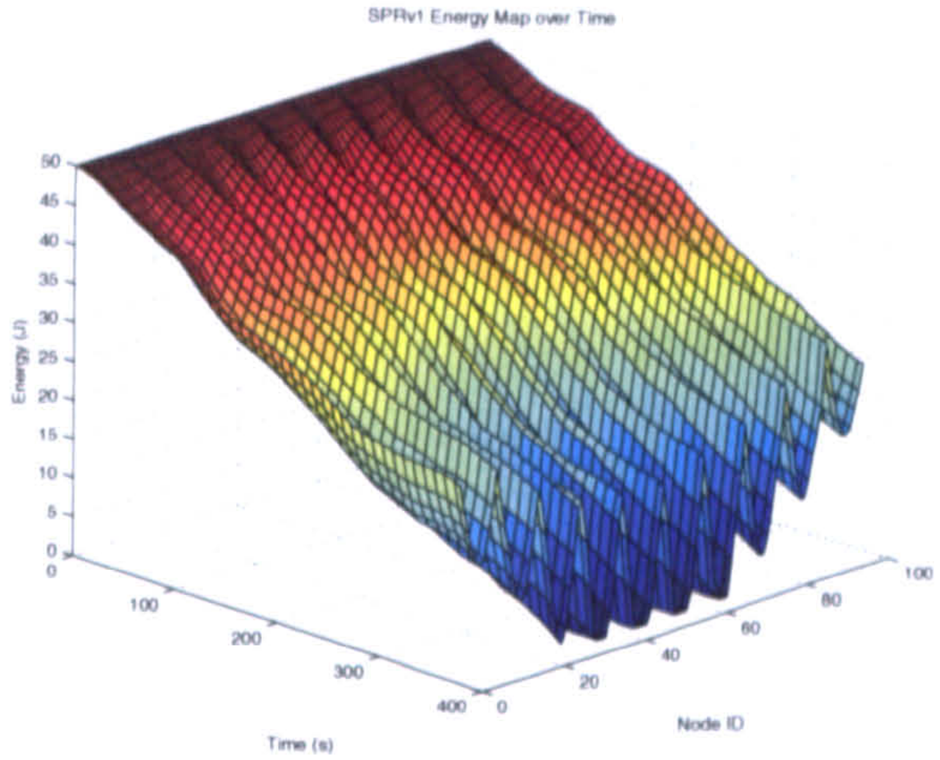
In order to explain the effect of cavity change, the traffic generator was examined closely. The traffic generator selects the source and destination of the connections uniformly among all the available nodes in the network before the simulation starts. At this point, all the nodes in the network have residual energy to initiate connections and inject traffic to the network. As the simulation progresses, nodes consume energy and their residual energy drops until they "die" and become permanently inactive. Inactive nodes are not able to complete any events and therefore they do not inject traffic to the network. Refer back to Figure 6.6 which shows the simulation timeline to visually represent the described effect. As number of active nodes decreases, the traffic injected in the network also decreases and consequently, the speed of call rejections decreases which results in a downwards concavity.

Finally, Figure 6.9 shows a three dimensional graph of the residual energy of all nodes against time for the three algorithms. The reader is reminded that the nodes are arranged in a topology similar to the one shown in Figure 6.3 but the grid order dimension is 10×10. Therefore nodes 1-10 are located on the first row from the bottom, while nodes 91-100 are on the first row from the top. Observing again the energy time-lines, it is clear that nodes located on the edges of the grid have higher residual energy, while nodes located towards the centre of the grid are exhausted.

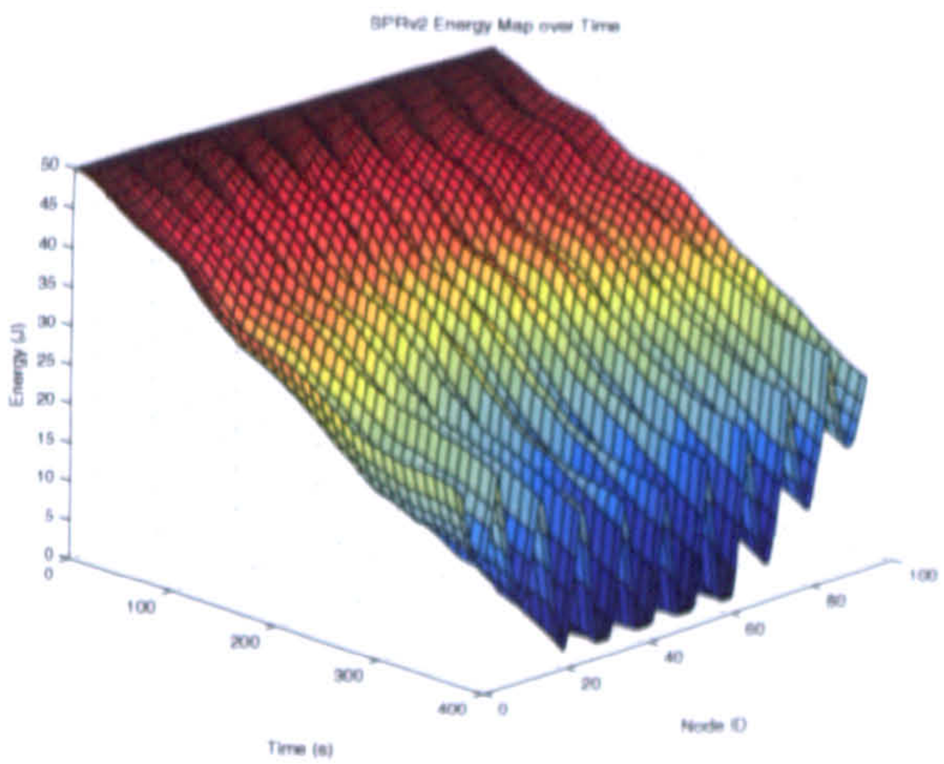
From the results presented here it can be observed that PTD does not always offer an improvement over the SPR; an apparent contradiction of the results obtained in Chapter 5.

The iterative model developed in Chapter 5 decrements the energy of the nodes by an integer unit every time an event occurs. As a consequence, the cost of the paths is also an integer value. Because the cost of the paths is integer, there is a high probability that more than two paths for a source destination pair will have the same cost, in which case the SPR discipline will select one at random; PTD and PTZ make more intelligent selections.

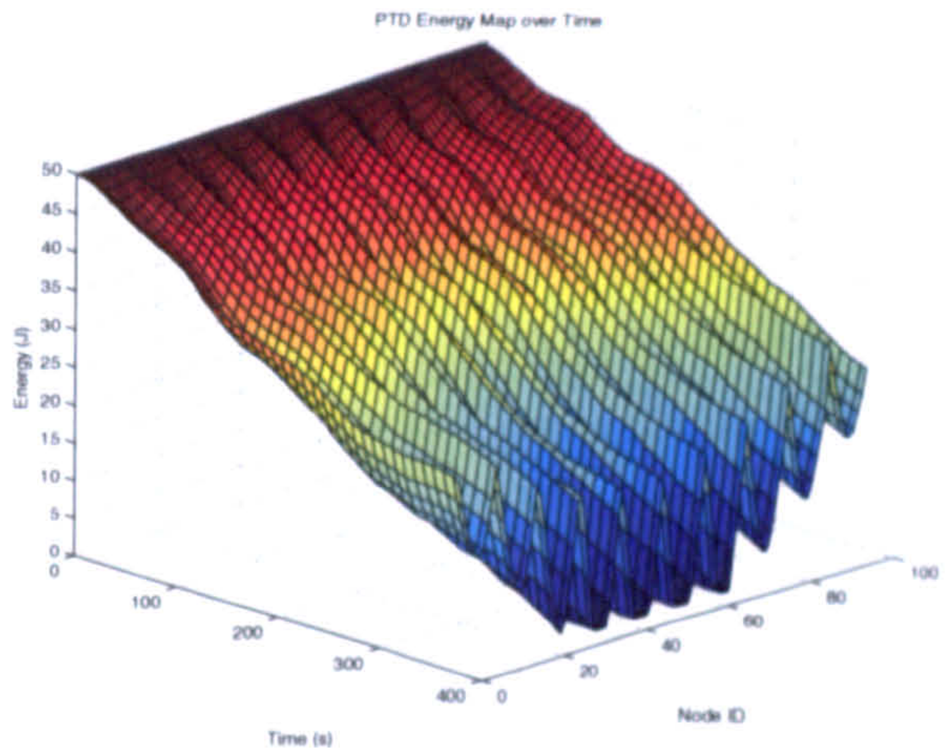
For the NS-2 model, the energy decrements are numbers belonging to the set of real numbers; $\mathbb{R}$. Additionally, each packet no longer is the same length because each overhead varies. Each DSR packet encapsulates the path within its



(a) SPRv1.



(b) SPRv2.



(c) PTD.

Figure 6.9: Energy timeline for the three algorithms.

header, therefore packets routed through paths of different lengths have variable packet sizes. Due to the packet size variability, the energy is not reduced from the nodes in discrete steps and, consequently, the total cost of the path is usually unique among the candidate paths and therefore SPR rarely selects a path at random. Additionally, it must be noted that SPR has the ability to diverge from the paths inside the rectangle. On the other hand, PTD uses a unity graph to determine the candidate paths and as a consequence, all the paths lie within the rectangle, until there are no longer any paths within it to satisfy the connection.

These differences represent the reason why in few occasions, SPR performed better than PTD. In fact, such cases occurred only when SPR selected paths lie outside the rectangle while PTD selected paths within it.

A more fair comparison of the routing disciplines would be to run SPR routing first and when multiple paths are available for selection, PTD and PTZ can be employed to perform a more intelligent selection. Additionally, in a real

implementation, each node has to obtain knowledge about the residual energy of the other nodes in the network via an information gathering protocol and consequently, the knowledge each node has will neither be instantaneous or high precision. There is a trade-off between saving energy by having up to date information when performing an intelligent decision and energy consumed to collect such information. The next section studies how precision affects the network lifetime.

6.1.4 Precision and Network Lifetime

In order to understand how precision affects the network lifetime, the SPR routing discipline is studied further and the Global Operations Director object is modified to report node residual energies in quantised levels. The quantisation method used is linear scalar quantisation. Other non-linear quantisation methods (particularly those having higher precision towards the lower end of the scale) can potentially offer prolonged lifetime, but require fine tuning to suit the particular application. In this work, simplicity takes precedence over other considerations.

If the residual energy level is quantised into M equally spaced energy levels ($M-1$ steps), then the spacing between them is $\delta e = E_0/(M-1)$, where E_0 is the maximum initial energy. The number of bits required to represent these distinct levels are $log_2(M)$ [114]. An example of the linear quantisation is shown in Figure 6.10. The following steps are taken to quantise the representation of nodal energy:

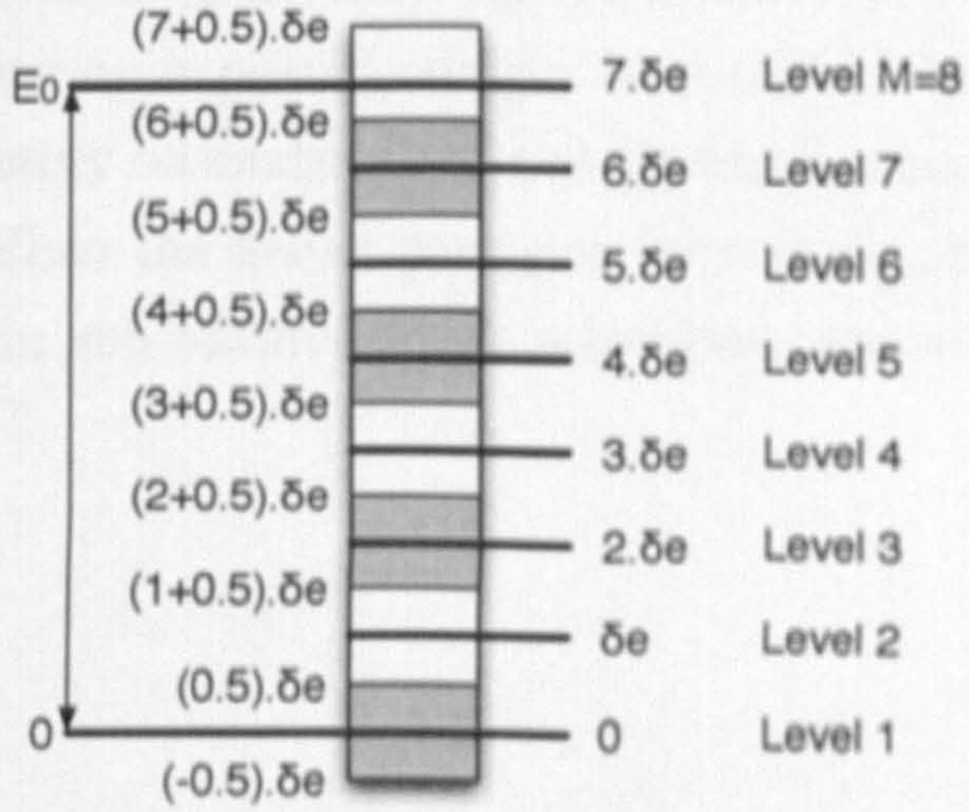


Figure 6.10: Linear Quantisation, for $E_0 = 7$ J, $M = 8$ steps, $\delta e = 1$ J.

- Define the number of energy levels M, i.e. $M = 8$

- Step size $\delta e = \frac{\max_{i=1}^{N}\left(E(N_i(0))\right)}{M-1}$, where N is the number of nodes in the network and $E(N_i(0))$ the energy of node N_i at time $t = 0$, i.e. $\delta e = 7/(8-1) = 1$ J.

- The quantised representation of the energy level of node N_x at time t ($E_Q(E(N_x(t)))$) is:

  $E_Q(E(N_x(t))) = round(\frac{E(N_x(t))}{\delta e}) \cdot \delta e$

  where $E(N_x(t))$ is the actual energy of node N_x at time t.

Using this quantisation method, a series of experiments were conducted to quantify the effect of quantisation to the network lifetime. In the first set of experiments, the number of connections is set to 3,000, the duration of each connection is uniformly distributed in [0-10] seconds and the data packet size is set to 512 bytes. The network topology is the same as before (10×10 grid) and the simulation duration is set to 2,000 seconds. The experiments were performed for 5, 10, 20 and 50 quantisation steps ($M = 6$, 11, 21 and 51 respectively) and no quantisation. The initial nodal energy is set to 50 J. All the experiments are carried out using the SPR routing discipline. Finally, the experiments are repeated for 10 random traffic profiles.

Figure 6.11 shows the lifetime achieved for various number of steps versus the rate. From the diagram it can be seen that there is a difference between network lifetimes for each quantisation value but this difference, although distinct, is not significant in absolute terms. As expected GOD with full precision achieves the longest lifetime and this is followed by 50, 20 10 and 5 steps/levels respectfully. The behaviour is entirely expected, simply because the larger the interval the least accurate the energy estimate is and the greater the margin for making poor routing decisions. When the lowest precision scheme is used (5 steps), the lack of precision results in sub-optimal route selections, which in turn impacts the lifetime.

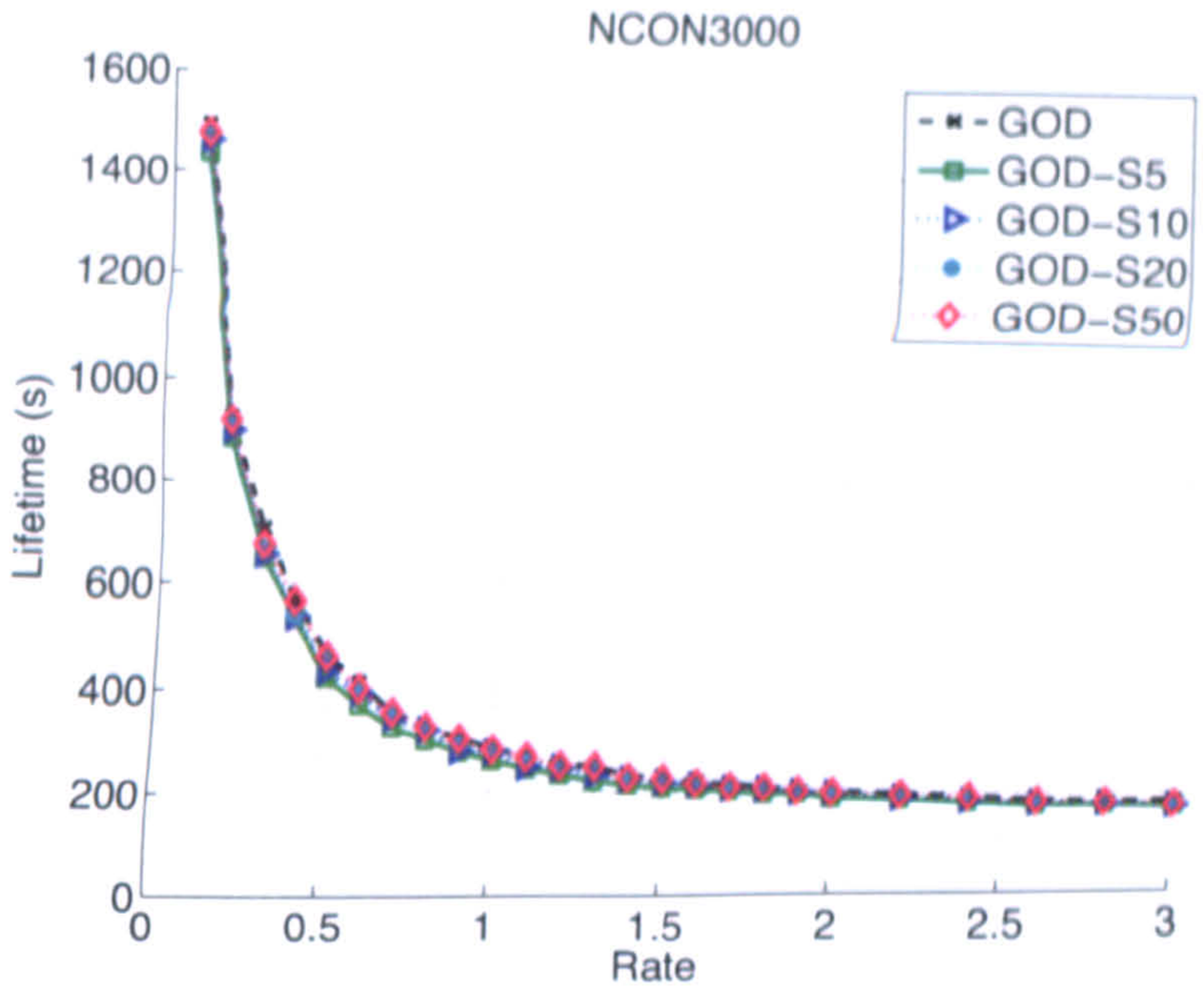


Figure 6.11: GOD lifetimes for Steps 5, 10, 20, 50 and GOD with full precision.

To further quantify the effects of quantisation level, the differences between lifetimes for 5-50 levels with reference to GOD with full precision is considered. Figure 6.12 is simply the difference between quantisation levels and the reference scheme at different rate values. The graph is plotted in terms of the best fit line, in order to look at the general trend. From this figure, it can be clearly seen that for low rates the lack of precision results in lower lifetime and as the rate increases the lifetime improvement drops. To explain this phenomenon consider the source/destination pair with n paths available between them. Independently of the path selection, the source and destination nodes will always be members of the selected path. When a connection is established between the source/destination pair, the load and the energy consumption is distributed across paths, but the source and destination are always going to deplete the same amount of energy, independently of the path selection. The high precision schemes will manage to distribute the energy depletion to the intermediate nodes but not for the source and destination.

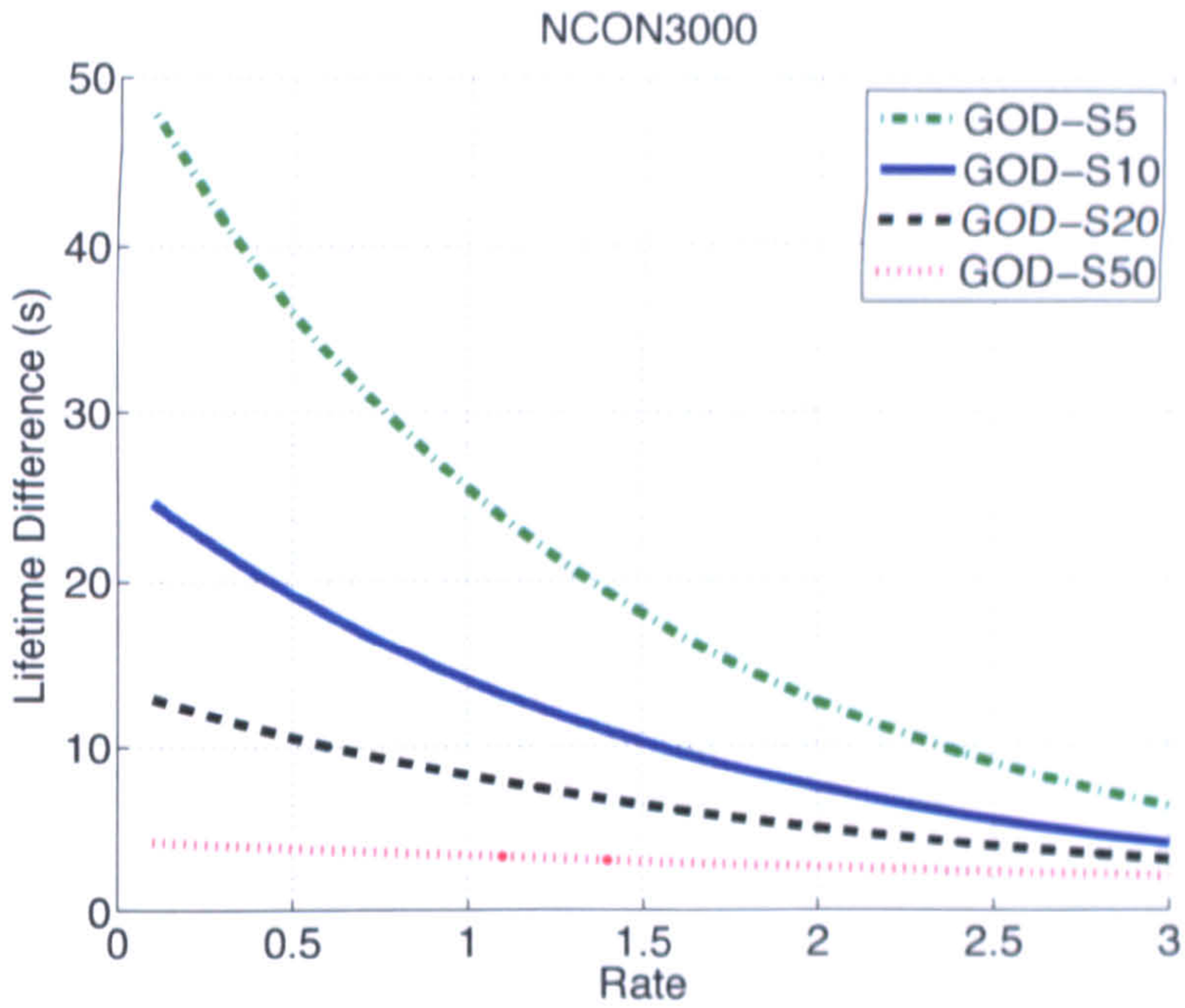
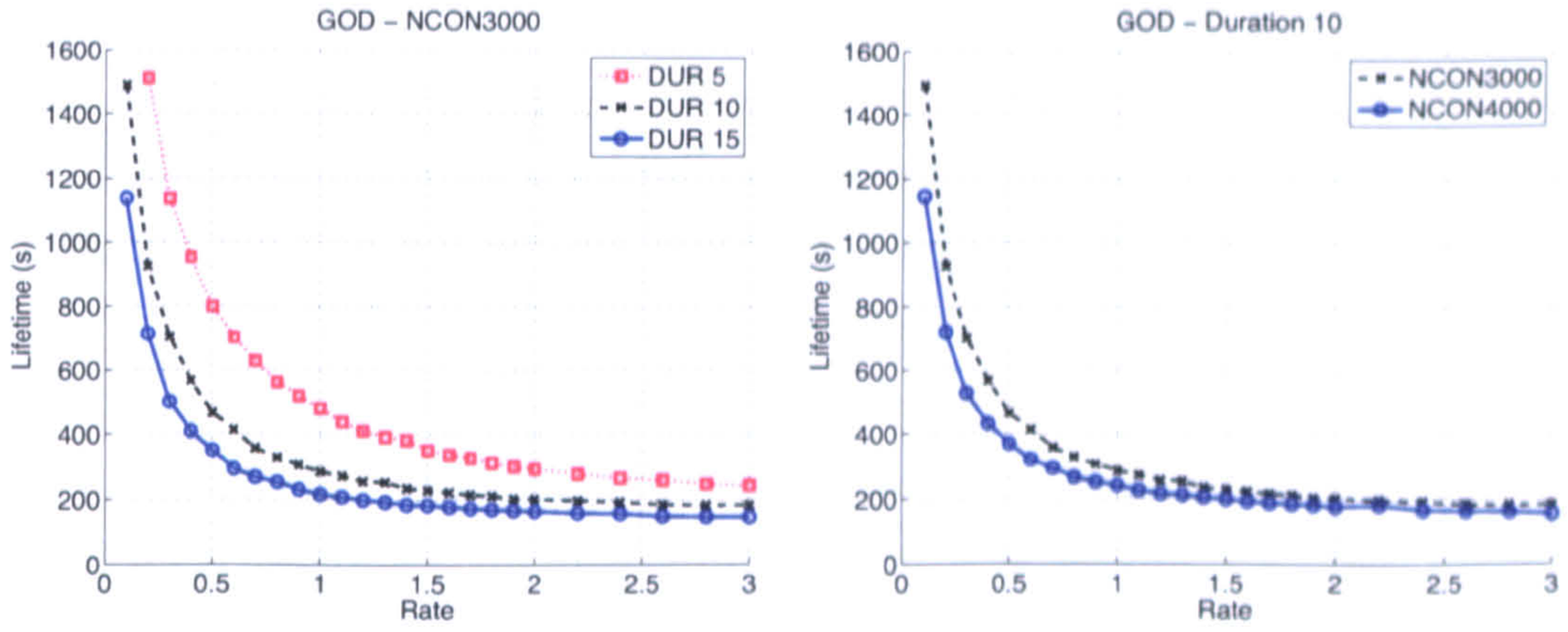


Figure 6.12: Lifetime difference between GOD for steps 5, 10, 20, 50 and GOD with full precision.

When the established connection has high rate, heavy energy depletion occurs on the source and destination, independently of how optimal the route selection is, which results in lower lifetime difference between the various precision schemes. Examining further trace-files, it was verified that when the first node dies, the dead node is also the source of connection.

The same set of experiments were repeated for: 1) maximum connection duration of 5 and 10 seconds, 2) for 4,000 connections with maximum connection duration of 10 seconds. The results of these experiments led to the same conclusions. Observing the network lifetime of GOD with full precision for variable maximum connection duration, as shown in Figure 6.13a, indicates that, when the maximum connection duration increases, the network lifetime decreases as expected. Figure 6.13b shows the lifetime of GOD with full precision for 3,000 and 4,000 number of connections and 10 seconds of maximum connection duration. Again, the same conclusion can be drawn; increasing the network load (number of connections), decreases the network lifetime as expected.



(a) GOD network lifetime for variable maximum duration.

(b) GOD network lifetime for variable number of connections

Figure 6.13: GOD network lifetime for variable network loads; variable maximum connection duration and variable number of connections.

Finally, GOD with full precision is compared against the traditional DSR in Figure 6.14 for 3,000 connections and duration of 10 seconds. These two lines represent the higher and the lower bounds for protocol design. The former uses optimal routing selection, while DSR is not energy aware.

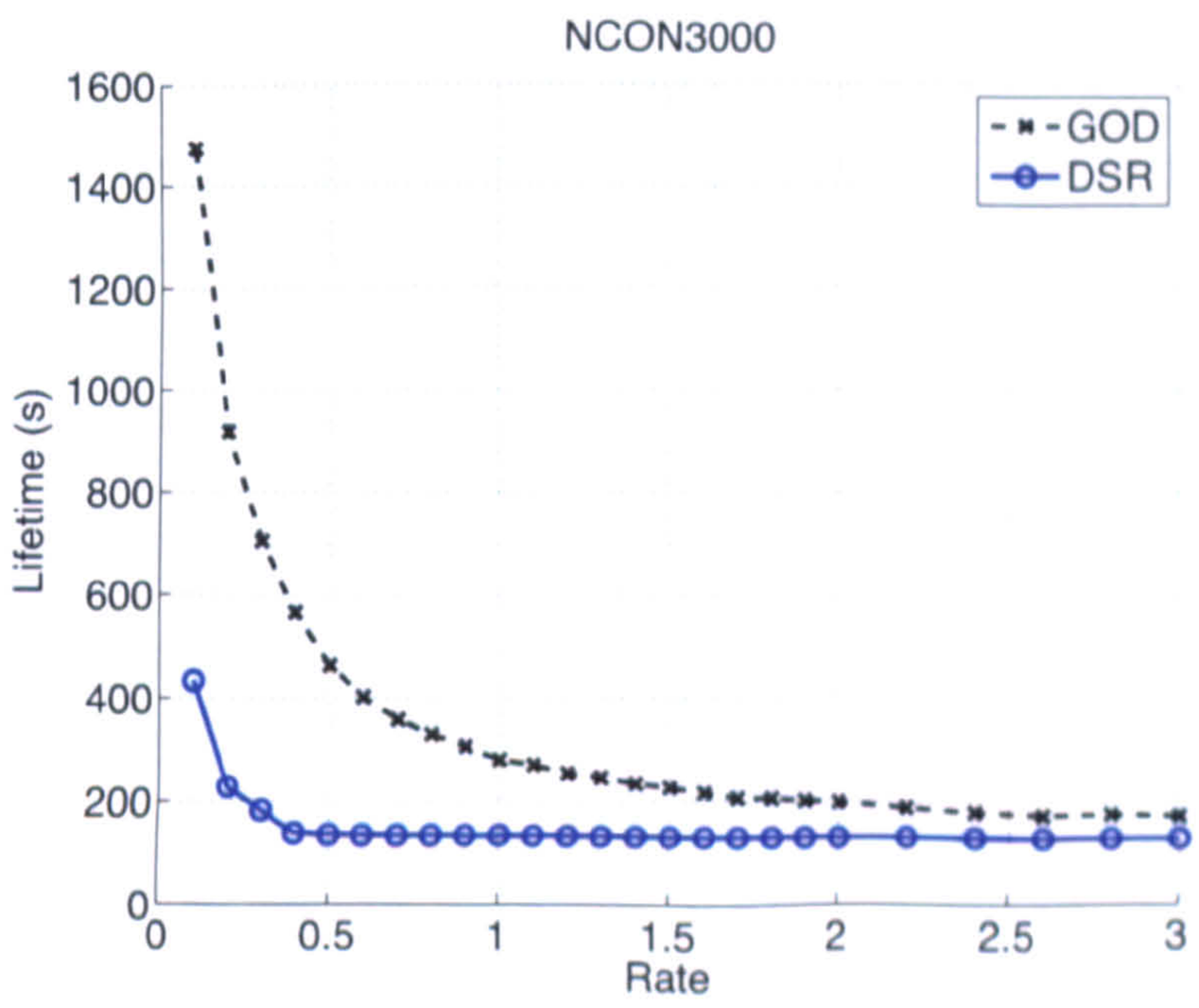


Figure 6.14: Network lifetime comparison between GOD and DSR for 3,000 connections and maximum duration of 10 seconds.

The remainder of this chapter discusses the steps taken to implement a signalling mechanism, to allow each node to collect energy information. This sig-

nalling mechanism is referred as the Information Gathering Protocol. This implementation stage is referred to as the partial knowledge stage because the nodes do not have knowledge of the residual energy of all nodes, but only a subset. In the first stage of the implementation, there is no energy cost associated with the signalling mechanism and any reduction in the network lifetime is simply as a result of the nodes only having energy related knowledge of a subset of the nodes, rather than all nodes within the network. Additionally, the energy levels are known with latency that results from the propagation delay associated when nodes exchanging signalling information. The second and final implementation stage introduces energy depletion for the exchange of the updates and its impact is discussed.

6.2 Information Gathering Protocol

6.2.1 Design Considerations

The initial NS-2 model assumed that all nodes have knowledge of the residual energy of all the other nodes in the network at any given time. Using this knowledge, each node constructs a graph to obtain the best path to route the traffic on per-packet basis. However, in more realistic scenario, the residual energy of all nodes will not be available. For this reason, it is necessary to include additional signalling in the model to collect information about the residual energy of the nodes in the network. Building such intelligence to the network raises the following issues (visualised in Figure 6.15):

1. Does each node in the network need to know the full or a subset of the energy map?
2. Do the source, destination, or intermediate nodes detect outdated information?
3. How will energy map updates be triggered; in a time-ordered or event-based fashion?
4. How is the energy map relayed to each node; using in-band or out-band signalling?
5. What is accuracy of the required energy map?

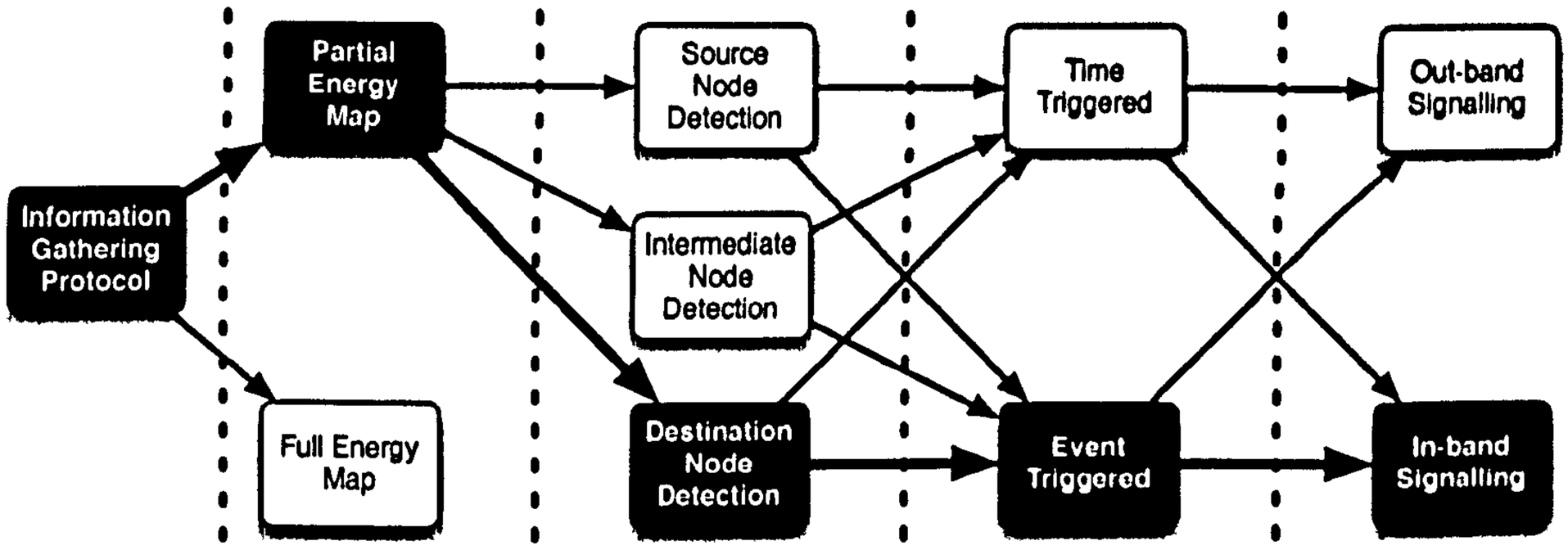


Figure 6.15: Design choices for the information gathering protocol.

Any combination of the design steps of Figure 6.15 represents a viable solution. The following sections discuss the choices made in the implementation of the information gathering protocol.

If each node collects information for the network as a whole (full energy map), then the amount of required signalling is significant and, consequently, defeats the purpose of energy efficiency, as well as increasing network congestion and possessing inherently poor scalability. Finally, the time required to communicate the full energy map to each node will be significant and, consequently, the collected information will quickly become out-of-date. Clearly not a practicable approach; significant effort for little return or degraded performance. The alternative approach is for each node to collect information for a sub-graph of the network. The aim is then to evaluate if the proposed PTD and PTZ schemes perform well with only partial knowledge of the energy map and consider the practical implementation of the proposed schemes.

From PTD and PTZ perspectives, the source node must have knowledge of the energy map in order to select a route for each packet. For example, consider the network in Figure 6.16. The source node S performs a route discovery and discovers paths A, B and C to destination D. In order for PTD and PTZ to select the most appropriate path for sending the traffic, the residual energies of the intermediate nodes $A_1 \rightarrow A_n$, $B_1 \rightarrow B_m$ and $C_1 \rightarrow C_k$, must be known to the source node. This information can be obtained by embedding additional information within the route request and route reply packets that are propagated during the route discovery procedure. The additional embedded information will enable the source node to construct a partial energy map of the network and perform routing decisions based on that map. However, the cache energy map of node S will eventually become outdated, as intermediate nodes might

be members of paths for other network connections (not shown in Figure 6.16) and their actual residual energies will be different from the residual energies recorded by node S. Such divergence between cached and actual residual energies could result in node S making inappropriate or inefficient routing decisions. For this reason, it is necessary to refresh the cache of node S. The issue is who should detect out-of-date (expired) cached information and, subsequently, take the necessary actions to refresh it. The choices can be classified as follows:

- Source node.
- Intermediate node.
- Destination node.

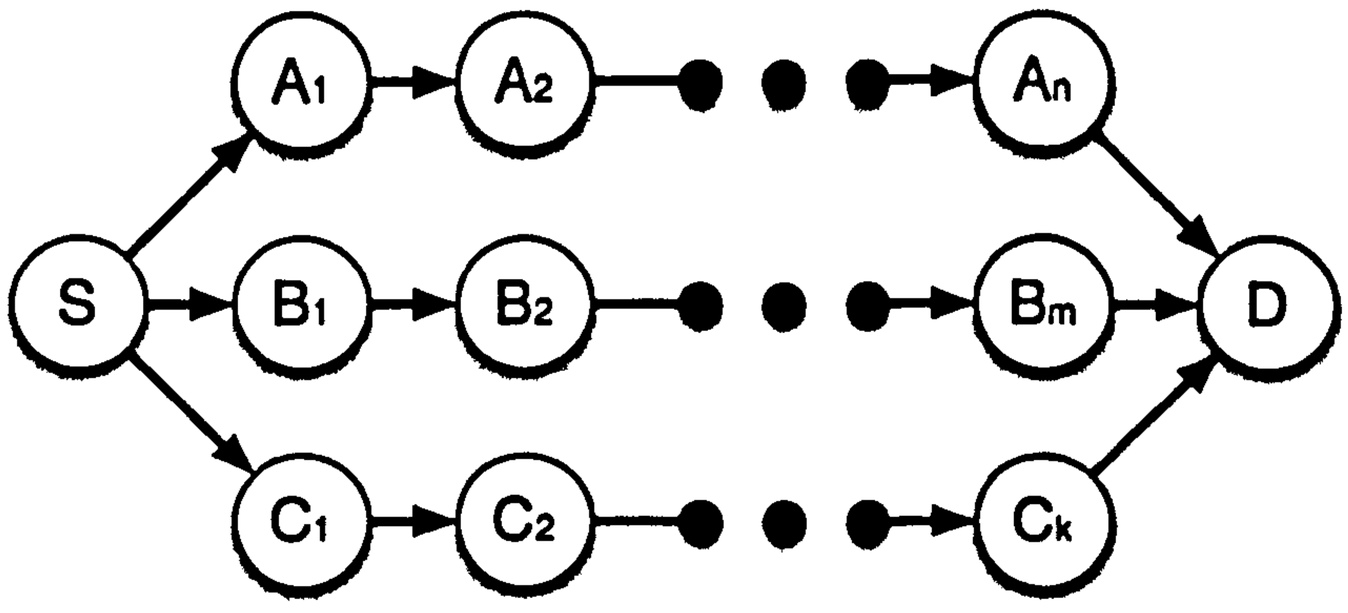


Figure 6.16: An example network region with 3 known paths from source (S) to destination (D).

For the first method, the source node must detect that cached information is out-of-date, but it does not have the means to know when expired information is cached. A solution would be to poll network nodes at regular time intervals for updates. Timer based polling however, raises the question of how frequent the polling should be. Too frequent polling will waste resources, while infrequent polling will result in usage of out-of-date information for longer than is necessary. A heavily loaded network will require frequent updates, while a lightly load network would need less frequent polling. A fixed time-window between polls is not efficient because the traffic load will not be constant throughout the operation of the network and hence this approach was not used.

The second approach would be to make intermediate nodes responsible for detecting expired cached information. Intermediate nodes do not have means to know the accuracy of the information cached by the source node. To overcome this problem, the source node can embed the known residual node energies in the packets. Consequently, intermediate nodes can determine if the information

cached by the source node is out-dated and trigger updates. However, a more important problem arises when this approach is used. Multiple intermediate nodes will detect outdated information and multiple updates will be triggered which will result in update "storms", which, in turn, will result in heavy additional energy expenditure and subsequent congestion. For this reason, such an approach was not considered.

Finally, the third method and the one used underpinning the experiments in this work, would be to make the destination node responsible for detecting out-of-date cached information. Using this method of detection, update storms are avoided, but it is still necessary for the source node to signal to the destination node the status of the cache (known costs).

Using destination node detection, the next design step is to decide when the destination node will trigger/send the updates. As previously mentioned, a time based trigger will be inappropriate, since a fixed frequency of updates will result either in unnecessary resource expenditure or inaccurate cached information for prolonged periods of time. If the time based trigger is used, the destination node should initiate the updates and not the source or intermediate nodes. In the source-based timer trigger, extra overhead will be inserted to the network as a result of the round trip of polling, while in the case of intermediate node based timer trigger, "storms" will be generated.

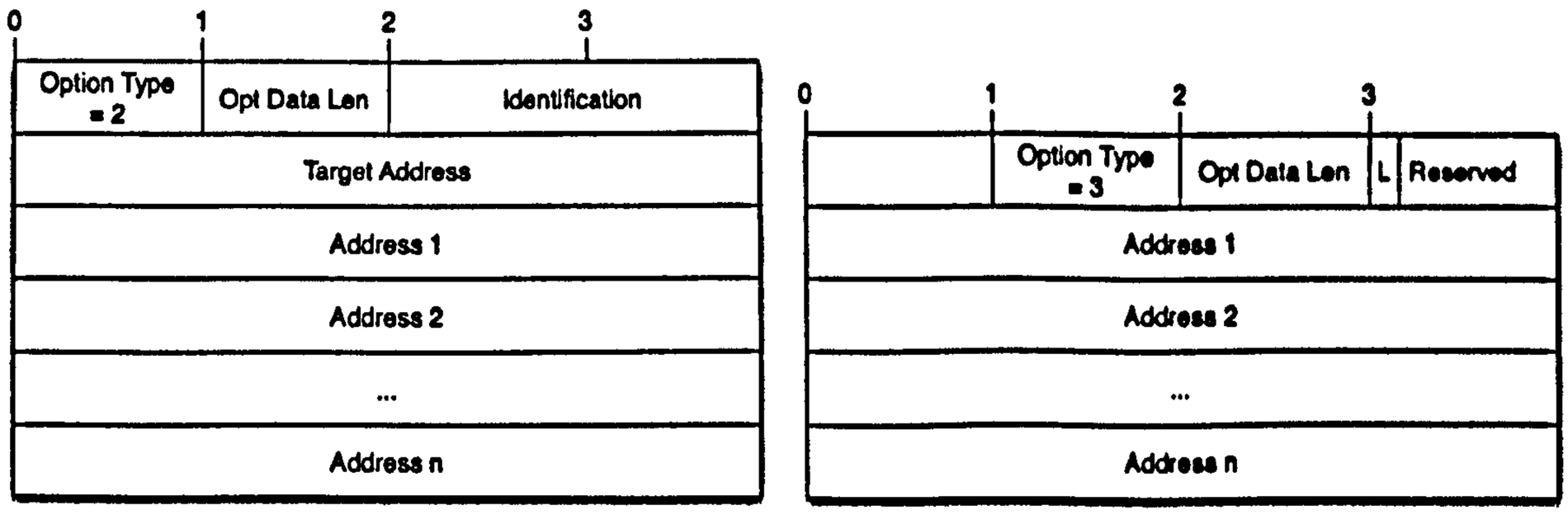
A more viable solution is to initiate updates based on a condition (event). The condition selected for triggering the updates is the divergence of the cached energies from the actual residual energies of the nodes. When the divergence reaches a specified threshold, an update is send to the source with the new residual information. The threshold used in the experiments is equal to unity, and it is essentially controlled by the precision used to report energies. For example, when a precision of 50 steps is used and the maximum initial energy is 50 J, then the step size is equal to 1 Joule and when the difference exceeds the 1 Joule threshold then an update is sent. On the other hand, when a precision of 10 steps is used, with the same maximum initial energy, the step size is 5 J and the unity threshold results, triggers updates when the path cost error exceeds 5 J.

The final design decision is the selection between in-band and out-band signalling. In-band signalling embeds information in existing packets, while out-band signalling uses dedicated packets to perform the communication platform of the protocol. Out-band signalling has the advantage of making the protocol autonomous and independent of the routing protocol implementation. However,

it imposes an additional overhead on the network. Since energy efficiency is the point of interest of this work, out-band signalling is not used, as it imposes additional overhead on the network, which increases the energy consumption in the capacity constraint ad-hoc networks. In-band signalling requires modification to the existing routing protocol to extend its messages to encapsulate the additional information required for the functionality of the information gathering routing protocol. The following sections describe the modifications to the DSR NS-2 code to implement SPR, PTD and PTZ with partial knowledge.

6.2.2 Route Discovery

The purpose of route discovery is to provide knowledge to the instigator about the status of the network such as topology and residual energy information. Initially, the source node floods the network with Route Request (RREQ) packets. The structure of the DSR RREQ packet is shown in Figure 6.17a. Intermediate nodes, upon receiving the packet, check the identification field to determine if they have previously processed this RREQ and, if not, the node appends its node ID to the packet and rebroadcasts it. The same procedure is followed by all intermediate nodes until the RREQ packet reaches the target of the discovery. The target then responds back with a Route Reply (RREP) packet. The structure of the RREP is shown in Figure 6.17b.



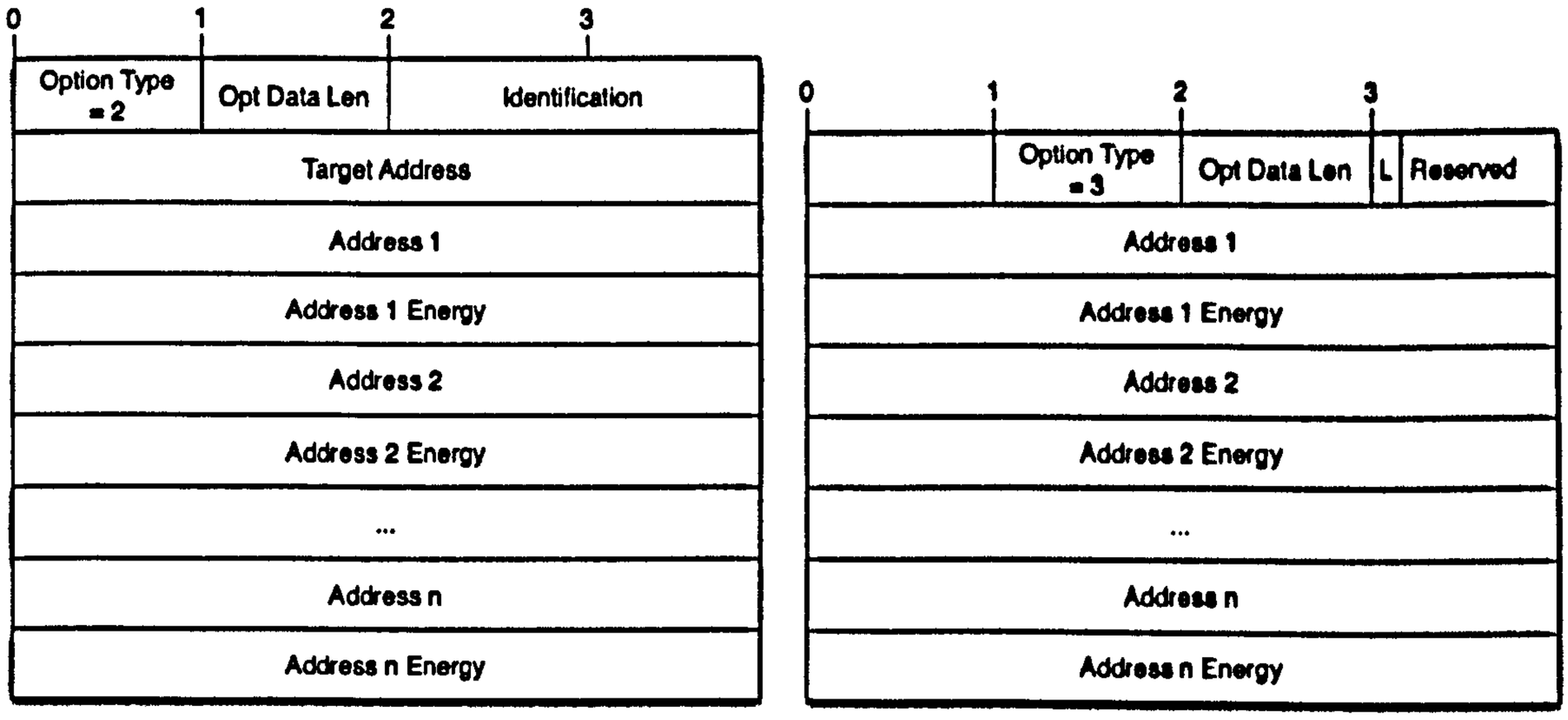
(a) DSR RREQ.

(b) DSR RREP.

Figure 6.17: Route Request and Route Reply packet structures for DSR.

The above procedure is modified as follows. The source node, upon sending the RREQ, appends its node ID together with its residual energy value. Intermediate nodes perform the same processing procedure as DSR with the only difference that nodes append their residual energy level in the RREQ packet in addition to their node ID. When the first RREQ packet reaches the target of

the discovery, the destination node constructs a graph $G_{t_1}(V_{t_1}, E_{t_1}) \subseteq G(V, E)$, where G(V,E) is the complete graph of the network, constructed by GOD as described in Section 6.1. The subscript t indicates that the graph is cached by the node t and the numerical subscript indicates the iteration (version) of the graph cache. The modified structure of the RREQ is shown in Figure 6.18a.



(a) Modified RREQ.

(b) Modified RREP.

Figure 6.18: Modified Route Request and Route Reply packet structures to accommodate the Information Gathering Protocol.

In order to avoid unnecessary delay in the route discovery procedure, the target of the discovery immediately responds to the RREQ with a route reply as per DSR, but uses the modified packet structure shown in Figure 6.18b. When the route reply packet reaches the instigator of the discovery, the instigator updates its own graph cache ($G_{s_1}(V_{s_1}, E_{s_1}) \subseteq G(V, E)$) with the information encapsulated in the route reply packet and starts sending any buffered traffic to destination.

In the meantime, the target of the discovery receives subsequent RREQ packets and then generates route replies to the instigator of the discovery if the route requests are node disjoint (as in Chapter 4). In addition, the destination node exploits the information encapsulated in RREQs and updates its graph cache and, subsequently, sub-graphs $G_{t_m}(V_{t_m}, E_{t_m})$ emerge. When it receives subsequent RREP packets, the instigator updates its graph cache with the new path and residual energies encapsulated in the reply packets. It must be also noted that all intermediate nodes, when they receive packets, extract the topology and residual energy information and update their own graph cache. Consequently,

as time elapses, the nodes gain knowledge of the network state in terms of parameters such as connectivity and energy map.

Using the above described mechanism, the instigator gains knowledge of the network status and is then able to perform route selection decisions. However, the cached information will date and can lead to inappropriate route selections. For this reason, it is necessary to provide a mechanism to refresh the cached information. As described in Section 6.2.1, the destination is responsible for detecting outdated information and updates are triggered when the threshold of allowed error (divergence) is reached. The next section describes how divergence is detected and how updates are triggered.

6.2.3 Outdated Information Detection & Updates

As mentioned in Section 6.2.1, the destination does not posses the means to directly detect outdated information from the source node and for this reason, the source node is required to signal the status of its cache to the destination. One possible method to facilitate such signalling would be for the source node to append to the data packets, the residual energies as known to it, and each intermediate node additionally append the actual residual energy level as shown in Figure 6.19. In such a way the destination node can detect out-dated information by comparing the energies appended by the source node and these appended by intermediate nodes.

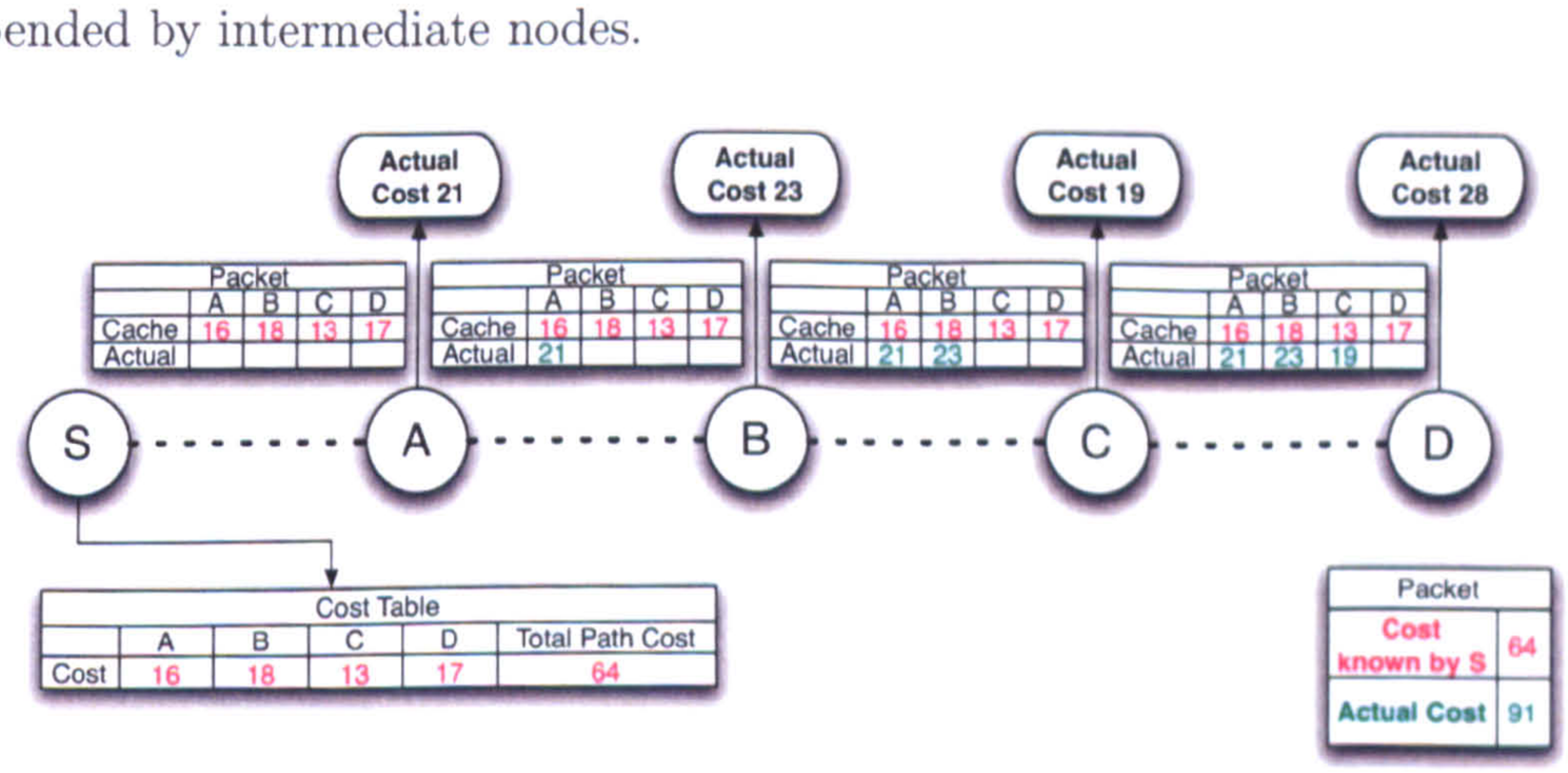


Figure 6.19: Use of cache and actual residual energies to determine out-of-date cached information.

A significant problem with this approach is the extended overhead added to the network. The embodiment of cache and known residual energies in every packet sent by the source node can result in a significant increase in packet

length and energy consumption. If the source node reports the energy with 256 *quantised level precision* (i.e. 1 byte long), then for a path which is $n-1$ hops long, an extra overhead of $\sum_{i=1}^{n-1}(n+i)$ bytes will be transmitted and the energy consumption in the network will increase. To clarify the enumeration of the additional bytes; the first node of the path will transmit n cached residual energies plus its own energy. At the next hop, the node will transmit the same number of additional bytes it received $(n+1)$, plus an additional byte for its own residual energy, resulting in total $n+1+1=n+2$ additional bytes. For the third hop, the additional bytes will be $n+3$ an etc. until one intermediate node before the end of the path transmits $n+n-1$ additional bytes. Adding, all additional bytes results in the sum presented above.

Since energy efficiency is the point of interest in this work, an alternative solution is used which embeds in the packet two values; the first value is the "known path cost" to the destination and the second value which will be increased every time the packet is forwarded (referred as the "actual path cost"). For example, consider the scenario shown in Figure 6.20. The source node calculates the total cost of the path based upon the cached information and appends this value to the packet along with the "actual cost", which is 0. Node A upon reception of the packet increases the "actual cost" to reach it (21) and then forwards the packet. The same procedure is followed by all intermediate nodes (B and C), and finally the packet reaches the destination node D. The destination node adds its cost to the "actual cost" (in the example this is 64) and compares this value with the "known path cost" (91) found encapsulated in the packet. Since the two costs are different, the destination node knows that the information cached by the source node is out-of-date and it can take the appropriate sequence of actions to send an update to the source.

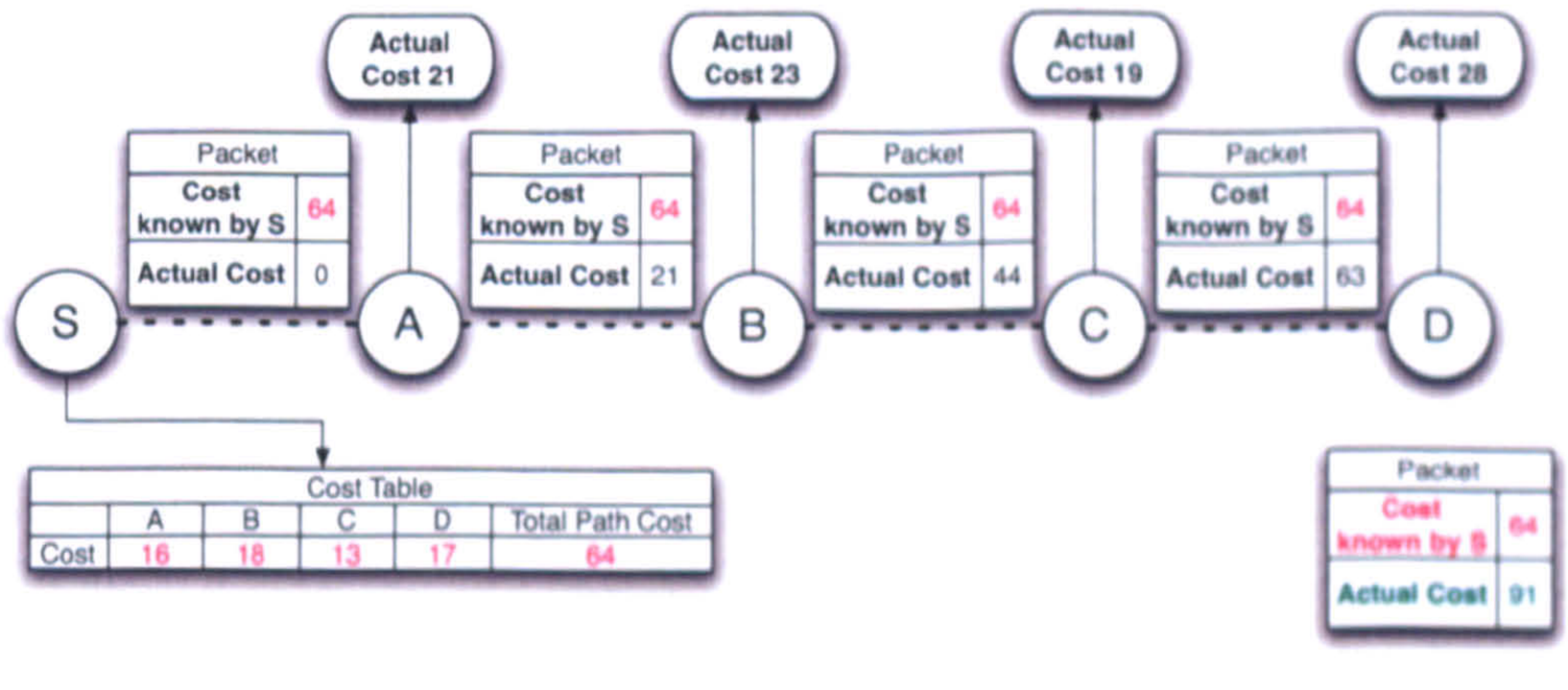


Figure 6.20: Use of sums to determine out-of-date cached information.

The known path cost and the actual path cost are reported with 2 bytes precision each and for a $n-1$ hop path, the additional overhead is $\sum_{i=1}^{n-1} 4 = 4 \cdot (n-1)$ bytes.

The third and final approach to notify the destination that the cached information is outdated, is to append in the packet the known cost to the destination; each intermediate node, before forwarding the packet to the next hop, will subtract from the known cost its own energy level and the resulting value will replace the known cost value in the packet. When the packet reaches the destination the value contained in the packet corresponds to the difference between the actual path cost and the cached cost by the source node. If the error is high then the destination node can take the appropriate actions to send an update to the source. Using this approach the overhead is reduced by a factor of two compared to the known and actual cost scheme. The total overhead transmitted is $\sum_{i=1}^{n-1} 2 = 2 \cdot (n-1)$ bytes for a path of $n-1$ hops. For the remainder of this thesis, it is this approach that will be applied and it will be referred to as the mechanism of appending cost error.

Figure 6.21 shows the overhead expressed as a function of the number of nodes for the three signalling approaches.

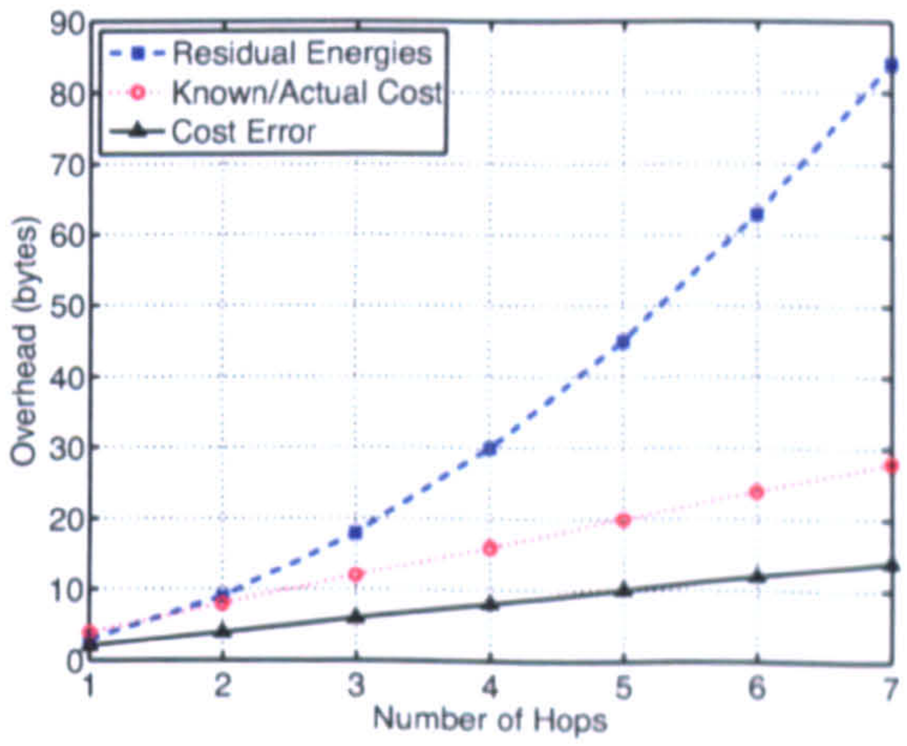


Figure 6.21: Overhead for the three signalling schemes versus number of hops.

From the graph of Figure 6.21, it can be seen that the approach of appending cost error and the known/actual costs scales linearly with the number of hops, while the method of appending the known residual energies does not. The average path length for various grid dimensions is shown in Figure 6.22, assuming that the network is uniformly loaded. From these two graphs (Figure 6.21 and 6.22), the average additional overhead can be estimated for all signalling schemes.

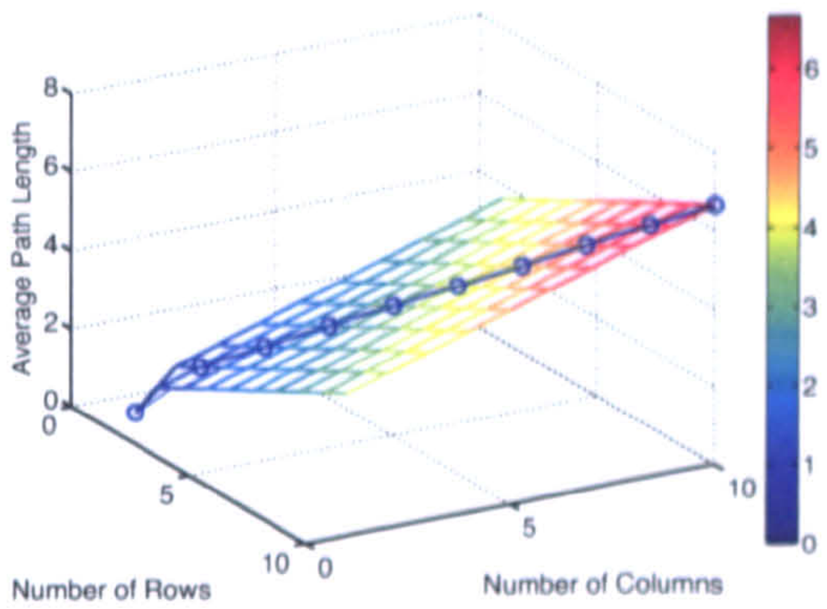
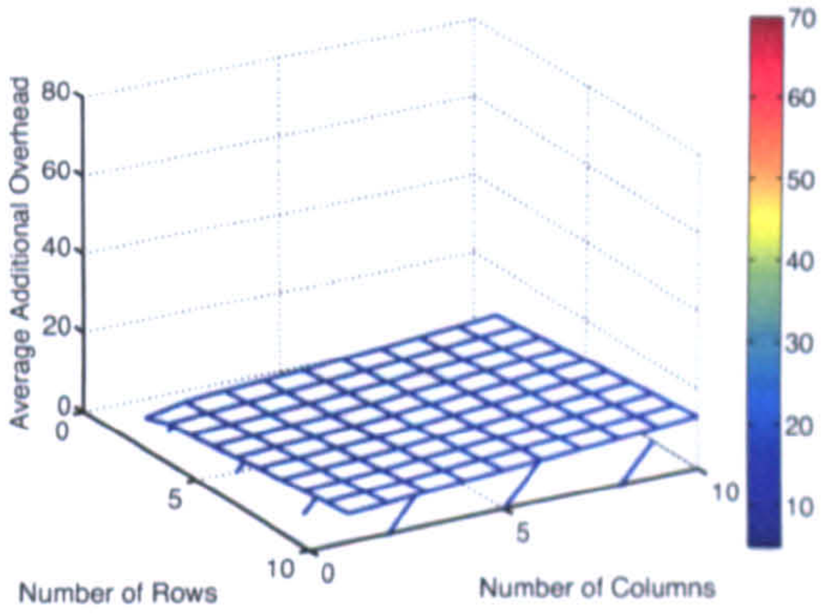
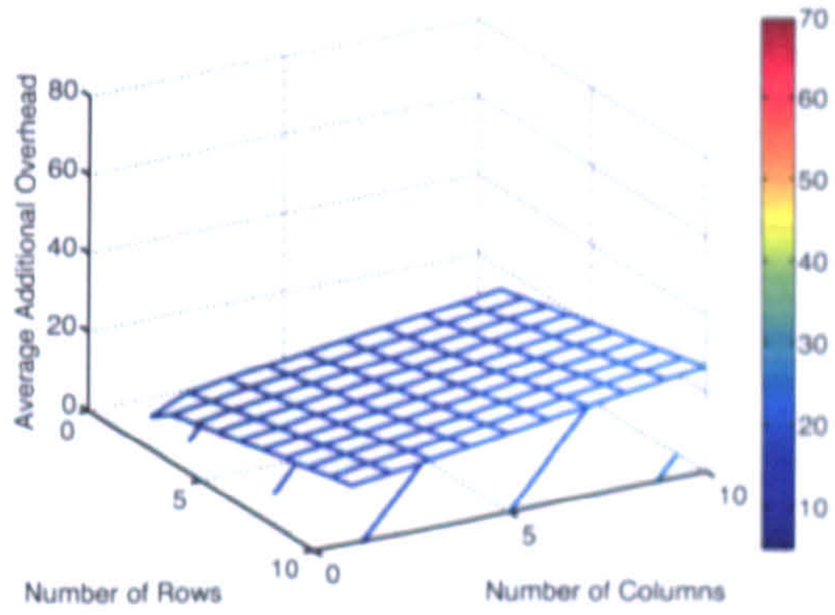


Figure 6.22: Average Path Length for varying grid dimensions.

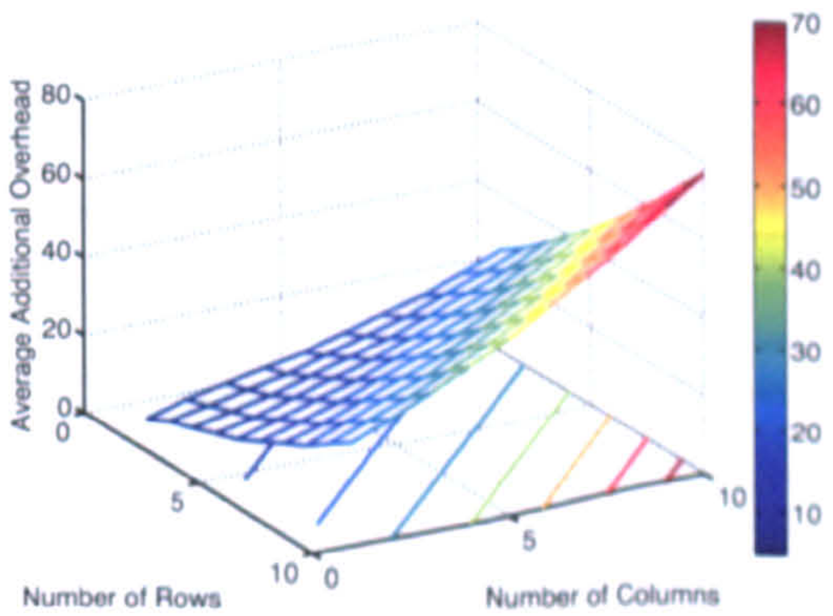
Figure 6.23b, 6.23b and 6.23c show the average additional overhead for three methods. For example, with a 10×10 grid, the method of cost error results in average packet increase of 13.33 bytes, the method of appending known/actual costs is 26.66 bytes, while by appending the residual energies the cost becomes 77 bytes. Consequently, it is clear that the preferable method of signalling is appending the cost error.



(a) Appending Cost Error.



(b) Appending Residual Energies.



(c) PTD.

Figure 6.23: Additional injected overhead for various grid dimensions.

6.3 Network Lifetime Performance using partial knowledge.

Section 6.1.4 presented the results of SPR using global knowledge for various precession levels. In this section, SPR operates using partial knowledge. Each node, obtains residual energy information for only a subset of nodes in the network using the information gathering protocol described in Section 6.2.1; partial knowledge stage.

Three sets of experiments are conducted. In the first set, the information gathering protocol is used to collect and maintain energy related information, but the nodes do not deplete any additional energy due to the additional bytes introduced by the signalling mechanism. This scheme will be referred as "SPR - NOCOST". The purpose of these experiments is to understand how partial knowledge affects the routing protocol's performance compared to global knowledge, rather than evaluating the effect of signalling in the network lifetime. In the second set of experiments, the assumption that signalling does not consume energy is removed and this is the first practical implementation of the routing protocol. It is shown that the lifetime performance is superior than that of DSR, and lower than Global Operations Director (GOD) and SPR-NOCOST. The third and final set, compares SPR with the two novel practical implementation of PTD and PTZ.

All three sets of experiments are conducted on a 10×10 grid for 3,000 connection with 10 seconds of maximum duration. The simulation time is set to 2,000 seconds and initial node energy is set to 50 J.

6.3.1 GOD versus SPR-NOCOST

Figure 6.24 shows the network lifetime versus rate, for SPR-NOCOST with various number of quantisation steps; 5, 10, 20 and 50 quantisation steps. Additionally, the graph shows the higher and lower design bounds, i.e. GOD with full precision and DSR respectively. The lifetime performance compared to DSR is improved to 700 seconds maximum under low loads, while for high loads the performance appears to be almost identical. SPR-NOCOST achieves 400-500 seconds lower lifetime compared to GOD under low rates and as the rate increases, the gap decreases. It is also notable that the lifetime difference of the variable precision is more notable for partial knowledge compared to the difference observed in Section 6.1.4 with GOD.

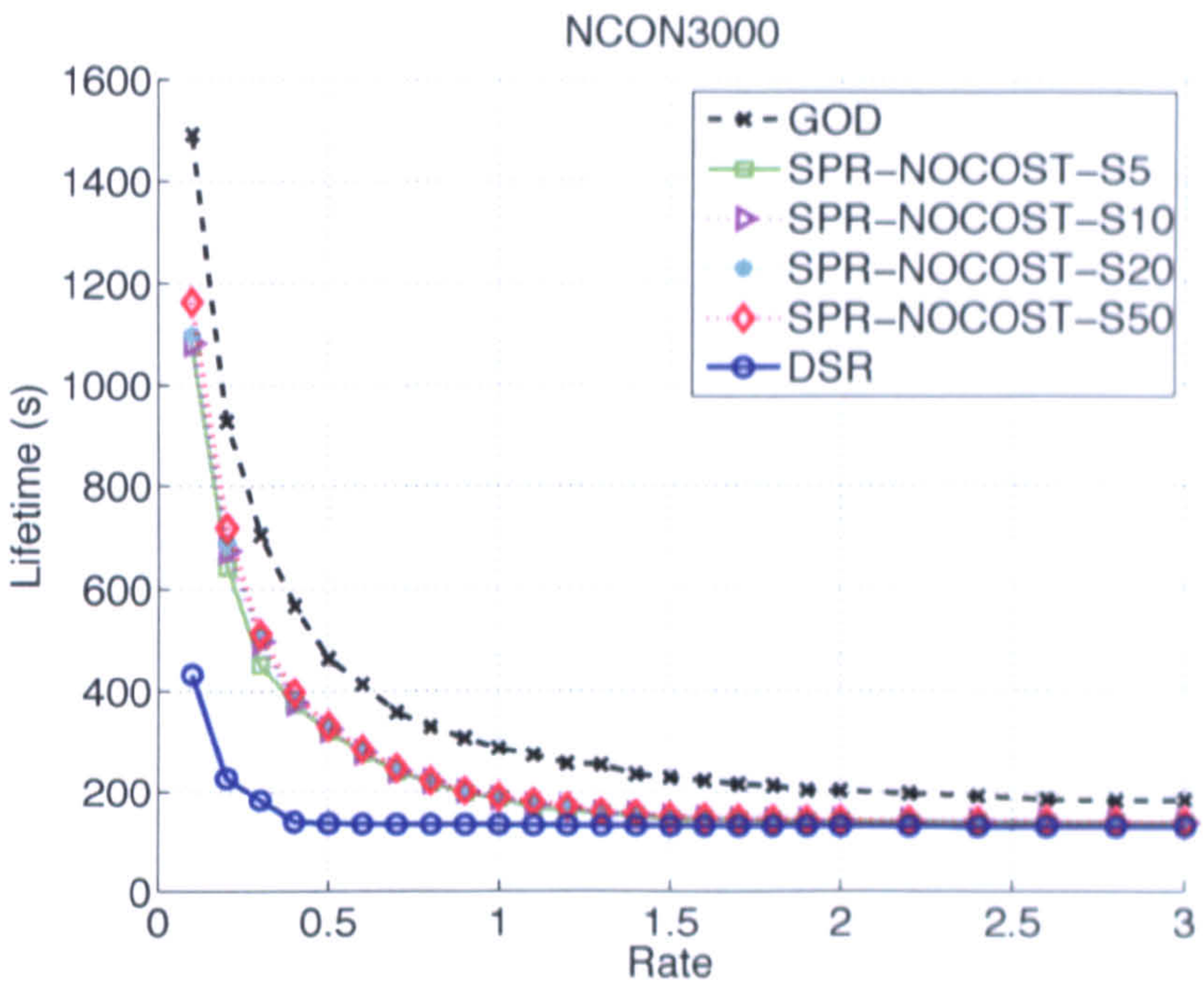


Figure 6.24: SPR-NOCOST lifetimes for variable number of steps 5, 10, 20, 50.

Again, the scale does not give clear view of the difference between the routing schemes and for this reason the absolute difference between the schemes and SPR-NOCOST-S50 is shown in Figure 6.25. SPR-NOCOST-S50 was selected as the reference scheme instead of GOD, because GOD achieves significant higher lifetime and the scale of the graph would obscure the difference between the precision levels. In addition, the graph shows where DSR is placed compared to the SPR-NOCOST. At low rates, the maximum difference between precision levels is clear and visible ranging between 40 and 100 seconds. 100 seconds is the difference between SPR-NOCOST-S50 and SPR-NOCOST-S5, while the difference of 40 seconds is between SPR-NOCOST-S50 and SPR-NOCOST-S20.

As the rate increases, the improvement drops and eventually becomes negligible. The reason why different precision levels achieve identical lifetimes is explained in Section 6.1.4. The source nodes are exhausted at the same time; the source node performs the same number of send and receive operations due to its positioning in the path (edge node), independently of the precision level. However, DSR does not balance the traffic at all; resulting in more immediate effect of the source node exhaustion, while the partial knowledge schemes mitigate this effect.

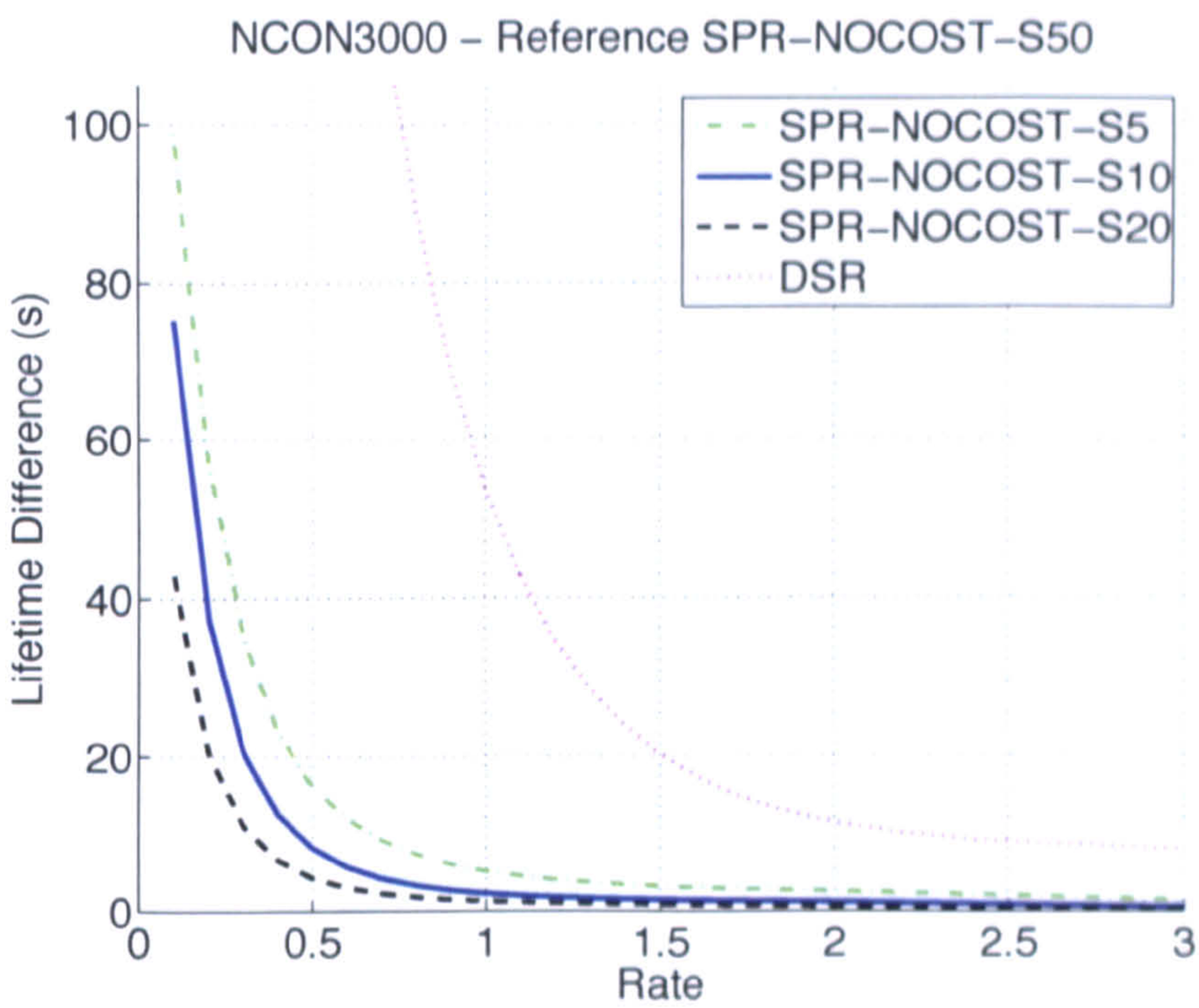


Figure 6.25: SPR-NOCOST lifetimes comparison using SPR-NOCOST-S50 as the reference scheme.

6.3.2 SPR-NOCOST versus Practical SPR

In this stage of the experimentation, the restriction that signalling does not deplete any energy resources is removed. This practical protocol implementation operates using partial knowledge and maintains energy related information using the information gathering routing protocol of Section 6.2.1. The network lifetime of SPR for various precision levels are shown in Figure 6.26. In low loads the lifetime is extended by almost 450 seconds maximum compared to DSR, while the lifetime is more than 600 seconds lower than that achieved by GOD. Compared to SPR-NOCOST the maximum difference is over 200 seconds.

Figure 6.27a shows the absolute difference in lifetimes using SPR-NOCOST-S50 as the reference scheme. From this graph it can be seen that schemes with different precision levels cross and in order to mitigate the difference and make the line more visible the lifetime difference is plotted using SPR-S50 as the reference scheme. It can be clearly seen that SPR algorithm extends the network lifetime throughout the range, with the improvement decreasing as the rate increases. However, comparing SPR for different precision schemes the picture is more complex.

Initially, at rates between 0.1 and 0.4 packets/second, SPR-S20 achieves the highest lifetime, followed by SPR-S50, SPR-S10 and finally SPR-S5. When the rate increases above 0.4 packets/second, the highest lifetime is achieved by

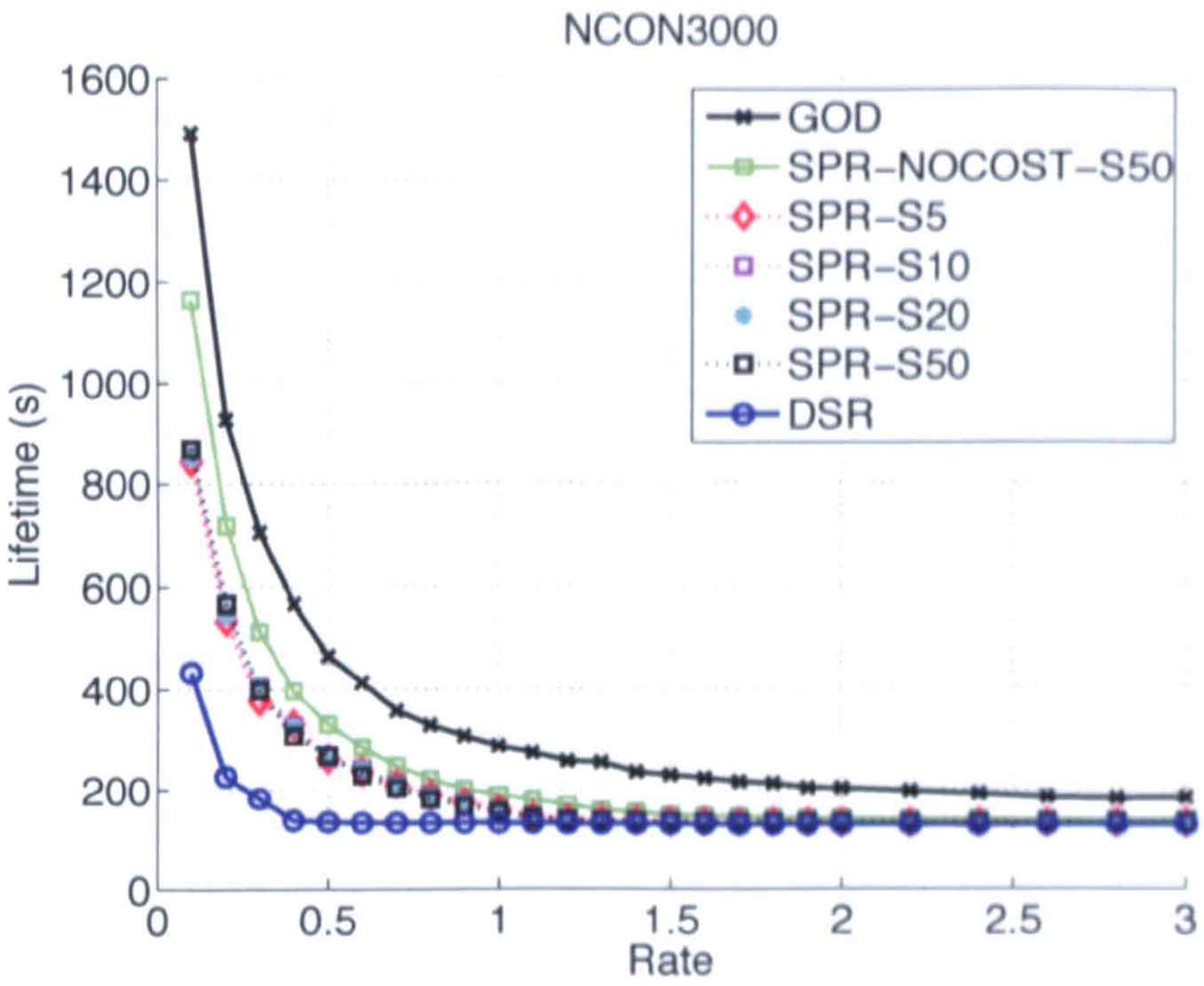
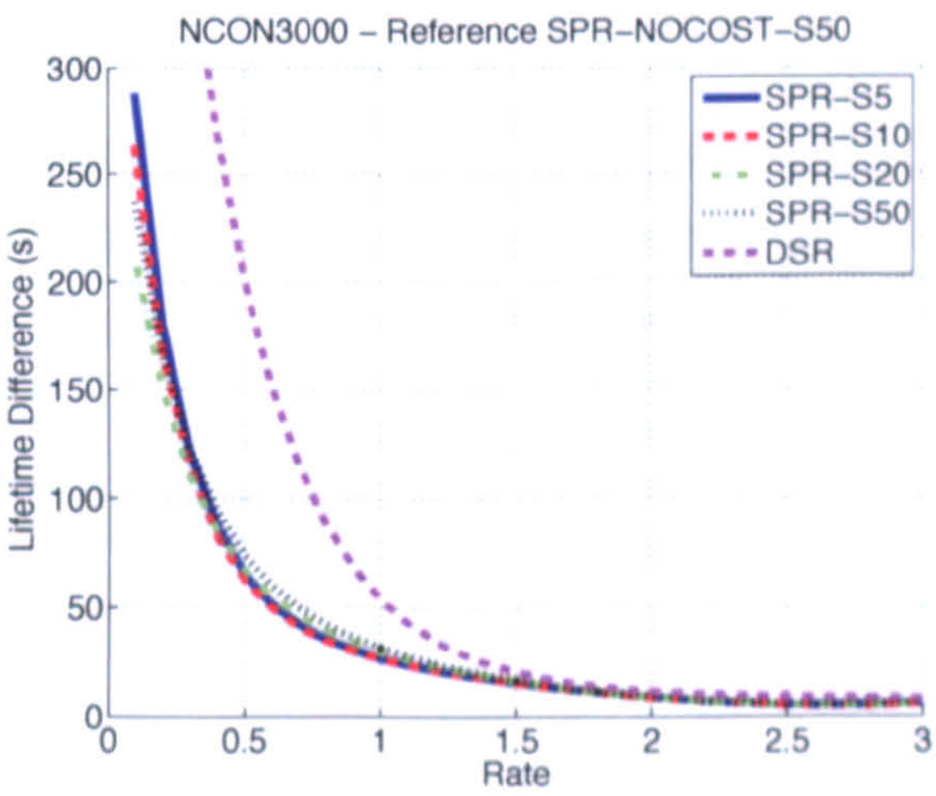
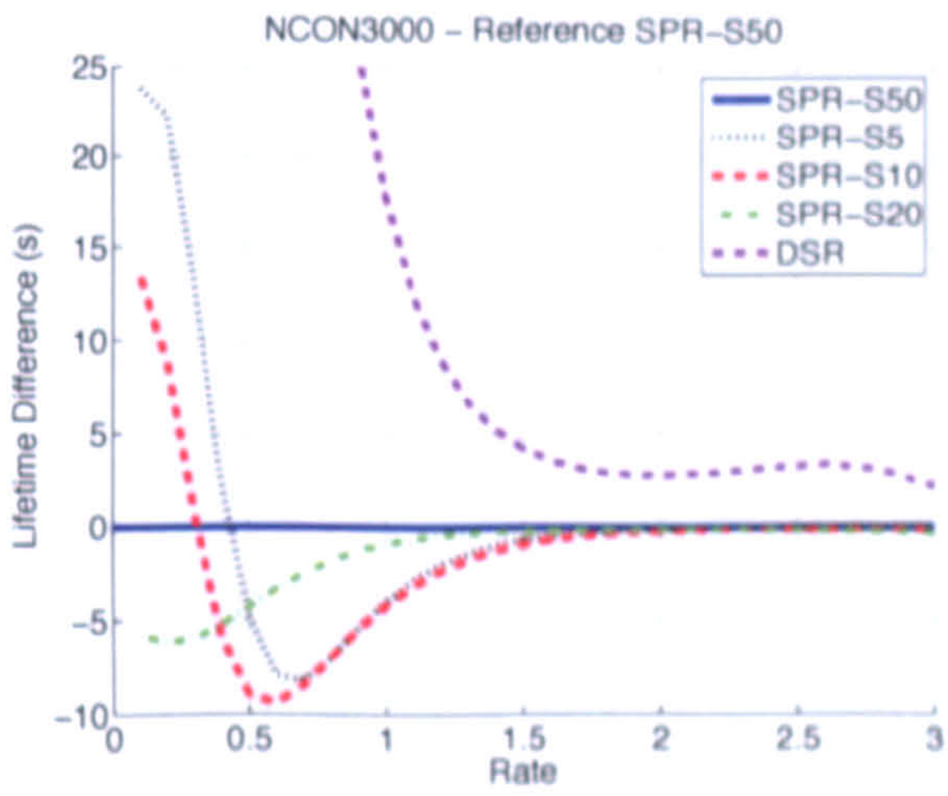


Figure 6.26: SPR lifetimes for variable number of steps 5, 10, 20, 50.



(a) Reference scheme SPR-NOCOST-S50.



(b) Reference scheme SPR-S50.

Figure 6.27: SPR lifetimes comparison.

SPR-S10, followed by SPR-S5, SPR-S20 and finally, SPR-S50.

Schemes with a different number of quantisation steps do not affect the overhead introduced in the data packets, because each data packet contains the same number of bytes for reporting the energy independently of the precision used. However, the unity threshold affects the frequency at which the destination node detects outdated information cached by the source. For example, the scheme SPR-S50 uses 50 quantisation levels and when the energy drops by 1 level, this corresponds to 1 Joule of actual energy depleted by any node along the path. 1 Joule of energy is consumed relatively fast, resulting in frequent updates

being triggered. As a result of these frequent updates, the energy consumption is increased and any additional merits of having precise information made available to the source node are countered.

Therefore, the result obtained for the various quantisation levels is expected, and the network designer has to decide on the appropriate number of quantisation levels to use when employing SPR to gain the maximum benefit. For example, in a wireless sensor network scenario, where the connection rate is 1 report per 2.5-10 seconds, the number of quantisation levels should be selected equal to 20, while networks which require connections of higher rates, the quantisation level of 10 could be used. One approach to accommodate all rates would be to have variable/adaptive number of quantisation levels. However, this approach may be impractical because nodes must keep track of the precision each path uses and the complexity increases significantly. An alternative solution would be to use a combination of constant thresholds and adaptive timers to trigger updates. The thresholds will detect outdated information, while timers will eliminate too frequent updates. The time window can be adaptive and change as a function of the connection rate. However, such schemes are outside the scope of this study, as these offer additional functionality of the information gathering routing protocol and the aim here is to simply illustrate that the proposed route selection schemes can extend the network lifetime.

6.3.3 Practical PTD

In this section, the lifetime achieved by the practical implementation of PTD for various quantisation steps is compared against each other and against SPR. Figure 6.28 shows the lifetime achieved by PTD.

The lifetime difference is shown in Figure 6.29 using PTD-S50 as the reference scheme. The picture is similar to that observed with SPR. At low loads, PTD-S10 achieves the highest lifetime and when the load exceeds 0.4 packets/second, the highest lifetime is achieved by PTD-S5. On the other hand, SPR requires 20 steps for low rates and 10 steps at loads higher than 0.4 packets/second. This indicates that PTD operates better at lower levels of precision; due to its ability to homogenise the energy map.

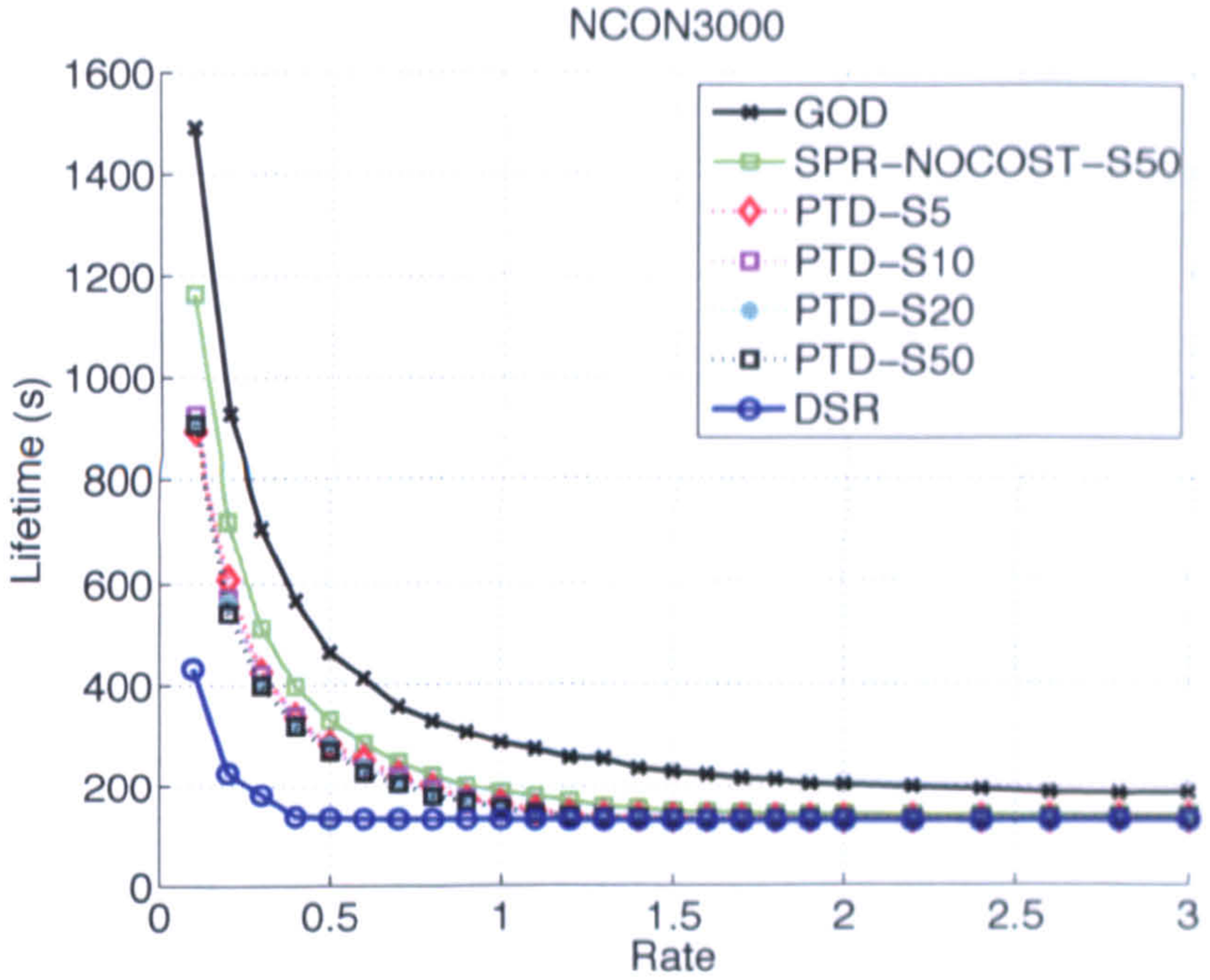


Figure 6.28: PTD lifetimes for variable number of steps 5, 10, 20, 50.

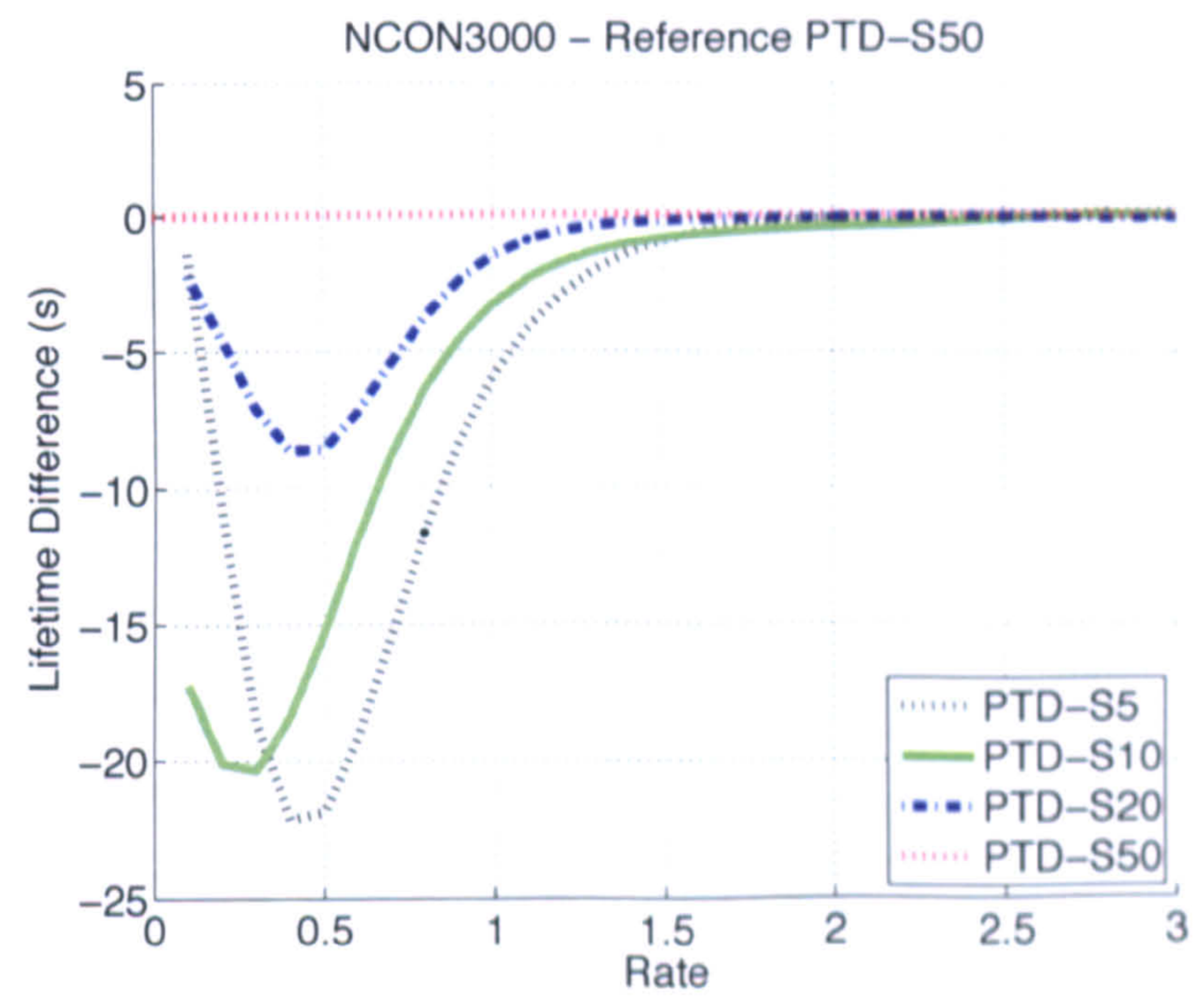


Figure 6.29: PTD lifetimes difference for various precision levels; Reference scheme PTD-S50.

A direct comparison between PTD and SPR is presented in Figure 6.30, where the difference of PTD minus SPR for various precision levels is shown. The maximum difference between PTD and SPR is for 10 quantisation levels and exceeds 50 seconds. At high rates the schemes converge, for the same reasons explained in Section 6.1.4 and 6.3.1.

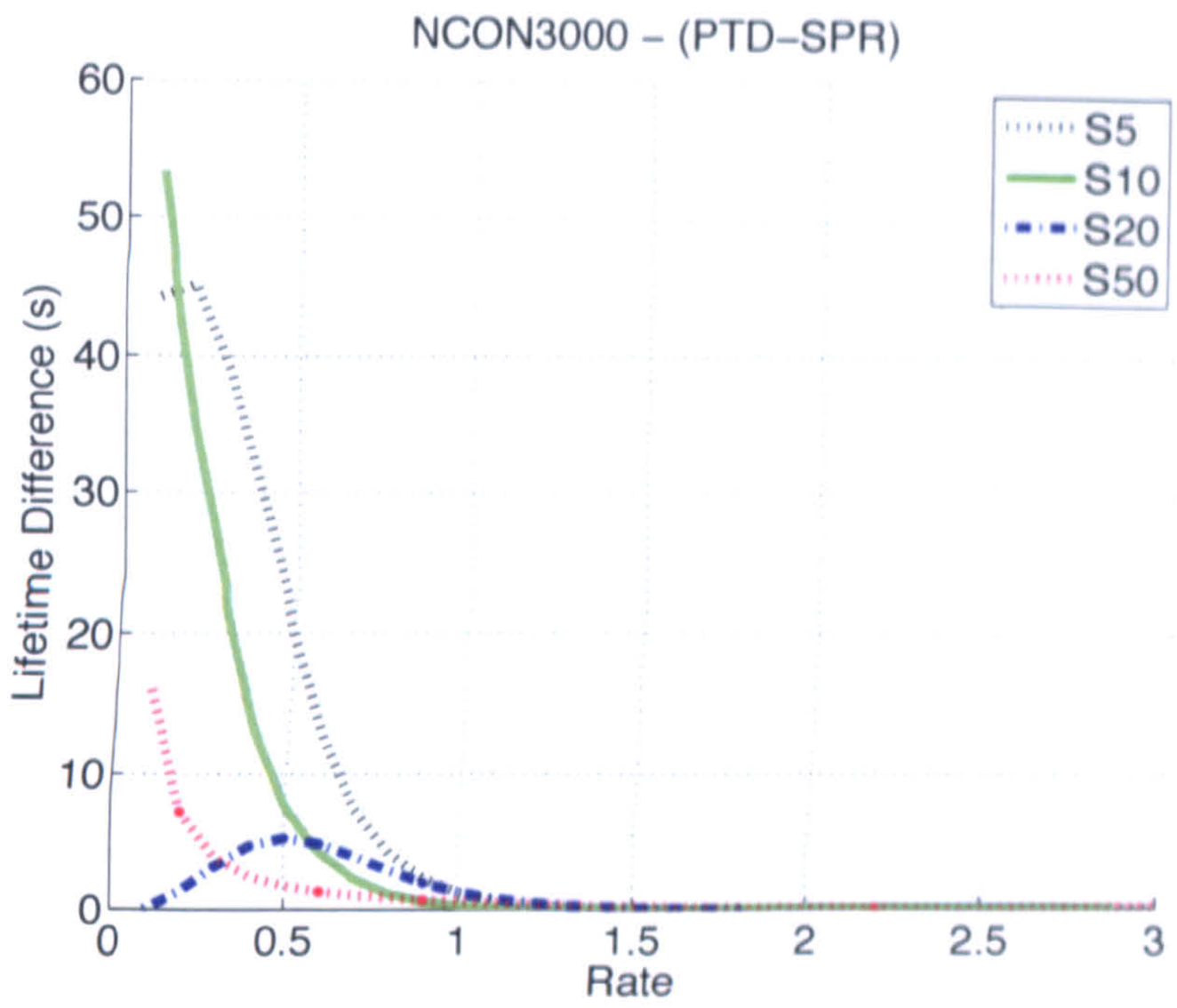


Figure 6.30: Lifetime Difference for PTD and SPR (PTD-SPR).

6.3.4 PTD versus PTZ

A final set of simulation were performed for the second proposed routing scheme, PTZ. The achieved lifetimes for variable numbers of precision steps is shown in Figure 6.31. As expected the lifetime achieved by PTZ falls between the lifetime of DSR and SPR-NOCOST.

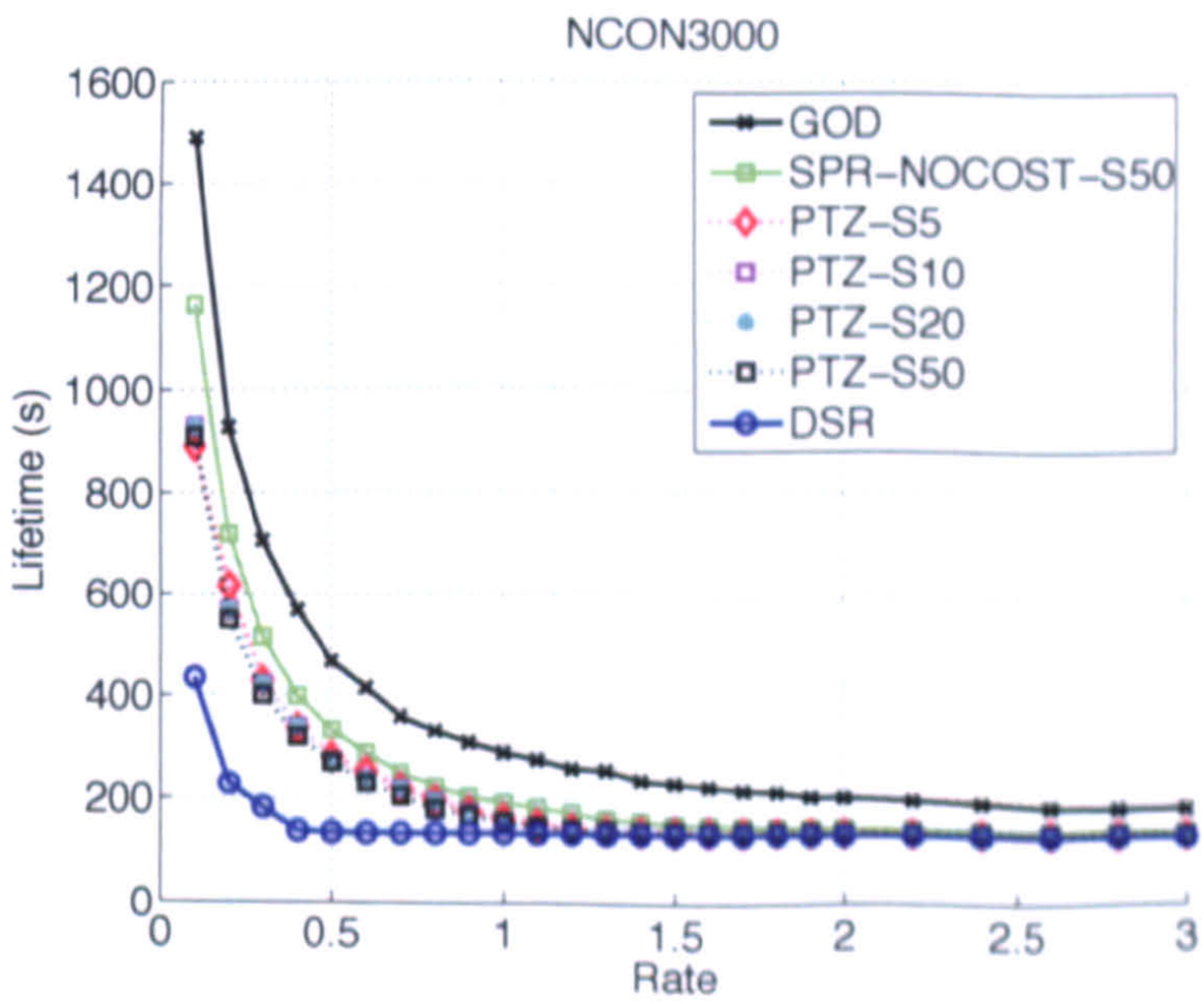


Figure 6.31: PTZ lifetimes for variable number of steps 5, 10, 20, 50.

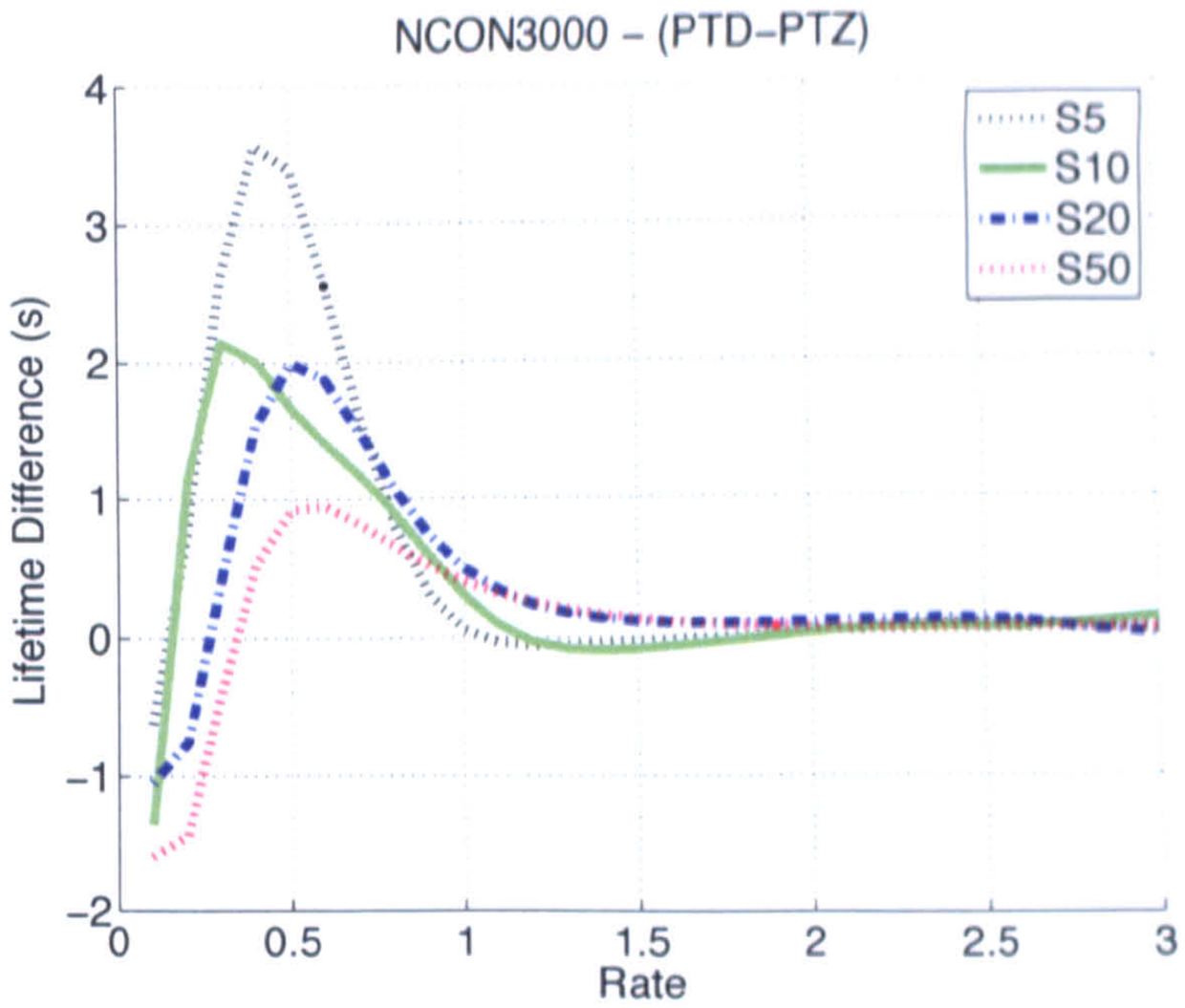


Figure 6.32: Lifetime Difference for PTD and PTZ (PTD-PTZ).

Figure 6.32 shows a direct comparison between PTD and PTZ and it can be seen that there is not a significant difference between these two algorithms, a conclusion which agrees with previous comparison of the two schemes in Chapter 5. The maximum difference of 3.57 seconds is observed to occur at rate of 0.4 packets/second for 5 quantisation steps. Figure 6.33 shows the difference between SPR and PTZ and here it is clear that the proposed scheme prolongs the network lifetime, with the maximum difference observed for 5 quantisation steps, followed by 10 quantisation steps.

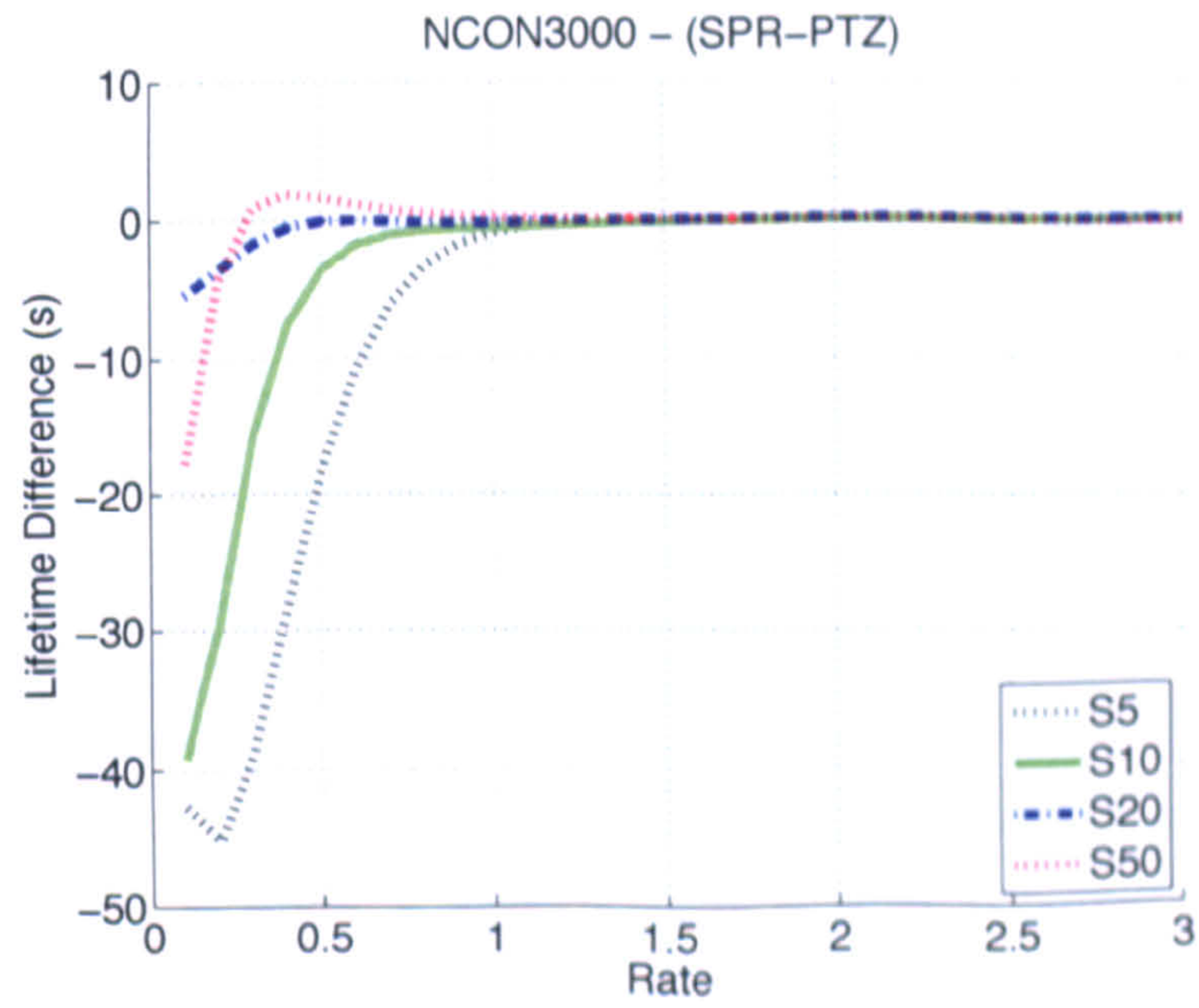


Figure 6.33: Lifetime Difference for SPR and PTZ (SPR-PTZ).

6.4 Summary

In this chapter, the two novel algorithms energy aware routing protocols, Point-To-Diagonal (PTD) and Point-To-Zero (PTZ) were ported to NS-2 simulator and it was shown that these prolong the network lifetime compared to DSR and Strongest Path Routing discipline.

The first stage of modelling assumed that all nodes know the residual energy of all other nodes with full precision. Initial comparison indicated that the benefits of PTD are marginal compared to SPR, when the residual energies are known to all nodes in full precision and without latency. This apparently contradicts the results obtained in Chapter 5 where PTD offered clear benefit compared to its counterpart. The reason behind this contradiction is that each path in SPR has unique numerical cost and PTD essentially becomes a shortest path scheme selecting paths based on hop count. However, in a practical protocol implementation, it can not be assumed that such global knowledge is available and it is costly to obtain and maintain; in which case PTD offers longer lifetime.

The partial knowledge for the practical protocol implementation uses a variety of techniques in an attempt to increase the energy efficiency. Residual energy information is obtained by appending quantised energy levels in the route request and route reply packets. Any additional overhead and energy drainage introduced by the route discovery process is balanced by the intelligent path selection, while the cached information is maintained via a destination detection scheme. The destination based detection scheme is used to detect outdated cached information and send updates to refresh cached information.

Finally, the effect of quantised energy levels is studied and it is shown that, when high precision is available and the knowledge is obtained without any additional energy cost, the network lifetime can increase. However, when the signalling messages consume energy, the merits of high precision are countered by the additional energy consumption to propagate these messages. As expected, there is a trade-off between knowing up-to-date information for performing intelligent routing decision and the cost paid to collect and maintain such information. To conclude, PTD and PTZ offer prolonged lifetime through energy homogenisation compared to DSR and SPR, when all the assumptions are removed and all protocols operate under realistic conditions.

Chapter 7

Conclusion

This chapter highlights the main contributions of the thesis and discusses directions for possible extensions to the research explored. Section 7.1 summarises the thesis, Section 7.2 presents the contributions and finally Section 7.3 concludes with directions for future research.

7.1 Thesis Summary

The thesis presents an in depth investigation into a range of issues pertaining to energy management and multipath operations within wireless ad-hoc networks. Each of the chapters address individual aspects and their contents are summarised below:

- Chapter 3 introduced fundamental concepts related to system performance evaluation and modelling. It is important to note that more than one technique should be used to ensure validity of results and any conclusion drawn. The chapter discussed in detail the techniques and methodology used to carry out the performance evaluation of the thesis such as transient removal, stability, traffic and mobility generation. Finally, a multipath routing scheme was proposed, which distributes the traffic load to multiple paths based on hop count. A simple topology was used to validate the simulation model against analytical calculations.

- In Chapter 4, extensive performance evaluation was performed for the proposed multipath routing protocol, Multipath Dynamic Source Routing with Node Disjoint Routes (MDSR-NDR), and its merits compared to its unipath counterpart were quantified. It was shown that MDSR-NDR performs better under static scenarios with significant improvements in

terms of goodput and delay, especially at low and intermediate loads. Two sets of mobile experiments were conducted; one varying the percentage of mobile nodes and one varying maximum nodal speed. In both cases, MDSR-NDR offered significantly higher goodput and lower delay, while a major routing efficiency improvement was observed throughout the rate axis. This is an interesting result as it contradicts the opinions expressed previously [59, 91] and does show that a multipath approach has significant additional merit.

The proposed scheme is unique compared to other schemes in the literature, because it does not modify the route propagation procedure and once paths are discovered, the routing scheme uses them in a per-packet allocation manner, compared to other schemes where they use the alternative paths solely to increase reliability, i.e. only when the primary path fails.

- Chapter 5 examined multipath routing from the energy and network lifetime perspective. Two new routing algorithms were proposed which maximise the lifetimes of wireless energy-constrained ad-hoc networks. The approaches are based on the homogenisation of energy consumption in the network as opposed to more traditional approaches where the paths are chosen based upon highest energy. The proposed schemes were compared with existing energy-aware protocols and it was shown that the two new algorithms enable full network connectivity for longer and mitigate against the effects of inevitable node outages caused by energy exhaustion and offer an improved likelihood of maintaining connectivity when nodes are lost.

- Further study of the two novel routing schemes was performed in Chapter 6. A new simulation-based model was built, which removes assumptions made in the previous modelling stage. Initially, the model assumed global knowledge which was obtained through the Global Operations Director (GOD) object, but the following assumptions were removed:

  - All candidate paths no longer reside within the rectangle formed by the source and destination pair and a graph search algorithm is used to form the candidate path set.
  - Nodal energy is no longer decreased by one unit every time a packet is relayed through a node. Energy decrements belong to the real number set $\mathbb{R}$, and depend upon the packet size. Additionally, energies of adjacent nodes are reduced for overhearing transmissions.

– Packets are no longer relayed instantly from source to destination.

Under these conditions, Point-To-Diagonal (PTD) does not always offer improvement over Strongest Path Routing (SPR); due to the unique cost of each candidate path for the reasons explained in Section 6.1.3.

The next section of the chapter studied how varying the precision of the information made available by GOD affects the network lifetime. As expected, higher precision results in better routing decisions, which in turn prolongs network lifetime. Additionally, the network lifetime performance of GOD with full precision and that offered by traditional DSR form the highest and lowest protocol design bounds.

Moving forward, an information gathering protocol was designed to remove the unrealistic assumption of global knowledge, which enables the routing protocols to operate in a distributed manner using partial knowledge. Initially, it was assumed that there is no energy depletion for the signalling introduced by the information gathering protocol and the effect of partial knowledge was studied. As expected, lack of instantaneous global energy map knowledge affects routing decisions, but the resultant lifetime lies between the lower and upper design bounds. Additionally, higher precision of the known information results in intelligent routing decisions which then extend network lifetime.

In the final stage of the implementation, the assumption that signalling does not deplete energy was removed and this practical protocol implementation was compared against its predecessor. At this stage, higher precision does not offer higher lifetimes because high precision results in rapidly outdated cached information which in turn, results in increased signalling and, consequently, increased energy consumption.

Finally, practical implementations of PTD and PTZ were compared to SPR and it was clearly shown that the proposed routing schemes prolong the network lifetime, compared to their traditional SPR and DSR counterparts, especially at low and intermediate loads.

7.2 Thesis Contributions

The thesis has provided proof by demonstration that multipath routing is advantageous in the capacity and energy constraint ad-hoc networks. The contributions can be summarised as follows:

- The importance of methodology to carry out research with accurate results and conclusions has been highlighted. Multiple techniques should be used to ensure validity of model and eliminate pitfalls in performance evaluation.

- A multipath on-demand routing scheme which exploits the route propagation procedure to discover multiple node disjoint paths and, subsequently, distribute the traffic to those paths has been proposed. The advantages of the proposed routing protocol are: 1) its simplicity, which allows easy implementation in existing networks, 2) it does not increase the route discovery overhead and, 3) it uses simultaneously the alternative discovered paths, unlike its predecessors, which do either one or the other. Additionally, the per-packet allocation scheme used selects paths with probabilities based on their length, which results in usage of the shortest path with higher probability. Finally, the use of path length in the selection cost function results in explicit knowledge of the cost by the source node and avoids additional signalling overheads which will be required by other schemes to maintain up-to-date costs.

- Two novel energy aware routing schemes were proposed, that prolong the network lifetime and are able to sustain connectivity after unavoidable node exhaustion occurs. The routing schemes required global energy map knowledge and base the routing decision on energy homogenisation, an approach not previously found in the literature.

- An extensive performance evaluation of the proposed routing schemes was performed under realistic conditions and have shown that the schemes prolong the network lifetime, even when they operate under partial knowledge of the energy map. It was shown that there is a trade-off between various levels of precision and network lifetime. Higher precision can results in marginally better routing decisions but increases the signalling and as a result the energy depletion is increased.

7.3 Future Work

This section discusses directions of future research and is categorised in two sections. These two sections discuss multipath routing from load distribution and energy awareness perspectives.

7.3.1 Multipath Routing for Load Distribution

Multipath routing offers improvements in terms of goodput, average end-to-end delay and routing efficiency. However, a series of effects arise when multipath routing is used. The most common of these effects is packet reordering. Data packets of the same flow can take different routes, which result in reordering; i.e. out of order delivery to the destination. This can be problematic when the receive buffers of the destination device are limited; quite common in wireless sensor networks. Packet reordering also affects the operation of the Transport Control Protocol (TCP) [115]. TCP can tolerate segment displacement of one or two positions. The effect of higher displacement is that the receiver generates duplicate acknowledgements, which cause the sender to invoke congestion control, which in turn reduces connection throughput. Finally, the multiple routes make the Round-Trip Time (RTT) measurements variable when these are measured over the multiple paths. Of course, TCP uses a smoothed RTT estimator [116], which is subsequently used to set the Retransmission Timeout (RTO) value.

To reduce packet reordering and improve TCP performance, the proposed routing scheme can be modified not to route every packet through a different path, but instead rotate the traffic periodically to the paths. The question then becomes, how much traffic should be routed through each path before switching to the next path. Coarse granularities will result in less packet reordering and increased TCP performance, while fine granularity results in better load distribution but increased reordering. Henceforth, it is necessary to quantify the trade-off between coarse and fine allocation granularity and identify any possible schemes which can adaptively adjust the allocation per path. Finally, TCP schemes specifically proposed in the literature for wireless ad-hoc networks [61, 62, 117] can be examined to quantify their performance in combination with the proposed multipath routing protocol.

MDSR-NDR uses a cost function based on path length to distributed the load to multiple paths. This cost function can become more comprehensive and include additional parameters such as the round trip times measured by TCP. The round trip time is an indication of congestion and can be used by MDSR-NDR to make more intelligent routing decision. This proposal is similar to Multipath Source Routing [82] but with the important advantage that it does not introduce additional signalling introduced by SRPing.

Having made all these modification to the MDSR-NDR simulation model, the MoteLab [42] testbed could be used to evaluate the schemes under a real ad-hoc network. MoteLab may not be an evaluation platform for mobile ad-hoc

networks but it does offer an inexpensive public resource to evaluate the proposed scheme and validate the simulation model. However, this task is not trivial as shown in other work carried out in the Broadband Networks group [118].

7.3.2 Multipath Routing for Energy Awareness

Further research can be carried out to quantify and evaluate further the merits of the proposed energy aware routing schemes. In addition to the evaluation of the proposed schemes under mobile scenarios, it is necessary to evaluate the schemes using energy models, which reflect more closely the physical behaviour of batteries. For example, models described in [47, 48, 49, 50, 119] consider non-linear effects, such as rate capacity and recovery effects. It is expected that the results will follow the same relative trends, but will better reflect the lifetime one should expect in a physical network.

In addition to the realistic battery models, energy harvesting devices, which convert ambient energy such as mechanical vibrations, wind, solar, acoustic and thermal energy to electrical energy, have attracted significant attention. The mechanics, which make these energy conversions possible, may not be directly related with the research presented in this thesis. However, it is interesting to evaluate the routing schemes under non-monotonically decreasing residual energy functions. From the routing protocols' perspective, network nodes which have energy scavenging capabilities, will have lower energy depletion rate and it is possible for their residual energy to increase, which in turn can affect the behaviour of the routing protocols. It is therefore necessary to evaluate the behaviour of the routing schemes under these conditions.

Among the high priority research directions would be to examine the routing schemes under non-linear quantisation schemes. As it was discussed in Section 6.1.4, non-linear quantisation can possibly prolong the network lifetime by introducing higher precision towards the lower end of energy scale. However, it was also shown that there is a trade-off between high precision and signalling to maintain up-to-date cached information and this trade-off must be examined further to determine the appropriate quantisation scheme.

Bibliography

[1] D. Johnson, "Routing in ad hoc networks of mobile hosts," in *Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on*, pp. 158–163, 8-9 Dec. 1994. (cited on page 1)

[2] A. Bateman, *Digital Communications*. Addison-Wesley, 1998. (cited on page 6)

[3] J. Proakis, *Digital Communications*. McGraw-Hill, 1983. (cited on page 6)

[4] H. Friis, "A note on a simple transmission formula," *Proceedings of the IRE*, vol. 34, pp. 254–256, May 1946. (cited on page 8)

[5] K. Fall and K. Varadhan, "The ns Manual (formerly ns Notes and Documentation)," The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, September 2003. (cited on pages xvii, 8, 10, 55, 70)

[6] "Bluetooth SIG." http://www.bluetooth.com. (cited on page 10)

[7] "ZigBee alliance." http://www.zigbee.org. (cited on page 10)

[8] "IEEE standards for local area networks: carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications," *ANSI/IEEE Std 802.3-1985*, 1985. (cited on page 11)

[9] "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1–1184, June 12 2007. (cited on pages 12, 13, 18, 21, 62)

[10] A. S. Tanenbaum, *Computer Networks*. Prentice Hall International, 2003. (cited on pages 12, 15)

[11] M. S. Gast, *802.11 Wireless Networks - The Definitive Guide*, ch. 15, pp. 311–342. O'Reilly, 2nd ed., April 2005. (cited on page 14)

[12] "IEEE Std 802.15.1 - 2005 IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. - Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)," *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pp. 1–580, 2005. (cited on page 14)

[13] Federal Communications Commission (FCC), "Revision of Part 15 of the Commission's Rules Regarding Ultra WideBand Transmission Systems." `http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-02-48A1.pdf`, February 2002. (cited on page 16)

[14] ECMA International, "Standard ECMA-368 High Rate Ultra Wideband PHY and MAC Standard, 1st Edition." `http://www.ecma-international.org/publications/standards/Ecma-368.htm`, December 2005. (cited on pages 16, 17)

[15] F. Nekoogar, *Ultra-Wideband Communications: Fundamentals and Applications.* Prentice Hall Communications Engineering and Emerging Technologies Series, Prentice Hall, 1 ed., August 2005. (cited on page 17)

[16] W. B. Heinzelman, *Application-Specific Protocol Architectures for Wireless Networks.* Phd, MIT - Massachusetts Institute of Technology, 2000. (cited on page 18)

[17] W.-Y. Choi and S.-K. Lee, "MAC Throughput Enhancement by Dynamic dot11RTSThreshold (doc.: IEEE 802.11-03/0509r0)." `https://mentor.ieee.org/802.11/documents`, July 2003. (cited on page 21)

[18] ISO Standards, "Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model (ISO/IEC 7498-1:1994)," 1994. (cited on page 27)

[19] S. M. Ballew, *Managing IP Networks with Cisco Routers*, ch. 5: Routing Protocol Selection. O'Reilly, 1997. (cited on pages xvii, 28, 32)

[20] E. Celebi, "Performance evaluation of wireless multi-hop ad-hoc network routing protocols," MSc, Bogazici University, 1998. (cited on page 34)

[21] P. P. Esnault, "IETF RFC4136: OSPF refresh and flooding reduction in stable topologies," July 2005. (cited on page 34)

[22] G. Pei, M. Gerla, and X. Hong, "LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility," in *MobiHoc '00: Proceedings of*

the 1st ACM international symposium on Mobile ad hoc networking & computing, (Piscataway, NJ, USA), pp. 11–18, IEEE Press, 2000. (cited on pages 34, 35)

[23] D. B. Johnson, D. A. Maltz, and Y. C. Hu, "IETF RFC4728: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," July 2004. (cited on pages 36, 41, 42)

[24] C. E. Perkins and E. Royer, "IETF RFC3561: Ad hoc On-Demand Distance Vector (AODV) Routing," July 2003. (cited on pages 36, 42)

[25] V. Park and S. Corson, "IETF Internet-Draft: Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification." `http://www.ietf.org/internet-drafts/draft-ietf-manet-tora-spec-04.txt`, July 2001. (cited on page 36)

[26] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Network*, vol. 16, pp. 11–21, Jul/Aug 2002. (cited on page 36)

[27] L. Kleinrock and K. Stevens, "Fisheye: A lenslike computer dispaly transformation," tech. rep., UCLA, Computer Science Department, 1971. (cited on page 36)

[28] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable routing strategies for ad hoc wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 17, pp. 1369–1379, Aug. 1999. (cited on pages xiii, 36, 37)

[29] I. Chakeres and C. E. Perkins, "IETF Internet-Draft: Dynamic MANET On-demand (DYMO) Routing." `http://www.ietf.org/internet-drafts/draft-ietf-manet-dymo-12.txt`, February 2008. (cited on page 37)

[30] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1567–1576, 23-27 June 2002. (cited on pages 41, 119)

[31] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pp. 90–100, 25-26 Feb. 1999. (cited on page 42)

[32] C. Perkins, E. Royer, S. Das, and M. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks," *Personal Communications, IEEE*, vol. 8, pp. 16–28, Feb. 2001. (cited on page 42)

[33] T. Clausen and P. Jacquet, "IETF RFC3626: Optimized Link State Routing Protocol (OLSR)." `http://www.ietf.org/rfc/rfc3626.txt`, October 2003. (cited on pages 44, 45)

[34] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pp. 3866–3875, 7-10 Jan 2002. (cited on page 45)

[35] A. Qayyum, "Wireless networks: Hiperlan," MSc, Universite de Paris-Sud Orsay, France, 1996. (cited on page 45)

[36] A. Hafslund, O. Kure, and K. Ovsthus, "Implementing and extending the optimized link state routing protocol," MSc, UniK - University Graduate Center - University of Oslo, 2004. (cited on page 46)

[37] "Gumstix Inc.." `http://www.gumstix.com/`. (cited on page 48)

[38] "Crossbow Technology Inc.." `http://www.xbow.com/`. (cited on page 48)

[39] "Moteiv Inc.." `http://www.moteiv.com/`. (cited on page 48)

[40] "Sentilla Corporation." `http://www.sentilla.com/`. (cited on page 48)

[41] Department of Electrical Engineering and Computer Science, "MoteLab: Harvard Sensor Network Testbed." `http://motelab.eecs.harvard.edu`. (cited on page 48)

[42] G. Werner-Allen, P. Swieskowski, and M. Welsh, "MoteLab: A Wireless Sensor Network Testbed," in *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN)*, 2005. (cited on pages 48, 179)

[43] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* Professional Computing, John Willey & Sons, 1991. (cited on pages xvii, 49, 50, 51, 64, 80, 87, 88)

[44] K. Watkins, *Discrete Event Simulation in C.* International Series in Software Engineering, London: The McGraw-Hill, 1st ed., 1993. (cited on pages 52, 64, 65, 80, 83)

[45] CMU Monarch, "The CMU Monarch project's wireless and mobility extensions to NS," 1998. (cited on page 55)

[46] T. S. Rappaport, *Wireless Communications: Principless & Practice.* Prentice Hall, 1995. (cited on page 56)

[47] G. Cox and L. Etzkom, "The effectiveness of battery-conserving protocols in wireless LANs," in *Advances in Wired and Wireless Communication, 2004 IEEE/Sarnoff Symposium on*, pp. 137–139, 26-27 Apr 2004. (cited on pages 61, 180)

[48] C. Chiasserini and R. Rao, "A model for battery pulsed discharge with recovery effect," in *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pp. 636–639, 21-24 Sept. 1999. (cited on pages 61, 180)

[49] V. Rao, G. Singhal, A. Kumar, and N. Navet, "Battery model for embedded systems," in *VLSI Design, 2005. 18th International Conference on*, pp. 105–110, 2005. (cited on pages 61, 180)

[50] K. Padmanabh and R. Roy, "Maximum lifetime routing in wireless sensor network by minimizing rate capacity effect," in *Parallel Processing Workshops, 2006. ICPP 2006 Workshops. 2006 International Conference on*, p. 8pp., 14-18 Aug. 2006. (cited on pages 61, 180)

[51] J. Banks, J. S. C. II, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation.* International Series in Industrial and Systems Engineering, Upper Saddle River, NJ 07458: Prentice Hall, 3rd ed., 2001. (cited on page 64)

[52] Pierre L'Ecuyer, "Good parameters and implementations for combined multiple recursive random number generators," *Operations Research*, vol. 47, no. 1, pp. 159–164, 1999. (cited on page 65)

[53] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," *ACM Transactions on Modeling and Computer Simulation*, vol. 8, pp. 3–30, January 1998. (cited on page 65)

[54] M. Matsumoto and Y. Kurita, "Strong deviations from randomness in m-sequences based on trinomials," *ACM Transactions on Modeling and Computer Simulation*, vol. 6, pp. 99–106, 1996. (cited on page 65)

[55] M. Matsumoto and T. Nishimura, *Monte Carlo and Quasi-Monte Carlo methods*, ch. A Nonempirical Test on the Weight of Pseudorandom Number Generators, pp. 381–395. Springer-Verlag, 2002. (cited on page 65)

[56] M. Matsumoto and T. Nishimura, "Sum-discrepancy test on pseudorandom number generators," *Mathematics and Computers in Simulation*, vol. 62, pp. 431–442, 2003. (cited on page 65)

[57] 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," *Mobile Computing and Networking, ACM/IEEE MobiCom*, pp. 85–97, October 1998. (cited on pages 67, 77)

[58] B. Divecha, A. Abraham, C. Grosan, and S. Sanyal, "Analysis of dynamic source routing and destination-sequenced distance-vector protocols for different mobility models," *ams*, vol. 0, pp. 224–229, 2007. (cited on page 67)

[59] P. Pham and S. Perreau, "Performance analysis of reactive shortest path and multipath routing mechanism with load balance," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 1, pp. 251–259, 30 March-3 April 2003. (cited on pages 69, 100, 116, 176)

[60] B. Bakshi, P. Krishna, N. Vaidya, and D. Pradhan, "Improving performance of TCP over wireless networks," in *Distributed Computing Systems, 1997., Proceedings of the 17th International Conference on*, pp. 365–373, 27-30 May 1997. (cited on page 71)

[61] P. Dimopoulos, P. Zeephongsekul, and Z. Tari, "Multipath Aware TCP (MATCP)," in *Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on*, pp. 981–988, 26-29 June 2006. (cited on pages 71, 179)

[62] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 19, pp. 1300–1315, July 2001. (cited on pages 71, 179)

[63] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing* (Imielinski and Korth, eds.), vol. 353, Kluwer Academic Publishers, 1996. (cited on page 77)

[64] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 2, pp. 1312–1321, 30 March-3 April 2003. (cited on pages 77, 78)

[65] C. Tachtatzis and D. Harle, "Performance evaluation of multi-path and single-path routing protocols for mobile ad-hoc networks," *International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS2008*, June 2008. (cited on page 91)

[66] R. Bhandari, *Survivable Networks Algorithms for Diverse Routing*. Kluwer Academic Publishers, 1999. (cited on pages 93, 94, 140, 141)

[67] A. Kist and R. Harris, "A heuristic to generate all best partially disjoint paths in a communications network," in *Communication Systems, 2002. ICCS 2002. The 8th International Conference on*, vol. 1, pp. 357–362, 25-28 Nov. 2002. (cited on page 93)

[68] A. Kist and R. Harris, "Note on the problem of partially link disjoint paths," in *Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, vol. 3, pp. 1680–1684, 15-18 Dec. 2003. (cited on page 93)

[69] T. Clinker, D. Mesko, G. Viola, and J. Tapolcai, "Routing with partially disjoint shared path (PDSP) protection," in *Next Generation Internet Networks, 2005*, pp. 47–52, 18-20 April 2005. (cited on page 93)

[70] J. Pu, E. Manning, and G. Shoja, "Routing reliability analysis of partially disjoint paths," in *Communications, Computers and signal Processing, 2001. PACRIM. 2001 IEEE Pacific Rim Conference on*, vol. 1, pp. 79–82, 26-28 Aug. 2001. (cited on page 93)

[71] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959. (cited on pages 93, 141)

[72] A. Nasipuri and S. Das, "On-demand multipath routing for mobile ad hoc networks," in *Eight International Conference on Computer Communications and Networks, 1999. Proceedings.*, pp. 64–70, 11-13 Oct. 1999. (cited on pages 98, 99)

[73] S.-J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *IEEE International Conference on Communications, 2001. ICC 2001.*, vol. 10, pp. 3201–3205, 11-14 June 2001. (cited on pages 98, 99)

[74] K. Wu and J. Harms, "Performance study of a multipath routing method for wireless mobile ad hoc networks," in *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*, pp. 99–107, 15-18 Aug. 2001. (cited on page 98)

[75] M. Marina and S. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Ninth International Conference on Network Protocols*, pp. 14–23, 11-14 Nov. 2001. (cited on pages 98, 99)

[76] R. Leung, J. Liu, E. Poon, A. L. C. Chan, and B. Li, "MP-DSR: a QoS-aware multi-path dynamic source routing protocol for wireless ad-hoc networks," *Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference on*, pp. 132–141, 2001. (cited on pages 98, 100)

[77] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A framework for reliable routing in mobile ad hoc networks," *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 1, pp. 270–280, 2003. (cited on page 98)

[78] V. Park and M. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, pp. 1405–1413, 7-11 April 1997. (cited on page 99)

[79] J. Raju and J. Garcia-Luna-Aceves, "A new approach to on-demand loop-free multipath routing," in *Computer Communications and Networks, 1999. Proceedings. Eight International Conference on*, pp. 522–527, 11-13 Oct. 1999. (cited on page 99)

[80] S.-J. Lee and M. Gerla, "AODV-BR: backup routing in ad hoc networks," in *Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE*, vol. 3, pp. 1311–1316, 23-28 Sept. 2000. (cited on page 99)

[81] R. Krishnan and J. Silvester, "Choice of allocation granularity in multipath source routing schemes," in *INFOCOM '93. Proceedings.Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking: Foundation for the Future. IEEE*, pp. 322–329, 28 March-1 April 1993. (cited on page 99)

[82] L. Wang, L. Zhang, Y. Shu, and M. Dong, "Multipath source routing in wireless ad hoc networks," in *Electrical and Computer Engineering, 2000 Canadian Conference on*, vol. 1, pp. 479–483, 7-10 March 2000. (cited on pages 99, 100, 179)

[83] L. Wang, Y. Shu, M. Dong, L. Zhang, and O. Yang, "Adaptive multipath source routing in ad hoc networks," in *IEEE International Conference on Communications, 2001. ICC 2001.*, vol. 3, pp. 867–871, 11-14 June 2001. (cited on page 99)

[84] L. Zhang, Z. Zhao, Y. Shu, L. Wang, and O. Yang, "Load balancing of multipath source routing in ad hoc networks," in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 5, pp. 3197–3201, 28 April-2 May 2002. (cited on page 100)

[85] S. Jung, N. Hundewale, and A. Zelikovsky, "Energy efficiency of load balancing in MANET routing protocols," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on*, pp. 476–483, 23-25 May 2005. (cited on page 100)

[86] S.-C. Huang and R.-H. Jan, "Energy-aware, load balanced routing schemes for sensor networks," *icpads*, vol. 00, p. 419, 2004. (cited on page 100)

[87] Y. Ping, B. Yu, and W. Hao, "A multipath energy-efficient routing protocol for ad hoc networks," in *Communications, Circuits and Systems Proceedings, 2006 International Conference on*, vol. 3, pp. 1462–1466, June 2006. (cited on page 100)

[88] R. Ma and J. Ilow, "Reliable multipath routing with fixed delays in manet using regenerating nodes," *lcn*, vol. 00, p. 719, 2003. (cited on page 100)

[89] N. Wisitpongphan and O. Tonguz, "Disjoint multipath source routing in ad hoc networks: transport capacity," in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 4, pp. 2207–2211, 2003. (cited on page 100)

[90] Z. Wu, H. Song, S. Jiang, and X. Xu, "Ant-based energy aware disjoint multipath routing algorithm in manets," in *Multimedia and Ubiquitous Engineering, 2007. MUE '07. International Conference on*, pp. 674-679, April 2007. (cited on page 100)

[91] Y. Ganjali and A. Keshavarzian, "Load balancing in ad hoc networks: single-path routing vs. multi-path routing," in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 1120–1125, 7-11 March 2004. (cited on pages 100, 101, 116, 176)

[92] S. Mueller and D. Ghosal, "Analysis of a distributed algorithm to determine multiple routes with path diversity in ad hoc networks," in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005. WIOPT 2005. Third International Symposium on*, pp. 277–285, 3-7 April 2005. (cited on page 100)

[93] W. Xu, P. Yan, and D. Xia, "Similar node-disjoint multi-paths routing in wireless ad hoc networks," in *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, vol. 2, pp. 731–734, 23-26 Sept. 2005. (cited on page 100)

[94] A. Nasipuri, J. Mandava, H. Manchala, and R. Hiromoto, "On-demand routing using directional antennas in mobile ad hoc networks," in *Ninth International*

Conference on Computer Communications and Networks, 2000. Proceedings., pp. 535–541, 16-18 Oct. 2000. (cited on page 100)

[95] S. Roy, S. Bandyopadhyay, T. Ueda, and K. Hasuike, "Multipath Routing in Ad Hoc Wireless Networks with Omni Directional and Directional Antenna: A Comparative Study," in *Distributed Computing. Mobile and Wireless Computing: 4th International Workshop, IWDC 2002 Calcutta, India, Proceedings*, vol. 2571/2002 of *Lecture Notes in Computer Science*, pp. 184–191, Springer Berlin / Heidelberg, 2002. (cited on page 100)

[96] C. Farinetto, D. Harle, C. Tachtatzis, and S. Zuyev, "Efficient routing for the extension of lifetime and quality of energy constrained ad hoc networks," *International Telecommunications Congress (ITC19)/Performance Challenges for Efficient Next Generation Networks (Eds.) X.J. Liang & Z.H. Xin,*, pp. 759–770, 2005. (cited on page 117)

[97] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for "Smart Dust"," in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, (New York, NY, USA), pp. 271–278, ACM Press, 1999. (cited on page 118)

[98] S. Jayashree and C. S. R. Murthy, "A taxonomy of energy management protocols for ad hoc wireless networks," *Communications Magazine, IEEE*, vol. 45, pp. 104–110, April 2007. (cited on page 118)

[99] M. Weiser, B. B. Welch, A. J. Demers, and S. Shenker, "Scheduling for Reduced CPU Energy," in *OSDI*, pp. 13–23, 1994. (cited on page 118)

[100] H. Shimada, H. Ando, and T. Shimada, "Pipeline stage unification: a low-energy consumption technique for future mobile processors," in *Low Power Electronics and Design, 2003. ISLPED '03. Proceedings of the 2003 International Symposium on*, pp. 326–329, 25-27 Aug. 2003. (cited on page 118)

[101] A. Savvides, S. Park, and M. B. Srivastava, "On modeling networks of wireless microsensors," in *SIGMETRICS '01: Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, (New York, NY, USA), pp. 318–319, ACM Press, 2001. (cited on page 118)

[102] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy-aware wireless microsensor networks," *Signal Processing Magazine, IEEE*, vol. 19, pp. 40–50, March 2002. (cited on page 118)

[103] E.-S. Jung and N. H. Vaidya, "A power control MAC protocol for ad hoc networks," in *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 36–47, ACM Press, 2002. (cited on page 119)

[104] M. Zawodniok and S. Jagannathan, "Energy-efficient rate adaptation MAC protocol for ad hoc wireless networks," in *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*, pp. 389–394, 7-9 April 2005. (cited on page 119)

[105] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 95–107, ACM Press, 2004. (cited on page 119)

[106] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: a hybrid MAC for wireless sensor networks," in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 90–101, ACM Press, 2005. (cited on page 119)

[107] K. Scott and N. Bambos, "Routing and channel assignment for low power transmission in PCS," in *Universal Personal Communications, 1996. Record., 1996 5th IEEE International Conference on*, vol. 2, pp. 498–502, 29 Sept.-2 Oct. 1996. (cited on page 119)

[108] C.-K. Toh, H. Cobb, and D. Scott, "Performance evaluation of battery-life-aware routing schemes for wireless ad hoc networks," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 9, pp. 2824–2829, 11-14 June 2001. (cited on pages 119, 121, 122, 123, 141)

[109] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, (New York, NY, USA), pp. 181–190, ACM Press, 1998. (cited on pages 121, 141)

[110] D. Kim, J. Garcia-Luna-Aceves, K. Obraczka, J.-C. Cano, and P. Manzoni, "Routing mechanisms for mobile ad hoc networks based on the energy drain rate," *Mobile Computing, IEEE Transactions on*, vol. 2, pp. 161–173, April-June 2003. (cited on pages 122, 123)

[111] R Development Core Team, "R: A language and environment for statistical computing." `http://www.R-project.org`, 2007. ISBN 3-900051-07-0. (cited on page 129)

[112] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992. (cited on page 131)

[113] G. K. Kanji, *100 Statistical Tests*. SAGE Publications, 2 ed., 1999. (cited on page 131)

[114] J. Dunlop and D. G. Smith, *Telecommunications Engineering*. Stanley Thornes, 3rd ed., 1994. (cited on page 151)

[115] C. M. Arthur, *The Impact of Packet Reordering on Internet Protocol Networks*. PhD thesis, University of Strathclyde, 2008 (to be published). (cited on page 179)

[116] W. R. Stevens, *TCP/IP illustrated (vol. 1): the protocols*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1993. (cited on page 179)

[117] C. Ma and K.-C. Leung, "Improving TCP reordering robustness in multipath networks," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pp. 409–410, 16-18 Nov. 2004. (cited on page 179)

[118] A. G. Jamieson, "A novel systems design approach to wireless sensor networks for industrial applications." EngD Thesis, Universities of Edinburgh, Glasgow, Heriot-Watt, Strathclyde, 2008 (to be published). (cited on page 180)

[119] M. Handy and D. Timmermann, "Simulation of mobile wireless networks with accurate modelling of non-linear battery effects," in *International Conference in Applied Simulation Modelling*, pp. 532–537, 2003. (cited on page 180)