

Investigating Credit Based Mechanisms for Enhancing Performance in Wireless Ad Hoc Networks

by

Ariel Shei Goldberg



Thesis presented in fulfilment of the requirements for the degree

Master of Science in Mathematical Sciences at

Stellenbosch University

Supervisor: Prof. Anthony E. Krzesinski

December 2012

Declaration

By submitting this thesis/dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

December 2012

Copyright 2012 Stellenbosch University

All rights reserved

Abstract

This thesis explores two key aspects of wireless ad hoc networks. The first aspect concerns the topic of stimulating cooperation between nodes in a wireless ad hoc network. The functionality of a wireless ad hoc network depends entirely on the willingness of nodes to relay messages on behalf of other nodes. Network functionality depends on ensuring cooperation between nodes, so that each node benefits from continued participation in the network. This suggests an important question: how can cooperation among individual nodes be managed to improve overall wireless ad hoc network performance?

The second aspect explored in this thesis concerns the concept of optimal resource utilisation. Wireless ad hoc networks are characterised by limited bandwidth and energy resources, which facilitates deployment in situations in which traditional infrastructure based networks are not practical. This suggests another important question: how can the use of the limited energy and bandwidth resources of wireless ad hoc networks be optimised?

This research relies on the concept of a credit-based market economy. Nodes in simulated ad hoc networks use credits to pay for the cost of sending their own traffic and earn credits by forwarding traffic on behalf of other nodes. We show that a credit-based market economy approach can be employed to stimulate and regulate cooperation between nodes in a wireless ad hoc network. We show that this approach can be implemented in a simple decentralised manner and that it has several variants depending on which node is considered to be paying for the service, what the price of each service should be and how we route packets around the network using information derived from the credit-based economy.

This thesis demonstrates that several variants of a credit-based scheme can be implemented in a packet based simulator and that these variants result in the stable operation of the network and improve the overall performance. The credit-based mechanisms also show significant improvement to network performance in resource constrained conditions and represent an effective means for optimising limited energy and bandwidth resource. The effectiveness of the credit-based mechanisms increases as the load on the networks increases.

Opsomming

Hierdie tesis ondersoek twee belangrike aspekte van draadlose ad hoc-netwerke. Die eerste aspek het betrekking op die onderwerp van 'n stimulerende samewerking tussen die nodusse in 'n draadlose ad hoc-netwerk. Dat die network funksioneer hang daarvan af om samewerking tussen die nodusse te verseker sodat elke nodus voordeel put uit voortgesette deelname in die netwerk. Dit dui op 'n belangrike vraag: Hoe kan die samewerking tussen die individuele nodusse bestuur word om die prestasie van 'n draadlose ad hoc-netwerk te verbeter?

Die tweede aspek wat in hierdie tesis ondersoek word, behels die konsep van optimale hulpbronbenutting. Draadlose ad hoc-netwerke word gekenmerk deur beperkte bandwydte- en energie-hulpbronne, wat ontplooiing bewerkstellig in situasies waar tradisionele infrastruktuurbaseerde netwerke nie practise moontlik is nie. Dit dui op 'n ander belangrike vraag: Hoe kan die gebruik van die beperkte energie- en bandwydte-hulpbronne van draadlose ad hoc-netwerke optimaal bestuur word?

Hierdie navorsing berus op die konsep van 'n krediet-gebaseerde markekonomie. Nodusse in gesimuleerde ad hoc-netwerke gebruik krediete om te betaal vir die versending van hul eie verkeer en nodusse verdien krediete deur die verkeer van ander nodusse aan te stuur. Ons wys dat die benadering van 'n krediet-gebaseerde markekonomie gebruik kan word om die samewerking tussen die nodusse in 'n draadlose ad hoc-netwerk te stimuleer en te reguleer. Ons wys dat hierdie benadering geïmplementeer kan word op 'n eenvoudige gedesentraliseerde wyse. Ons ondersoek verskeie variasies van die benadering, na gelang van watter nodus oorweeg word om vir die diens te betaal, wat die prys van elke diens moet wees en hoe inligting aangeleid kan word van die krediet-gebaseerde ekonomie gebruik kan word om pakkies in die netwerk te roeteer.

Hierdie tesis toon dat verskeie variante van 'n krediet-gebaseerde skema geïmplementeer kan word in 'n netwerksimulator en dat hierdie variante die stabiele bedryf van en algehele verbetering in die prestasie van die network tot gevolg het. Die krediet-gebaseerde mechanismes toon 'n beduidende verbetering in hulpbronbenutting en netwerkprestasie in omgewings met beperkte hulpbronne en verteenwoordig 'n doeltreffende manier om die beperkte energie- en bandwydte-hulpbronne optimaal te benut. Laastens, die doeltreffendheid van die krediet-gebaseerde mechanismes word verhoog as die las op die netwerke word verhoog.

Acknowledgements

To my supervisor Prof. Krzesinski: thank you for your tireless guidance, support and encouragement during this thesis. I have learnt a tremendous amount from you over the last four years and I appreciate your exceptionally valuable contribution to my education.

To my parents Colleen and Steve: thank you for your infinite encouragement and support. You are both outstanding role models of how to serve your communities and humanity as professionals and I hope that this degree takes me one step closer to replicating in some small way the wonderful contributions you have made during your careers.

Table of Contents

Declaration	2
Abstract	3
Opsomming	4
Acknowledgements	5
Table of Contents	6
List of Figures.....	9
List of Tables	10
Glossary of terms	11
1. Introduction	12
2. Literature Review	14
2.1. Wireless Ad Hoc Networks	14
2.1.1. Applications for Wireless Ad Hoc Networks.....	15
2.1.2. Current Status of Real World Wireless Ad Hoc Networks	18
2.1.3. Routing Protocols in Wireless Ad Hoc Networks	20
2.1.4. Table Based Protocols.....	22
2.1.5. On Demand Protocols	24
2.1.6. Hybrid Protocols.....	28
2.1.7. Energy Conserving Routing in Wireless Ad Hoc Networks	29
2.2. Mechanisms for Stimulating Fairness in Wireless Ad Hoc Networks	31
2.2.1. Reputation Mechanisms	31
2.2.2. Credit Based Mechanisms	31
3. Methodology	34
3.1. Overview	34
3.2. The Basic Schemes	34
3.2.1. Free for All.....	34
3.2.2. Tit for Tat.....	35
3.3. The Credit Based Schemes.....	36
3.3.1. The Basic Credit-Based Scheme.....	36
3.3.2. Origin Pays with Constant Pricing.....	38
3.3.3. Origin Pays with Congestion Based Pricing	40
3.3.4. Destination Pays with Constant Pricing	42
3.3.5. Destination Pays with Congestion Based Pricing	44

3.3.6.	Origin Pays with Energy Based Pricing	45
3.3.7.	Origin Pays with Credit Based Pricing	48
3.4.	Routing Metric Based Schemes	50
3.4.1.	Congestion Based Routing	50
3.4.2.	Energy Based Routing	50
3.4.3.	Credit Based Routing	51
3.5.	The Simulator	52
3.5.1.	Node Structure in ns2	52
3.5.2.	Wireless Propagation Model	54
3.5.3.	Medium Access Control and Contention-Based Access Using the DCF in ns2	55
3.5.4.	Routing Protocol Implementation.....	59
3.5.5.	Energy Model Implementation	59
3.6.	Measures of Network Performance	60
3.6.1.	Packet Success Probability	60
3.6.2.	Throughput Rate Per Node	60
3.6.3.	Total Average Latency	60
3.6.4.	Energy Rate Per Node	60
3.6.5.	Network Fairness	61
3.7.	The Network Topology.....	61
3.8.	The Experiments.....	63
3.8.1.	Experiment 1: Performance across varying packet arrival rates	63
3.8.2.	Experiment 2: What is the effect of the redistribution mechanism.....	63
3.8.3.	Experiment 3: What is the effect of the redistratate	63
3.8.4.	Experiment 4: What is the effect of the target credit balance	64
3.8.5.	Experiment 5: What is the effect on the use of resources in the network.....	64
4.	Results	65
4.1.	Experiment 1: Performance across varying packet arrival rates	65
4.1.1.	Which Credit Based Scheme is Most Effective at Improving Network Performance..	65
4.1.2.	Performance of the Basic Schemes: <i>FFA</i> and <i>TFT</i>	70
4.1.3.	Performance of the Credit Based Schemes: <i>OP-const</i> and <i>OP-cong</i>	76
4.2.	Experiment 2: What is the effect of the redistribution mechanism	79
4.2.1.	<i>OP-const</i> : the effect of the redistribution mechanism	80
4.2.2.	<i>OP-cong</i> : the effect of the redistribution mechanism.....	81
4.3.	Experiment 3: What is the effect of the redistratate.....	81
4.3.1.	<i>OP-const</i> : the effect of redistribution rate	82
4.3.2.	<i>OP-cong</i> : the effect of redistribution rate	83
4.4.	Experiment 4: What is the effect of the target credit balance	84
4.4.1.	<i>OP-const</i> : the effect of target credit balance	85
4.4.2.	<i>OP-cong</i> : the effect of target credit balance	86

4.5.	Experiment 5: What is the effect on the use of resources in the network	87
4.5.1.	Low packet arrival rates	88
4.5.2.	Low packet arrival rates	90
4.5.3.	Medium packet arrival rates	95
4.5.4.	High packet arrival rates	99
5.	Conclusions	103
6.	Recommendations for Further research.....	108
7.	References.....	109

List of Figures

Figure 1: Basic structure of a wireless node in ns2	53
Figure 2: Node layout for 50 node network.....	62
Figure 3: Overall network Packet Success Probability for three representative packet arrival rates ...	66
Figure 4: Overall network TRPN (pps) for three representative packet arrival rates	68
Figure 5: Overall network TAL (s) for three representative packet arrival rates	69
Figure 6: Overall network ERPN (J/s) for three representative packet arrival rates	70
Figure 7: Packet Success Probability vs packet arrival rate for <i>FFA</i> and <i>TFT</i>	71
Figure 8: TRPN (pps) vs packet arrival rate for <i>FFA</i> and <i>TFT</i>	72
Figure 9: TAL (s) vs packet arrival rate for <i>FFA</i> and <i>TFT</i>	73
Figure 10: ERPN (J/s) vs packet arrival rate for <i>FFA</i> and <i>TFT</i>	74
Figure 11: rate and reciprocity vs packet arrival rates for <i>FFA</i> and <i>TFT</i>	75
Figure 12: PSP and TRPN (pps) for <i>FFA</i> , <i>OP-const</i> and <i>OP-cong</i> schemes vs packet arrival rate	77
Figure 13: TAL (s) and ERPN (J/s) for <i>FFA</i> , <i>OP-const</i> and <i>OP-cong</i> schemes vs packet arrival rate.	78
Figure 14: Reciprocity and rate of <i>FFA</i> , <i>OP-const</i> , and <i>OP-cong</i>	79
Figure 15: <i>OP-const</i> : PSP, TRPN(pps), TAL(s) and ERPN(J/s) for redistribution vs no redistribution	80
Figure 16: <i>OP-cong</i> : PSP, TRPN(pps), TAL(s) and ERPN(J/s) for redistribution vs no redistribution .	81
Figure 17: <i>OP-const</i> : PSP, TRPN (pps), TAL (s) and ERPN (J/s) for varying redistratate.....	83
Figure 18: <i>OP-cong</i> : PSP, TRPN (pps), TAL (s) and ERPN (J/s) for varying redistratate	84
Figure 19: <i>OP-const</i> : PSP, TRPN (pps), TAL (s) and ERPN (J/s) for varying target credit balance	86
Figure 20: <i>OP-cong</i> : PSP, TRPN (pps), TAL (s) and ERPN (J/s) for varying target credit balance.....	87
Figure 21: PSP for high packet arrival rates for <i>FFA</i> , <i>TFT</i> , <i>OP-const</i> and <i>OP-cong</i>	89
Figure 22 PSP, throughput (packets) and energy (Joules) of <i>FFA</i> : low packet arrival rates.....	91
Figure 23 PSP, throughput (packets) and energy (Joules) of <i>TFT</i> : low packet arrival rates.....	92
Figure 24 PSP, throughput (packets) and energy (Joules) of <i>OP-const</i> : low packet arrival rates	93
Figure 25 PSP, throughput (packets) and energy (Joules) of <i>OP-cong</i> : low packet arrival rates	94
Figure 26 PSP, throughput (packets) and energy (Joules) of <i>FFA</i> : medium packet arrival rates	95
Figure 27 PSP, throughput (packets) and energy (Joules) of <i>TFT</i> : medium packet arrival rates	96
Figure 28 PSP, throughput (packets) and energy (Joules) of <i>OP-const</i> : medium packet arrival rates	97
Figure 29 PSP, throughput (packets) and energy (Joules) of <i>OP-cong</i> : medium packet arrival rates.	98
Figure 30 PSP, throughput (packets) and energy (Joules) of <i>FFA</i> : high packet arrival rates	99
Figure 31 PSP, throughput (packets) and energy (Joules) of <i>TFT</i> : high packet arrival rates	100
Figure 32 PSP, throughput (packets) and energy (Joules) of <i>OP-const</i> : high packet arrival rates	101
Figure 33 PSP, throughput (packets) and energy (Joules) of <i>OP-cong</i> : high packet arrival rates....	102

List of Tables

Table 1 PHY Characteristics of MAC802.11b network in ns2	58
Table 2 95% confidence interval halfwidths for <i>FFA</i> , <i>TFT</i> , <i>OP-const</i> and <i>OP-cong</i>	67

Glossary of terms

AODV: ad hoc on demand distance vector (page 27)

BSC: base station controller (page 14)

BTS: base transceiver station (page 14)

DBF: Distributed Bellman-Ford algorithm (page 22)

DP-cong: destination pays with congestion related pricing (page 44)

DP-const: destination pays with constant pricing (page 42)

DSDV: destination-sequence distance-vector (page 22)

DSR: destination source routing (page 24)

FFA: free for all (page 34)

FFA-battery-routing: energy-based-routing (page 50)

MSC: mobile switching centre (page 14)

OP-cong: origin pays with congestion related pricing (page 40)

OP-cong-routing: origin pays with congestion related pricing and congestion based routing (page 50)

OP-const: origin pays with constant pricing (page 38)

OP-const-cred-routing: origin pays with credit based routing (page 51)

OP-credit: origin pays with credit related pricing (page 48)

OP-energy: origin pays with energy related pricing (page 45)

PSP: packet success probability (page 60)

TAL: total average latency (page 60)

TFT: tit for tat (page 35)

TORA: temporarily-ordered routing algorithm (page 25)

TRPN: throughput rate per node (page 60)

USAR: urban search and rescue (page 15)

1. Introduction

Ad hoc wireless networks are collections of wireless mobile nodes that form dynamic temporary networks without the use of any existing network infrastructure or centralized administration [1]. The concept of a wireless ad hoc network was developed in the 1970's, when DARPA created the first example of a wireless ad hoc network (called Packet Radio NETwork, PRNET). The ad hoc wireless network concept has been an active area of research since then and continues to be so today. The development of wireless ad hoc networks has branched into multiple specialised applications, such as sensor networks, mesh networks, vehicular ad hoc networks and disaster relief networks.

The aftermath of Hurricane Katrina recently demonstrated the utility of ad hoc networks. When Katrina struck New Orleans on the 29th of August 2005, the only network that continued functioning was a wireless ad hoc network, which aggregated the city's security camera data. This network became the primary means of communication in New Orleans, carrying business and consumer VoIP traffic for months while traditional communication systems were being repaired. Wireless ad hoc networks have been used to measure the soil temperature in forests and jungles, to keep swarms of robots in communication with one another and to provide Internet access to members of wireless user groups throughout cities around the world.

This thesis explores two key aspects of wireless ad hoc networks. The first aspect concerns the important topic of stimulating cooperation between nodes in a wireless ad hoc network. The functionality of a wireless ad hoc network depends entirely on the willingness of nodes to relay messages on behalf of other nodes. In some contexts, all the nodes may belong to a single authority that clearly motivates nodes to act on behalf of one another. In other contexts, nodes belonging to multiple authorities may require coordinated motivation to act on behalf of one another. Network functionality depends on ensuring cooperation between nodes, so that each node benefits from continued participation in the network. This suggests an important question: how can cooperation among individual nodes be managed to improve overall wireless ad hoc network performance?

The second aspect explored in this thesis concerns the concept of optimal resource utilisation. Wireless ad hoc networks are characterised by limited bandwidth and energy resources, which facilitates deployment in situations in which traditional infrastructure based networks are not practical. This suggests another important question: how can the use of the limited energy and bandwidth resources of wireless ad hoc networks be optimised?

This thesis addresses both of these important questions. Conceptually, this research relies on the concept of a credit-based market economy. Nodes in simulated ad hoc networks use credits to pay for the cost of sending their own traffic and earn credits by forwarding traffic on behalf of other nodes. The concept of a credit-based scheme in wireless ad hoc networks was proposed by Butyán et al. [2] for the purposes of stimulating cooperation between nodes. The notion was further explored by Crowcroft et al. [3] and Göbel et al. [4]. I extend these studies by implementing the credit-based scheme in ns2, which is one of the most widely used network simulators available today [5].

This thesis makes conceptual and empirical contributions. Conceptually, we show that a credit-based market economy approach can be employed to stimulate and regulate cooperation between nodes in a wireless ad hoc network. We show that this approach can be implemented in a simple decentralised manner and that it has several variants depending on which node is considered to be paying for the service, what the price of each service should be and how we route packets around the network using information derived from the credit-based economy.

The basic outline of the credit-based scheme is as follows. Each node is assigned a target credit balance. Each time a node sends or receives a packet it is considered to be performing a service. Nodes earn credit by performing services on behalf of other nodes. Nodes spend credit in order to send their own packets. In this way, nodes exchange credits for services resulting in a market economy. If a node does not have sufficient credit to send its own packet, the packet will be dropped. A decentralised credit redistribution mechanism is implemented to ensure the stability of the system. We will show that the credit-based schemes provide both a means of managing constrained resources in a network and motivating nodes to act on behalf of one another.

This thesis shows how the credit-based schemes can be implemented in a packet-based simulation model and used to measure the effectiveness of the several variants in improving network performance and network efficiency. A method to determine which variants of the credit-based schemes are most effective will also be illustrated, facilitating an evaluation of the practical value of credit-based schemes in real world ad hoc networks. A number of metrics are described for evaluating the performance of the network. Packet success probability (PSP) is a measure of the probability of a packet being delivered to its destination. Throughput rate per node (TRPN) is a measure of how much throughput the average node is capable of achieving during a simulation. Total average latency (TAL) is a measure of the average latency experienced by a packet when passing through the network and energy rate per node (ERPN) is a measure of the rate of energy use for the average node. We define two further metrics called the reciprocity and the fairness. A high reciprocity means all the nodes perform equal amounts of service on behalf of one another. This means that there is a balanced use of resources across the network. A high fairness means that the performance of the nodes is equivalent.

The remainder of this thesis is organised as follows: Chapter 2 presents a literature review of wireless ad hoc networks and credit mechanisms for stimulating cooperation. Chapter 3 describes the credit mechanisms used in this study and outlines their implementation in ns2. The network performance measures and experiments conducted are described in detail. Results are presented in Chapter 4 and conclusions are discussed in Chapter 5. Recommendations for further research are given in Chapter 6.

2. Literature Review

2.1. Wireless Ad Hoc Networks¹

In 1973, the Defence Advanced Research Projects Agency (DARPA) initiated research on the feasibility of using packet switched, store-and-forward radio communication to provide reliable computer communications [6]. It was the beginning of wireless ad hoc networks and it was called PRNET, a Packet Radio NETwork that reached the size of about 50 nodes and allowed some nodes to be mobile [7]. The follow-up project of PRNET was SURAN (SURvivable Adaptive Networks, 1983–1990).

For the SURAN project, a number of routing algorithms, an in-lab emulator and a real-world demonstrator based on custom made hardware were developed [8]. With the advent of personal computers and other viable communications equipment in the early 90's the idea of commercialized ad hoc networks took off. At the same time the idea of a collection of mobile nodes was proposed at several research conferences. Since then the area of mobile ad hoc networks has been an active field of research and continues to be so today [9].

Communication networks are traditionally built in much the same way as a network of roads. An initial outlay of capital is made, the infrastructure and paths are laid down and then these fixed paths are maintained and upgraded as the traffic increases. The infrastructure branches in a tree-like manner from large centres out towards the serviced area. Any communication initiated from the serviced area will generally traverse upwards in the branched hierarchy and then downwards in the correct branch to reach its destination.

Even in a wireless cellular network, if two people make a telephone call in adjacent rooms from one another, their call will still take the same path it would have had they been hundreds of kilometres apart. They will each talk to a *Base Transceiver Station (BTS)*, which will route their traffic towards the *Base Station Controller (BSC)* which in turn will route the traffic towards a *Mobile Switching Centre (MSC)*. At the MSC, the call will be connected. The same is true of the packet switched Internet and most other communication networks. Ultimately, the fixed infrastructure of the network is laid down and communication can only happen along these fixed paths.

There are times however, when there is no fixed infrastructure and the time and effort required to roll out a network is not available. An example would be enabling soldiers to communicate on a battlefield or enabling rescue workers to communicate in the 72 hours after an earthquake has destroyed the traditional communications networks.

¹ Another term often used to refer to Ad Hoc Wireless networks is that of a Mobile Ad Hoc Network or MANET. A MANET refers specifically to a wireless ad hoc network in which the nodes are mobile and move around freely. A MANET is required to be able to adjust to changes in the topology of nodes, with nodes being able to enter and leave the network without disturbing its operation. MANETs are designed to be immediately deployable in areas without fixed infrastructure

An ad hoc network is a collection of wireless mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralized administration [1]

The individual nodes within the network (end user clients) cooperatively route traffic back and forth amongst one another to create routes from any one individual node in the network to another. There is no need for an underlying infrastructure as the nodes themselves become the communication medium for transporting data. If node A and node B are not within range of one another and node A wishes to send a message to node B, node A can send its packets to an intermediate node C which will forward the packets to node B. The nodes can communicate via means of multi hop routes so long as there are intermediate nodes between the source and the destination to be able to forward the packets.

Wireless ad hoc networks are decentralized. Their operation does not depend on any centralized authority. This makes them resilient to change and well suited to certain applications.

2.1.1. Applications for Wireless Ad Hoc Networks

Disaster Relief Networks

The need for effective communications in a disaster relief situation is clear. Whether it is directing a team of fire fighters through a collapsed building, uploading medical data from field units or controlling specialized search and rescue robots, the ability to exchange information is vital and can greatly enhance the success of the disaster relief effort. Traditional communication networks are often unable to meet the communications requirements of first responders due to infrastructure damage or congestion. Satellite communications provides a means for people on the ground to talk to the outside world, but these very quickly become congested and are not effective for enabling first responders to communicate and coordinate their own activities. Wireless Ad Hoc Networks provide an effective solution for a dedicated disaster relief network in the absence of fixed infrastructure.

MANETs are quick and easy to deploy and can function in the absence of a power network. Nodes consisting of a simple laptop computer equipped with a wireless interface card and a petrol generator for power could be dropped off at strategic positions to form an ad hoc wireless network. For voice communication, a simple two way radio network would be easier to deploy, but the ad hoc wireless network would have the advantage that it could be used to transfer data as well as just voice. A command and control centre could send building blueprints, maps and any other applicable data to workers in the field.

Another promising field of application for wireless mobile ad hoc networks in disaster relief situations is that of *Urban Search and Rescue (USAR)* robotics. The use of robots in USAR is an idea filled with great potential and wireless ad hoc networks would be a fundamental component of many USAR robotic solutions as the robots will need a data communication channel to upload video and other data they are collecting as well as for controllers to be able to drive them. One possibility is the notion of a swarm of small insect like robots that will be released onto a damaged building to search for and assess the conditions and possibility of finding survivors. These insect like robots would each be

equipped with a wireless network card and would form a MANET by which they would communicate with one another and a central controller.

Sensor Networks

A sensor network is composed of a large number of sensor nodes, which are densely deployed either inside the area to be monitored or very close to it [10]. Recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications and digital electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate over short distances. The nodes can be randomly distributed throughout or near to the area to be monitored and upload their measurements to a central gateway or storage node via a self-organized ad hoc network.

Sensor networks are used for a wide range of applications. They are a relatively recent technology, emerging within the last decade, but they have seen a steady increase in applications and interest over the last ten years as indicated by Google Scholar's steadily increasing hit rate for the term "sensor network" [11]. Sensor network applications can be categorized into military, environmental, health, home and other commercial areas.

Possible military applications for sensor networks are [10]:

- Monitoring friendly forces equipment and ammunition: sensors can be placed on military equipment (vehicles, weapons, ammunition etc.) to monitor the equipment's status, condition and position. Information can be forwarded to a central hub and a full status view can be achieved.
- Battlefield surveillance: sensor networks can be quickly deployed over critical areas to monitor the activities of enemy troops.
- Targeting: sensor networks can be incorporated into the weapons guidance systems.
- Battle damage assessment.
- Nuclear, biological and chemical (NBC) attack detection and reconnaissance: sensor networks can be used to detect hazardous materials.

Possible environmental applications for sensor networks are [12]:

- Cattle monitoring: sensor networks can be used to record the position of cattle over time and the soil moisture within a paddock
- Ground water quality monitoring: sensor networks can be used to monitor the salinity, water table level and rate of extraction of a borehole
- Rainforest monitoring: sensor networks can be used to provide information on soil moisture, temperature, wind speed etc. of natural environments
- Lake water quality monitoring: a project in Brisbane Australia used sensor networks to measure the vertical temperature profile of a large body of storage water.

Possible health applications for sensor networks are [13]:

- Monitoring in mass casualty disasters: triage becomes challenging as the numbers of victims increases. Sensor networks can be used to monitor many victims simultaneously and to monitor the health status of first responders.
- Vital sign monitoring in hospitals: wireless sensing technology helps address various drawbacks associated with wired sensors that are commonly used in hospitals and emergency rooms to monitor patients.
- At-home and mobile aging: wireless network sensors embedded in people's living spaces or carried on the person can collect information about personal physical, physiological, and behavioural states and patterns in real-time.
- Assistance with motor and sensory decline: traditional assistive devices such as canes, crutches, walkers, and wheel chairs can collate information from built-in and external sensors to provide the users with continual personalized feedback and guidance towards the correct usage of the devices.
- Large-scale in-field medical and behavioural studies: Body-worn sensors together with sensor-equipped Internet-connected smartphones have begun to revolutionize medical and public health research studies by enabling behavioural and physiological data to be continually collected from a large number of subjects as they lead their day to day lives.

Possible home applications for sensor networks are [10]:

- Smart sensor nodes and actuators can be embedded in appliances, such as vacuum cleaners, micro-wave ovens, refrigerators, and VCRs. These sensor nodes inside domestic devices can interact with each other and with the external networks via the Internet or Satellite.

Other commercial areas range from vehicle tracking to interactive museums to managing inventory. The technology of sensor networks is still relatively new and many applications will become available as the ability of sensor nodes matures.

Vehicle Ad Hoc Networks

Vehicle ad hoc networks (VANETs) are a special case of mobile ad hoc networks. They can be formed between vehicles with vehicle to vehicle (V2V) communication or between vehicles and an infrastructure with vehicle to infrastructure (V2I) communication [14]. There are two types of VANET applications. Applications that increase safety on the roads are called safety applications and applications that provide some kind of value added service are called user applications.

Studies [15] show that about 60% roadway collisions could be avoided if the driver of the vehicle was provided warning at least one-half second prior to a collision. Wireless communication technologies are promising to significantly reduce the delay in propagating emergency warnings. Instead of relying on brake lights and line of sight, there is potential for cars to signal an emergency situation via a V2V VANET and inform other drivers of a potential hazard.

There is also a multitude of applications that attempt to reduce vehicular traffic by informing drivers of problems on the road [16] [17] and less congested routes to their destination.

User applications can provide road users with information, advertisements and entertainment during their journey. These may involve connecting the vehicle and its occupants to the Internet [18], or peer-to-peer applications in which motorists share music or information about their destinations.

Again, VANETs are an emerging technology and present significant challenges before they will become widespread.

2.1.2. Current Status of Real World Wireless Ad Hoc Networks

There are many examples of real world MANETs, wireless mesh networks and sensor networks. Most implementations of wireless ad hoc networks however, have been set up for test purposes. There are examples of commercially functioning ad hoc networks, but these have not yet become widespread. The world has experienced a communications revolution since the 90's and wireless ad hoc networks are not intended to compete with technologies such as ADSL, WiFi, UMTS and WiMax. Wireless ad hoc networks have specific roles to play in specialized technologies where infrastructure based communications are too expensive and impractical to set up.

Below is a summary of real world implementations of wireless ad hoc networks.

Sensor Networks

The most commercially advanced form of wireless ad hoc networks are sensor networks. There are a number of companies manufacturing and running sensor networks today [19] [20] [21]. This is due to the fact that sensor networks operate on a much smaller scale and have a far more focused task than that of other ad hoc networks.

Some experimental sensor networks are summarized below.

One of the largest sensor networks was deployed in the context of the ExScal project [22]. The network consisted of more than 1000 sensor nodes and about 200 802.11b nodes that served as backbone network, creating a 2-tier network structure [8].

In [23] the performance of a network of 70 MICA2 motes equipped with dual-axis magnetometers was evaluated. The application of the sensor network was to monitor enemy targets without endangering personnel. The ability to deploy unmanned surveillance missions, by using wireless sensor networks, is of great importance for the military.

The authors of [24] present a sensor network with up to 91 nodes intended to collect votes from congress participants. The network allows users to respond to questions by pressing either a yes or a no button on a remote and determines the result of the vote. Users may change their response over time, requiring a continuous tally. The measurements were performed in a laboratory environment on a grid topology.

Mesh Networks

The purpose of wireless mesh networks is to provide multi-hop access to the internet. They are composed of stationary nodes equipped with radio hardware and connected to the main power supply system.

There are many instances of wireless mesh networks set up within cities to provide private access to the Internet. These are private networks run by wireless enthusiasts. The wireless nodes are used to connect individuals who then share an Internet connection at one of the nodes via a traditional means such as ADSL. Because there are many more users sharing the link to the Internet, users are able to benefit from the economy of scale and get better prices on their Internet connection.

A good example of a wireless mesh network is Melbourne Wireless Inc, which describes itself as “a non-profit group aiming to establish a fast, free, metropolitan area wireless network based on existing off-the-shelf 802.11 based equipment” [25]. There are more than 100 operational nodes in the Melbourne wireless network. Other examples of mesh networks are the wireless network in the city of Leiden [26] and the South African Wireless User Group (WUG) [27].

There are also examples of wireless mesh networks set up for scientific purposes. The MIT roofnet project [28], the Champaign-Urbana Community Wireless Network (CUWiN) [29] and SMesh at Johns Hopkins University [30] are examples of these. They are all used for investigating various aspects of wireless ad hoc networks such as routing protocols, medium access and physical layer radio propagation.

Some interesting examples of wireless mesh networks are the wireless mesh network deployed in downtown New Orleans to support surveillance cameras [31]. After Hurricane Katrina struck, this was the only communications network still working. It was used for VoIP communications and broadband access while the cities communications infrastructure was rebuilt. Wireless mesh networks inherent decentralized design results in these types of networks being tremendously resilient.

The city of Portsmouth bus service implemented a wireless mesh network which informs passengers in real time on the expected arrival and departures times of their buses [32]. On board devices collect real time information such as speed and location of over 300 buses. The data is then uploaded to a centralized server which distributes it to 45 LCD displays around the city allowing passengers to see whether their bus is on time or not. The system is part of a city wide effort to promote public transport and reduce carbon emissions.

Commercially available mesh products [33] [34] are available today and an amendment to the IEEE 802.11 standard is dedicated to enabling mesh networking. IEEE 802.11s is a draft IEEE 802.11 amendment for mesh networking, defining how wireless devices can interconnect to create a WLAN mesh network, which may be used for static topologies and ad-hoc networks [35].

The One Laptop Per Child (OLPC) project included a 802.11s wireless network card in the XO-1 laptop so that children benefitting from the program will be able to connect to one another and to the Internet and greatly enhance their learning experience.

MANETs

One means of conducting research on MANETS is via simulations. Popular simulation packages are ns2, OPNET, GlomoSIM and QualNet.

Some experiments set up real world MANET test beds. In [36] a network was set up which was a combination of up to 5 ground nodes, 3 UAV nodes and 2 laptop-based nodes in a seven square kilometre flat outdoor area. In [37] the experimenters set up a 33 node MANET on a rectangular athletic field of size 225m-by 365m using notebook computers. The simulation ran for an hour testing four routing algorithms for 15 minutes each. This is one of the largest outdoor tests of wireless routing ever conducted.

The only existing testbed for mobile ad-hoc networks used to a larger extent is the ad-hoc protocol evaluation testbed (APE) [8]. APE is a Linux distribution which can be booted from CD on notebooks. In [38] experiments with up to 37 nodes are reported.

A good example of a real world MANET is the CENTIBOTs project [39]. Funded by DARPA, the CENTIBOTs project is aimed to design, implement, and demonstrate a computational framework for the coordination of very large robot teams, consisting of at least 100 small, resource-limited mobile robots (CENTIBOTs), on an indoor reconnaissance task. The project was designed to investigate the collective behaviour of robots and their ability to perform tasks as a team autonomously. To do this they need to communicate with one another so that they can coordinate their resources to accomplish their task.

2.1.3. Routing Protocols in Wireless Ad Hoc Networks

The problem of routing data through an ad hoc network is significantly more complex than that of routing data through a fixed infrastructure based network. The changing topology of the network requires that routing protocols be able to manage and adapt routes in real time; the limited resources of the mobile nodes, both in terms of battery power and network bandwidth, require routing protocols to be very efficient; and the vast range of applications ad hoc wireless networks can be used for places application specific requirements on routing protocols such as QoS service level agreements or minimum latency for data through the network. Along with all of these requirements is the need for routing protocols to be scalable as the number of nodes grows.

A number of important characteristics of routing protocols are outlined below [40].

- Routing protocols must be distributed: ad hoc networks are by definition decentralized. Routing protocols must be implemented on individual nodes in the network without a centralized authority, and the protocol must be scalable. This means that the protocol will continue to work as the number of nodes involved increases.
- Freedom from broadcast storms and routing loops: a broadcast storm is caused by a single routing update triggering all nodes in the network to send out their own routing updates which in turn triggers more nodes to send out more updates and so on. This occurs in RIPV2 networks and will utilize a lot of resources on the part of the nodes for no

benefit. Wireless ad hoc networks cannot afford the expenditure of resources due routing loops and broadcast storms.

- Efficient use of bandwidth: the routing updates which are necessary for a routing protocol to work consume network bandwidth and node battery power needed for the transmission of other data. It is important to minimize the overhead of a routing protocol by minimizing the number of routing updates and other control packets necessary for its operation.
- Fast route convergence and adaptability: ad hoc wireless networks present a very dynamic system and routing protocols need to be able to react quickly to the changes that occur in the network topology.
- Optimization of network metrics: the quality of a network can be measured in many different ways depending on the purpose of the network. Certain routing protocols can be designed to optimize certain metrics. There is often a trade-off between network metrics. An example could be the network latency vs. throughput. At a certain point, increasing the network throughput could have an adverse effect on the network latency and the design of the routing protocol must be taken into account to deliver the required performance.

It is important for wireless ad hoc routing protocols to avoid the condition of a network becoming partitioned. We refer to the network being partitioned when there are islands of nodes which cannot reach other islands of nodes due to the absence of a route between them. The performance of the network declines rapidly once the network is partitioned because the packets from nodes in one partition are undeliverable to nodes in other partitions.

Routing protocols designed for traditional fixed infrastructure networks have been applied to the ad hoc routing problem along with entirely new routing schemes developed specifically for the ad hoc environment. The result is that wireless ad hoc routing protocols can be categorized into 3 types.

- Table based protocols where each node sends periodic updates to inform other nodes in the network of its own routing table.
- On demand protocols where a route discovery procedure is performed when a route to a destination node is needed.
- Hybrid protocols which are a mix of both table based and on demand protocols.

All ad hoc routing protocols can be characterized as being one of the above. Much research has been conducted on developing specialized routing protocols for certain conditions and applications of ad hoc networks. The protocols that we are interested in for the purposes of this thesis are those dealing with energy efficient routing.

First we describe the most common routing protocols found in the literature. We then describe some of the routing protocols that deal with energy optimization in wireless ad hoc networks. Energy is a critical resource in wireless ad hoc networks and is very important to preserve for overall network performance.

2.1.4. Table Based Protocols

The table based protocols are modified forms of the fixed line routing protocols, adapted to meet the more stringent conditions of ad hoc networks. Traditional routing protocols fall into two categories. The first category include the distance-vector routing protocols such as RIP V2 and EIGRP. These protocols use the number of hops from the source to the destination as the only routing metric to determine which route data should travel through a network. The second category are the link-state routing protocols such as OSPF and IS-IS which use the number of hops as well as the bandwidth of the links to measure the “least cost” route data should take to traverse a network.

Link-state routing protocols require each router to keep an updated map of the topology of the network, a task which is seen as too burdensome for nodes in an ad hoc wireless network. The table based routing protocols used in ad hoc networks are derived from distance vector protocols. The problem is that RIP V2 is susceptible to the formation of temporary routing loops and a counting to-infinity problem. Both of these conditions result in an excess of packets exchanged on the network as the routing protocol resolves the routing loop. This is acceptable in traditional fixed line networks, but in the wireless ad hoc environment bandwidth is too scarce a resource to be able to sustain temporary routing loops.

Destination-Sequence Distance-Vector (DSDV)

Destination-Sequence Distance-Vector (DSDV) is an example of a table based routing protocol. *DSDV is an adaptation of the traditional distance vector routing protocol with the enhancement of sequence numbers in routing updates to avoid temporary routing loops [41].*

DSDV utilizes the Distributed Bellman-Ford (DBF) algorithm to route packets through the network. *The DBF algorithm computes single-source shortest paths in a weighted digraph [42].* Each node keeps a routing table which specifies the direction (which neighbour) and distance (number of hops) towards any other node in the network. A packet being routed through the network is routed at each individual node to the best “next hop” which will take it towards its destination.

The routing tables are populated with information from periodic and triggered routing advertisements which the nodes distribute through the network. The mechanism for this is that each node is required to advertise its current routing table to each of its neighbours. Each of the neighbours will in turn advertise their routing table to their neighbours and so the routing information for any particular node will propagate through the network. The routing table specifies the next hop towards the destination as well as the number of hops required to reach the destination. The number of hops is the simplest routing metric to use, but it is not the only routing metric.

Routing updates are sent out periodically or upon a triggered event such as a link going down or an update to a node’s routing table. The frequency of these routing advertisements must be such that the routing tables of all nodes are up to date. In traditional distance vector routing protocols, temporary routing loops are known to occur in a network due to nodes updating their routing table information based on out of date routing advertisements. Mobile nodes will receive the same routing

advertisements many times as there are many possible routes for the advertisements to propagate through the network, and if a node updates its routing table based on an advertisement packet that has taken a very long time to reach the node, it can cause a broadcast storm of routing packets and adversely affect the network performance.

Routing loops occur in networks when outdated routing information persists. Consider two nodes X and Y on opposite ends of a mobile ad hoc network. Initially node X will send out routing update 1 specifying that it has a pathway to network Z. This routing update will propagate through the network towards node Y via multiple paths. Once the routing update reaches node Y via path 1, node Y will update its own routing table to specify that network Z is up and reachable. Now, should network Z go down, node X will again send out a triggered routing update 2 which will propagate through the network towards node Y. This update will again reach node Y very quickly via path 1 and inform node Y that network Z is no longer reachable. After this event has occurred, the original routing update 1 specifying that network Z is reachable arrives at node Y via path 2. Node Y updates its own routing table with this information and believes that network Z is reachable, when in reality it is not. Because there are no sequence numbers in the routing updates, node Y has no way of distinguishing between new and out dated routing updates.

DSDV includes in the routing advertisement a sequence number generated by the originating host. This sequence number enables mobile hosts receiving routing advertisements to determine whether a particular route is out of date compared to a previous route it may have received. A routing table will only update a route if it has the most recent sequence number attached to it, or if it has the same sequence number and the routing metric is lower than the currently held route. Routes with older sequence numbers are discarded.

Perkins et al. give the following account for the loop free nature of DSDV in [41]:

“At all instants, the DSDV protocol guarantees loop free paths to each destination. To see why this property holds, consider a collection of N mobile hosts forming an instance of an ad-hoc style network. Further assume that the system is in steady-state, i.e. routing tables of all nodes have already converged to the actual shortest paths. At this instant, the next node indicators to each destination induce a tree rooted at that destination. Thus, routing tables of all nodes in the network can be collectively visualized as forming N trees, one rooted at each destination. In the following discussion, we’ll focus our attention on one specific destination x and follow the changes occurring on the directed graph $G(x)$ defined by nodes i and arcs (i, p_i^x) where p_i^x denotes the next-hop for destination x at node i. Operation of DSDV algorithm ensures that at every instant $G(x)$ is loop-free, or rather, it is a set of disjoint directed trees. Each such tree is rooted either at x or at a node whose next-hop is nil. Since this property holds with respect to each destination x, all paths induced by routing tables of DSDV algorithm are indeed loop free at all instants.”

This modification of DSDV to the classical DBF algorithm ensures that there is no occurrence of routing loops in the network and this is a critical requirement of an ad hoc wireless networking protocol.

Periodic routing advertisements will not update all of the routing tables in the network fast enough for the routing protocol to respond effectively to changes in the network topology (a link between two mobile nodes going down due to the movement of one of the mobile nodes). As well as periodic routing updates, whenever a mobile node detects that a link has gone down it will immediately broadcast a routing information packet update. The routing information packet will assign an infinite metric (any value that is greater than the maximum allowed metric) to the failed link to indicate that this link is no longer valid.

There are two types of triggered updates that can be sent out upon the detection of a topology change. The first is called a “full dump” and will carry all the available routing information. The second is called an “incremental dump” and will only carry the information changed since the last full dump. When movement in the network becomes frequent and the amount of information in an incremental dump would take up more space than a single network protocol data unit (NPDU), a full dump is sent instead. The incremental dump is used to minimize the protocol overhead on the network.

Damping fluctuations are also used to minimize the number of protocol administration packets on the network. This involves any particular node waiting a specific amount of time before advertising a new route due to the possible imminent arrival of another route with the same sequence number and a better routing metric.

DSDV is a simple enhancement of one of the most prevalent and successful routing protocol mechanisms in infrastructure based networks. The most important change to DSDV is the addition of destination sequence numbers to all of the routes. This guarantees that the network will be loop free at all times. The greatest advantage of DSDV is that it is simple to implement and does not require a great deal of processing on the part of the mobile nodes.

The disadvantages of DSDV are that it requires periodic routing updates which immediately result in a higher protocol overhead compared to other on demand protocols. It also requires the fine tuning of some parameters that can be difficult to select as their optimum values will change depending on the size of the network. These parameters are the periodic update interval, the maximum value of the “settling time” for a destination and the number of update intervals which may transpire before a route is considered “stale”.

2.1.5. On Demand Protocols

On demand routing protocols have a lower overhead. In this form of routing protocol, a route is only “discovered” when it is needed. If node A has a packet to send to node B, it will first check if it has routing information for that node. If it does not, it will initiate a route discovery procedure. The route discovery procedure consists of broadcasting a discovery packet which will propagate through the network informing intermediate nodes that node A is seeking a route towards node B. If any intermediate nodes have a route to node B, they will respond to the route discovery with a route reply informing the source node of their route. If intermediate nodes do not have the route in their own table, they will forward the packet. The packet will eventually propagate to the destination node which will send back a route reply. The source node will aggregate all the route replies that it receives and

select the best route for routing packets towards the destination. All intermediate nodes that forward discovery and reply packets during the exchange will also be able to learn routing information from the exchange.

DSR and TORA are examples of on demand protocols.

Destination Source Routing (DSR)

DSR is a routing protocol that works on the principle of the source node specifying the complete route (all hops along the path towards the destination) as opposed to the packets being forwarded to the "next hop" according to each intermediate node's routing table [43] [44] [45].

Source routing is a routing technique in which the sender of a packet determines the complete sequence of nodes through which to forward the packet; the sender explicitly lists this route in the packet's header, identifying each forwarding "hop" by the address of the next node to which to transmit the packet on its way to the destination.

There are no periodic route advertisements in the protocol. Instead, when a host needs a route to another host, it dynamically determines one based on cached information or on the results of a route discovery procedure.

Each node in the network maintains a route cache of available routes to all destinations in the network. If a packet needs to be sent to a certain node, the source node consults its route cache. If a route is found, the packet is sent with the list of nodes it needs to traverse listed in the header. If there is no available route in the route cache, the source node initiates a route discovery procedure to find a route to the destination node. This route will then be stored in the node's cache for the next time it needs to send a packet to that particular destination. Each route has associated with it an expiration period after which the entry is deleted from the cache.

The route discovery procedure consists in the source host broadcasting a route request packet. The route request packet contains the destination node for the route request as well as the source node and a route record detailing the intermediate nodes that the packet has traversed. When the route request packet reaches the destination, its route record will record the route by which packets need to be sent from the source to the destination. Each route request packet also contains a request id set by the source node. This is to ensure that old routing requests do not confuse the routing protocol. Only recent and up to date route requests are propagated through the network.

When the destination receives a route request, it responds with a route reply packet detailing the path that was followed to get to the destination. The route reply can be sent in one of three ways. If the destination has a route available in its route cache back towards the source, it can send the route reply via this route. If it does not have a cached route available, it can reverse the route record accumulated in the route request packet and hope that the intermediate links between nodes are symmetrical (that is to assume that if a node A can reach an adjacent node B, then the adjacent node B will also be able to send data to the original node A). The last option is for the route reply message to be piggy-backed on another route request message.

When a node has a route to a destination, it will also have a route to each one of the intermediate nodes along the route. Their routes will simply be the original route truncated at that particular node. Also, if the nodes have their NICs configured for promiscuous mode, they can gather routes from any other route request or route response packets that they overhear. By populating their own routing tables with this information they will reduce the amount of protocol that would be used to discover routes. Each node also maintains the operation of each of its routes. If a particular node moves in such a way that a link along a route is broken, the node will detect this and initiate another route discovery procedure to find another route to the destination.

The advantages of DSR are that it does not require periodic advertisements. This reduces the protocol overhead in the network and conserves the battery life of individual nodes as they do not have to participate in regular updates. The nodes can go into a sleep state if there is no traffic on the network. DSR is also very quick to react to changes in topology - at least for small networks (50 nodes). The process of finding a route is direct and rapid and only involves the nodes that are involved in the routing of the data.

The disadvantages of DSR are that it may not be scalable for networks involving very many nodes. The reason for this is that the entire route is stored in the message header. As the number of nodes in a network grows and the size of the routes subsequently grows with it - so too will the size of the packets transmitted on the network.

Temporally-Ordered Routing Algorithm (TORA)

TORA is an on demand routing protocol using a “link reversal” algorithm to direct traffic towards a destination node.

TORA is one of a family of protocols using a “link reversal” algorithm [46]. The protocol discovers multiple routes on demand and considers the optimality of the route as being of secondary importance when compared to the speed with which routes are established and minimizing the protocol overhead to do so. Essentially the action of the protocol is to give links between certain nodes the correct orientation such as to route data from the source to the destination along those links.

The protocol performs three basic functions. Route creation, route maintenance and the erasing of routes. The protocol accomplishes these functions through the use of three control packets. Query (QRY), update (UPD) and clear (CLR). The on demand property of the protocol means that it only creates routes when a source has data to transmit to a destination. There is no need for periodic updates and this significantly reduces the protocol overhead. The analogy for the protocol is that of water running from a source to a destination. The role of the protocol is to adjust the “height” of the nodes such that the data (water) flows from the source to the destination.

It does this by propagating a “temporally-ordered sequence of diffusing computations”. The algorithm runs on all nodes causing them to alter their own height depending on their interactions with neighbouring nodes. The result is a set of correctly directed links between nodes that data can flow through to reach its destination. There is no concern in the algorithm for finding an optimal path from

source to destination. Instead, the algorithm creates as many viable paths as possible for the data to flow through. This results in an increased effective bandwidth between the source and the destination than if just a single optimal path was configured.

"A logically separate version of the protocol is run for each destination to which routing is required" [46]. This means that each node will have a number of instances of the protocol running on it, one for each destination it is routing towards. The height of any node is related to the destination it is routing towards, and any node can have many "height" values, one for each destination. The direction of the links between neighbouring nodes is also related to the destination so that each destination node results in a certain orientation for each link. Packets are identified by their final destination and are routed along the links accordingly.

"At any given time, an ordered quintuple $H_j = (\tau_i, oid_i, r_i, \delta_i, i)$ is associated with each node $i \in N$. Conceptually, the quintuple associated with each node represents the height of the node as defined by two parameters: a reference level and a delta with respect to the reference level. The reference level is represented by the first three values in the quintuple while the delta is represented by the last two values. The first value representing the reference level, τ_i , is a time tag set to the "time" of the link failure. . The second value, oid_i , is the originator-ID (i.e. the unique ID of the node which defined the new reference level). The third value, r_i , is a single bit used to divide each of the unique reference levels into two unique sub-levels. This bit is used to distinguish between the original reference level and its corresponding, higher reflected reference level. The first value representing the delta, δ_i , is an integer used to order nodes with respect to a common reference level. This value is instrumental in the propagation of a reference level. How δ_i is selected will be clarified in a subsequent section. Finally, the second value representing the delta, i , is the unique ID of the node itself." [46].

Some means of synchronization is necessary on the network, the precision of which depends on the rate of change of topology. For instance, if the average time between link failures is on the order of minutes, then achieving a clock synchronization on the order of seconds is very likely sufficient.

The advantages of TORA are that it is an on demand protocol and that route creation is only required when the source has data that it wishes to transmit. The protocol also simplifies the reaction to topological changes by localizing the reaction to changes to a small set of nodes around the point of change. It is not necessary for control packets to propagate throughout the network and effect change in a hierarchical fashion.

TORA is a scalable algorithm as it executes in a distributed fashion. Each node only needs to keep information about adjacent nodes. The protocol is loop free and typically establishes many routes between a source and destination.

A problem with other link reversal algorithms is that they often cannot detect partitions in the network. TORA has been designed to erase routes so that upon detection of a network partition, all links (in the portion of the network which has become partitioned from the destination) must be undirected to

erase invalid routes. This makes TORA far more efficient at reacting to route failures than other link reversal algorithms.

2.1.6. Hybrid Protocols

Hybrid protocols are a combination of the periodic table based and on demand routing protocols.

Ad Hoc On Demand Distance Vector (AODV)

The AODV routing protocol is a combination of DSR and DSDV [46] [47] in that it combines the on demand route request and route maintenance mechanism from DSR and the hop by hop routing with sequence numbers from DSDV.

When a node needs to transmit data to a destination, it queries its routing cache. If the route is not there, it initiates a route request as in DSR. The route request is flooded through the network until it finds either the destination or an intermediate node that has a route to the destination. As each node forwards the route request packet, it sets up a temporary reverse route. This temporary reverse route is simply one hop back towards the node that it received the original route request from. This temporary route is sustained long enough for the route request to traverse the network and potentially produce a route reply back towards the source.

Should the route request packet find the destination, it will respond with a route response. This will be propagated back to the source along the reverse route recorded during the route request. Eventually, the route response will return to the source and a route will be set up between it and the destination. This route is not stored in the node's route cache as a list of nodes the packet must traverse, as it is in DSR. The route is recorded as a metric and the next hop towards which packets must be sent if they are to reach the destination. The entire route is stored in all the intermediate nodes between the source and destination in that their own routing tables have been configured with the "next hop" to route packets.

Distance metrics are associated with each next hop so that different routes can be compared and the best route chosen. And as in DSDV, a sequence number is associated with each route to avoid the formation of routing loops in the network. Links between nodes are monitored by either regular hello packets or link layer acknowledgements. Perkins [48] found the link layer mechanism to be the most effective means of maintaining links. When a link is detected as being down, an unsolicited route reply message is propagated upstream of the broken link with an infinite metric value for that link. Nodes upstream using that link can initiate another route discovery procedure to regain a route to the destination.

AODV is an on demand protocol and thus minimizes the protocol overhead in the network by avoiding using regular updates packets. The protocol is loop free by utilizing destination sequence numbers as in DSDV and is more scalable than DSR as the packet header does not grow as the number of nodes within routes and within the network increase.

2.1.7. Energy Conserving Routing in Wireless Ad Hoc Networks

Energy conserving routing attempts as its primary objective to find the most energy efficient routes within an ad hoc network. Ad hoc networks are usually constrained in terms of energy. It is one of the most important resources to conserve.

Minimum Total Transmission Power Routing (MTPR)

If nodes are able to adjust their transmission power, they can select routes for which the total energy required for a packet to be delivered can be minimized. MTPR is an example of this. The total power is calculated for each route by adding the amount of power required at each hop, and then the route that requires the least amount of power is selected [40]. This is not strictly true, as the routes that require the least amount of power usually result in the most hops. This is due to the fact that power is dependent on the distance between nodes. Two short transmissions take considerably less power than one long transmission. However routes that consist of more hops are more prone to failures and introduce more network delay.

To overcome this problem the cost of receiving a packet at the receiving nodes is taken into account and added to the route metric. This increases with the number of hops and thus results in shorter routes than if the transmit power were considered alone.

MTPR is a useful means of conserving the overall energy of the network, but it has a major shortcoming in that it does not consider individual nodes energy levels at all. An unintended side effect of MTPR is that it causes the batteries of nodes along the most energy efficient routes to be depleted quicker, which in turn results in network partitioning earlier and minimizes the overall lifetime and throughput performance of the network. The routing protocol finds the most energy efficient routes and then sends the majority of traffic along these routes. Because nodes that are positioned along the most energy efficient routes handle more traffic, they tend to use up their batteries sooner and the overall performance of the network suffers.

A more effective metric for network performance is rather to conserve the lifetime of the network. This leads to the next protocol in the field of energy aware routing.

Minimum Battery Cost Routing (MBCR)

Minimum battery cost routing attempts to prolong the overall lifetime of the network. This is achieved by selecting routes according to the amount of battery resources available along the route. It no longer matters which is the most energy efficient route available, but rather which route does the network have the most resources to support.

Consider the energy metric $f_i(c_i^t)$ as the battery cost function of node n_i . One possible choice for $f_i(c_i^t)$ is [40]

$$f_i(c_i^t) = \frac{1}{c_i^t}$$

As the battery capacity of node i decreases, so the value of the cost function $f_i(c_i^t)$ will increase. The battery cost for route R_j is defined as

$$R_j = \sum_{i=0}^{D_{j-1}} f_i(c_i^t)$$

The route battery cost consists of the sum of the battery costs of all nodes along that route.

Since battery capacity is directly incorporated into the routing protocol, this metric prevents nodes from being overused, thereby increasing their lifetime and the time until the network is partitioned [49]. This results in a more “fair” distribution of routes, and thus traffic, across the network and increases the overall network lifetime as nodes in the centre of the network are less likely to run out of power before other nodes. The result is that the battery utilization of all nodes is more equitable.

The protocol however does not consider individual node’s battery capacity, but rather the battery capacity of an aggregated route. It is still possible that those routes containing nodes with very little battery capacity will be selected, and that these individual nodes will run out of battery capacity before other nodes in the network. The next routing protocol takes this into consideration.

Min-Max Battery Cost Routing (MMBCR)

To ensure that no node is overused, a route can be selected according to the node along that route that has the lowest battery. The battery cost R_j for route j is defined as [40]

$$R_j = \max_{i \in \text{route}_j} f_i(c_i^t)$$

Similarly, the desired route i can be obtained from the equation

$$R_i = \min\{R_j | j \in A\}$$

Where A is the set containing all possible routes.

The route will be selected according to the minimum battery capacity of one of the nodes along the route. The route selected is not the most energy efficient route. The result of this protocol is that nodes in the network with the least energy will be protected.

Summary of Energy Aware Routing Protocol Performance

Energy aware routing mechanisms have been shown to improve the overall lifetime of the network [50] [40] [49]. This does not mean that the performance of the network is necessarily improved for other performance metrics. Often the extension of the lifetime of the network will result in poorer performance of other network metrics such as throughput and latency. It is important to consider the purpose and design requirements of the network when selecting a routing protocol. There are many examples of wireless ad hoc networks in which energy is a critical resource and needs to be optimised above all other metrics. The lifetime of the network as a whole is obviously a critical metric to maximize as once the network is partitioned or ceases to function, the wireless ad hoc network and the application using it cannot fulfil its purpose.

2.2. Mechanisms for Stimulating Fairness in Wireless Ad Hoc Networks

Wireless ad hoc networks require nodes to forward traffic on behalf of other nodes. Along any route, the intermediate nodes forwarding packets on behalf of the source and destination expend their own limited resources and derive no benefit from doing so. This might not be advisable when considered from an individual node's point of view, yet the willingness of nodes to forward traffic on behalf of one another is critical to the operation of the network.

There are two contexts in which it is necessary to stimulate the cooperation of nodes. The first context is that of a network in which individual nodes do not belong to the same authority and are acting on their own behalf. The second context is that of a network in which all nodes belong to the same authority and are programmed to act on one another's behalf. The means for encouraging cooperation amongst nodes can be split into two categories. There are reputation based systems and credit based mechanisms [51]. These are detailed below.

2.2.1. Reputation Mechanisms

Reputation mechanisms deal primarily with the case in which individual nodes do not belong to the same authority. In these types of networks, each node needs to be stimulated to forward packets on behalf of other nodes because they do not have any inherent reason for cooperating.

Trust and reputation mechanisms for avoiding non-forwarding nodes have been proposed in [51] [52]. These enable networks to improve their throughput by avoiding individual nodes which do not participate in forwarding. The non-forwarding nodes receive a bad reputation and as result are not assisted by any nodes in the network.

This provides an incentive for a node not to be "selfish" and in effect stimulates cooperation. The reputation based mechanism is also a defence against hostile nodes attempting to sabotage the network performance by taking part in the route discovery mechanism and then not forwarding packets.

2.2.2. Credit Based Mechanisms

The second means of stimulating cooperation is by means of credit based incentive mechanism [53] [2] [3]. This method of stimulating cooperation can be used in both types of network contexts mentioned previously.

Nuglets

For networks in which all nodes are self-interested units and do not report to a centralized authority, a virtual currency called "nuglets" is proposed [2]. In the "packet purse" model, intermediate nodes earn nuglets by forwarding packets on behalf of other nodes and source nodes are required to spend nuglets in order to send traffic. Before sending a packet, the source node must estimate the number of hops required to reach the destination. The appropriate amount of nuglets is then placed into the "packet purse" and as the packet traverses the various nodes on the route, each node will extract a

“payment” for the service rendered. If a node does not have sufficient nuglets to pay for a certain route, then it cannot send that packet. In this way, if nodes do not cooperate they will not be able to send their own traffic.

A tamper resistant security module is necessary to prevent nodes from dealing disingenuously with nuglets. This would be similar to a GSM SIM card and would be required by any node to gain access to the network. Once the node is authenticated on the network, all nuglet transactions would need to be verified through the SIM module.

A “packet trade model” is also outlined as an alternative to the “packet purse” model. In this case, a packet purse starting at zero at the source node accumulates nuglets along the route. Once the packet reaches the destination node, the nuglets accumulated must be paid for if the destination is to receive the packet. If the destination node does not have enough nuglets to pay for the delivery of the packet, then the packet is dropped at the destination. This scheme is wasteful in that the resources of the network are used to deliver the packet all the way along the route, before the decision is made as to whether or not it can be delivered.

Optimized Throughput by Means of Incentives

For networks in which the nodes are answerable to a centralized authority, there is another reason to stimulate cooperation. In this context, we do not need to worry about the possibility of a selfish node sabotaging the network, but rather how we can maximize the trade-off decision individual nodes need to make between forwarding packets on behalf of other nodes and conserving their own resources.

How does an intermediate node choose to expend its own resources on behalf of other nodes? How does a network of nodes maximize the trade-off between conserving the network resources and the vital function of forwarding traffic without which the network will fail?

Usually the default option is for all nodes to forward all packets on behalf of other nodes regardless of their own resources. But this is not necessarily the most efficient solution or the most effective use of resources. If nodes do not discriminate and instead forward all packets, this can lead to a situation in which the nodes near the centre of the network that are likely to be present in many routes end up expending their resources quicker than nodes at the edge of the network. The central nodes will run out of battery earlier and the network will be partitioned even while there are unutilized resources at the edge of the network.

In [3], pricing mechanisms provide a means of guiding a network to its optimal operating point. The concept of setting up a pricing mechanism that will optimally distribute resources is investigated by means of fluid flow simulations. A pricing model is based on the two critical resources of a node, namely the bandwidth and the energy. Similarly to the model in [2], source nodes are charged to send packets and intermediate nodes are able to earn credits. The prices for transmitting a packet across a link depend on the bandwidth and energy of the transmitting node. Intermediate nodes are rewarded for forwarding traffic, thus creating incentives for collaboration [54]. The rewards at each link

are continuously adjusted such that the network converges to an operating point that maximizes the overall data throughput rates.

The authors of [4] use the idea of a credit mechanism to create several models to optimize network performance. The models seek to optimize the fair usage of network resources by addressing the problem in which centralized nodes expend too many resources forwarding packets on behalf of other nodes. There are three basic incentive schemes presented. The first does not regulate the willingness of nodes to forward packets on behalf of other nodes at all. It is called free-for-all. The second scheme dictates that a node shall not forward a packet on behalf of the community (defined as all the other nodes) unless it has received an equivalent amount of service. This scheme is called tit-for-tat.

The third represents a scheme between the previous two. It consists of a credit mechanism in which each node in the network is assigned a target credit balance. Source nodes spend credits to send packets while intermediate nodes charge credits each time they forward a packet. At a random time interval a credit redistribution mechanism takes place in which nodes with excess credit (more credit than the target credit balance) are taxed and nodes with deficient credit (less credit than the target credit balance) are granted extra credit. The credit redistribution mechanism ensures that the system tends towards a stable operating point and ensures that the situation in which the central nodes expend their resources quicker than edge nodes is less severe. There are four variants of the credit mechanism scheme listed below:

1. Source pays with constant pricing
2. Destination pays with constant pricing
3. Source pays with congestion related pricing
4. Destination pays with congestion related pricing

The credit mechanism is a means of distributing resources around the network. If a node does not contribute sufficiently to the performance of the network by forwarding packets on behalf of other nodes, it is not able to send its own packets. Nodes are incentivized to participate in sharing their own resources to improve the overall performance of the network.

By setting the target credit balance and the rate at which credits are redistributed, it is possible to control the degree to which the network enforces the fair use of resources. These parameters cause the credit mechanism to act along a range between the free-for-all and tit-for-tat schemes. If redistribution happens too frequently and the target credit balance is set too high, the network will tend towards free-for-all because there will be no situation in which nodes do not have enough credit to send their own packets. If the target credit balance is set too low and the redistribution mechanism does not happen frequently enough, only nodes that are earning credit will be able to send packets. The network will tend towards the tit-for-tat case.

The incentive schemes are shown in [4] to be capable of affecting the network performance and stimulating the fair use of resources in the network.

3. Methodology

3.1. Overview

Mobile ad hoc networks (MANETs) are often required to operate under energy constrained conditions. Examples of this would be to provide communications in disaster relief situations or to extend network coverage to areas where normal infrastructure based networks are not viable. One means of improving the performance of MANETs is to improve the efficiency with which the nodes energy resources are used. This section proposes various credit-based incentive schemes modelled on those in [4] to stimulate the fair use of energy resources across the network. These schemes are outlined in Section 3.2 and Section 3.3. Routing metric based schemes are an extension of the credit based schemes in that they implement principles of the credit-based schemes as well as intelligently route traffic through the network to enhance performance. The routing metric based schemes are outlined in Section 3.4. All schemes investigated in this thesis deal with the case in which nodes belong to the same authority.

All schemes are implemented in ns2. Ns2 is a discrete event simulator widely used for the purposes of investigating packet based networks. Ns2 provides a reliable foundation for simulating the performance of MANETs using the mac802.11 and DSDV protocols as these are standard libraries in the ns2 default package. The credit based mechanisms are implemented in the routing and transport layers of the individual nodes. All credit-based schemes are implemented using the standard ns2 DSDV library in which the routing metric is based on hop count. The routing metric based schemes are implemented in ns2 by making adjustments to the routing metric parameter in the ns2 DSDV library. By means of adjusting the routing metric parameter we are able to modify the properties of the DSDV routing protocol. The details of this are described in Section 3.5.

The performance of the network is measured by the Packet Success Probability (PSP), Throughput Rate Per Node (TRPN), Total Average Latency (TAL) and Energy Rate Per Node (ERPN). The fairness of the network is measured using the rate fairness and the reciprocity. These measures are described in Section 3.6.

The network topology is described in Section 3.7 and the experiments are described in Section 3.8.

3.2. The Basic Schemes

3.2.1. Free for All

The first incentive scheme is called free-for-all (FFA). FFA is defined as follows: *nodes do not discriminate on how much service they have received from or provided to the network. All nodes will relay all packets on behalf of all other nodes.*

In FFA the central nodes tend to expend their battery resources quicker than other nodes due to the fact that there is more transit traffic passing through them. The central nodes tend to shut down earlier and cause the network to partition.

The *FFA* protocol does not result in the fair use of energy resources across the network. Because there is no restriction on which nodes can send packets, we anticipate high packet loss at high packet arrival rates due to insufficient buffer space at the node packet queues or due to radio interference.

3.2.2. Tit for Tat

The second incentive scheme is called tit-for-tat (*TFT*) and it is the most stringent of all the credit-based incentive schemes. *TFT* is defined as follows: a node will only receive service from the network (all the other nodes) if it has performed an equivalent amount of service in return.

Let $prov_i$ represent the number of packets node i has serviced on behalf of the network and rec_i represent the number of packets that the network has serviced on behalf of node i . N_{margin} is a small positive constant and $cost$ is the number of times a packet will need to be received and forwarded by intermediary nodes before it reaches its destination.

A node is defined to perform a service on behalf of the source node if it receives a packet or forwards a packet.

In *TFT* a source node will only send a packet if the following condition is true:

$$prov_i + N_{margin} - rec_i \geq cost$$

For a route r hops long, the cost of transporting the packet from the source to the destination is $2r$. If the condition is not true and the node has not provided enough service to the network in comparison to the service it has received, the packet will be dropped. This restricts nodes that do not perform enough service on behalf of the network from sending their own packets.

The tit-for-tat algorithm

At the source $node_i$:

```
procedure TFT( ) begin
    if (provi + Nmargin - reci < cost ) {
        drop packet
    } else {
        if (queue full) {
            drop packet
        } else {
            queue packet
            reci = reci + cost
            provi = provi + 1
        }
    }
end
```

At the transit $node_j$:

```
procedure TFT( ) begin
    receive_packet{
        provj = provj + 1
    }
```

```

if (queue full) {
    drop packet
} else {
    forward_packet{
        provj = provj + 1
    }
}
end

```

At the destination $node_j$:

```

procedure TFT( ) begin
    receive_packet{
        provj = provj + 1
    }
}
end

```

Each time a node j receives a packet it increments $prov_j$ by one. Each time a node j forwards a packet it increments $prov_j$ by one.

Nodes at the edge of the network do not have many transit routes passing through them. As a result these nodes do not have many opportunities to perform services on behalf of the network. For this reason, $prov_j$ is low for these nodes and many packets arriving at these nodes are dropped due to the TFT restrictions. Nodes on the edge of the network tend to perform poorly in TFT .

3.3. The Credit Based Schemes

The credit-based schemes lie between FFA and TFT . We require a balance between FFA in which the central nodes are required to expend their resources unfairly and TFT in which the performance of the edge nodes is severely degraded. The credit-based schemes provide a means for controlling the degree to which nodes will serve one another and enable the network to achieve a more optimal balance between FFA and TFT . The current research proposes and investigates six variants of the basic credit-based scheme. The variations are due to changing the node responsible for payment of the service and changing the pricing of the service. The basic credit-based scheme and the six variations are described below.

3.3.1. The Basic Credit-Based Scheme

Each node is assigned a target credit balance. Each time a node sends or receives a packet it is considered to be performing a service. Nodes earn credit by performing services on behalf of other nodes. Nodes spend credit in order to send their own packets. In this way, nodes exchange credits for services resulting in a market economy.

The introduction of pricing raises the question as to which node should pay for the data transmission. Either the originating node or the destination node will pay the transit nodes on a multi-hop route to compensate the transit nodes for the use of their resources. In [2] both the origin pays and the destination pays were investigated. In practice, the decision as to who has to pay should depend

upon who benefits from a successful packet transmission. The originating node may benefit (consider a user who sends an email) or the destination node may benefit (consider a node that downloads regular stock exchange updates from a server) or both nodes may benefit.

Another important consideration is how much a node should pay for a service. One option is to have a constant pricing mechanism in which the cost of a service is fixed for all nodes. This makes the implementation of the mechanism simpler, but the prices do not necessarily reflect the availability of resources that are used to provide the service.

Congestion related prices are based on the packet queue at the node performing the service. Let n_i represent the maximum number of packets node i can have in its packet queue and $n_i(t)$ represent the number of packets at time t in the packet queue. The price of a service at node i is given by

$$\text{unit_cong_price} = n_i(t)/n_i$$

If the node is congested and has many packets in its queue the *unit_cong_price* will be close to 1. If the node is not congested and has few packets in its queue the *unit_cong_price* will be close to 0. This makes sense because a node that is not highly congested is not using a lot of resources, and therefore its resources should be cheaper than a node which is highly congested and using a lot of its resources.

It is also possible to create a unit price based on the node energy. Let $e_i(t)$ represents the energy available at node i at time t and e_i represents the initial energy of node i . The price of a service at node i is given by

$$\text{unit_energy_price} = 1 - e_i(t)/e_i$$

This is the inverse of the congestion related price because energy is a resource that we would want to conserve. The unit energy price is 0 at the beginning of the simulation when the instantaneous energy of the node $e_i(t)$ is the same as the initial energy e_i . As the energy of the node decreases the *unit_energy_price* will get closer and closer to 1. As the instantaneous energy of the node decreases the price of it performing a service increases.

As the network functions, some nodes will begin to accumulate credit as they earn more credit than they spend and other nodes will begin to lose credit as they spend more credit than they earn. A credit redistribution mechanism ensures stability of the system. The redistribution mechanism is a decentralised algorithm. If a node has credit in excess of the target credit balance, it will destroy a certain percentage of that excess credit. If a node has less credit than the target credit balance, that node will create a certain percentage of its deficit credit. The overall affect is that of a flow of credit from the wealthier nodes to the poorer nodes. It also provides a means for credit lost from the system due to dropped packets to be regained.

The redistribution mechanism works as follows. Consider a credit redistribution event at *node i* at time t . Let δ denote the discount rate in units of time^{-1} . Let Δ denote the time interval between successive discount events. If *node i* is selected at random then the interval between successive

discount events at *node i* will be random. Let B denote the target credit balance and let $B_i(t)$ denote the credit balance at *node i* at time t . At every redistribution event the credit at *node i* is adjusted as follows:

$$B_i(t + \Delta) = B_i(t) + \delta\Delta(B - B_i(t))$$

Thus nodes that possess an amount of credit that exceeds the target balance will destroy a fraction of the surplus per unit time, while nodes whose credit balance is less than the target credit balance will create a fraction of the deficit per unit time.

There are two variables to consider in the credit redistribution mechanism.

δ is the redistribution rate. It is the fraction of excess or deficit credit that must be created or destroyed per unit time, expressed as a percentage per second. If the discount interval is small, the discount rate will be adjusted appropriately per discount interval. If δ is too high, then the network will tend towards FFA as the redistribution mechanism will keep all nodes credit balances close to the target balance and all nodes will be able to send. If δ is too low, then the network will tend towards TFT as the redistribution mechanism will be ineffective and there will be a few wealthy nodes and many poor nodes that will be unable to send packets.

Δ is called the redistribution interval and represents the interval between successive redistribution events. If the redistribution events are not performed frequently enough, it will have no effect and the protocol will tend towards TFT behaviour. Only wealthy nodes will have enough credit to transmit packets while poorer nodes will have no credit and will not be able to transmit packets at all.

Credit redistribution ensures that under-provisioned nodes are able to send some traffic, while at the same time providing over-provisioned nodes a mechanism for disbursing their credits for the common good rather than accumulating them. Credit redistribution ensures that the system is stable and robust and that the total amount of credit converges to the sum of the target credit balances of all nodes. Nodes which enter or leave the network or credits which are lost to the system by dropped packets have a temporary impact on the performance of the network protocol.

A node that has a surplus of credit will destroy a fraction of that surplus over time, while a node that has a deficit will create a fraction of that deficit over time.

The six variations of the credit-based scheme are described below.

3.3.2. Origin Pays with Constant Pricing

The third incentive scheme is called origin-pays-constant-pricing (*OP-const*). *OP-const* is defined as follows: *the origin node is responsible for paying for the delivery of packets and all services are of constant cost at one unit per service*.

OP-const is implemented as follows:

Each node is assigned a target credit balance. If a node needs to send a packet to a destination r hops away, it will need to have $2r$ credits. This is the amount of credit it will cost to send the packet

because each hop along the path will deduct 2 credits. Each intermediary node will deduct 1 credit for receiving the packet and 1 for forwarding it on. The source node will deduct 1 credit from itself for sending the packet and the destination node will charge 1 credit for receiving the packet.

If the source node does not have enough credit, it will drop the packet and not expend any resources on sending it.

The origin-pays-constant-pricing algorithm

At the source $node_i$:

```
procedure OP-const( ) begin
    route_cost = 2r
    if ( nodei.credit < route_cost ) {
        drop packet
    } else {
        if ( queue full ) {
            drop packet
        } else {
            queue packet
            nodei.credit = nodei.credit - route_cost
            packetpurse = packetpurse + route_cost
            forward packet
            nodei.credit = nodei.credit + 1
            packetpurse = packetpurse - 1
        }
    }
end
```

At the transit $node_j$:

```
procedure OP-const( ) begin
    if ( packetpurse < 1 ) {
        drop packet
    } else {
        receive packet {
            packetpurse = packetpurse - 1
            nodej.credit = nodej.credit + 1
        }
    }
    if ( packetpurse < 1 ) {
        drop packet
    } else {
        if ( queue full ) {
            drop packet
        } else {
            forward packet{
                packetpurse = packetpurse - 1
                nodej.credit = nodej.credit + 1
            }
        }
    }
}
```

```

    }
end

```

At the destination $node_k$:

```

procedure OP-const( ) begin
    if ( $packet_{purse} < 1$ ) {
        drop packet
    } else {
        receive packet {
             $node_k.credit = node_k.credit + packet_{purse}$ 
        }
    }
end

```

If the originating node has enough credit it will place the required number of credits into the packet purse and each time the packet is serviced by a node, the correct amount of credit will be deducted as payment for the service. The node that performs the service on the packet will add the deducted credit to its own credit balance. This credit will be required when it needs to send its own packets.

When the packet reaches the destination node it should have 1 credit left in its packet purse to pay for the last service which is the reception by the destination node. If there is any excess credit, due to the route length changing (becoming shorter) while the packet was in transit, the excess credit is passed onto the destination node.

The credit redistribution mechanism is performed in a decentralised manner.

3.3.3. Origin Pays with Congestion Based Pricing

The fourth incentive scheme is called origin-pays-congestion-pricing (*OP-cong*). *OP-cong* is defined as follows: *the origin node is responsible for paying for the delivery of packets and the cost of a service is based on the congestion at the servicing node*. This scheme is very similar to *OP-const*. The only difference is that the price of a service is related to the congestion at the node performing the service. Congestion prices are based on the number of packets in a node's queue. They are calculated as follows:

Let n_i represent the maximum number of packets node i can have in its packet queue and $n_i(t)$ represent the number of packets at time t in the packet queue. The price of a service at node i is given by

$$unit_cong_price = n_i(t)/n_i$$

The *unit_cong_price* varies from 0 for uncongested nodes to 1 for highly congested nodes.

OP-cong is implemented as follows:

Each node is assigned a target credit balance. If a node needs to send a packet to a destination r hops away, it will need to have a maximum of $2r$ credits. This is the amount of credit it will cost to

send the packet if each $unit_cong_price$ at each node is 1. The total cost of sending the packet will be less than $2r$ credits depending on the level of congestion at the transit nodes. The excess credit remaining in the packet purse will be passed onto the destination node.

If the source node does not have enough credit it will drop the packet and not expend any resources on sending it.

The origin-pays-congestion-pricing algorithm

At the source $node_i$:

```
procedure OP-cong( ) begin
    unit_cong_price =  $n_i(t)/n_i$ 
    route_cost = 2r
    if ( nodei.credit < route_cost ) {
        drop packet
    } else {
        if ( queue full ) {
            drop packet
        } else {
            queue packet
            nodei.credit = nodei.credit - route_cost
            packetpurse = packetpurse + route_cost
            forward packet
            nodei.credit = nodei.credit + unit_cong_price
            packetpurse = packetpurse - unit_cong_price
        }
    }
end
```

At the transit $node_j$:

```
procedure OP-cong( ) begin
    unit_cong_price =  $n_j(t)/n_j$ 
    if ( packetpurse < unit_cong_price ) {
        drop packet
    } else {
        receive packet {
            packetpurse = packetpurse - unit_cong_price
            nodej.credit = nodej.credit + unit_cong_price
        }
    }
    if ( packetpurse < unit_cong_price ) {
        drop packet
    } else {
        if ( queue full ) {
            drop packet
        } else {
            forward packet{
                packetpurse = packetpurse - unit_cong_price
            }
        }
    }
end
```

```

        nodejcredit = nodejcredit + unit_cong_price
    }
}
end

```

At the destination $node_k$:

```

procedure OP-cong( ) begin
    unit_cong_price =  $n_k(t)/n_k$ 
    if (packetpurse < unit_cong_price) {
        drop packet
    } else {
        receive packet {
            nodekcredit = nodekcredit + packetpurse
        }
    }
end

```

If the originating node does not have enough credit to send the packet, the packet will be dropped at the source.

If the originating node has enough credit it will place the required credit into the packet purse and send the packet onwards. At each transit node the price of the service will be calculated according to the number of packets in the node's queue. This will usually be a fraction of a credit. Because the packet purse is filled with enough credit to account for the maximum amount of credit it could be charged along the route, it may reach the destination node with some excess credit. This is the credit that was not spent along the route. This excess credit is then passed on to the destination node.

This results in a flow of credit from the source to the destination. In the case of asymmetrical communication, it would be possible to create the notion of change. Each node could keep track of the amount of change owed to each other node, and this change could be returned as an excess sum added to the packet purse destined for that node. For our purposes this is not necessary, as the communication between nodes is symmetrical and the flow of credit from source to destination is balanced out by this symmetrical communication.

The credit redistribution mechanism is performed in a decentralized manner.

3.3.4. Destination Pays with Constant Pricing

The fifth incentive scheme is called destination-pays-constant-pricing (*DP-const*). *DP-const* is defined as follows: *the destination node is responsible for paying for the delivery of packets and all services are constant at one unit per service*. There are applications in which the destination benefits more from the successful receipt of a packet than the source does. A stock exchange application that regularly downloads updates on the stock market is one example.

DP-const is implemented as follows:

If a node needs to send a packet, it will set the amount of credit in the packet purse to 0. The node will send the packet and at each node along the path where the packet receives a service, the node servicing the packet will update its own credit by the constant service price of 1 and record in the packet purse the amount that the destination node needs to pay for delivery of the packet. As the packet traverses the route it will accumulate credit in its packet purse. When the packet reaches the destination node, the destination node will deduct the amount of credit in the packet purse from its own credit as payment for the delivery of the packet.

The destination-pays-constant-pricing algorithm

At the source $node_i$:

```
procedure DP-const( ) begin
    if (queue full) {
        drop packet
    } else {
        queue packet
         $packet_{purse} = 0$ 
        forward packet
         $node_i.credit = node_i.credit + 1$ 
         $packet_{purse} = packet_{purse} + 1$ 
    }
end
```

At the transit $node_j$:

```
procedure DP-const( ) begin
    receive packet
     $node_j.credit = node_j.credit + 1$ 
     $packet_{purse} = packet_{purse} + 1$ 
    if (queue full) {
        drop packet
    } else {
        forward packet
         $node_j.credit = node_j.credit + 1$ 
         $packet_{purse} = packet_{purse} + 1$ 
    }
end
```

At the destination $node_k$:

```
procedure DP-const( ) begin
    if (  $node_k.credit < packet_{purse}$  ) {
        drop packet
    } else {
        receive packet
         $node_k.credit = node_k.credit - packet_{purse}$ 
    }
end
```

Transit nodes along the path receive credits for performing services and the packet purse records the amount of service that the packet received in traversing the route.

When the packet reaches the destination node, the amount of credit in its packet purse will be the amount of credit that the destination node needs to pay to receive that packet. If the destination node has enough credit in its own credit balance, it will deduct that amount of credit from its own balance and receive the packet. If the destination node does not have enough credit in its own balance, the packet will be dropped.

The principle flaw in the destination pays model is that all the resources needed to transport the packet from the source to the destination are used, but the packet might still not be successfully received. The model does not help conserve resources because it is only known after the resource are used whether or not the destination can afford for those resources to be used.

3.3.5. Destination Pays with Congestion Based Pricing

The sixth incentive mechanism is called destination-pays-congestion-pricing (*DP-cong*). *DP-cong* is defined as follows: *the destination node is responsible for paying for the delivery of packets and the cost of a service is based on the congestion at the servicing node*.

DP-cong is implemented as follows:

If a node needs to transmit a packet, it will reset the packet purse to 0. As the packet traverses a route it will receive service from the transit nodes. These nodes charge the packet a congestion related price for the service performed and the cost will be added to the packet purse as it traverses the route.

Let n_i represent the maximum number of packets node i can have in its packet queue and $n_i(t)$ represent the number of packets at time t in the packet queue. The price of a service at node i is given by

$$\text{unit_cong_price} = n_i(t)/n_i$$

If the destination node does not have enough credit to pay for the delivery of the packet, the packet will be dropped.

The destination-pays-congestion-pricing algorithm

At the source node_i :

```
procedure DP-const( ) begin
    unit_cong_price = n_i(t)/n_i
    if (queue full) {
        drop packet
    } else {
        queue packet
        packet_purse = 0
        forward packet
        node_i.credit = node_i.credit + unit_cong_price
```

```

    packetpurse = packetpurse + unitcong_price
}
end

```

At the transit $node_j$:

```

procedure DP-const( ) begin
    unitcong_price =  $n_i(t)/n_i$ 
    receive packet
    nodejcredit = nodejcredit + unitcong_price
    packetpurse = packetpurse + unitcong_price
    if (queue full) {
        drop packet
    } else {
        forward packet
        nodejcredit = nodejcredit + unitcong_price
        packetpurse = packetpurse + unitcong_price
    }
end

```

At the destination $node_k$:

```

procedure DP-const( ) begin
    unitcong_price =  $n_i(t)/n_i$ 
    if ( nodekcredit < packetpurse ) {
        drop packet
    } else {
        receive packet
        nodekcredit = nodekcredit - packetpurse
    }
end

```

Each time a node performs a service on behalf of a packet it increases its own credit balance by the congestion price and adds that amount of credit to the packet purse. As the packet traverses the route it pays the transit nodes for their services and accumulates the charge of sending the packet along the route. Once it reaches the destination, the amount of credit in its packet purse is the amount of credit the destination node must pay in order to receive the packet.

If the destination node has enough credit, it will decrement its own credit balance by that amount and receive the packet. If the destination node does not have enough credit, the packet will be dropped.

3.3.6. Origin Pays with Energy Based Pricing

The seventh incentive scheme is called origin-pays-energy-pricing (*OP-energy*). *OP-energy* is defined as follows: *the origin node is responsible for paying for the delivery of packets and the cost of a service is based on the energy at the servicing node*.

The unit price based on the servicing node's energy is calculated as follows:

Let $e_i(t)$ represents the instantaneous energy of node i and e_i represents the initial energy of node i . The price of a service at node i is given by

$$\text{unit_energy_price} = 1 - \frac{e_i(t)}{e_i}$$

Initially, $e_i(t)$ will be close to e_i and so the unit energy price will be close to 0. This is because when the node has all its energy, energy is not a scarce commodity and the price of using this resource is low. As $e_i(t)$ declines however, the unit energy price will tend closer and closer to 1. As the energy of the node decreases, so the price of servicing a packet becomes higher as the resource required for that service becomes more valuable.

OP-energy is implemented as follows:

Each node is assigned a target credit balance. If a node needs to send a packet to a destination r hops away, it will need to have a maximum of $2r$ credits. This is the amount of credit it will cost to send the packet if each *unit_energy_price* at each node is 1. The total cost of sending the packet will be less than $2r$ credits depending on the energy level at the intermediary nodes. The excess credit in the packet purse will be passed onto the destination node.

If the source node does not have enough credit it will drop the packet and not expend any more resources on sending it.

The origin-pays-energy-pricing algorithm

At the source node_i :

```
procedure OP-energy( ) begin
```

```

unit_energy_price = 1 - e_i(t)/e_i

route_cost = 2r
if ( node_i.credit < route_cost ) {
    drop packet
} else {
    if ( queue full ) {
        drop packet
    } else {
        queue packet
        node_i.credit = node_i.credit - route_cost
        packet_purse = packet_purse + route_cost
        forward packet
        node_i.credit = node_i.credit + unit_energy_price
        packet_purse = packet_purse - unit_energy_price
    }
}
end
```

At the transit node_j :

```
procedure OP-energy( ) begin
```

```

unit_energy_price =  $1 - \frac{e_j(t)}{e_j}$ 

if ( $packet_{purse} < unit\_energy\_price$ ) {
    drop packet
} else {
    receive packet {
         $packet_{purse} = packet_{purse} - unit\_energy\_price$ 
         $node_j.credit = node_j.credit + unit\_energy\_price$ 
    }
}
if ( $packet_{purse} < unit\_energy\_price$ ) {
    drop packet
} else {
    if (queue full) {
        drop packet
    } else {
        forward packet{
             $packet_{purse} = packet_{purse} - unit\_energy\_price$ 
             $node_j.credit = node_j.credit + unit\_energy\_price$ 
        }
    }
}
}
end

```

At the destination $node_k$:

```

procedure OP-energy( ) begin
     $unit\_energy\_price = 1 - \frac{e_k(t)}{e_k}$ 
    if ( $packet_{purse} < unit\_energy\_price$ ) {
        drop packet
    } else {
        receive packet {
             $node_k.credit = node_k.credit + packet_{purse}$ 
        }
    }
}
end

```

If the originating node does not have enough credit to send the packet it will be dropped at the source.

If the originating node does have enough credit, it will place the required credit into the packet purse and send the packet onwards. At each intermediary node the price of the service will be calculated according to the servicing node's energy. This will be a fraction of a credit. When the packet reaches the destination it may have excess credit left in its packet purse. This excess credit will be given to the destination node.

3.3.7. Origin Pays with Credit Based Pricing

The eighth incentive scheme is origin-pays-credit-pricing (*OP-credit*). *OP-credit* is defined as follows: *the origin node is responsible for paying for the delivery of packets and the cost of a service is based on the credit at the servicing node.*

The unit price based on the servicing node's credit is calculated as follows:

Let $c_i(t)$ represents the instantaneous credit of *node i* and c_i represents the target credit balance of the node. The price of a service at node *i* is given by

$$\text{unit_credit_price} = \max\left(0.1 - \frac{c_i(t)}{c_i}\right)$$

Initially, $c_i(t)$ will be close to c_i and so the unit credit price will be close to 0. If any node has a $c_i(t)$ greater than c_i , its unit credit price will be equal to 0. So it will not earn any credit from assisting other nodes. As the $c_i(t)$ of a node decreases its unit credit price will get closer to 1. The less credit a node has, the more credit it earns from its services.

OP-credit is implemented as follows:

Each node is assigned a target credit balance. If a node needs to send a packet to a destination *r* hops away, it will need to have a maximum of $2r$ credits. This is the amount of credit it will cost to send the packet if each *unit_credit_price* at each node is 1. The total cost of sending the packet will be less than $2r$ credits depending on the credit level at the intermediary nodes. The excess credit in the packet purse will be passed onto the destination node.

If the source node does not have enough credit it will drop the packet and not expend any resources on sending it.

The origin-pays-credit-pricing algorithm

At the source *node_i*:

```
procedure OP-credit( ) begin
```

```

unit_credit_price = 1 - c_i(t)/c_i

route_cost = 2r
if ( node_i.credit < route_cost ) {
    drop packet
} else {
    if ( queue full ) {
        drop packet
    } else {
        queue packet
        node_i.credit = node_i.credit - route_cost
        packet_purse = packet_purse + route_cost
        forward packet
        node_i.credit = node_i.credit + unit_credit_price
        packet_purse = packet_purse - unit_credit_price
    }
}
```

```

        }
    }
end

```

At the transit $node_j$:

```

procedure OP-credit( ) begin
    unit_credit_price =  $1 - \frac{c_j(t)}{c_j}$ 

    if ( $packet_purse < unit\_credit\_price$ ) {
        drop packet
    } else {
        receive packet {
            packet_purse =  $packet_purse - unit\_credit\_price$ 
            nodej.credit =  $node_j.credit + unit\_credit\_price$ 
        }
    }
    if ( $packet_purse < unit\_credit\_price$ ) {
        drop packet
    } else {
        if (queue full) {
            drop packet
        } else {
            forward packet{
                packet_purse =  $packet_purse - unit\_credit\_price$ 
                nodej.credit =  $node_j.credit + unit\_credit\_price$ 
            }
        }
    }
}

```

end

At the destination $node_k$:

```

procedure OP-credit( ) begin
    unit_credit_price =  $1 - \frac{c_k(t)}{c_k}$ 
    if ( $packet_purse < unit\_credit\_price$ ) {
        drop packet
    } else {
        receive packet {
            nodek.credit =  $node_k.credit + packet_purse$ 
        }
    }
}

```

If the originating node does not have enough credit to send the packet it will be dropped at the source.

If the originating node does have enough credit, it will place the required credit into the packet purse and send the packet onwards. At each intermediary node the price of the service will be calculated according to the servicing node's credit. This will be a fraction of a credit. When the packet reaches

the destination it may have excess credit left in its packet purse. This excess credit will be given to the destination node.

3.4. Routing Metric Based Schemes

Ns2 implements a version of the DSDV routing protocol. Routing updates are periodically promulgated through the network and the routing tables of each node are populated according to the details specified in Section 2.1.4. In standard DSDV the routing metric is based on hop count, but it is possible to alter this routing metric. The basic mechanics of the routing protocol (such as updates and sequence number etc) will still be based on the ns2 implementation of DSDV, but the routes chosen by the protocol will vary as the routing metric changes. We implement the following routing metric based schemes by modifying the routing metric parameter in the standard ns2 DSDV library. All previous credit based schemes have been based on the standard DSDV implementation which uses hop count as the routing metric.

We investigate the following three routing metrics.

3.4.1. Congestion Based Routing

The first of the routing based metric schemes is called congestion-based-routing (*OP-cong-routing*) and it is an extension of the *OP-cong* scheme. *OP-cong-routing* is defined as follows: *the origin node is responsible for paying for the delivery of packets and the cost of a service is based on the congestion at the servicing node. The routing metric of the DSDV routing protocol is modified from the standard hop count to a congestion based metric that routes traffic along less congested routes.*

We would like to avoid congested nodes so we make the routing metric proportional to the congestion being experienced at a particular node. As a node becomes more congested, its routing metric will become higher and traffic will be encouraged to flow along routes with less congested nodes. This will have a balancing effect on the congestion in the network.

The routing metric at each node will be defined as

$$f_i(\text{cong}_i^t) = \text{cong}_i^t$$

Where cong_i^t is defined as the congestion of node n_i at time t .

$$\text{cong}_i^t = n_i(t)/n_i$$

The basics of the *OP-cong* scheme will still be operating with regards to the sending, relaying and reception of packets. The routing metric being influenced by the congestion at a particular node is an enhancement that should reduce congestion in the network and improve overall performance.

3.4.2. Energy Based Routing

The second routing based scheme is called Energy-based-routing (*FFA-battery-routing*). *FFA-battery-routing* is defined as follows: *all nodes forward all traffic on behalf of all other nodes. The*

routing metric of the DSDV routing protocol is modified from the standard hop count to an energy based metric that routes traffic away from nodes with less energy.

Energy-based-routing (*FFA-battery-routing*) is designed to encourage the use of routes that traverse nodes with unused energy resources. As a node loses energy, it becomes less attractive as a transit node. The *FFA-battery-routing* routing metric is inversely proportional to the energy of the node. As the energy of the node decreases the routing metric increases.

The routing metric at each node will be defined as

$$f_i(b_i^t) = \frac{1}{b_i^t}$$

Where b_i^t is defined as the battery capacity of node n_i . The battery cost for route $R_i(t)$ for route i at time t is defined as

$$R_i(t) = \sum_{j \in \text{route}_i} f_i(b_i^t)$$

As (b_i^t) decreases the value of $f_i(b_i^t)$ will increase. Thus routes containing nodes with low battery capacity will be less likely to be chosen as the routing metric will be high. If all nodes have equal amounts of battery capacity, this metric will become equivalent to the hop count metric as the overall route cost will equal the sum of $f_i(b_i^t)$ from each node. Once nodes in the centre of a network start to use up their energy, this routing metric will choose longer routes in terms of hop count, along the outside of the network where nodes have more energy. This will effectively spread the use of resources of the network more evenly.

Nodes at the centre of the network will last for a longer time as nodes at the periphery share the load of the network. The overall effect will be that the network will last longer as it will not become partitioned.

3.4.3. Credit Based Routing

The final routing metric to be investigated is called credit-based-routing-metric (*OP-const-cred-routing*). *OP-const-cred-routing* is defined as follows: *the origin node is responsible for paying for the delivery of packets and the cost of a service is constant at one unit. The routing metric of the DSDV routing protocol is modified from the standard hop count to a credit based metric that routes traffic towards nodes that have less credit.*

OP-const-cred-routing is a combination of the credit incentive schemes and the energy routing metrics discussed earlier. We can use the information inherent in the pricing mechanism to route packets. By allocating credit to nodes and then requiring nodes to earn and spend that credit in the sending of packets we are setting up a free market system which allocates a value to network resources.

By basing the credit price for rendering a service on the resource of a particular node, we are effectively assigning a value to that resource in the network. An example would be the energy of a node. If the credit price of a node performing a service on behalf of another node depends on the energy capacity of the servicing node, then the credit level of each node will be a representation of that node's ability to buy energy resources from the network.

Let c_i^t represent the credit of node i at time t . We therefore make the routing metric

$$f_i(c_i^t) = \frac{1}{c_i^t}$$

The routing metric is inversely proportional to the credit level of each node. The cost of route $R_i(t)$ for route i at time t is defined as

$$R_i(t) = \sum_{j \in \text{route}_i} f_i(c_i^t)$$

If a node has a high credit level, it means that this node is performing more service on behalf of the network than the network is performing on its behalf. The node is earning more credits than it is spending. If a node has a low credit level, it means this node is performing less service on behalf of the network than the network is performing on its behalf. The node is spending more credit than it is earning. We would like our routing mechanism to route packets towards nodes that have lower credit mechanisms to stimulate the fair use of resources across the network.

3.5. The Simulator

Ns2 is a discrete event simulator developed by UC Berkley for the purposes of investigating wired networks. Towards the late 90's extensions for simulating wireless ad hoc networks were added by the Monarch group at Carnegie Mellon University and the simulator has become a widely used tool for investigating wireless ad hoc networks.

We use the default values of ns2 to create a simulation of the experimental scenarios described in Section 3.7. The workings of ns2 are described below.

3.5.1. Node Structure in ns2

Ns2 is a packet-based simulator. Ns2 models the sending of DATA packets, ACK packets, DSDV routing packets, and RTS/CTS packets. Each packet is an object derived from the base object Event. Figure 1 shows the structure of a wireless node in ns2.

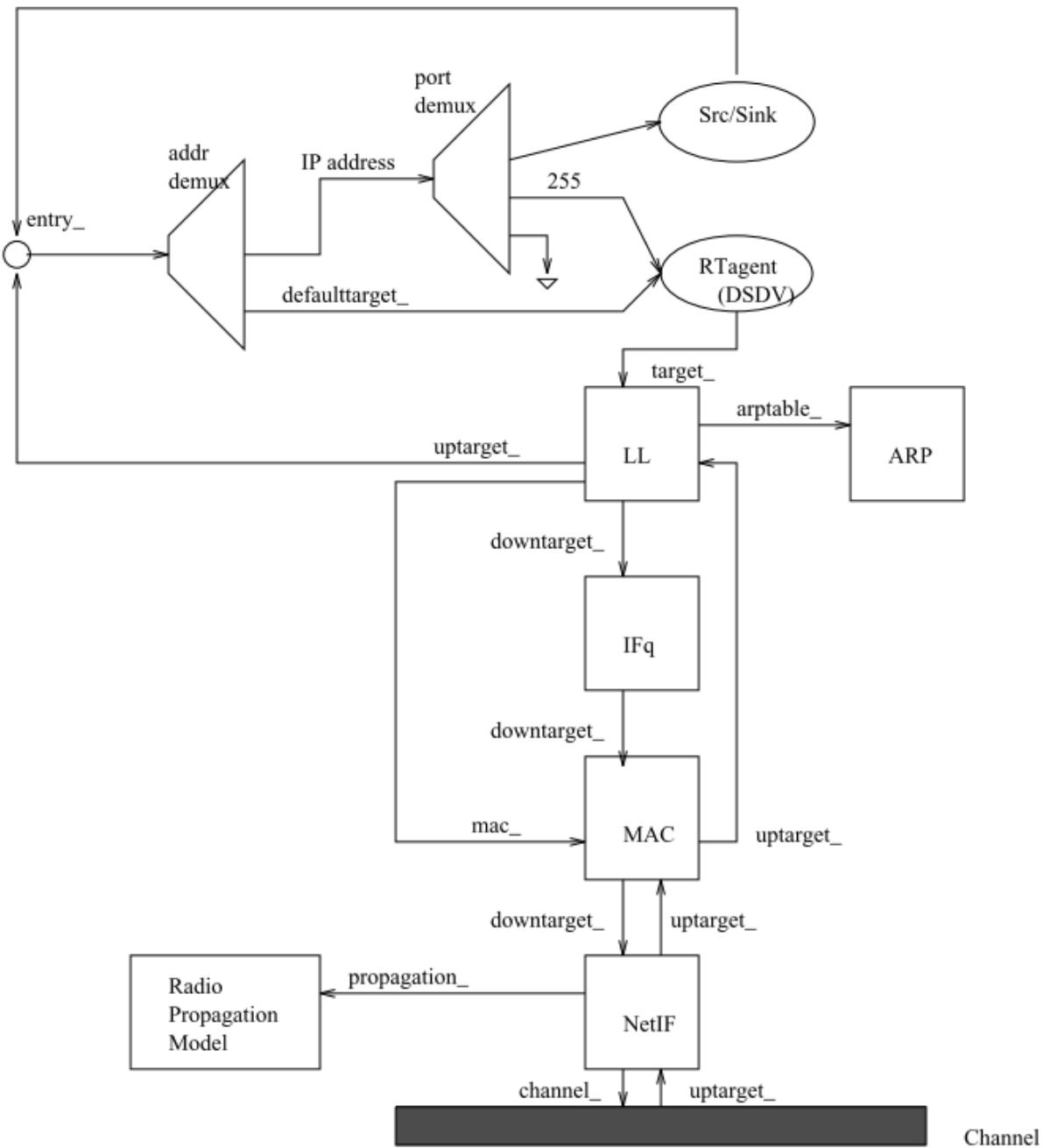


Figure 1: Basic structure of a wireless node in ns2

An ns2 node is comprised of a number of objects. The routing protocol is implemented in the Routing Agent module which is attached to port 255 of the node. Packets received by the node at `entry_` are passed to an address classifier. If the destination address of the packet does not match the node address, the packet is passed on to the routing agent where it is forwarded according to the node's routing table. If the destination address of the packet matches the node address, the packet is passed on to the port classifier. The port classifier will determine the particular service agent object to pass the packet to. Multiple service agents can be attached to a single node simulating the multiple applications that can be running on a single networked host.

Any packets sent from the agent itself will also pass through the address classifier. Since the node will not send packets to itself, the packets will then move onto the routing agent module and be routed accordingly.

The routing agent module is attached to a network interface which simulates layers 1, 2 and 3 of the OSI model. The LL object implements link layer operations and the ARP object keeps an ARP table of all nodes within wireless transmission range. The IFQ implements a First-In-First-Out (FIFO) queue that can store up to 50 packets. The MAC object implements the medium access control protocol the node uses to access the wireless medium.

3.5.2. Wireless Propagation Model

The wireless channel object has a list of all nodes attached to the wireless channel. Using a radio propagation model and the positions of the sending and receiving nodes, the wireless channel object calculates the received power of a packet at the destination node. The wireless channel implements a Two Ray Ground propagation model which is based on the Friis Free-Space equation for short distances and the Two Ray Ground equation for longer distances [55].

The Friis Free-Space equation is defined by

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

Where d is the distance between the transmitter and the receiver. P_t is the transmitted signal power. G_r and G_t are the antenna gains of the receiver and the transmitter respectively, L is the system loss and λ is the wavelength from the frequency by

$$\lambda = \frac{3 \times 10^8}{frequency} = \frac{3 \times 10^8 m/s}{914 \times 10^6 Hz} = 0.328m$$

It is common to select $G_r = G_t = 1$ and $L = 1$ in ns2 simulations.

The Two Ray Ground equation is defined by

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

Where h_r and h_t are the heights of the receive and transmit antennas respectively. The received power at a node is calculated using the Free-Space equation when the distance is less than the crossover distance, and the Two Ray Ground equation when the distance is greater than the crossover distance. The crossover distance is defined as the distance at which the two equations calculate an equal received power and is given by

$$d = \frac{4\pi h_r h_t}{\lambda}$$

The propagation model is used to determine the received power of a transmission depending on the distance between the nodes. Once the received power is known, two parameters at the receiving node determine the power level at which a packet can be received or cause interference. If the

receiving power level of the signal is above CThreshold, the transmission will cause interference with any other transmissions that node is receiving. If the power level is above RXThreshold, the transmission can be received and decoded. The RXThreshold is always higher than the CThreshold because if the power level of a transmission is strong enough for the node to receive the packet, it will be strong enough to cause interference. Signal to Noise Ratio (SINR) is not modelled in the ns2 radio model.

We use the ns2 default values for CThreshold and RXThreshold. These are set as follows

$$CThreshold = 1.559 \times 10^{-11} W \text{ which results in an interference distance of } 564m$$

$$RXThreshold = 3.652 \times 10^{-10} W \text{ which results in an interference distance of } 249m$$

If the receiving power is strong enough for the packet to be received, the packet is sent to the node interface to the MAC layer. The MAC layer will then deal with the packet according to the MAC 802.11 protocol.

Each time a node is determined to have received a packet, the packet is sent to the MAC layer and the MAC layer state is changed. There are 7 possible states for the MAC layer. They are: IDLE, RECV, SEND, COLL, RTS, CTS and ACK. If the MAC layer is any state other than IDLE, the transmission medium is considered to be busy.

3.5.3. Medium Access Control and Contention-Based Access Using the DCF in ns2

We configured ns2 so that the nodes implement the HR/DSSS of the IEEE 802.11-2007 otherwise known as mac802.11b. This amendment of the 802.11 standard was chosen because this was the first commercially practical implementation of the 802.11 protocol and provides the functionality necessary for testing the credit based mechanism. The 802.11b standard uses a slotted Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) to share access to the wireless medium.

There is a physical CS mechanism provided by the PHY layer and a virtual CS mechanism implemented on each node.

Random backoff Procedure in ns2

All nodes implement the backoff procedure as specified in the standard.

If a node wishing to send a packet senses the transmission medium with its PHY CS and its virtual CS and finds the medium to be BUSY, it will defer sending until the medium becomes IDLE. Once the medium becomes IDLE, it will wait for one DIFS interval to elapse and then it will implement the backoff procedure. Each node has a backoff timer which holds the time that the node must wait during the contention period before sending. This timer is only allowed to decrement during contention periods. A contention period begins once the transmission medium becomes IDLE and one DIFS period elapses during which the medium remains IDLE for the duration of the DIFS. If the medium becomes busy before the backoff timer reaches zero, it is paused at its current value and allowed to resume once another DIFS period has elapsed.

If the node enters a contention period and the backoff timer is not initiated (does not have a value), then a random number will be selected and the backoff timer will be initiated. From there on, the backoff timer will be decremented during contention periods. The first time the backoff timer is set, a random number is selected in the range 0 to aCwmin. Each time the backoff timer is reset the range is doubled. This happens until the range is from 0 to aCwmax. Once the range reaches this value, it remains at this value for subsequent resets of the backoff timer. The increase in the backoff timer period each time the timer is reset is a means of accommodating congestion on the radio interface. The backoff timer is reset when there is a collision, and each time there is a collision the nodes involved will wait longer before they retransmit.

If a node's backoff timer does reach zero and the node transmits a packet and the packet timeout interval elapses before the node gets a responding packet, the node will increment the appropriate retry counter and increase the contention window size and retransmit the packet. If the retry counter exceeds the maximum retries, the packet will be dropped.

Request to Send and Clear to Send in ns2

The virtual CS mechanism is implemented by assigning a Network Allocation Vector (NAV) to each node and updating the NAV with the information in the duration fields of RTS/CTS and DATA and ACK packets. The NAV is implemented as a timer in each node and as long as the timer is not zero, the virtual CS is considered to show the medium as busy.

When the node has a packet to send, it first checks the state of the MAC and the NAV. If the MAC state is anything except the IDLE state or if the NAV is not zero then the transmission medium is considered to be busy and the node defers sending until the medium state is changed to IDLE and the NAV timer is zero.

The dot11RTSThreshold attribute is implemented in an ns2 node and can be configured by the end user. This value works as specified in the standard in that all non-broadcast DATA packets longer than the threshold value will require a RTS/CTS exchange before they are transmitted. If a DATA packet is smaller than the dot11RTSThreshold value, then no RTS/CTS handshake is necessary. It is possible to turn off RTS/CTS entirely for a wireless network by configuring dot11RTSThreshold to be 2346 bytes (maximum size a DATA packet can be). In this case all DATA packets are smaller than the dot11RTSThreshold value and will not require a RTS/CTS handshake.

Ns2 simulates the sending of RTS/CTS packets which are 20 and 14 bytes long respectively. A node receiving an RTS packet will implement the actions required of it by the 802.11 protocol.

If it is the intended recipient for the RTS packet it will check its PHY CS and virtual CS. If the PHY CS and virtual CS both return that the medium is IDLE, then the node will respond with a CTS packet indicating that the sending node can begin transmitting. If either the PHY CS or the virtual CS indicates that the medium is busy, the node will not reply with a CTS and the sending node's CTSTimeout will elapse. The sending node will then send another RTS packet after incrementing its Short Retry Counter (ssrc) and increasing its contention window size. Each station implements both

the Long and the Short Retry counters. If the maximum retry counts for either counter is reached the packet that the node intends to send will be dropped. The maximum retry counts are configurable and are set to the values for HR/DSSS MAC 802.11 as can be seen in Table 1.

If the node receiving the RTS is not the intended recipient for the RTS packet but rather a neighbouring node, it will set its NAV according to the value in the duration field of the RTS packet.

BasicRate and DataRate in ns2

Ns2 implements the basicRate and dataRate parameters as specified in the mac802.11b standard. These are important for calculating the transmission times for packets and calculating how long the medium is reserved for. For instance a DATA packet that is 1500 bytes long will take

$$\frac{1500 * 8}{11} = 1090\mu s$$

The basicRate is the rate at which the header information in the MAC frame is sent, and the dataRate is the rate that the data payload is sent at.

MAC Level Positive Acknowledgements in ns2

All DATA packets sent in ns2 require positive ACK packets in response to confirm the successful transmission of the packet. When a node sends a packet, it begins a timer with the value

$$ACKTimeout = SIFS + ACK \text{ transit time}$$

If this timer reaches zero, the node assumes that the sent packet was lost and attempts to retransmit.

If a node receives a successful DATA packet it will reply to the sending node with an ACK packet. The credit based mechanism is not implemented on an ACK packets.

Interframe Spacing in ns2

Ns2 implements the Distributed Interframe Space (DIFS), Short Interframe Space (SIFS) and Extended Interframe Space (EIFS) as specified in the standard. Between packets within one atomic exchange, nodes will wait a SIFS before replying. When a node sends a packet, it will check the transmission medium using its PHY CS and virtual CS and when the transmission medium becomes IDLE it will wait for a DIFS before sending.

If there is a collision, the standard requires that the nodes involved should defer access to the medium for a period of one EIFS before entering the contention period. This is implemented in ns2.

Our simulation only concerns the DCF mechanism for mobile ad hoc networks with no infrastructure component to the networks. For this reason we will not require the PCF Interframe Space (PIFS).

PLCP and PMD Layers in ns2

In ns2 the MPDUs are mapped into PPDUs which can be transmitted by the PMD layer. This involves adding a preamble and PLCP header to the packet so that receiving nodes will be able to detect the node transmission and lock onto the signal and decode the bit stream.

The preamble and PLCP header affect the transmit time for a packet as follows. The preamble is 72 bits long, and the PLCP header is 48 bits long. The preamble is transmitted at 1Mbps (basicRate) and the PLCP header is transmitted at 2Mbps (dataRate). The extra time added to the transmission of a packet is then $72\mu s + 24\mu s = 96\mu s$. This is implemented in ns2.

Fragmentation and Reassembly in ns2

Fragmentation and reassembly are implemented in ns2. The fragmentation Threshold by default is set to 2346 which is the maximum size a DATA packet can be, so it is effectively disabled by default and this is how we have configured our simulation to run.

PHY Characteristics of MAC802.11b in ns2

Table 1 shows the settings of variables in ns2 to simulate a mac802.11b network.

Characteristic	Value
SlotTime	$20\mu s$
SIFS	$10\mu s$
DIFS	$SIFS + 2 * SlotTime = 50\mu s$
Preamble	72bits
PLCP Header	48bits
basicRate	1Mbps
dataRate	11Mbps
aCwmin	31
aCwmax	1023
RTSThreshold	0
Short Retry Limit	7
Long Retry Limit	4

Table 1 PHY Characteristics of MAC802.11b network in ns2

3.5.4. Routing Protocol Implementation

DSDV is implemented in the Routing Agent module of a mobile node. This is shown in Figure 1. The routing agent performs all the layer 3 routing of packets in a mobile node. It is the default target of the address classifier object and is also attached to port 255 of the port classifier object.

Packets are routed towards the routing agent object in one of two ways.

- DATA packets at the address classifier with a destination address other than the address of the node are routed to the default target. This can be packets received from other nodes that need to be forwarded on toward their destination, or packets originating from one of the agents attached to the node itself.
- DSDV packets pass through the address classifier and onto the port classifier and are routed to port 255 because they are identified as DSDV routing packets

The DSDV agent has a reference to a Routing Table object. The routing table stores route objects with the following properties

- Destination address
- Next hop
- Link cost metric
- Sequence Number
- Time the route was last changed
- Time when a new sequence number was heard

Periodic updates are sent out at the periodic update period which is set to 15s by default. The MAC layer object will report a link down if it increments through its short or long retry counters without receiving any positive ACK packets to confirm the reception of the packet. This will trigger the lost_link procedure in the DSDV object which will then assign a very high metric to that route and send out an update. There is also a value to specify the number of periodic updates that must be missed before a link is considered to be lost. This is set to 3 by default in ns2.

The DSDV implementation receives routing updates from other nodes and configures the metric of the routes according to hop count. If the route is forwarded, it adds 1 to the route metric to account for itself as an extra hop in reaching the destination.

3.5.5. Energy Model Implementation

The energy model in ns2 is implemented by an Energy Model object associated with each node. The energy model is assigned an initial node energy in our simulation of 50 Joules. We use the ns2 default values of transmit power set to 0.5W and the receive power set to 0.001 W. The energy used for a transmission is calculated by multiplying the transmission time by the transmission power.

Nodes will enter an idle time when they are not receiving or transmitting. During this time there is no energy lost.

3.6. Measures of Network Performance

To measure the effectiveness of the various incentive mechanisms, we define the following metrics to measure the network's performance.

3.6.1. Packet Success Probability

Packet Success Probability (PSP) defined by

$$P_i = \frac{S_i}{N_i}$$

Where S_i is the number of packets successfully delivered from source node i and N_i is the number of packets originating from source node i .

Average PSP defined by

$$\text{Avg PSP} = \frac{\sum_i P_i}{\text{nodes}}$$

Where nodes is the number of active nodes in the simulation.

3.6.2. Throughput Rate Per Node

Throughput rate per node (TRPN) is defined by

$$T = \frac{\sum_i S_i}{\text{nodes} \times \text{time}}$$

Where S_i is the number of packets successfully delivered from source node i over the course of the simulation, nodes is the number of active nodes in the simulation and time is the time traffic has been flowing through the simulation.

3.6.3. Total Average Latency

Total average latency (TAL) is defined by

$$T = \frac{\sum_i L_i}{\text{nodes}}$$

Where L_i is the average time taken for packets to be successfully delivered from node i over the course of the simulation, nodes is the number of active nodes in the simulation.

3.6.4. Energy Rate Per Node

Energy rate per node (ERPN) is defined by

$$E = \frac{\sum_i E_i}{\text{nodes} \times \text{time}}$$

Where E_i is the amount of energy used at node i over the course of the simulation, nodes is the number of active nodes in the simulation and time is the time traffic has been flowing through the simulation.

3.6.5. Network Fairness

To measure the network fairness we use Jain et al.'s [4] measurement of fairness of n competing flows defined by

$$f_{rate}(x_1, \dots, x_n) = \frac{(\sum_i x_i)^2}{n \sum_i x_i^2}$$

Where x_i represents the data rate of flow i . The fairness ranges from $1/n$ in the case of least fairness (where all flows except one are zero) to 1 in the case of most fairness (where all flows are of equal size). Using this we defined two measures of network fairness.

The first measure of fairness is f_{rate} defined by

$$f_{rate}(P_1, \dots, P_n) = \frac{(\sum_i P_i)^2}{n \sum_i P_i^2}$$

f_{rate} represents the fairness in data rate allocation where P_i denotes the PSP of node i . It is a measure of fairness of service each node receives from the network. Low values of f_{rate} means that the network provides unfair service to the respective nodes. High values of f_{rate} mean that the network provides fair service to each node.

The second measure of fairness is $f_{reciprocity}$ defined by

$$f_{reciprocity}(Q_1, \dots, Q_n) = \frac{(\sum_i Q_i)^2}{n \sum_i Q_i^2}$$

$f_{reciprocity}$ represents the fairness in resource provisioning where Q_i is called the reciprocity of node i where

$$Q_i = \frac{\sum_{i \neq j} c_i(j)}{\sum_{i \neq j} c_j(i)}$$

$c_i(j)$ is the number of packets that node i has transmitted or received on behalf of node j . Thus $f_{reciprocity}$ represents the fairness with which nodes are serving one another in the network. If $f_{reciprocity}$ is high, then all nodes are serving one another approximately equally. If $f_{reciprocity}$ is low, then some nodes are serving others far more than those nodes are serving them.

3.7. The Network Topology

The network models were configured with $N = 50$ nodes. There are two network topologies.

The first network topology used for Experiment 1 is when all nodes are static. In this case the nodes are placed at the centre of their respective cells. There are 49 cells (7×7) so the 50th node is placed randomly in the central cell of the topology. Each cell is 125m wide so that two nodes at the centres of adjacent cells will be 125m apart. This is well within the transmission range of 250m.

The second network topology used for subsequent Experiments 2, 3,4 and 5 is when half the nodes are placed in the centre of their respective cells and the rest of the nodes move according to the random waypoint model [47]. In this model, the node randomly chooses a destination point in space and moves towards this point with a randomly chosen constant speed. When it reaches the chosen destination, it stops and waits there for a randomly chosen time. Then, it chooses a new destination and speed, and starts to move again. These steps are repeated until the end of the simulation. In our simulations, the nodes choose their speed between 0 m/s and 5 m/s uniformly. The average pause time is 150s [4].

We distinguish between two localities of nodes in our experiments. The central locality is highlighted in dark grey in Figure 2 and the edge locality is highlighted in light grey. We evaluate the network metrics for the overall network, for the central nodes and for the edge nodes separately so as to be able to distinguish the different conditions and behaviours at various positions in the network.

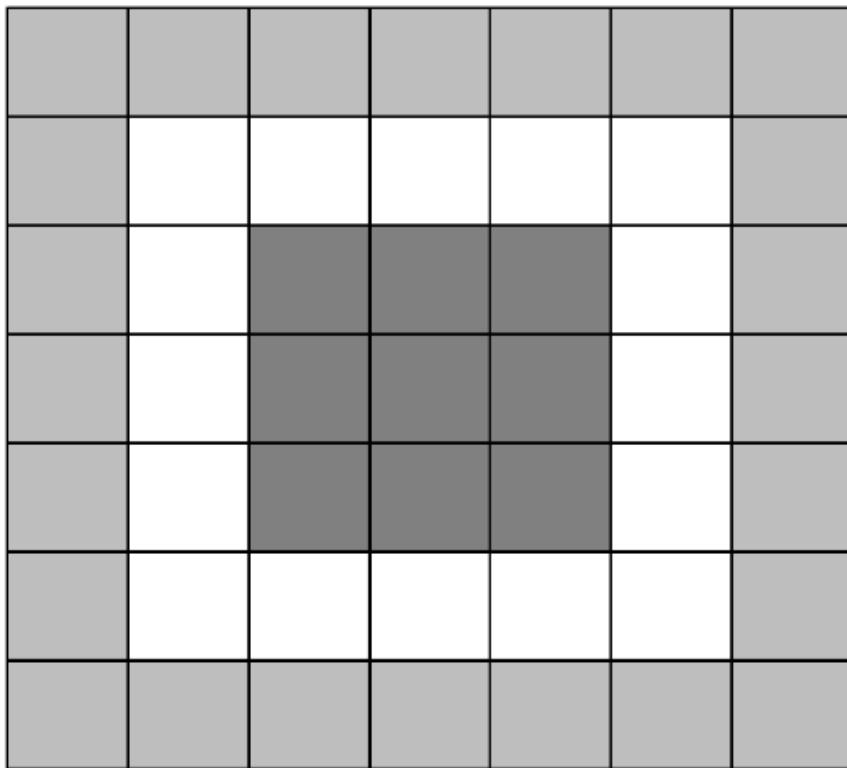


Figure 2: Node layout for 50 node network.

Each node has a bandwidth of 11Mbps and a battery power of 0.5W. Each node has a packet queue that can store 50 packets. Each node attempts x packet transmissions per second. The transmissions take place at the instants of a Poisson process. Each transmission selects a random destination node. RTS/CTS is used when transmitting frames larger than the RTS Threshold. The DSDV routing protocol is modelled in ns2.

Periodic updates are regularly flooded across the network and used to build routing tables at each node. The default routing metric for DSDV is least hop count (LHC) routing. It is possible to change

the routing metric while still using the underlying DSDV mechanism. Each node routes on a next hop basis. If a link between two nodes becomes unusable, the node attempting to send across that link will experience a packet drop due to a packet retry exceeded and a triggered routing update will be sent out to inform other nodes that the link is no longer usable. DSDV is known to be a very dynamic routing protocol in that routes are constantly changing.

Each packet is 1554 bytes long. Each simulation models 200,000 packet transmissions.

3.8. The Experiments

3.8.1. Experiment 1: Performance across varying packet arrival rates

The aim of the first experiment is to investigate the performance of the various incentive schemes across varying packet arrival rates.

All schemes are applied in a model of a 50-node network. Nodes are immobile and are located at the centres of their respective $125m \times 125m$ cells on a $875m \times 875m$ plane. Least hop count routing is used. The node batteries are continuously recharged so that the batteries are not depleted. The performance of the protocols is investigated at packet arrival rates varying from 2 to 16 packets per second. Each experiment is independently replicated 15 times.

3.8.2. Experiment 2: What is the effect of the redistribution mechanism

The aim of the second experiment is to investigate the effect of the redistribution mechanism on the network performance.

All schemes involving credit redistribution (*OP-const*, *OP-cong*, *OP-credit*, *OP-const-cred-routing*, *OP-energy* and *OP-cong-cong-routing*) are applied in a model of a 50-node network. The nodes are initially placed at random on a $875m \times 875m$ plane and move according to a random waypoint model. Node batteries are constantly recharged. Each experiment is independently replicated 15 times.

In the first iteration of the experiment the redistribution mechanism is disabled, and in the second iteration it is enabled.

3.8.3. Experiment 3: What is the effect of the redistratate

The aim of the third experiment is to investigate the effect of the redistratate on the networks performance.

All schemes involving credit redistribution (*OP-const*, *OP-cong*, *OP-credit*, *OP-const-cred-routing*, *OP-energy* and *OP-cong-cong-routing*) are applied in a model of a 50-node network. The nodes are initially placed at random on a $875m \times 875m$ plane and move according to a random waypoint model. Node batteries are constantly recharged. Each experiment is independently replicated 15 times.

The network performance is measured for packet arrival rates of 2, 6 and 12 pps. The network performance is presented for values of redistribution interval varying from 0.01 to 0.5.

3.8.4. Experiment 4: What is the effect of the target credit balance

The aim of the fourth experiment is to investigate the effect of the target credit balance on the network performance.

All schemes involving credit redistribution (*OP-const*, *OP-cong*, *OP-credit*, *OP-const-cred-routing*, *OP-energy* and *OP-cong-cong-routing*) are applied in a model of a 50-node network. The nodes are initially placed at random on a $875m \times 875m$ plane and move according to a random waypoint model. Node batteries are constantly recharged. Each experiment is independently replicated 15 times.

The network performance is measured for packet arrival rates of 2, 6 and 12 pps. The network performance is presented for values of target credit balance varying from 10 to 1600.

3.8.5. Experiment 5: What is the effect on the use of resources in the network

The aim of the fifth experiment is to investigate the effect of the credit mechanisms on the conservation of resources of the network.

All schemes are applied in a model of a 50-node network. Nodes are immobile and are located at the centres of their respective $125m \times 125m$ cells on a $875m \times 875m$ plane. Node batteries are not recharged. Each node has a limited amount of energy. When a node runs out of energy, that node switches off and the simulation continues without it.

The network performance is measured for packet arrival rates of 2, 6 and 12 pps. We monitor the instantaneous PSP and average energy through the simulation to evaluate the effect that the various protocols have on how resources are shared.

4. Results

4.1. Experiment 1: Performance across varying packet arrival rates

The aim of the first experiment is to investigate the performance of the various incentive schemes as a function of the packet arrival rate.

All schemes are applied in a model of a 50-node network. Nodes are immobile and are located at the centres of their respective $125m \times 125m$ cells on a $875m \times 875m$ plane. Least hop count routing is used. The value of N_{margin} for TFT is 20. All credit-based schemes have a target credit balance of 100, a redistribution rate (δ) value of 0.05 and a redistribution interval (Δ) of 1s. The node batteries are continuously recharged so that the batteries are not depleted. The performance of the protocols is investigated at packet arrival rates varying from 2 to 16 packets per second. Each experiment is independently replicated 15 times and the 95% confidence interval half-widths are computed. The results displayed are the average of these 15 replications. The confidence intervals are not illustrated in the figures as they are too small. Each replication runs for a total of 1500s. Nodes do not send any data for the first 500 seconds of the simulation. This time allows the routing protocol to establish stable conditions before the network begins carrying traffic.

4.1.1. Which Credit Based Scheme is Most Effective at Improving Network Performance

We investigated eight credit-based schemes for enhancing the network's performance. These schemes lie somewhere between the overly restrictive *TFT* and the *FFA* scheme which is not restrictive enough.

In this section we analyse the performance of the various credit based schemes and identify which schemes lead to the best network performance.

Figures 3, 4, 5 and 6 display the overall network PSP, TRPN, TAL and ERPN respectively for the various protocols for three representative packet arrival rates. Packet arrival rates of 2, 8 and 16 packets per second represent the network under light load, medium load and heavy load conditions.

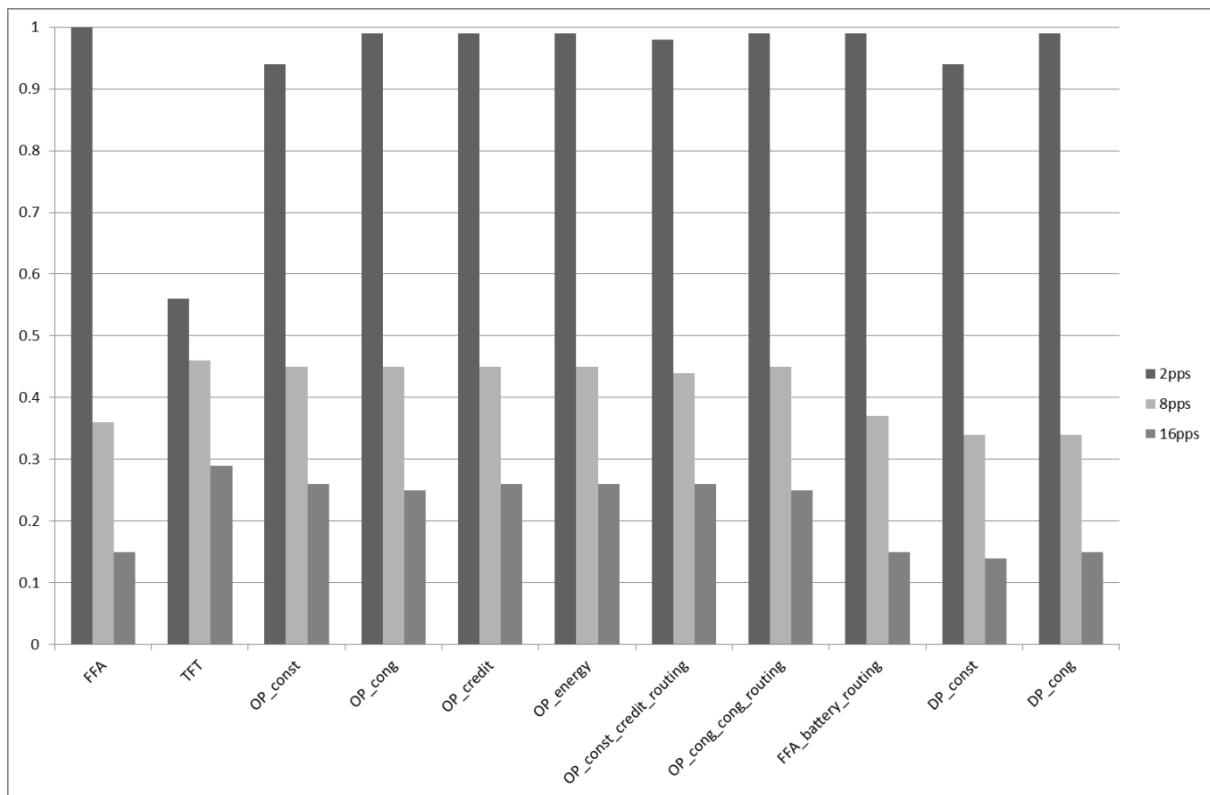


Figure 3: Overall network Packet Success Probability for three representative packet arrival rates

The performance of *FFA* is good for low packet arrival rates. This is represented by a high PSP and TRPN when compared to the other schemes. As the packet arrival rates increase however, the performance of *FFA* declines sharply. *TFT* performs poorly for low packet arrival rates as nodes are limited by the restrictive *TFT* conditions, but as packet arrival rates increase the PSP and TRPN of *TFT* decrease less rapidly than *FFA*. Under medium load conditions of 8pps, the performance of *TFT* is better than *FFA*. *TFT* also results in a low TAL on the network for all packet arrival rates as it reduces congestion. This represents a significant improvement in network performance and we will see is one of the major benefits of *TFT* and the credit-based mechanisms we propose in this thesis.

Rate	Credit Mechanism	Avg PSP	95% conf interval half-widths	Rate	Credit Mechanism	Avg PSP	95% conf interval half-widths
2	FFA	1.00	0.000101	2	TFT	0.56	0.001929
4	FFA	0.79	0.000728	4	TFT	0.57	0.001795
6	FFA	0.50	0.001194	6	TFT	0.54	0.001194
8	FFA	0.36	0.001192	8	TFT	0.46	0.000799

10	FFA	0.27	0.001508	10	TFT	0.40	0.000375
12	FFA	0.22	0.001089	12	TFT	0.35	0.000366
14	FFA	0.18	0.000998	14	TFT	0.32	0.000335
16	FFA	0.15	0.000932	16	TFT	0.29	0.000564
2	OP_const	0.94	0.000260	2	OP-cong	0.99	0.000143
4	OP_const	0.76	0.000576	4	OP-cong	0.75	0.000722
6	OP_const	0.58	0.000950	6	OP-cong	0.56	0.001209
8	OP_const	0.45	0.000628	8	OP-cong	0.45	0.000577
10	OP_const	0.37	0.000634	10	OP-cong	0.38	0.007372
12	OP_const	0.32	0.000321	12	OP-cong	0.32	0.000534
14	OP_const	0.28	0.000563	14	OP-cong	0.28	0.000349
16	OP_const	0.26	0.000555	16	OP-cong	0.25	0.000460

Table 2 95% confidence interval halfwidths for *FFA*, *TFT*, *OP-const* and *OP-cong*

The first credit-based mechanism is *OP-const*. The PSP and TRPN of *OP-const* are only slightly less than that of *FFA* at low packet arrival rates. As the load on the network increases the PSP and TRPN of *OP-const* for packet arrival rates of 8 and 12pps are comparable to that of *TFT*. The TAL of *OP-const* is also significantly lower than that of *FFA* although not as low as that of *TFT*. *OP-const* then represents the excellent performance of *FFA* at low packet arrival rates while maintaining the advantageous performance of *TFT* at higher packet arrival rates.

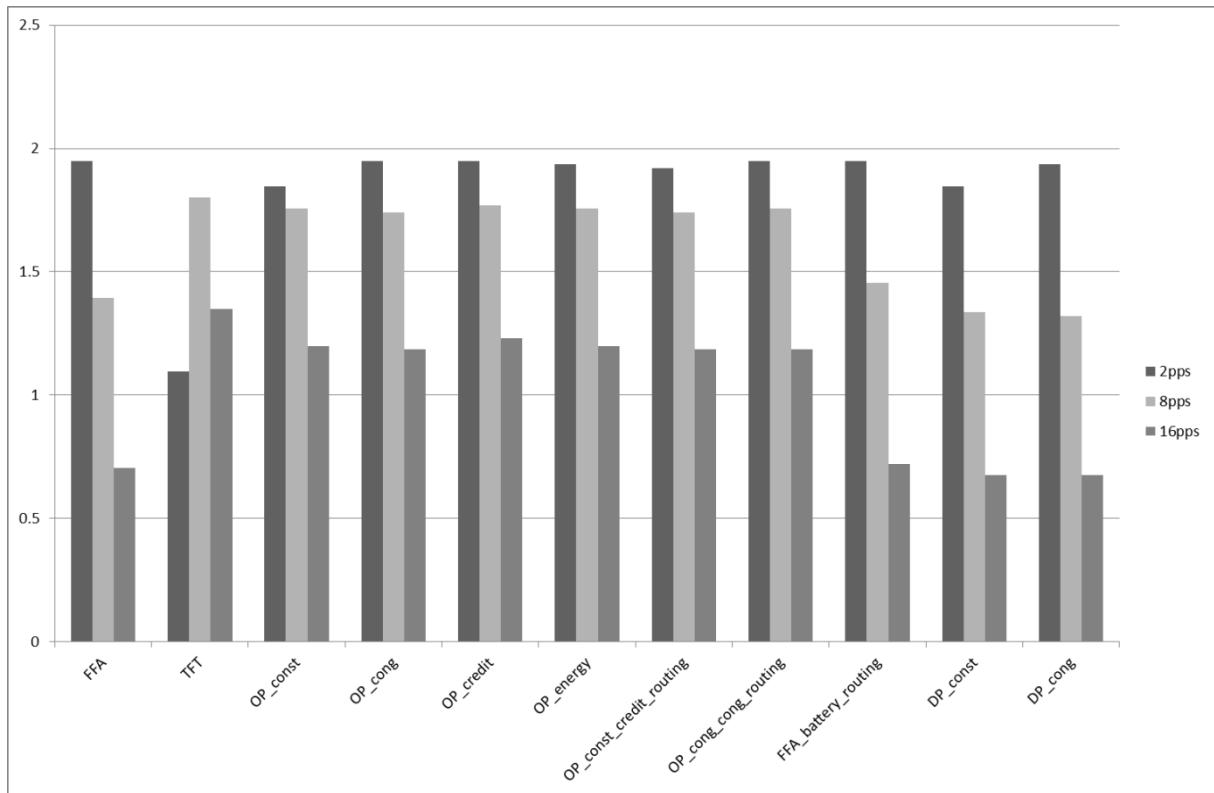


Figure 4: Overall network TRPN (pps) for three representative packet arrival rates

OP-cong is an elaboration of *OP-const* in that the prices per service are based on the congestion conditions at the serving node. At low packet arrival rates the packet queues at nodes are generally empty and so the price per service is low. This results in less restriction on the sending of packets at low arrival rates in *OP-cong* and improved performance under low network load conditions. *OP-cong* shows an improvement in PSP and TRPN over *OP-const* at packet arrival rates of 2pps but maintains the same performance at *OP-const* at higher packet arrival rates. The TAL of *OP-cong* also shows an improvement over *OP-const* for higher packet arrival rates as the congestion based pricing reduces the congestion on the network.

The remaining elaborations of the *OP-const* protocol are *OP-credit* and *OP-energy*. In these schemes the prices of services are based on the credit and energy levels of the nodes. For all packet arrival rates the PSP and TRPN is comparable to that of *OP-cong*. The TAL of *OP-energy* is the same as *OP-cong* while *OP-credit* results in a slightly higher TAL. Neither of these protocols shows an enhancement over the performance of *OP-cong*.

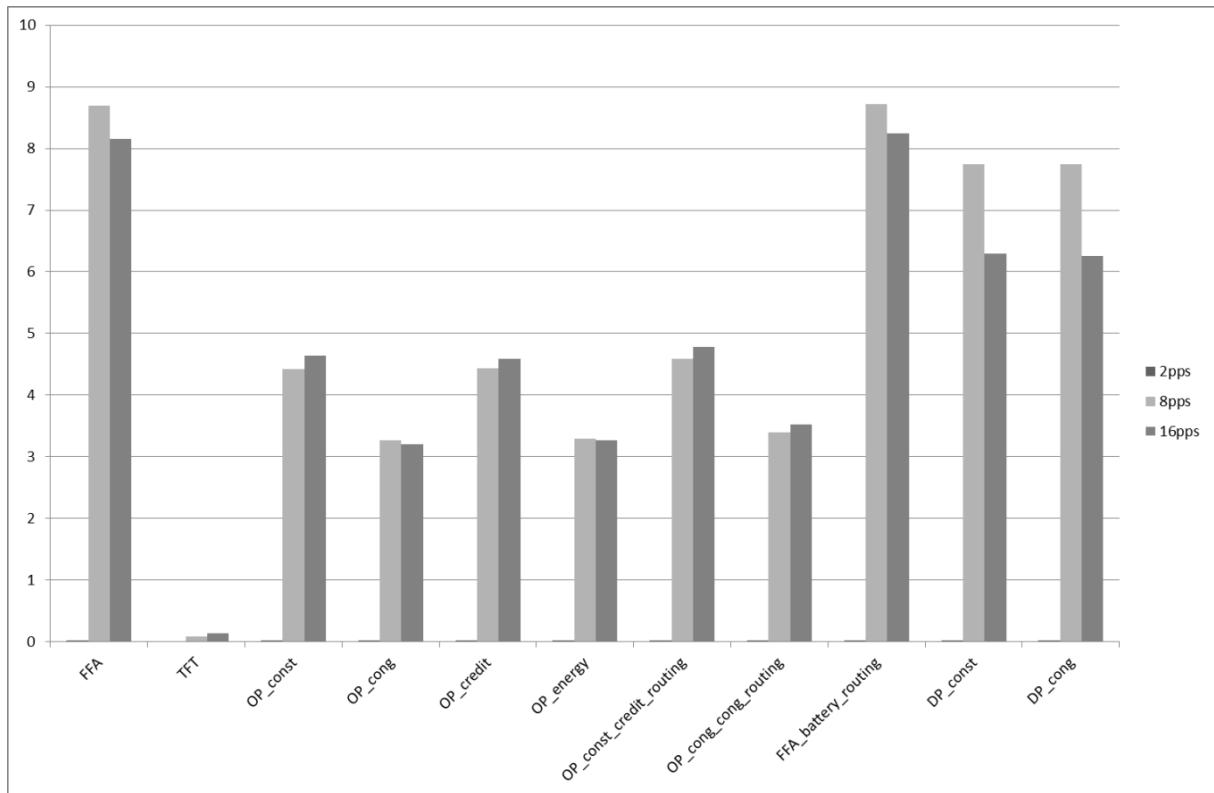


Figure 5: Overall network TAL (s) for three representative packet arrival rates

It is interesting to note that the credit-based protocols with enhancements to the routing metric do not show an improved performance over *OP-cong*. *OP-const-credit-routing* and *OP-cong -routing* have the same PSP and TRPN as *OP-cong* for all packet arrival rates. The TAL of *OP-cong -routing* is also the same as *OP-cong* for all packet arrival rates while *OP-const-credit-routing* shows a slightly higher TAL. These results show that using the congestion and credit information at a node to inform the routing of packets around the network has little effect. The value added by the credit-based mechanisms comes from determining whether an originating node should be allowed to send a packet.

We can also see that *FFA-battery-routing* has degraded performance at higher packet arrival rates. This is due to the fact that this scheme does not provide any intelligence as to whether packets should be sent or not, but rather attempts to route them in an energy saving manner. Figure 5 shows that the TAL of *FFA-battery-routing* is higher than all the other schemes due to the longer (in terms of hop count) energy saving routes it will use. We also see in Figure 6 that the ERPN is higher than the other schemes. This is due to the fact that *FFA-battery-routing* attempts to balance the use of energy across the various nodes of the network. This results in longer routes being selected. Routes that use nodes at the periphery of the network instead of the shortest path through the centre.. The result is that the overall energy use of the network increases, but that the energy use across the various nodes is more balanced.

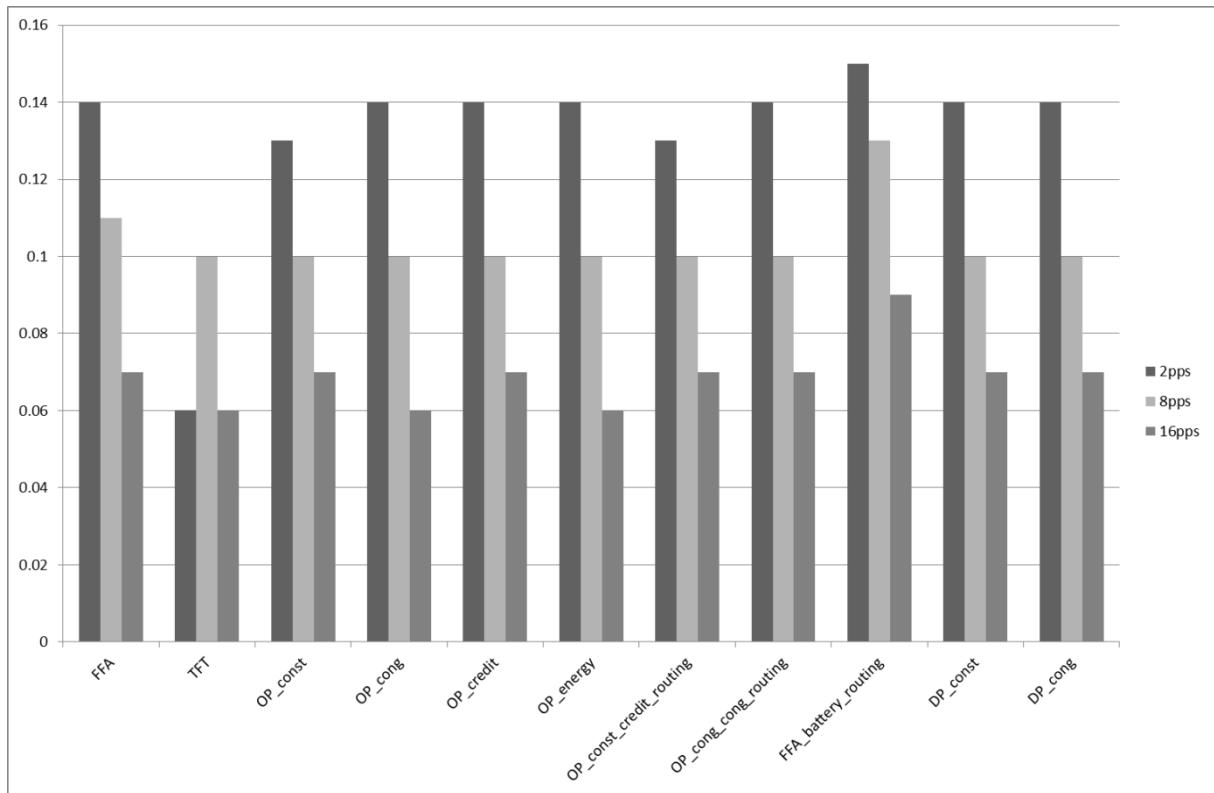


Figure 6: Overall network ERPN (J/s) for three representative packet arrival rates

The performance of the destination pays schemes (*DP-const* and *DP-cong*) is below that of the other credit schemes. We expect this to be the case as the destination pays schemes do not provide added value to the network. Network resources are used to carry a packet to its destination at which point it may be dropped due to the destination not having enough credit to pay for the packet. The resources that are intended to be conserved by the credit scheme are still used regardless of whether the packet reaches its destination or not.

When considering the performance of the various protocols we will focus on *OP-const* and *OP-cong*. *FFA* results in excellent performance under low load conditions but poor performance under high load conditions. *TFT* has poor performance under low load conditions but shows an improved performance under high load conditions over that of *FFA*. *TFT* also show improved TAL. *OP-const* combines the positive effects of the *FFA* and *TFT* protocols by resulting in good performance at low and high packet arrival rates. *OP-cong* further improves performance at low packet arrival rates. The remaining variants of the credit-based schemes do not show any improvement over the performance of *OP-cong*. For this reason we select *OP-const* and *OP-cong* as the protocols for further analysis in this thesis.

4.1.2. Performance of the Basic Schemes: *FFA* and *TFT*

In this section we present the results of the two basic incentive schemes *FFA* and *TFT*.

Figure 7 shows the performance of the *FFA* and *TFT* schemes for varying packet arrival rates. The overall PSP is the overall average PSP of all nodes in the 50 node network. The central PSP is the average PSP for the 10 centrally located nodes and the edge PSP is the average PSP for the 27 nodes located on the edge of the network. It is clear that at low packet arrival rates (less than 4pps) the *FFA* scheme performs far better than *TFT*, especially for nodes at the edge of the network.

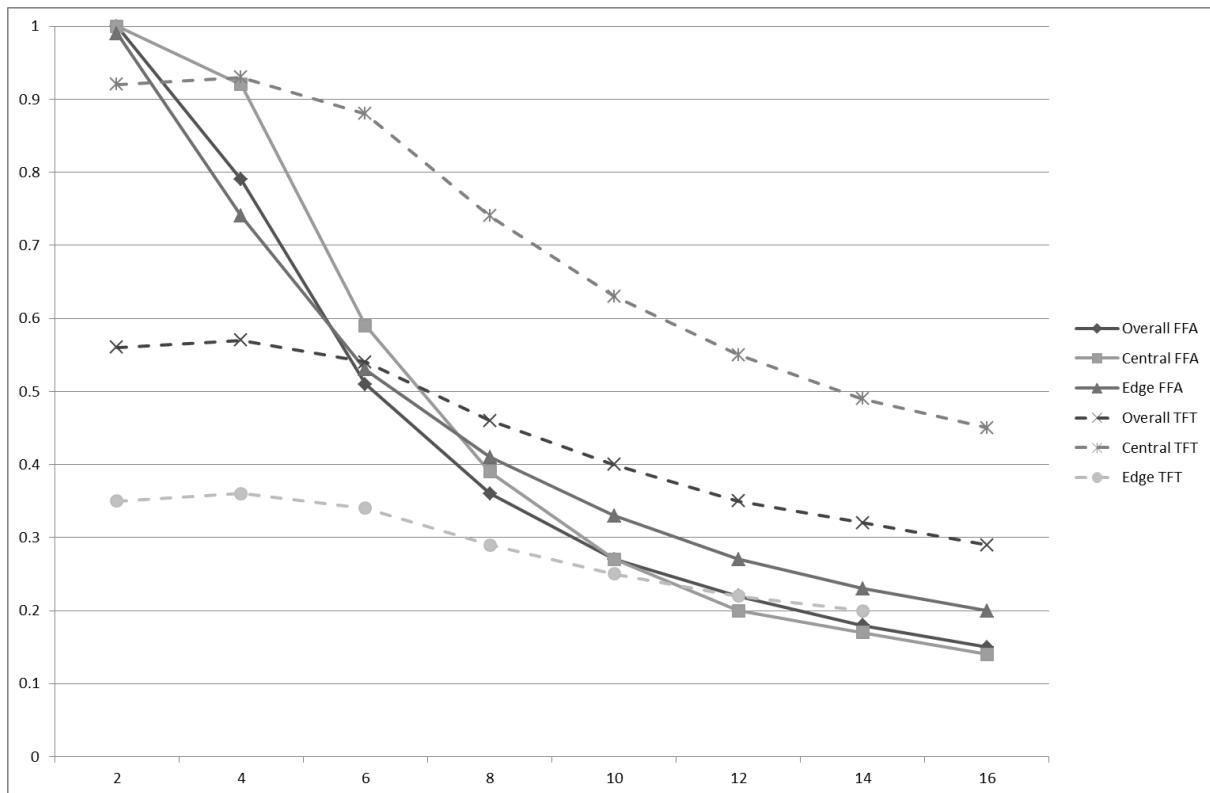


Figure 7: Packet Success Probability vs packet arrival rate for *FFA* and *TFT*

At low packet arrival rates, nodes at the edge of the network do not get many opportunities to serve transit traffic. Because they don't serve traffic from other nodes, they don't earn the right to send their own traffic. The edge nodes drop many packets due to the *TFT* restrictions. Nodes at the centre of the network get many opportunities to serve other nodes and generally are not hindered by the *TFT* restrictions.

At low packet arrival rates, the network has enough capacity to handle the traffic load and *FFA* performs well. As the packet arrival rate increases, the performance of *FFA* declines rapidly. At 6 pps the overall network PSP with *FFA* is around 0.5. As the packet arrival rate increases further this PSP drops even more. For packet arrival rates of 10 and above, the network is non-functional with a PSP of 0.3 or less.

The *TFT* protocol also experiences a decline in performance as the packet arrival rates increase, but its decline is less severe and at high packet arrival rates (higher than 10pps) *TFT* overall outperforms *FFA*. The overall network performance of *TFT* at low arrival rates is poor due the fact that edge

nodes get very little opportunity to provide service to other nodes, and as a result are severely hindered by the *TFT* restrictions.

Figure 8 shows the average throughput rate per node (TRPN) for the *FFA* and *TFT* schemes. Again it is clear that under the *FFA* scheme, the overall performance of the network is far better than *TFT* at low packet arrival rates. *TFT* has a comparable performance for nodes which are located in the centre of the network. This is because these nodes have many opportunities to serve passing traffic and are never hindered by the *TFT* restrictions. Nodes located at the edge of the network however, bring down the overall TRPN as these nodes are severely hampered by the *TFT* restrictions.

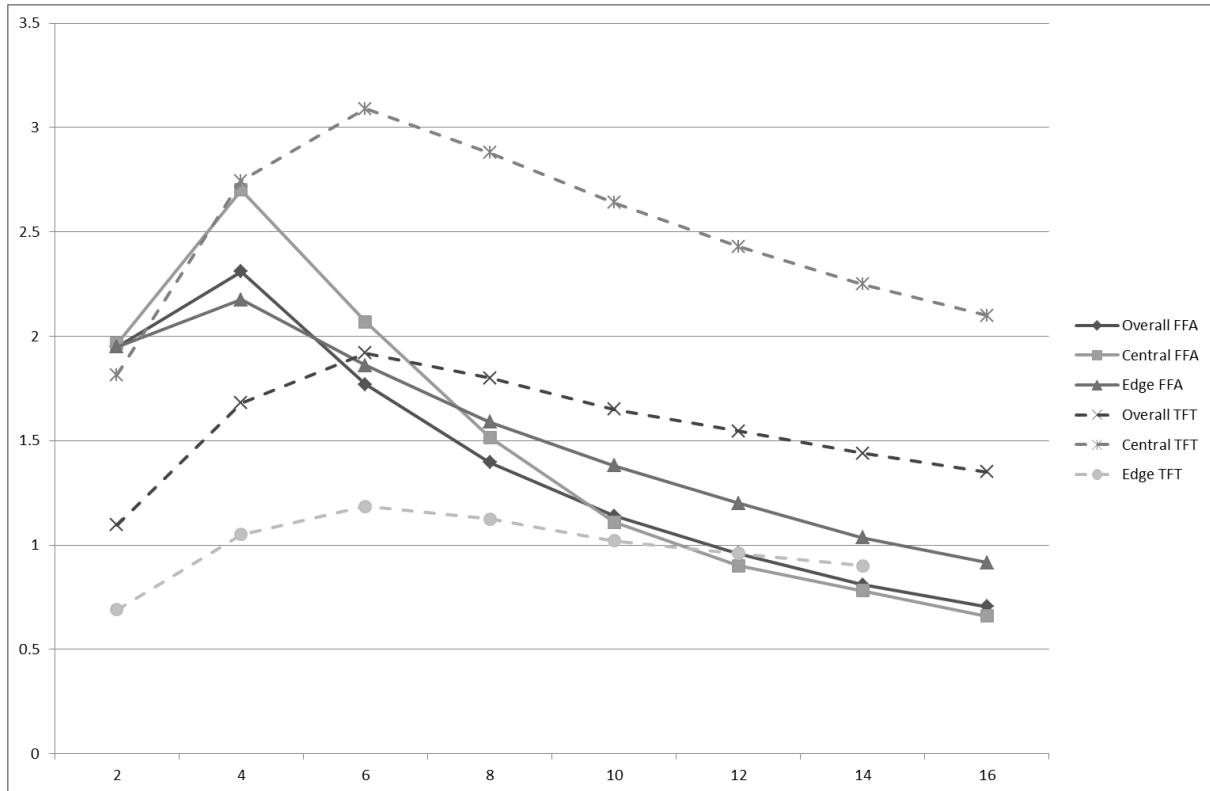


Figure 8: TRPN (pps) vs packet arrival rate for *FFA* and *TFT*

As the packet arrival rate increases towards 4 packets per second, the TRPN for both *FFA* and *TFT* increases. As the packet arrival rate increases further however, the TRPN begins to fall under the *FFA* scheme. It declines most rapidly for nodes located at the centre of the network because these nodes experience the most congestion. Packets are lost due to the fact that too many nodes are attempting to transmit. The limited resources of the air interface become oversubscribed and the number of collisions and back offs due to collisions increase rapidly. The more congested the air interface becomes; the less effectively the bandwidth of the nodes is used resulting in a decrease in throughput.

The *TFT* scheme restricts the number of packets that are released into the network due to the strict sending criteria at the source node. This means that the network starts to become congested at 6 pps as can be seen in Figure 8. The *TFT* scheme works particularly well for nodes at the centre of

the network with the TRPN of a central node under *TFT* being significantly higher than the TRPN for a central node under *FFA* at all arrival rates greater than 6 pps. This improved performance is due to the fact that under the *TFT* scheme edge nodes admit few packets into the network and the central nodes are to that extent less congested.

Figure 9 shows that the Total Average Latency (TAL) is significantly higher in *FFA* for all nodes regardless of their position in the network. This can be attributed to the congestion that is caused by more nodes trying to access the air interface resulting in longer packet queues and thus a longer overall time for packet delivery. The *TFT* scheme reduces the number of packets sent into the network at the source node and thus reduces the congestion and use of the limited air interface resources. Less packets colliding on the air interface and less packets accumulating in node queues greatly reduce the overall network latency. The positive effect of *TFT* on network TAL can be seen as one of the primary benefits it brings to the network's performance.

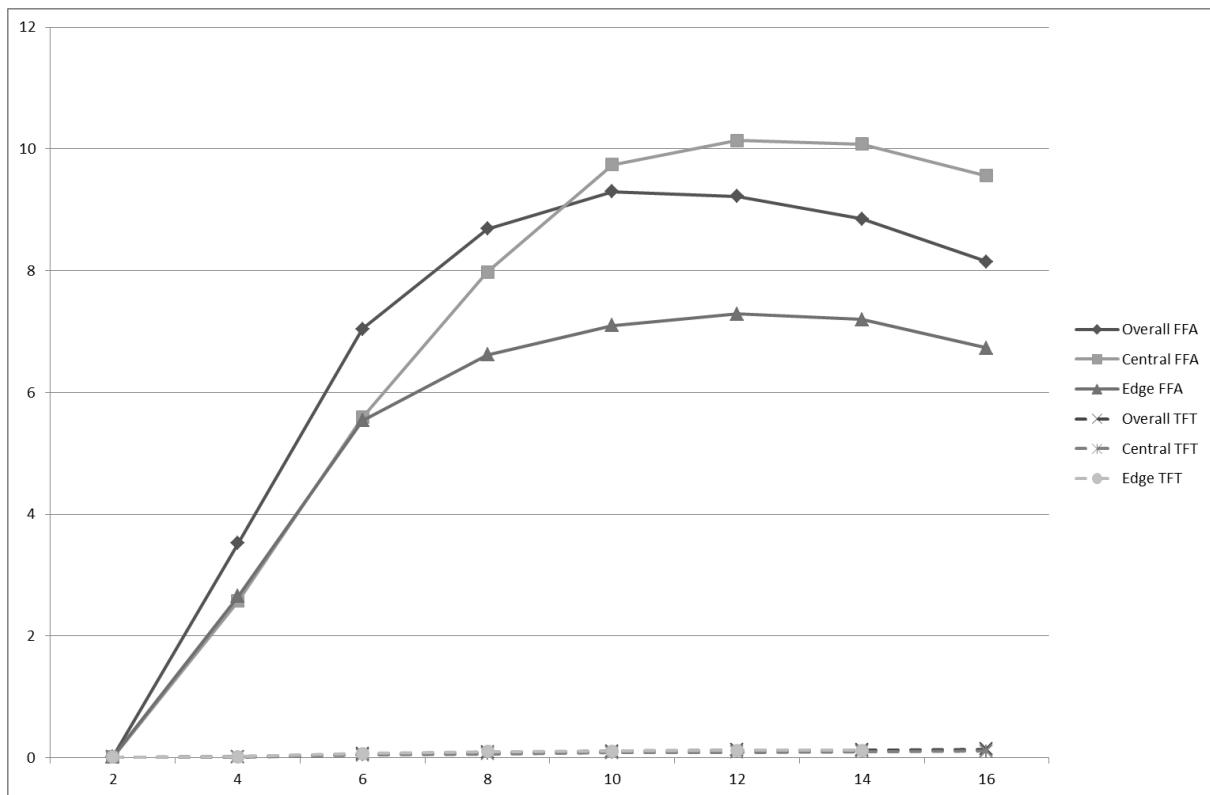


Figure 9: TAL (s) vs packet arrival rate for *FFA* and *TFT*

We see that the network latency of *FFA* reaches its maximum at around 10pps. It does not increase with higher packet arrival rates because at this stage the network is saturated. Higher packet arrival rates cause the packets to be dropped at the packet queue of the first node. The air interface does not become more congested and the latency thus does not increase. The TAL begins to decrease at packet arrival rates higher than 10pps because the proportion of traffic flow from edge node to edge node becomes higher as the congestion at the centre of the network increases. Because the proportion of traffic from edge node to edge node becomes more significant, the overall TAL of all

successfully delivered packets becomes lower. Traffic that would usually traverse longer routes is no longer successful and does not count toward the TAL. It is interesting to note that the centrally located nodes experience higher latency's than the peripheral nodes. This is due to the fact that central nodes experience more congestion on the air interface because there are more nodes within their transmission range.

TFT is very effective in eliminating congestion as it causes packets to be dropped at the source node. This results in no congestion on the air interface and at packet queues and results in very low network latencies.

Figure 10 shows the energy rate per node (ERPN) versus packet arrival rate. The ERPN is a measure of how quickly the nodes are using their energy. It is the total energy used by the nodes divided by the simulation time divided by the number of nodes.

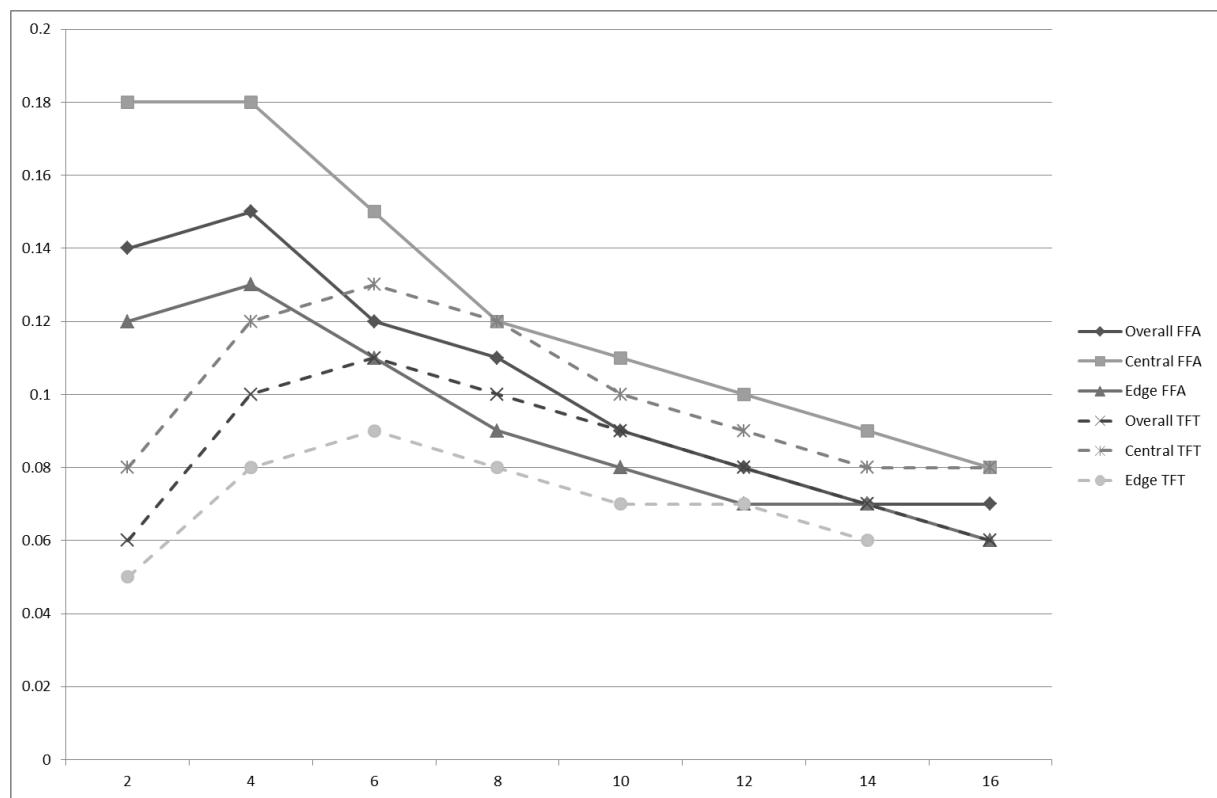


Figure 10: ERPN (J/s) vs packet arrival rate for *FFA* and *TFT*

In the *FFA* scheme, the nodes use their energy more rapidly than in the *TFT* scheme. The nodes in the centre of the network use their energy more rapidly because they are using their resources to transmit their own packets as well as transit traffic from other nodes. These nodes receive more transit traffic than nodes at the edge because more routes are likely to go via these central nodes. ERPN follows the same trend as the TRPN as it is the amount of traffic that a node handles that determines how much energy it uses. In the *FFA* scheme this peaks at a packet arrival rate of 4pps and in the *TFT* scheme it peaks at 6pps, both the same packet arrival rates for which the TRPN peaked.

The ERPN is an important parameter when we consider networks with finite resources. When a node's battery power is not constantly recharged, the use of energy resources becomes a key aspect to the continued operation of the network. If the nodes in the centre of the network rapidly use all their resources, they will cease to function and cause the network to become partitioned. Once the network is partitioned, packets will be undeliverable across the network even though there is energy available in the edge nodes.

The fairness of the network can be evaluated using the f_{rate} and $f_{reciprocity}$ measures mentioned in Section 3.6. f_{rate} is a measure of the fairness of flows or PSP of each node. A value of 1 means that all node PSP's are equal. As the value decreases so the variation in the node's PSP's increases. FFA has a high f_{rate} for low packet arrival rates as all nodes have very high PSP's because all traffic is successfully delivered to the destination. As the packet arrival rate increases however, the f_{rate} decreases as resources on the network become constrained. The edge nodes suffer more from this than the central nodes and as a result the variation in node PSP's increases. TFT has a low f_{rate} regardless of the packet arrival rate. Even at 2pps, the f_{rate} is just above 0.8. This is due to the TFT mechanism which disadvantages edge nodes.

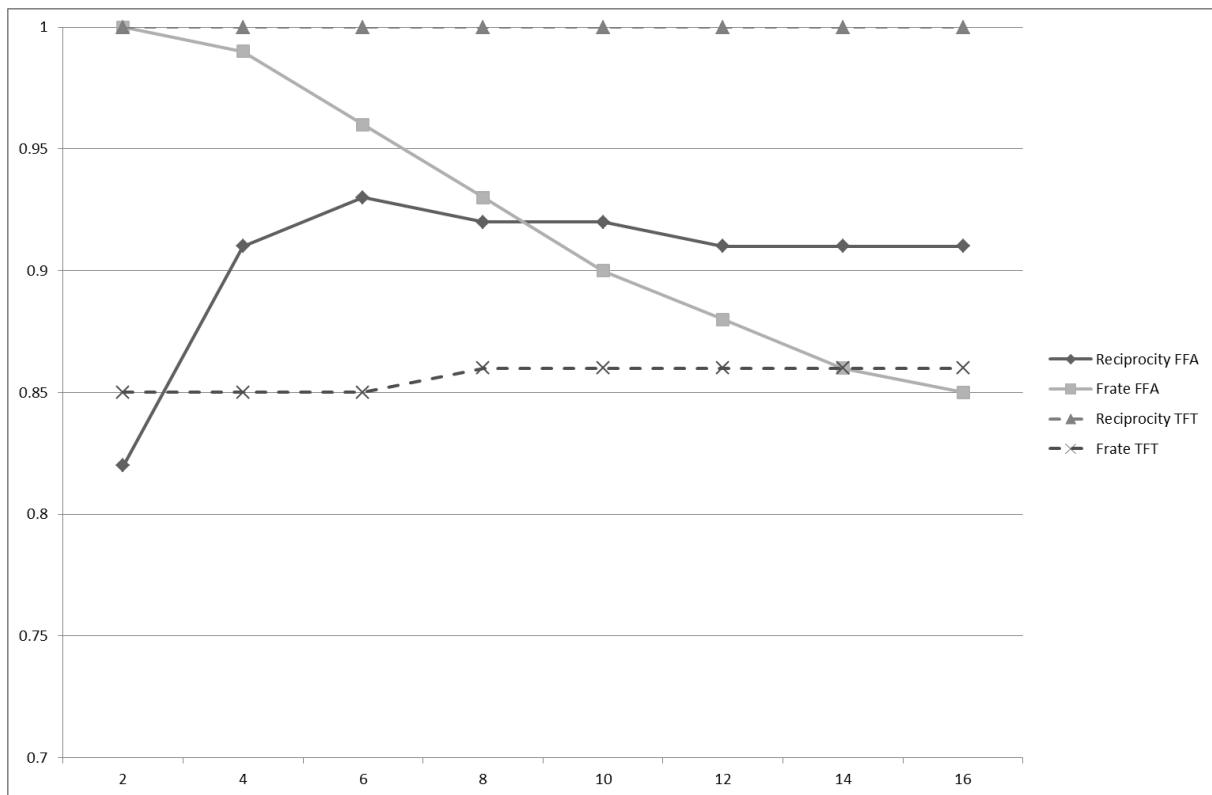


Figure 11: rate and reciprocity vs packet arrival rates for FFA and TFT

The $f_{reciprocity}$ represents the fairness with which nodes are serving one another in the network. In the TFT scheme, the $f_{reciprocity}$ is almost 1 for all values of packet arrival rate. This is because the TFT scheme forces nodes to serve one another to the same degree, so nodes ensure that they serve one another fairly. In FFA, the $f_{reciprocity}$ is low for low packet arrival rates because nodes in the

centre perform far more service than nodes at the edge. The $f_{reciprocity}$ increases as the packet arrival rate increases because nodes in the center experience congestion and are unable to relay traffic as they did before. This means that nodes at the edge, which experience less congestion, are able to provide more service as the overall fairness and reciprocity of the network increases.

The credit based systems we propose in Chapter 3 lie somewhere in between *FFA* and *TFT*. What we want to achieve is a network that performs well at low packet arrival rates as *FFA* does, but has a less severe performance decline as packet arrival rates increase. We require a scheme that combines the good throughput and PSP performance of *FFA* but also has the advantage of the high $f_{reciprocity}$ and fairness and low TAL exhibited by *TFT*.

4.1.3. Performance of the Credit Based Schemes: *OP-const* and *OP-cong*

In this section we present the results for the credit based schemes that were found to be most effective in stimulating cooperation and enhancing network performance. These were found to be *OP-const* and *OP-cong*. The various other schemes (*OP-energy*, *OP-credit*, *OP-const-cred-routing*, *OP-cong-cong-routing*, *DP-const*, *DP-cong* and *FFA-battery-routing*) were found to provide negligible enhancements over *OP-cong*. We investigate the performance of these credit based schemes under the same network conditions as the previous *FFA* and *TFT* experiments.

Figure 12 illustrates the PSP and TRPN of the *OP-const* and *OP-cong* credit mechanisms for all nodes averaged across the network, the central nodes and the edge nodes respectively. The figure shows that for centrally located nodes, the *OP-const* scheme has a positive effect on the performance of the nodes. From a packet arrival rate of 6pps and higher it causes an improvement of 20% in the PSP over the *FFA* scheme. The *OP-cong* scheme also has a positive impact on the PSP and TRPN for centrally located nodes. This is due to the fact that both the *OP-const* and *OP-cong* schemes favour nodes in the centre of the network because nodes in the centre of the network experience many opportunities to earn credit and thus are generally unhindered by the credit mechanisms.

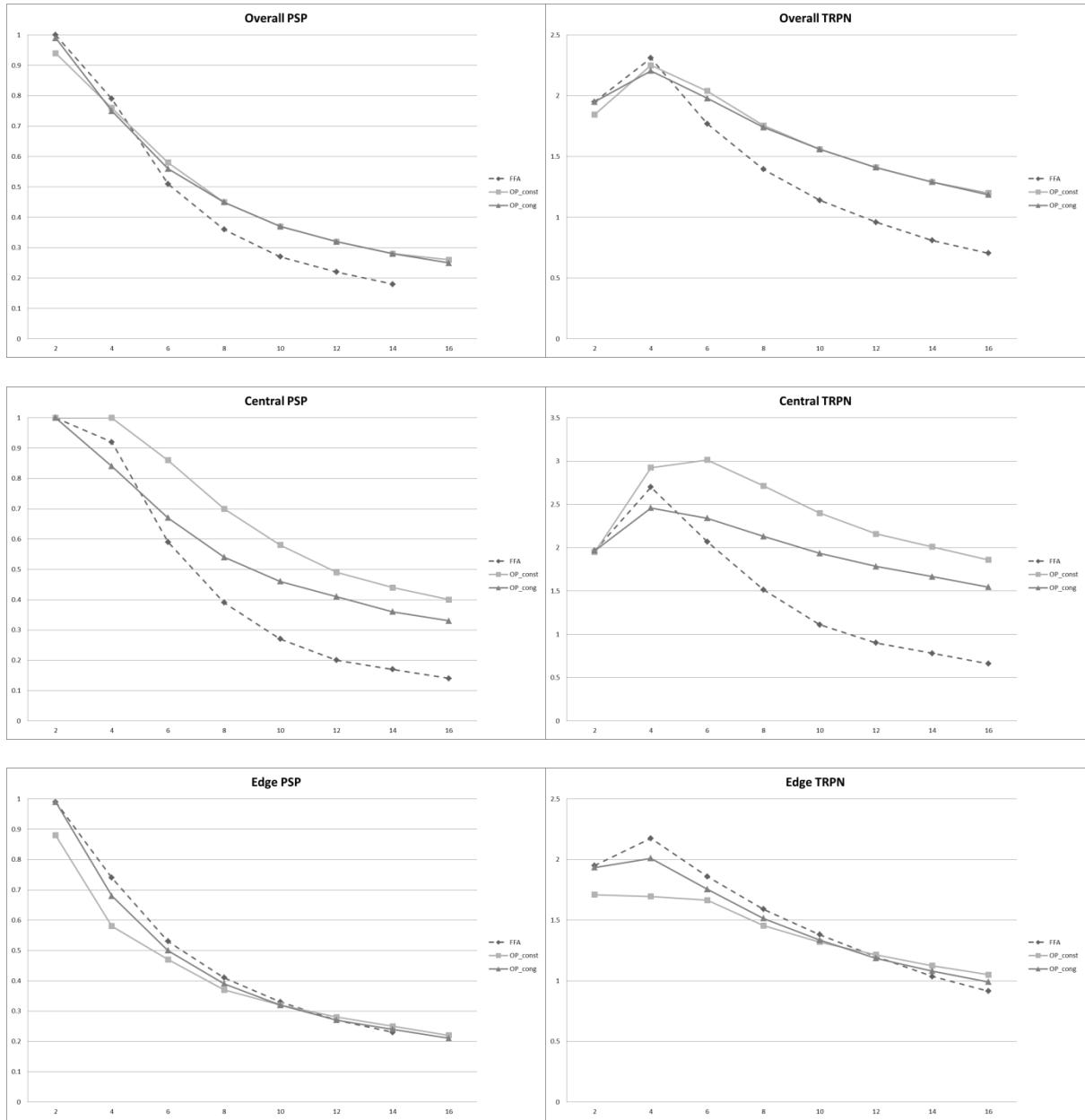


Figure 12: PSP and TRPN (pps) for *FFA*, *OP-const* and *OP-cong* schemes vs packet arrival rate

For the nodes on the edge, the PSP and TRPN improvement is less clear. In fact, the PSP and the TRPN of both *OP-const* and *OP-cong* is decreased at low packet arrival rates. This is due to the fact the credit mechanisms tend to disadvantage nodes on the periphery due to the fact that they do not experience many opportunities to earn credit. For higher packet arrival rates the performance is the same.

The result for the overall network PSP and TRPN is that both the *OP-const* and *OP-cong* schemes improve the performance of the network to the same degree. The *OP-cong* scheme has a slightly better performance than *OP-const* at lower packet arrival rates. This is because at low packet arrival rates, the *OP-cong* scheme approximates the *FFA* scheme because all queues are generally empty and thus the price per service is generally close to zero.

Figure 13 illustrates the effect of the *OP-const* and *OP-cong* schemes on total average latency (TAL) and energy rate per node (ERPN). As we can see, for all nodes both centrally and peripherally located, the *OP-const* and *OP-cong* schemes improve the network latency. The *OP-cong* scheme results in the best latency on the network as the pricing is directly related to the congestion at a node. The *OP-cong* scheme charges a high price to use nodes that are congested. The result is that packets traversing congested links tend to get dropped at the originating node more often freeing up those resources. This leads to less latency on the network.

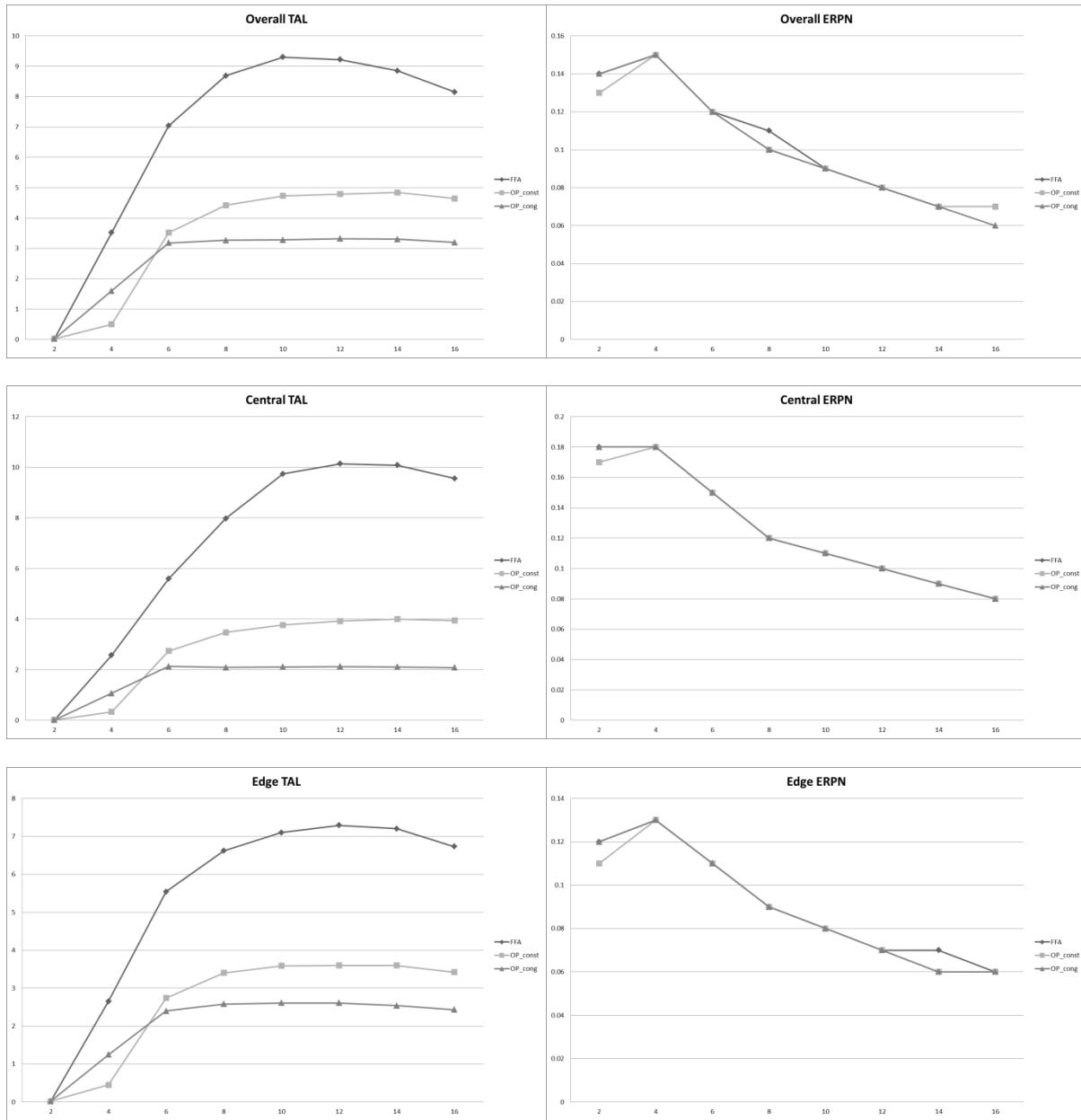


Figure 13: TAL (s) and EPN (J/s) for *FFA*, *OP-const* and *OP-cong* schemes vs packet arrival rate

The *OP-const* and *OP-cong* schemes have little effect on the EPN of any of the nodes, whether averaged across the whole network or considering the central or edge nodes separately.

Figure 14 shows the reciprocity and rate of the *OP-const* and *OP-cong* schemes compared to *FFA*. *OP-const* enforces a higher reciprocity by limiting nodes that do not serve the network from sending their own packets. The reciprocity of *OP-const* is not as severe as *TFT*. *OP-cong* has a very similar reciprocity to *FFA* for all values of packet arrival rate. All reciprocity values are high for all schemes.

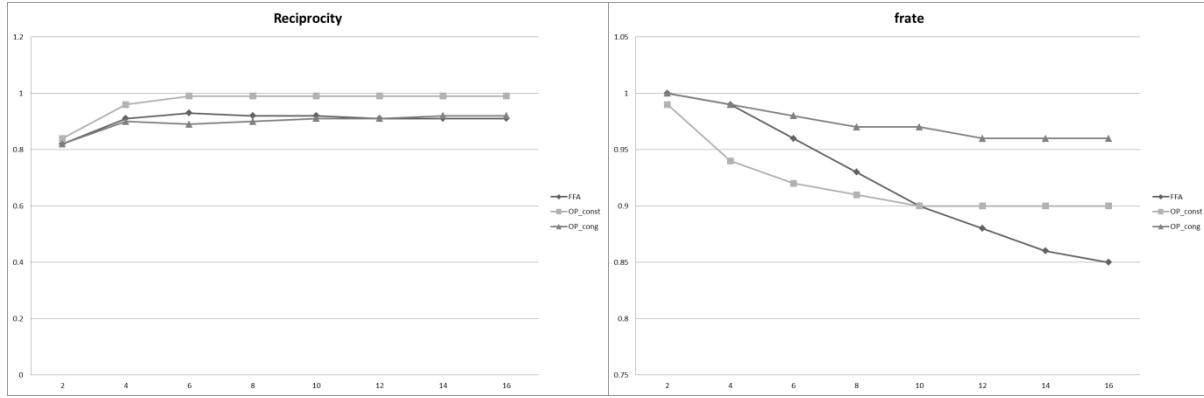


Figure 14: Reciprocity and rate of *FFA*, *OP-const*, and *OP-cong*

The fairness in PSP (rate) is the same for all schemes at low packet arrival rates. This is because at low arrival rates, there is no load on the network and all nodes have a PSP of 1. As the packet arrival rate increases, the rate of the *FFA* scheme drops steadily. The *OP-cong* scheme also exhibits this behaviour, but at a less rapid decline. The *OP-const* scheme drops more sharply at first, but then steadies out at a rate value of 0.9 which it maintains as the packet arrival rate increases. These results indicate that the *OP-const* and *OP-cong* schemes have a positive influence on the reciprocity and rate of the network since the network uses its resources more fairly amongst the various nodes.

OP-const improves network performance in two respects. As packet arrival rates increase, it results in higher PSP and TRPN over that of *FFA*. It also greatly improves the TAL of the network for packet arrival rates. *OP-cong* shows an improvement in the PSP and TRPN of *OP-const* for low packet arrival rates and comparable PSP and TRPN for higher packet arrival rates. The TAL under *OP-cong* is also improved over that of *OP-const*.

4.2. Experiment 2: What is the effect of the redistribution mechanism

The aim of the second experiment is to investigate the effect of the redistribution mechanism on network performance.

All schemes involving credit redistribution (*OP-const*, *OP-cong*, *OP-cred*, *OP-energy*, *OP-const-cred-routing* and *OP-cong-routing*) are applied in a model of a 50-node network. The nodes are initially placed throughout a $875m \times 875m$ plane and move according to a random waypoint model. The node batteries are constantly recharged. Each experiment is independently replicated 15 times and the 95% confidence interval half-widths are computed. The results displayed are the average of these 15 replications. The confidence intervals are not illustrated in the figures as they are too small. Each replication runs for a total of 1500s. Nodes do not send any data for the first 500 seconds of the

simulation. This time allows the routing protocol to establish stable conditions before the network begins carrying traffic.

In the first iteration of the experiment the redistribution mechanism is disabled, and in the second iteration it is enabled. A comparison of the network performance of *OP-const* and *OP-cong* is presented below as these are the two protocols we have identified as being most effective in stimulating cooperation and improving network performance.

4.2.1. *OP-const*: the effect of the redistribution mechanism

Figure 15 shows the effect of the redistribution mechanism on the *OP-const* protocol. We can see that for PSP and TRPN, the redistribution mechanism has a positive effect at low arrival rates. This is due to the fact that few packets are exchanged at low arrival rates, and thus some nodes do not get the opportunity to earn credit. The redistribution mechanism ensures that all nodes have at least some credit with which to send packets.

At higher packet arrival rates, the credit in the system is distributed by virtue of the fact that the nodes are exchange packets (and thus credit) frequently. In the case in which nodes are mobile, the movement of the nodes is in itself a redistribution mechanism where nodes at the edge of the network with low credit balances move to the centre of the network and acquire credit by relaying packets on transit routes.

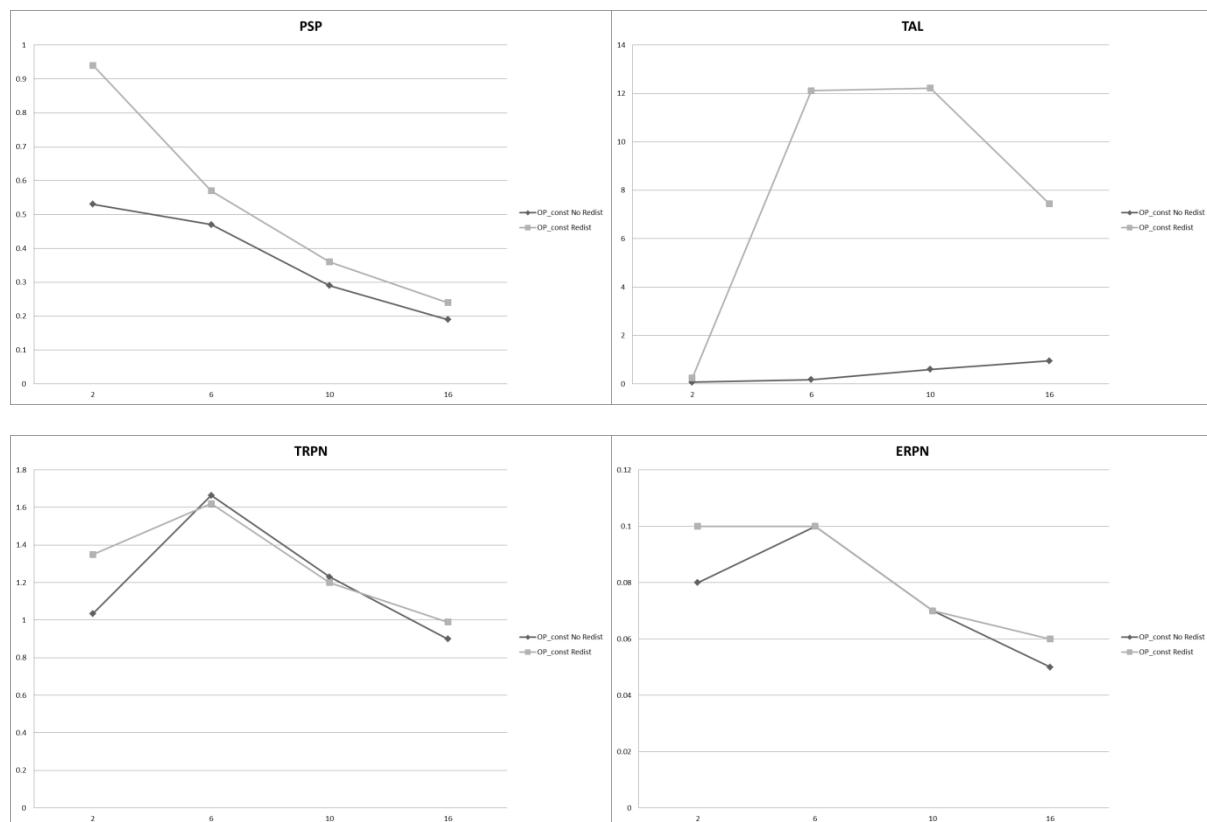


Figure 15: *OP-const*: PSP, TRPN(pps), TAL(s) and ERPN(J/s) for redistribution vs no redistribution

Note that the redistribution mechanism has a significant effect on the TAL of the network. When the redistribution mechanism is enabled it causes the TAL to increase for higher packet arrival rates. When the redistribution mechanism is enabled, there are fewer nodes that are inhibited from sending packets and thus there are more packets in the network giving rise to congestion and delays. The ERPN is hardly affected by the redistribution mechanism.

4.2.2. OP-cong: the effect of the redistribution mechanism

Figure 16 shows the results of the redistribution and no redistribution experiment for the *OP-cong* protocol. In this protocol we see an improvement in the PSP and TRPN for all packet arrival rates.

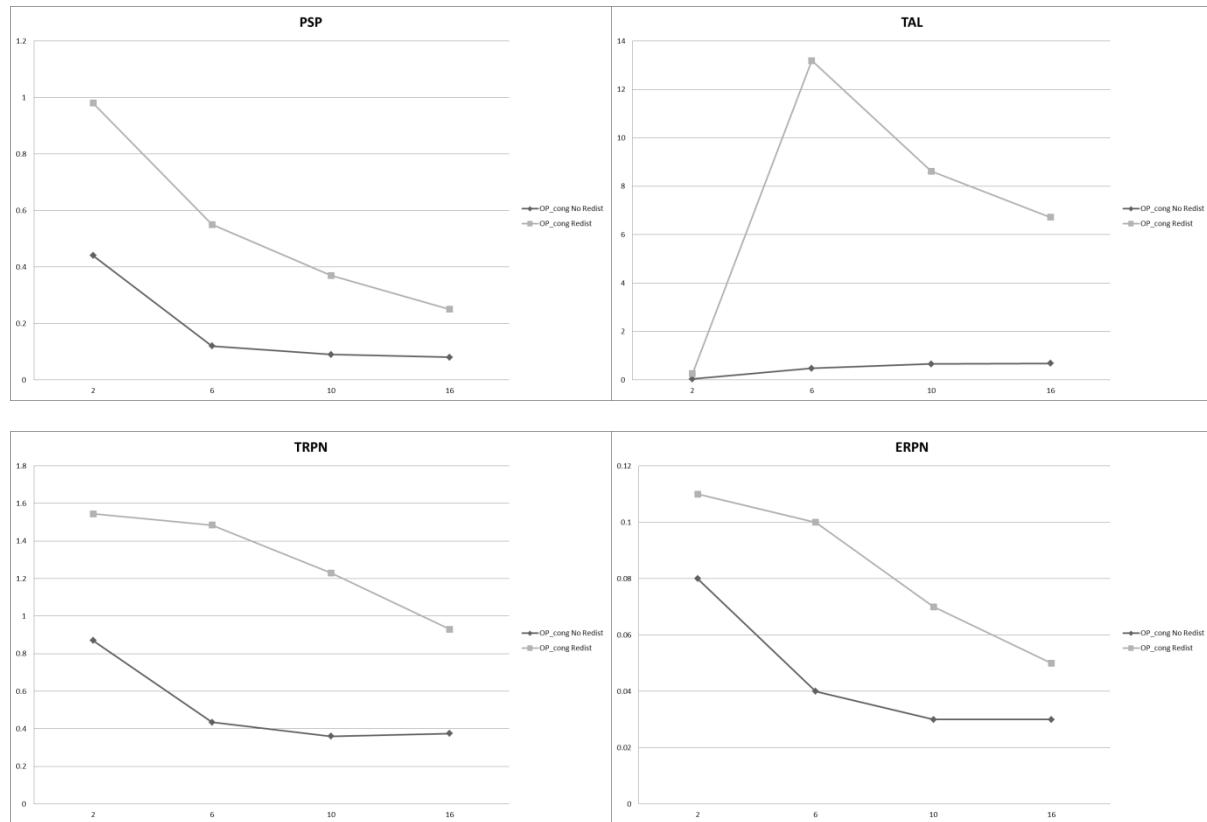


Figure 16: *OP-cong*: PSP, TRPN(pps), TAL(s) and ERPN(J/s) for redistribution vs no redistribution

The TAL is increased with the redistribution mechanism enabled as more packets are released onto the network. The ERPN is also increased because nodes are expending more energy in sending and receiving the increased traffic load on the network.

The redistribution mechanism is an important aspect of the credit-based mechanisms. For both *OP-const* and *OP-cong* it results in better network performance when it is enabled.

4.3. Experiment 3: What is the effect of the redistratate

The aim of the third experiment is to investigate the effect of the redistratate parameter on network performance.

All schemes involving credit redistribution (*OP-const*, *OP-cong*, *OP-cred*, *OP-energy*, *OP-const-cred-routing* and *OP-cong-cong-routing*) are applied in a model of a 50-node network. Nodes are scattered throughout a $875m \times 875m$ plane and are moving according to a random waypoint model. Node batteries are constantly recharged. Each experiment is independently replicated 15 times and the 95% confidence interval half-widths are computed. The results displayed are the average of these 15 replications. The confidence intervals are not illustrated in the figures as they are too small. Each replication runs for a total of 1500s. Nodes do not send any data for the first 500 seconds of the simulation. This time allows the routing protocol to establish stable conditions before the network begins carrying traffic.

The network performance is measured for packet arrival rates of 2, 6 and 12 pps. The network performance is presented for varying values of redistribution rate varying from 0.01 to 0.5. The results for *FFA*, *OP-const* and *OP-cong* are presented below as these were the two most effective protocols for enhancing the network performance.

The redistribution rate was defined in Chapter 3 as follows. Let B denote the target credit balance and let $B_i(t)$ denote the credit balance at node i at time t . At every redistribution event, the credit at node i is adjusted as follows:

$$B_i(t + \Delta) = B_i(t) + \delta\Delta(B - B_i(t))$$

where δ is called the redistribution rate. It is the fraction of excess or deficit credit that is created or destroyed per unit time. Δ is the redistribution interval. It is the time interval between successive redistribution events.

For low values of redistribution rate, the amount of credit redistributed is low and give rise to inequality in the credit balance of the nodes. This results in some nodes having a surplus of credit and being able to send all their packets, while other nodes have a credit deficit and are unable to send their packets. When the parameter is low, the network tends to behave like the *TFT* protocol. Some nodes are limited in what they can send, while other nodes by virtue of their location in the network are able to send all their packets.

As δ increases, so the redistribution of the credits amongst the nodes becomes more effective and the disparity between nodes becomes less pronounced. When δ is high, credit is constantly being redistributed around the network and there is little inequality between the nodes. All nodes in unfavourable locations are frequently reimbursed with credit and as a result all nodes tend to have enough credit to send their packets. For high values of δ , the network tends to behave like the *FFA* protocol.

4.3.1. *OP-const: the effect of redistribution rate*

Figure 17 presents the results of varying δ for the *OP-const* protocol. For low packet arrival rates we can see that the PSP and TRPN increase as the δ increases. This is because the network tends to behave like the *FFA* protocol, and for low packet arrival rates this protocol provides the best performance.

The network performance is good across a wide range of δ values. We can see this in the consistent PSP, TRPN and ERPN for δ values ranging from 0.01 to 0.5. The TAL increases for lower values of δ as very low values of δ cause the credit based scheme to behave like TFT and thus reduce congestion. Once the δ value is greater than 0.05 the TAL becomes more consistent for changing values of δ .

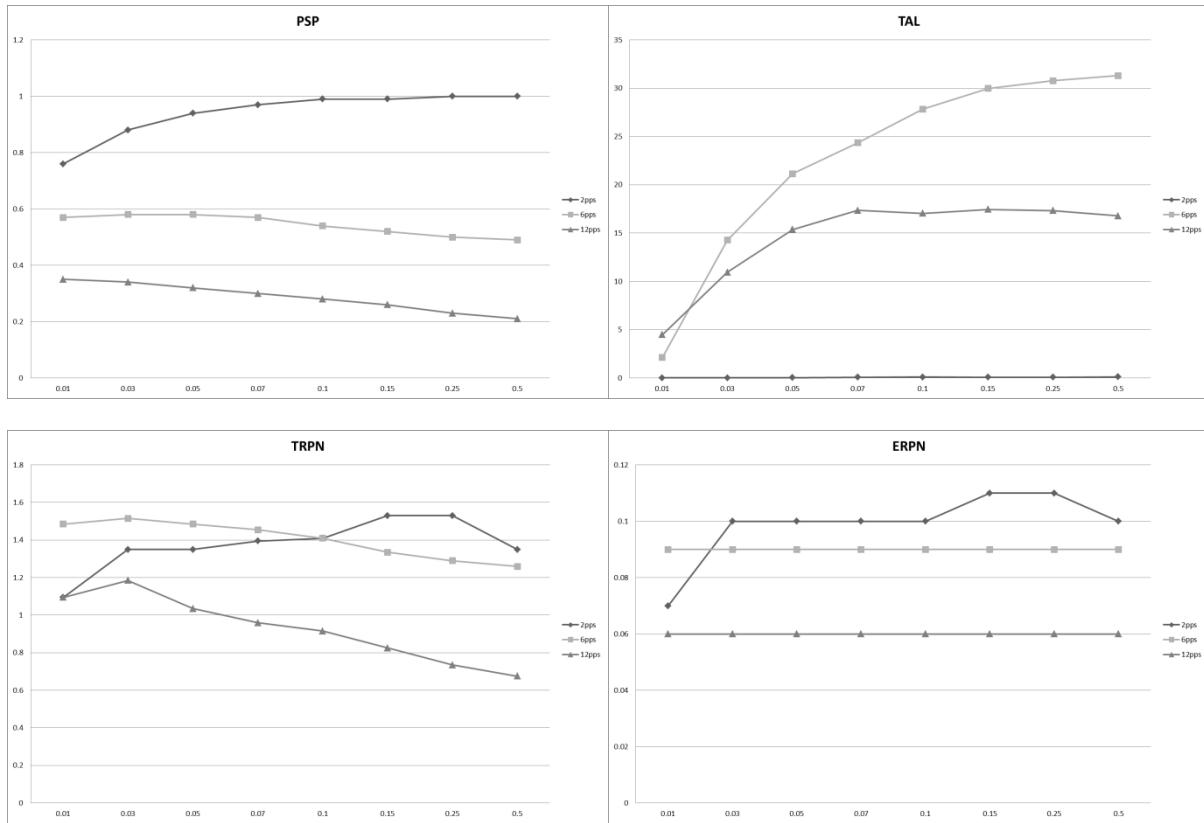


Figure 17: OP-const: PSP, TRPN (pps), TAL (s) and ERPN (J/s) for varying redistratate

The fact that network performance is good across a wide range of δ values is an important characteristic of the redistribution mechanism because it shows that a generic value of δ will result in acceptable network performance and that each network implementing the redistribution and credit based mechanisms will not need to be specifically tuned. We select a δ value of 0.05 for the remaining experiments in this thesis.

4.3.2. OP-cong: the effect of redistribution rate

Figure 18 presents the results for varying δ for the OP-cong protocol. The effect of the redistribution rate in the OP-cong scheme is very similar to that of OP-const in the previous section.

The PSP, TRPN and ERPN are all consistent for δ values ranging from 0.01 to 0.5. The TAL increases for values of δ less than 0.05 as these low values of tend to cause the network to behave like TFT as very little credit is redistributed through the network.

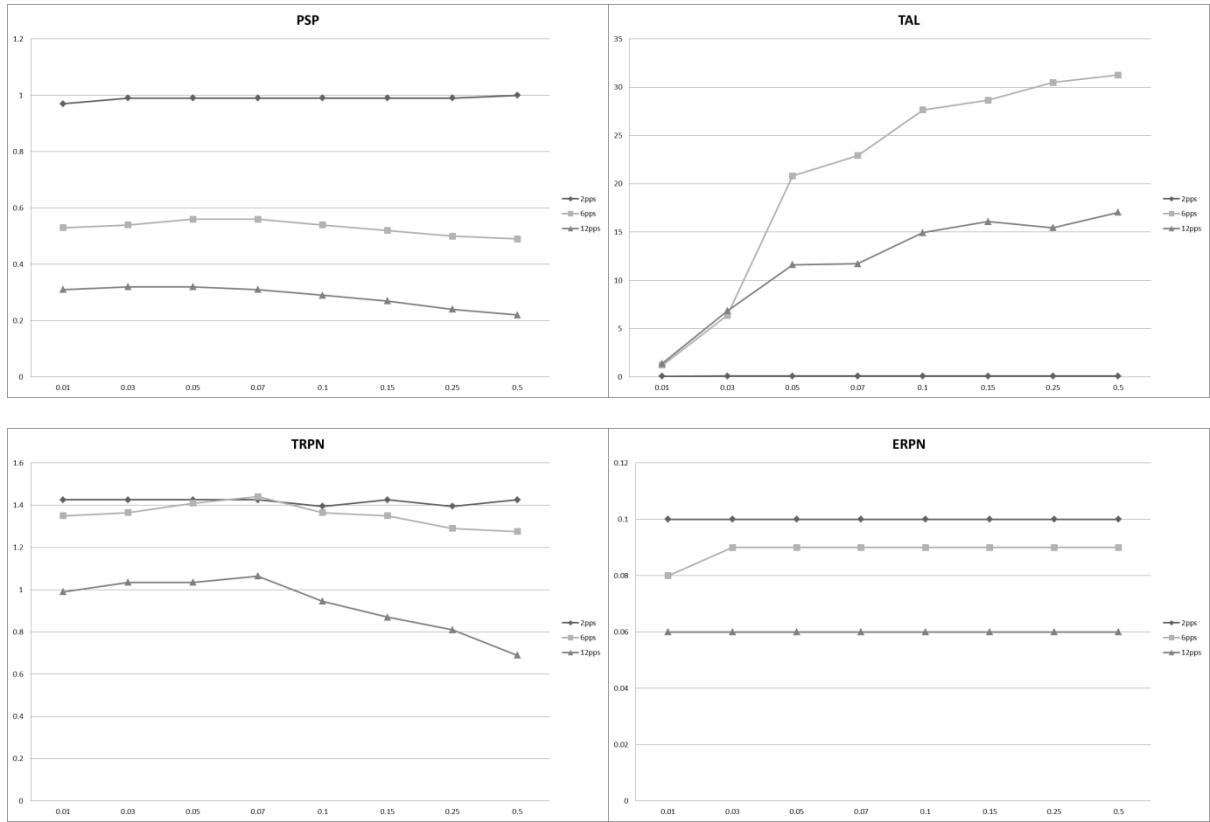


Figure 18: OP-cong: PSP, TRPN (pps), TAL (s) and ERPN (J/s) for varying redistratate

The consistency of the network performance across a wide range δ of values shows that the *OP-cong* protocol does not need to have the δ specifically tuned for each network it is implemented in, but rather that a generic value can be expected to achieve good performance. We select a δ value of 0.05 for the remainder of the experiments in this thesis.

4.4. Experiment 4: What is the effect of the target credit balance

The aim of the fourth experiment is to investigate the effect of the target credit balance on network performance.

All schemes involving credit redistribution (*OP-const*, *OP-cong*, *OP-cred*, *OP-energy*, *OP-const-cred-routing* and *OP-cong-cong-routing*) are applied in a model of a 50-node network. Nodes are scattered throughout a $875m \times 875m$ plane and are moving according to a random waypoint model. Node batteries are constantly recharged. Each experiment is independently replicated 15 times and the 95% confidence interval half-widths are computed. The results displayed are the average of these 15 replications. The confidence intervals are not illustrated in the figures as they are too small. Each replication runs for a total of 1500s. Nodes do not send any data for the first 500 seconds of the simulation. This time allows the routing protocol to establish stable conditions before the network begins carrying traffic.

The network performance is measured for packet arrival rates of 2, 6 and 12 pps. The network performance is presented for varying values of target credit balance varying from 10 to 1600. The results for *FFA*, *OP-const* and *OP-cong* are presented below as these were the two most effective protocols for enhancing the network performance.

When the target credit balance is low the credit mechanism tends to behave towards *TFT*. This is because the constraints on the nodes are tight as each node has only a little credit with which to send packets. As target credit balance increases, this constraint becomes less stringent. For high values of target credit balance, the credit mechanism tends to behave towards *FFA* because every node has enough credit to always send all packets.

4.4.1. *OP-const*: the effect of target credit balance

Figure 19 shows the effect of varying the target credit balance.

For low packet arrival rates, the PSP and TRPN increases with increasing target credit balance. This is because at low packet arrival rates, the network is not under any pressure with regards to resources and the most effective protocol in terms of network performance is *FFA*. As the target credit balance increases, so the credit protocol tends to behave more towards *FFA* and the PSP and TRPN improve.

For higher packet arrival rates, this is not the case. We can see for packet arrival rates of 6 and 12 that the PSP and TRPN improve up to a certain target credit balance and then begin to decrease as the target credit balance continues to increase. This is due to the fact that at higher packet arrival rates an excess of credit can admit too many packets to the network, giving rise to congestion.

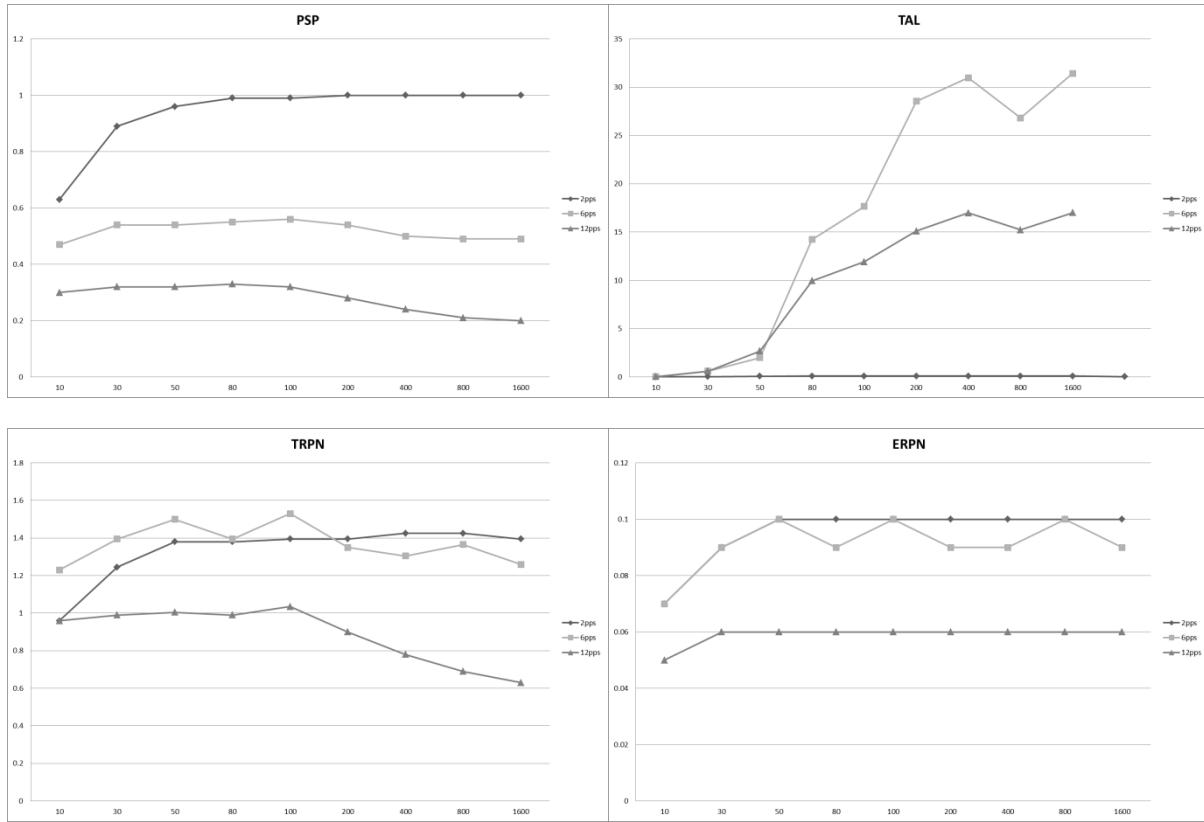


Figure 19: *OP-const*: PSP, TRPN (pps), TAL (s) and ERPN (J/s) for varying target credit balance

From figure 19 this appears to be when the target credit balance is approximately 100.

As the target credit balance increases so the restriction on nodes sending their packets becomes less severe. With an increasing number of packets being sent into the network, there will be an increase in latency.

The ERPN increases as the amount of traffic in the network increases. This can be seen until the target credit balance increases beyond the value of 100. At higher values than this, congestion in the network means that the throughput of the network stays the same with increasing target credit balance and so the ERPN becomes constant.

4.4.2. *OP-cong*: the effect of target credit balance

Figure 20 shows the effect of varying the target credit balance on the *OP-cong* protocol.

The credit mechanism tends to behave like *TFT* for low values of the target credit balance and like *FFA* for high values of the target credit balance. The behaviour exhibited by the *OP-cong* protocol is similar to that of the *OP-const* protocol. For low packet arrival rates the PSP and TRPN increase with increasing target credit balance as the protocol tends towards *FFA*.

For high packet arrival rates we see an optimal target credit balance around 100 where the PSP and TRPN peak. For target credit balances higher than this, the PSP and TRPN begin to decrease with increasing packet arrival rates.

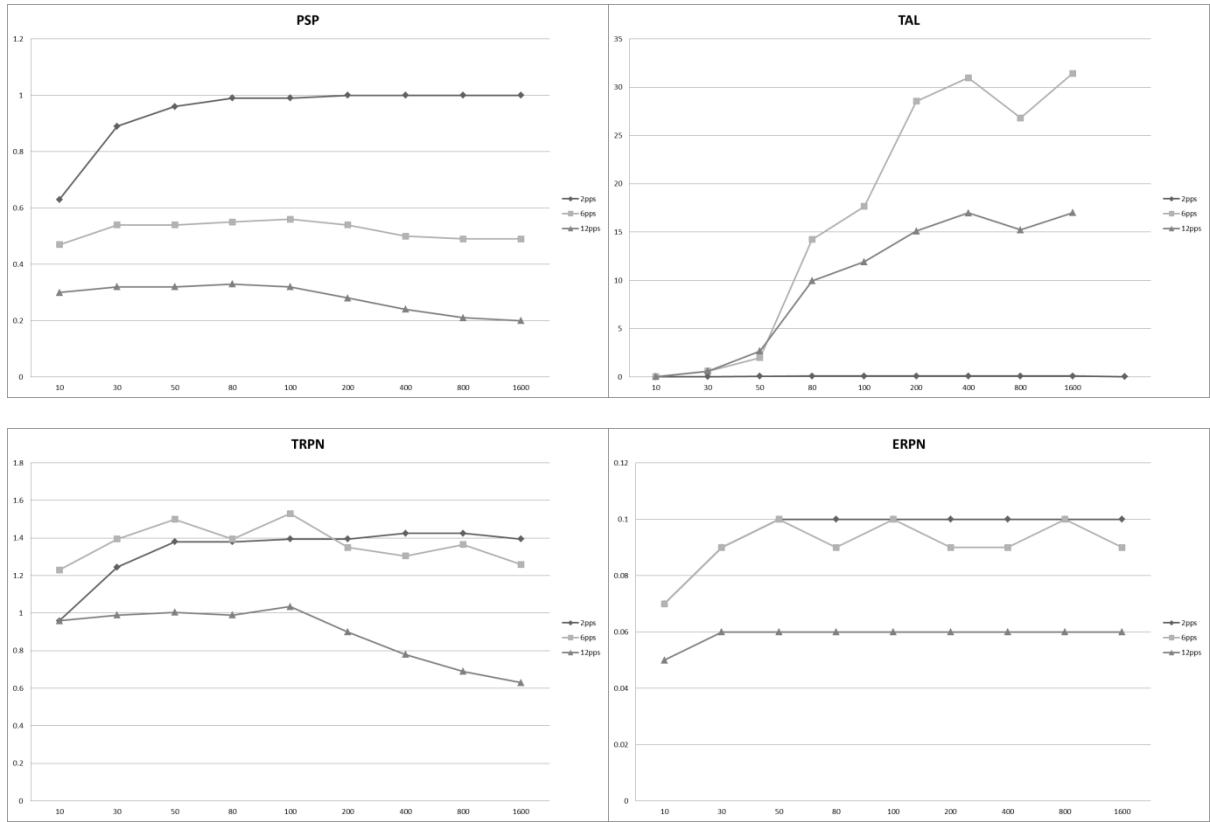


Figure 20: OP-cong: PSP, TRPN (pps), TAL (s) and ERPN (J/s) for varying target credit balance

The TAL increases as the target credit balance increases due to the fact that more packets are being sent into the network and thus create more congestion. The ERPN increases until the optimal target credit balance value and then remains stable with increasing values of target credit balance.

The target credit balance affects the performance of the credit-based schemes. For low values of target credit balance the schemes are highly restrictive and tend toward *TFT* behaviour. For high values of target credit balance the schemes become less restrictive and tend toward *FFA*. For the 50-node network model under investigation, for both *OP-const* and *OP-cong* the optimal value of target credit balance is 100.

4.5. Experiment 5: What is the effect on the use of resources in the network

The aim of the fifth experiment is to investigate the effect of the credit mechanisms on the conservation of resources of the network. Experiments are not replicated to confirm statistical accuracy of the results. The objective of this experiment is to obtain a view into the dynamics of the network during the course of a single simulation and get some insight into how well the credit based schemes conserver network resources.

All schemes are applied in a model of a 50-node network. Nodes are immobile and are located at the centres of their respective $125m \times 125m$ cells on a $875m \times 875m$ plane. Node batteries are not recharged. Each node has a limited amount of energy. When a node runs out of energy, that node

switches off and the simulation continues without it. Nodes do not send any data for the first 500 seconds of the simulation. This time allows the routing protocol to establish stable conditions before the network begins carrying traffic. This can be seen in Figures 21 to 33 as the instantaneous PSP and Total Packets Sent remain at zero for the first 500s.

The network performance is measured for packet arrival rates of 2, 6 and 12 pps. We observe the instantaneous PSP, the overall network throughput and the average energy per node to understand the effect that the various protocols have on how resources are shared. The results for *FFA*, *TFT*, *OP-const* and *OP-cong* are presented below.

4.5.1. Low packet arrival rates

The credit based schemes have a favourable impact on the use of resources in the network. Figure 21 shows the instantaneous PSP of the network for a packet arrival rate of 12pps for the *FFA*, *TFT*, *OP-const* and *OP-cong* schemes.

In *FFA* we see that the network performance is poor with a stable PSP around 0.2. This is due to the high congestion that results under the *FFA* scheme at high packet arrival rates. The resources of the network are used quickly as there are no restrictions on nodes attempting to send traffic. At approximately 630s the nodes begin to run out of energy and by 670s there are only 12 operational nodes left. The PSP of the network drops to close to zero once the nodes begin running out of energy. This occurs at approximately 680s. There is a small amount of activity remaining shown by the PSP of 0.05 for the remainder of the network's lifetime. This is due to the exchange of packets amongst neighbouring nodes.

TFT results in the nodes lasting for longer than any of the other schemes. We see that the nodes die off at a slower rate and in clusters. At 1000s there are still 20 operation nodes left in the network. The performance of the network however, is completely dependent on all nodes being present. As soon as the first cluster of nodes run out of energy, the PSP drops off to practically zero. In the *TFT* scheme the network becomes non-functional at approx. 650s.

The credit based schemes enhance the use of network resources by lengthening the lifetime of the nodes in the network and maintaining the networks functionality once the first clusters of nodes begin running out of energy.

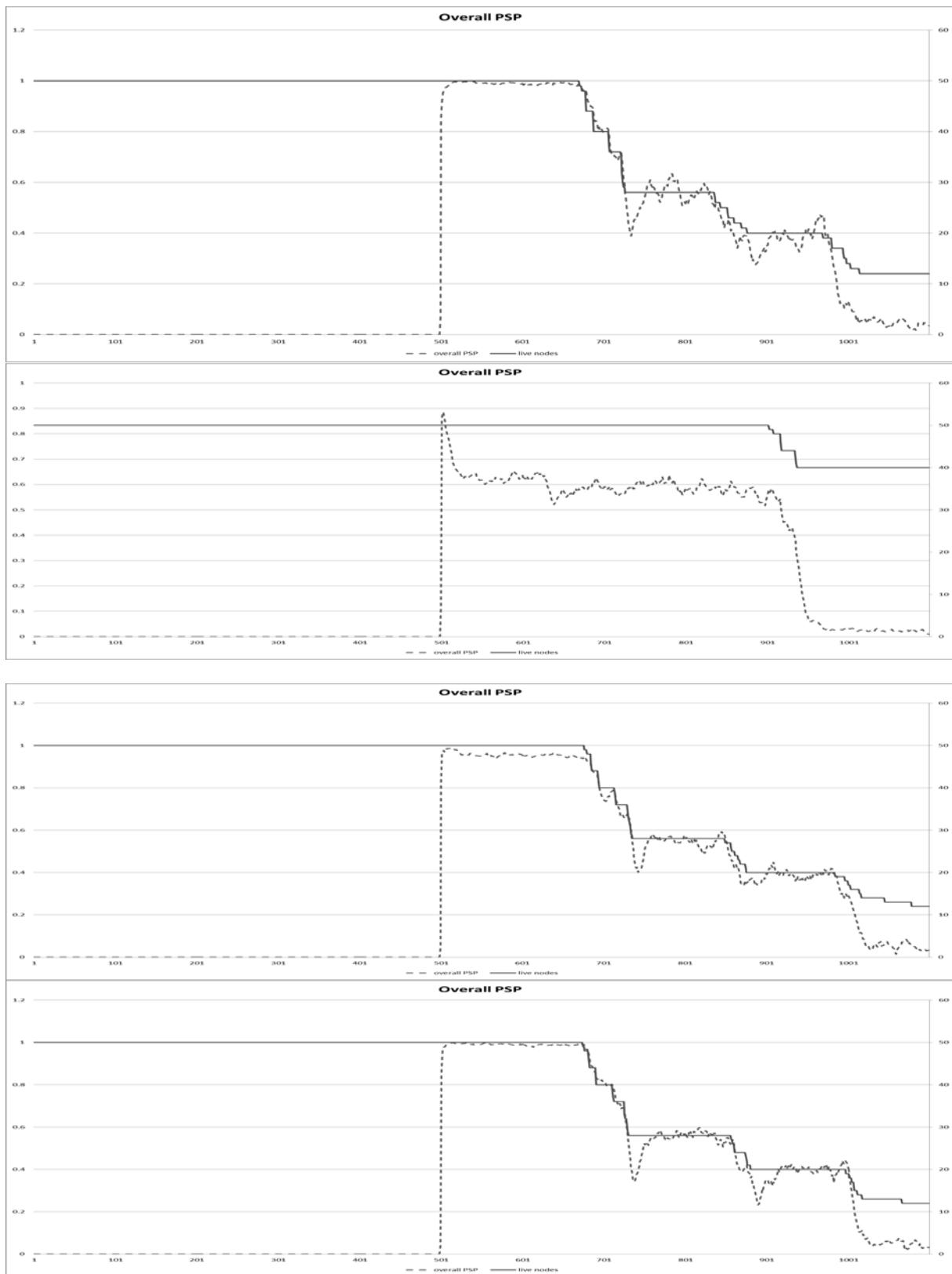


Figure 21: PSP for high packet arrival rates for *FFA*, *TFT*, *OP-const* and *OP-cong*

In *OP-const* we can see that the nodes die off in a staggered manner and that the PSP of the network continues to show activity. In this scheme, the network reaches a state of 12 operational nodes at

approximately 710s. The PSP of the network shows that there is traffic being carried by the network up until this point. Once there are only 12 operational nodes left in the network, the *OP-const* scheme goes into the same state as that of *FFA* in which packets are only exchanged between neighbouring nodes a PSP of 0.05 is maintained for the remainder of the networks lifetime.

In *OP-cong* the nodes also die off in a staggered manner. The network resources last slightly longer than the *OP-const* scheme because the *OP-cong* scheme reduces congestion (and thus the wasteful use of network resources) more effectively than *OP-cong*. The network reaches a state of 12 operational nodes at approximately 790s.

The *TFT* scheme shows an improvement in conserving network resources over *FFA* as there are more nodes that remain alive in the network. This does not translate into the network performing for longer as the PSP falls to zero at 650s in *TFT* compared to it falling to practically zero at 690s in *FFA*. This is due to the face that *TFT* is dependent on all nodes being operational for the network to perform.

The credit based schemes provide an enhancement in the conservation of network resources and in slightly extending the operational lifetime of the network as they result in more nodes remaining operational and the network PSP remaining above zero. In *OP-const*, the PSP falls to close to zero at approximately 710s and in *OP-cong* the PSP falls to close to zero at 790s.

4.5.2. Low packet arrival rates

At low packet arrival rates the resources of the network are not under strain. The *FFA* protocol results in good network performance. The *TFT* protocol results in degraded network performance due to the strict restrictions it imposes on nodes. The *OP-const* and *OP-cong* protocols have similar performance to the *FFA* protocol because at low packet arrival rates these credit based mechanisms mimic *FFA*. *OP-cong* mimics *FFA* even more closely because at low packet arrival rates the prices of services are close to zero because there is no congestion.

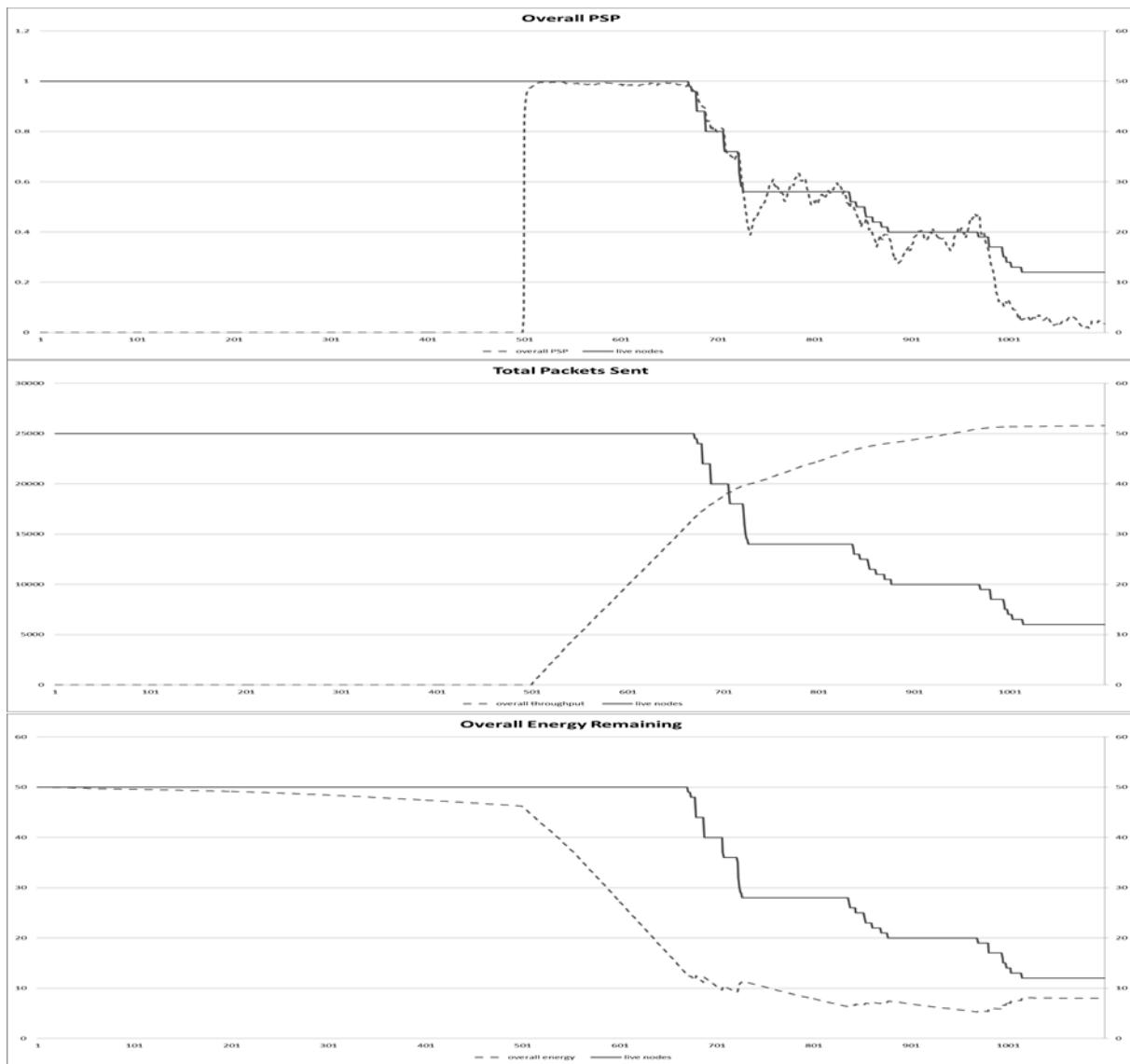


Figure 22 PSP, throughput (packets) and energy (Joules) of *FFA*: low packet arrival rates

Figures 22 to 25 show the PSP, throughput and average energy vs time for a packet arrival rate of 2 pps for protocols *FFA*, *TFT*, *OP-const* and *OP-cong* respectively. At low packet arrival rates *FFA*, *OP-const* and *OP-cong* have similar behaviour. The overall PSP is high while all nodes are still alive. The first cluster of nodes die at around 637s. When the first cluster of nodes run out of energy the overall PSP drops to around 0.5. The first cluster of nodes to die are all located in the centre of the network. This is to be expected because these central nodes carry many routes and thus expend a lot of their resources on serving other nodes. They are the nodes that will run out of energy first. These nodes are a critical part of the network due to the fact that they carry so many routes and once they die the overall network PSP drops significantly.

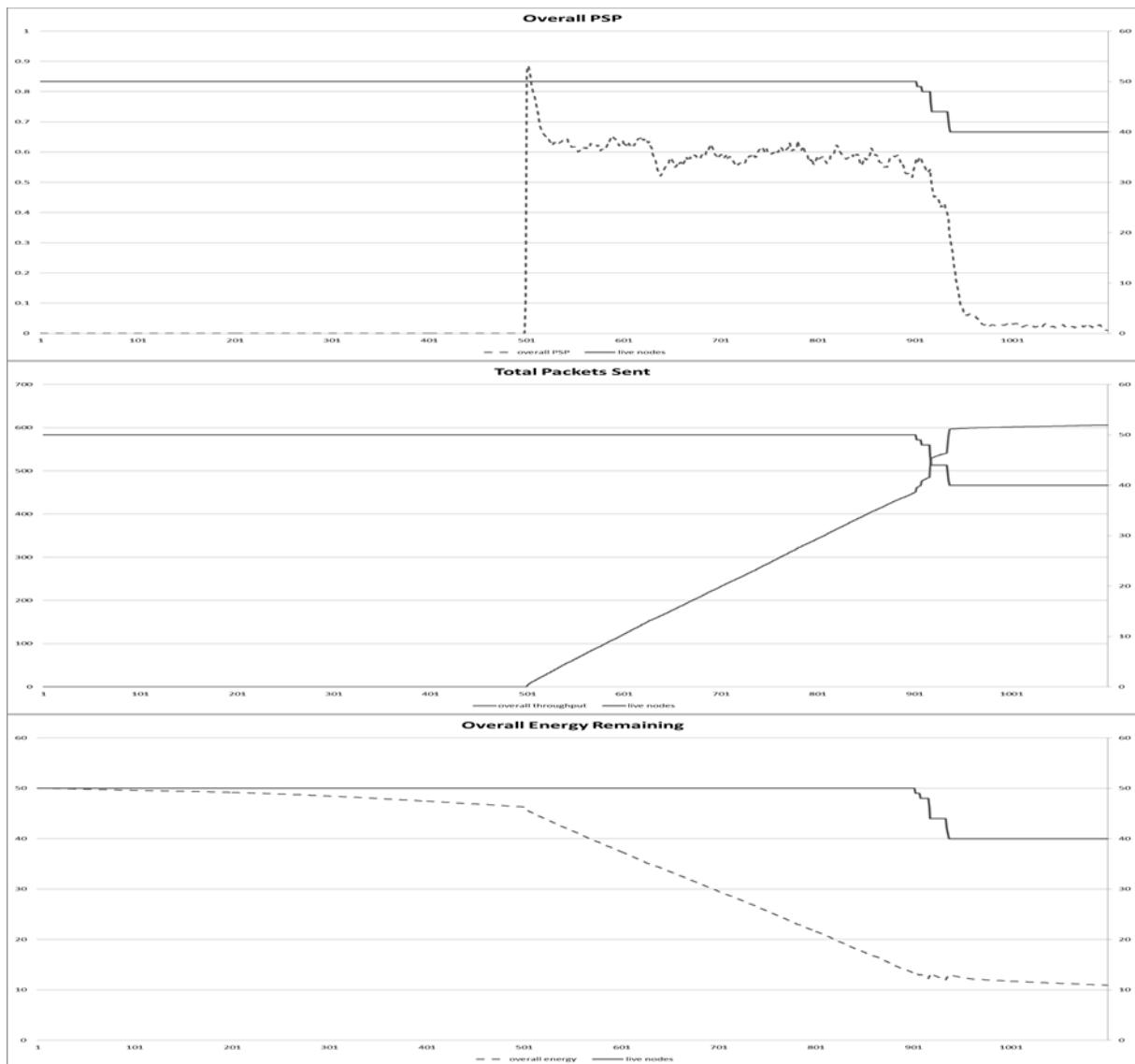


Figure 23 PSP, throughput (packets) and energy (Joules) of *TFT*: low packet arrival rates

In the *TFT* simulation it takes much longer for the first set of nodes to die. The central nodes die first, but they die at around 937 seconds, much later than in the *FFA*, *OP-const* and *OP-cong* simulations. This is due to the fact that the *TFT* mechanism is very restrictive and reduces the amount of transit traffic the central nodes carry on behalf of other nodes. This comes at a price though. When all nodes are alive the overall network PSP in the *TFT* simulation is significantly lower than the *FFA*, *OP-const* and *OP-cong* case.

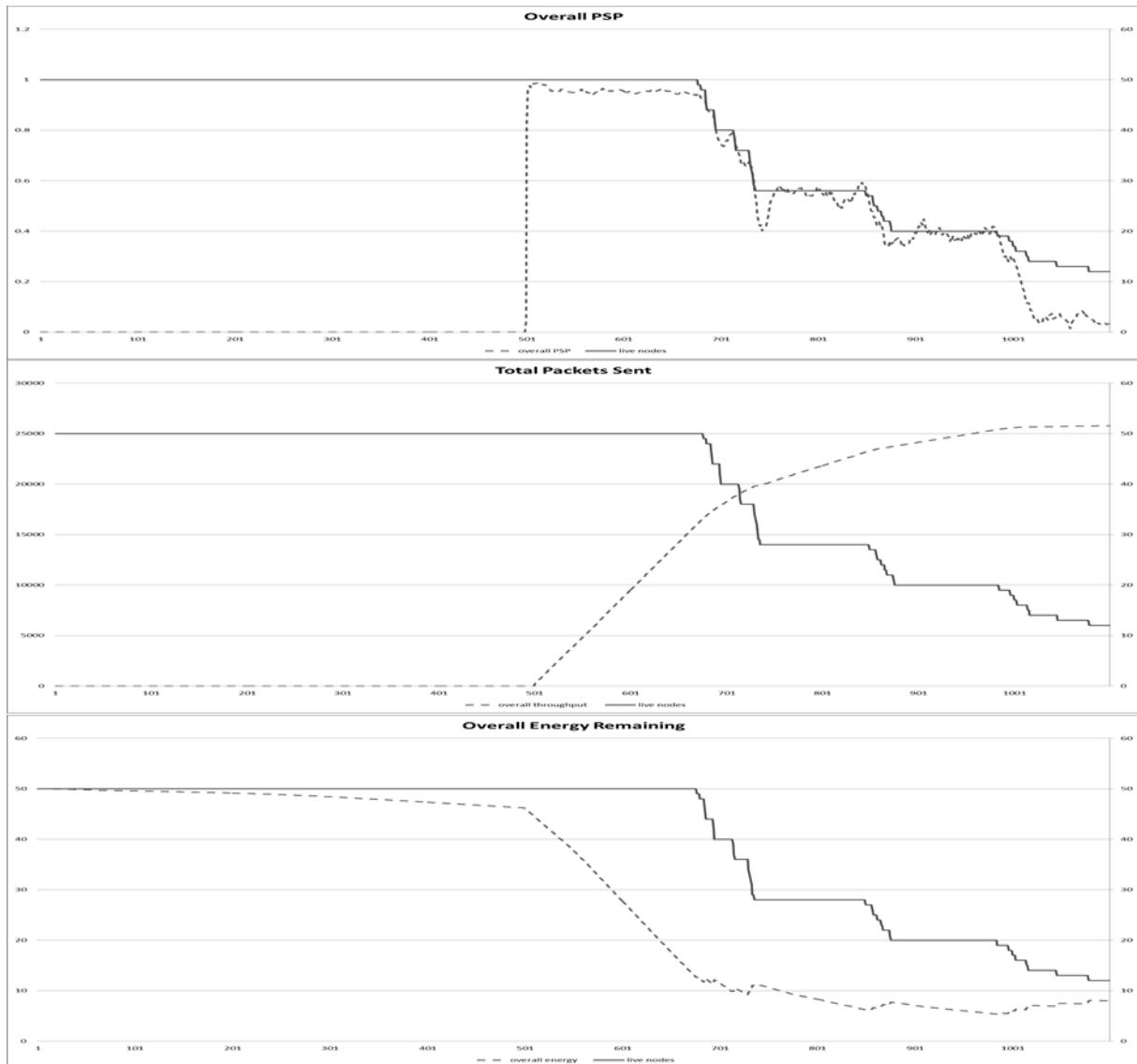


Figure 24 PSP, throughput (packets) and energy (Joules) of OP-const. low packet arrival rates

In *FFA*, *OP-const* and *OP-cong* at around 950s only 12 nodes (all of them on the periphery) are still alive and the network PSP is very low. This is due to the fact that the network becomes partitioned and the only traffic that can successfully flow is between neighbouring nodes.

In *TFT* this condition is reached at a similar time (913 s) but is a very different situation. There are then 40 nodes with energy in the *TFT* stable state, but none of them can send traffic and so these nodes are essentially dead. In the credit-based schemes the network remains functional after the first cluster of nodes die because credit is redistributed by means of the redistribution mechanism.

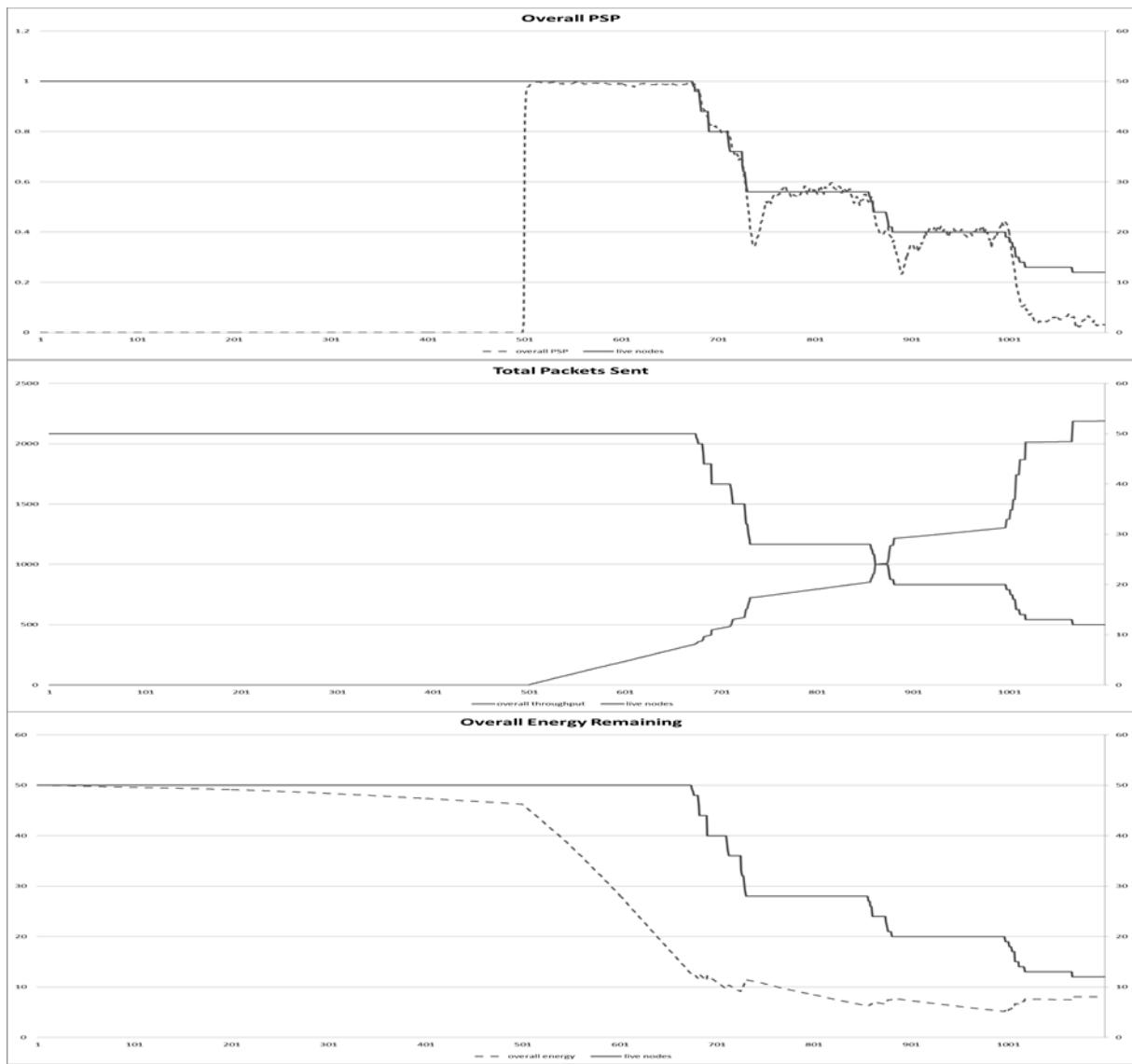


Figure 25 PSP, throughput (packets) and energy (Joules) of OP-cong: low packet arrival rates

It is interesting to note that the nodes die off in clusters causing the number of live nodes in the network to be a step function. Eventually a state is reached in which the network is disjoint and no nodes can send any traffic. At this point, the energy use declines to close to zero as only neighbouring nodes exchange packets.

The rate of energy use is clearly less severe in *TFT*. There is also only one occasion where a cluster of nodes die. Once this occurs, the network becomes non-functional: the edge nodes are unable to service other nodes and thus earn the right to send their own traffic.

At low packet arrival rates, *TFT* extends the lifetime of the nodes in the network but this comes at a cost. The PSP is significantly lower and the overall throughput at the end of the simulation is slightly less when compared to *FFA*, *OP-const* and *OP-cong*.

At low packet arrival rates the total packets sent is close to 25000 packets for all protocols.

4.5.3. Medium packet arrival rates

In the case of medium packet arrival rate of 6pps the resources of the network are under some strain.

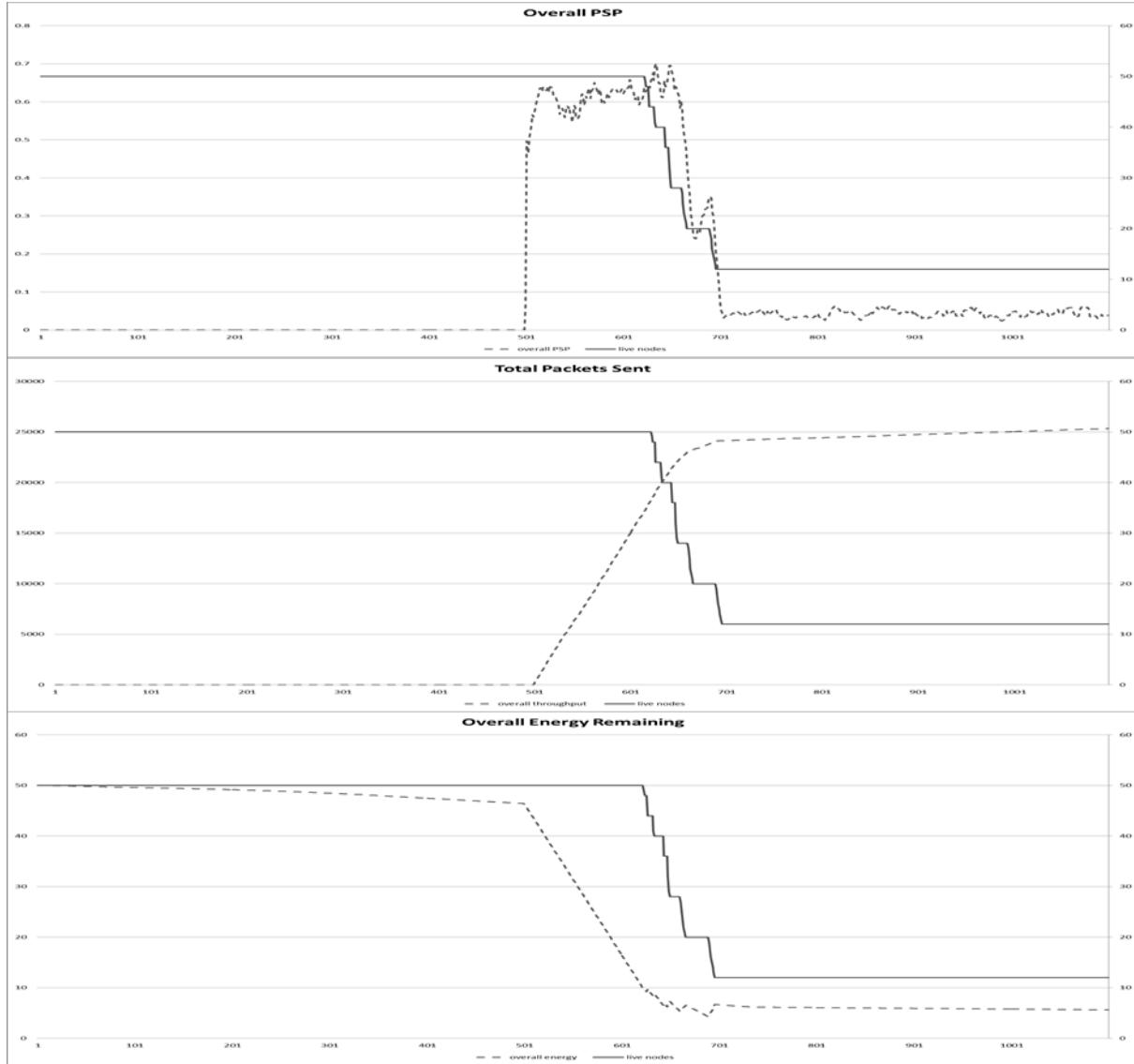


Figure 26 PSP, throughput (packets) and energy (Joules) of *FFA*: medium packet arrival rates

Figures 26 to 29 show the PSP, throughput and average energy versus time for a packet arrival rate of 6 pps for protocols *FFA*, *TFT*, *OP-const* and *OP-cong* respectively. We see that the *FFA* protocol results in an overall PSP of approx. 0.6 while all the nodes are still alive. The first cluster of nodes dies at around 637s in the *FFA* protocol. Once this cluster of nodes dies the rest of the nodes in the network run out of energy very quickly. Eventually there are only 12 edge nodes left operating by 700s.

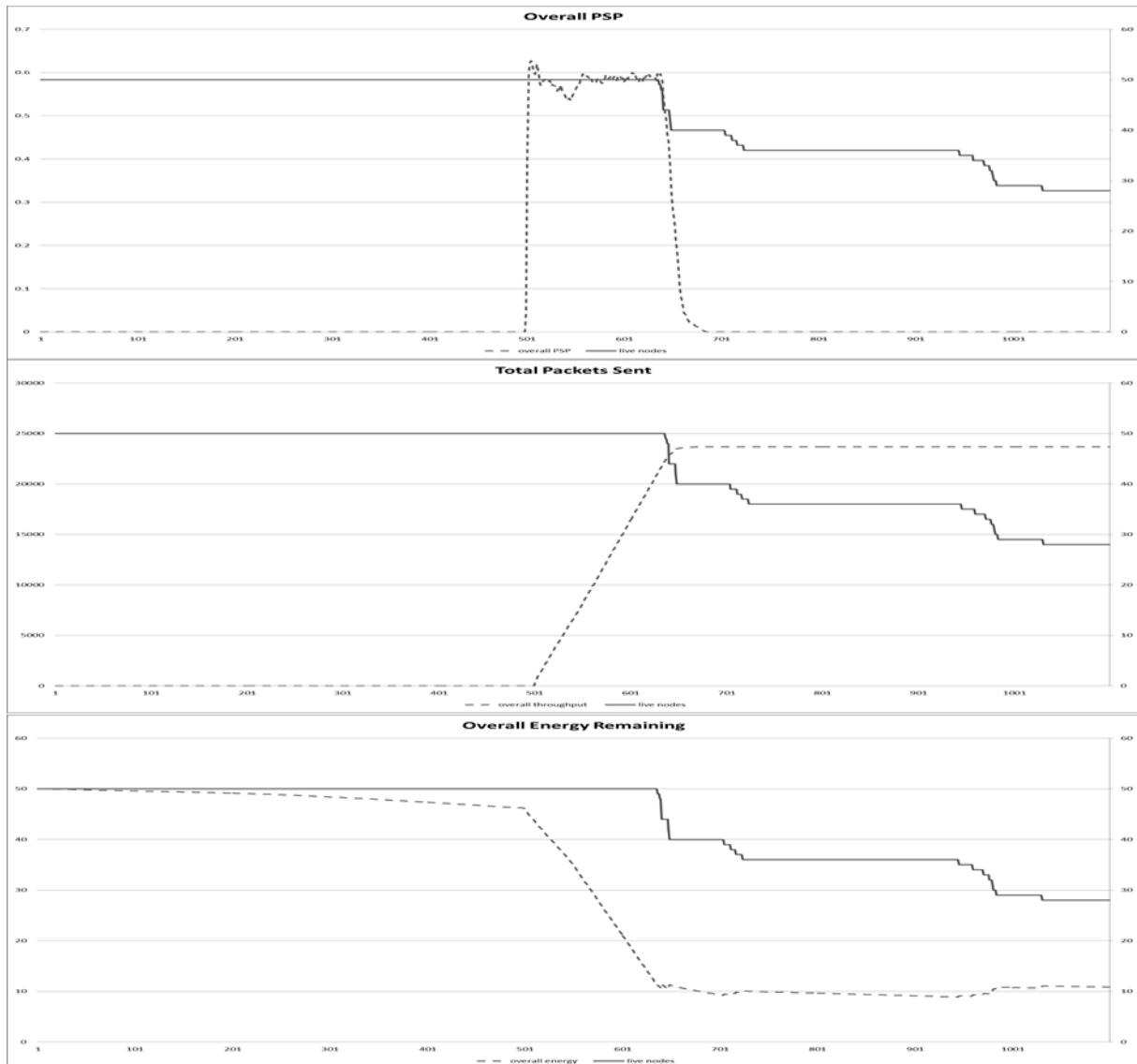


Figure 27 PSP, throughput (packets) and energy (Joules) of *TFT*: medium packet arrival rates

In *TFT* we see that the overall PSP of the network while all nodes are still alive is also approximately 0.6. The central nodes run out of energy at 680s lasting 50s longer than the *FFA* case. Once the first cluster of nodes has run out of energy in *TFT*, the network is non-functional. The overall PSP falls to near zero as soon as the central cluster of nodes run out of energy. The overall lifetime of the network is less than in the *TFT* case.

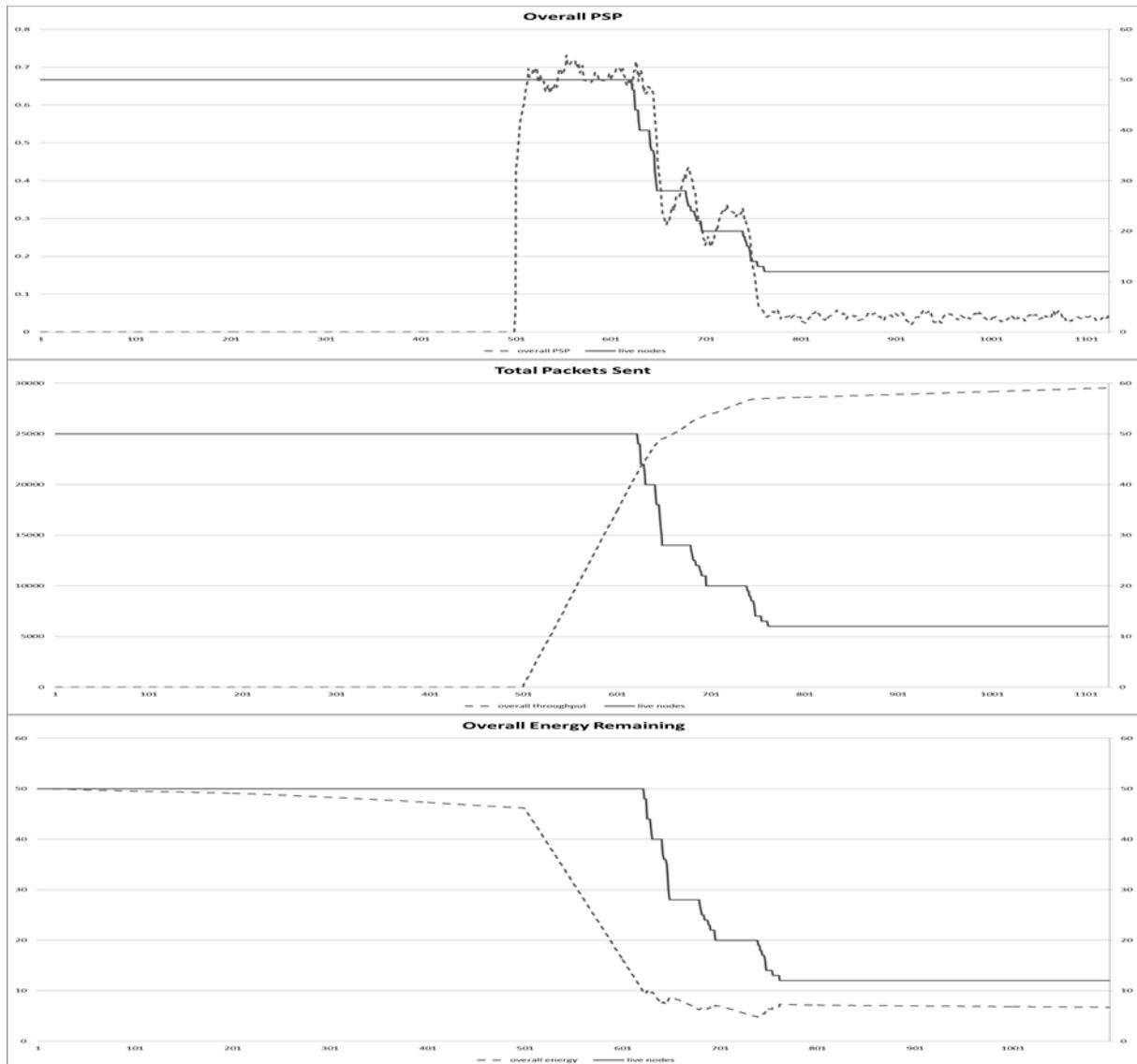


Figure 28 PSP, throughput (packets) and energy (Joules) of *OP-const*. medium packet arrival rates

In the *OP-const* simulation, the overall PSP of the network is slightly improved over that of *FFA*. While all the nodes are alive the overall PSP is approximately 0.65.

The first cluster of nodes dies at a similar time to *FFA* and *TFT* but the degradation in the network is less severe. The overall PSP falls to around 0.35 and fluctuates there before it falls to practically zero when the last cluster of nodes die. *OP-const* provides a slight improvement to the network's performance over *FFA* at 6pps, and the network lasts slightly longer.

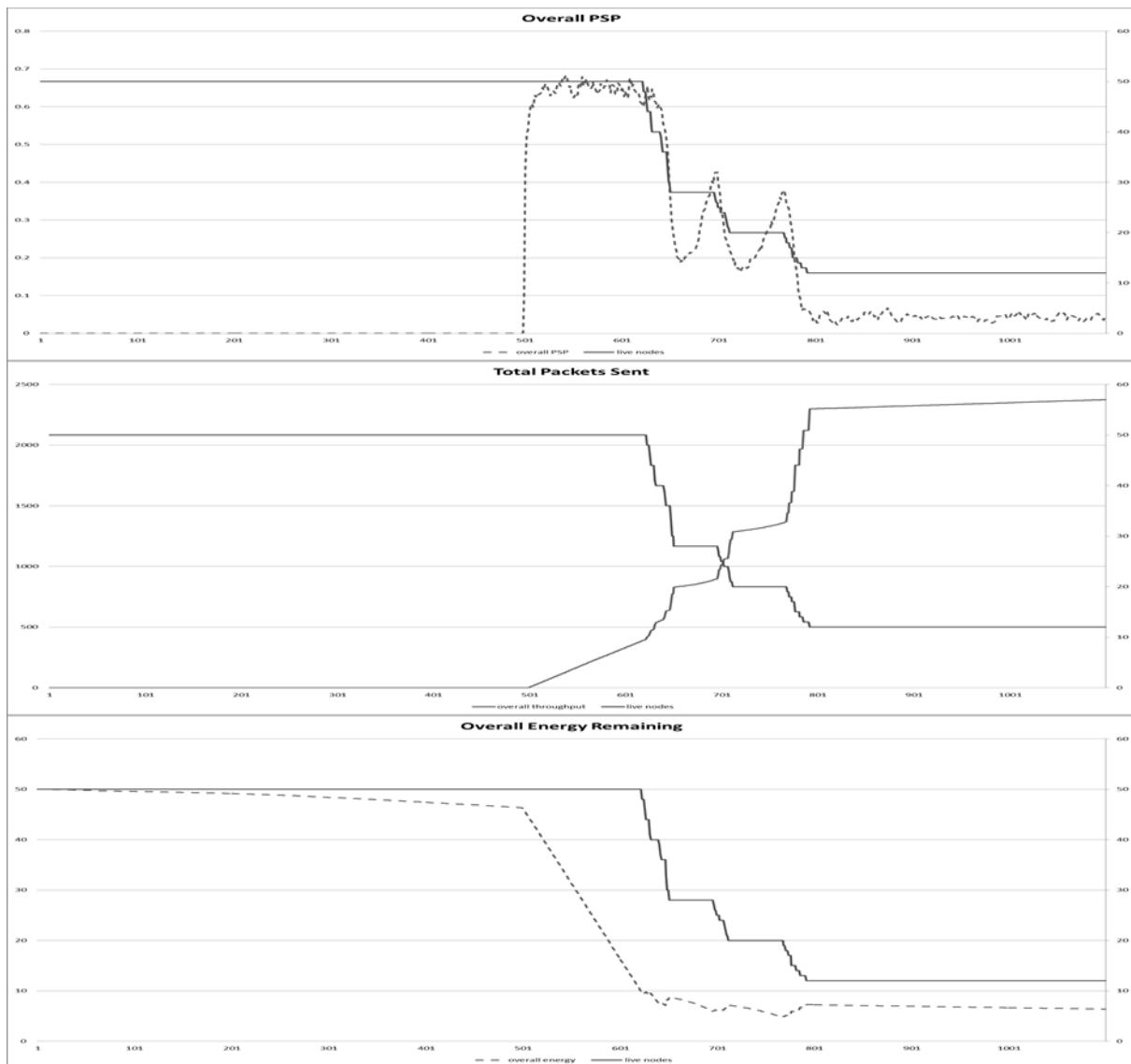


Figure 29 PSP, throughput (packets) and energy (Joules) of OP-cong: medium packet arrival rates

OP-cong also has an improved overall PSP when compared with *FFA* at the beginning of the simulation when all the nodes are alive. The overall PSP is around 0.65 as in the case for *OP-const*.

The first cluster of nodes die at around the same time as previous protocols but the network continues to function for longer than in the case of *OP-const*. The overall PSP fluctuates around 0.3 until the last cluster of nodes run out of energy at 790s and the network becomes non-functional.

At medium packet arrival rates the overall throughput for the *FFA* and *TFT* protocols is comparable at around 25000 packets. The *OP-const* and *OP-cong* protocols improve the overall network throughput up to 34000 packets showing that as the network becomes more strained by the increasing amount of traffic, the credit-based mechanisms add value by conserving the resources of the network in an intelligent manner.

4.5.4. High packet arrival rates

At high packet arrival rates of 12pps the network is under considerable strain. There is more load on the network and as a result the nodes use up their energy more quickly.

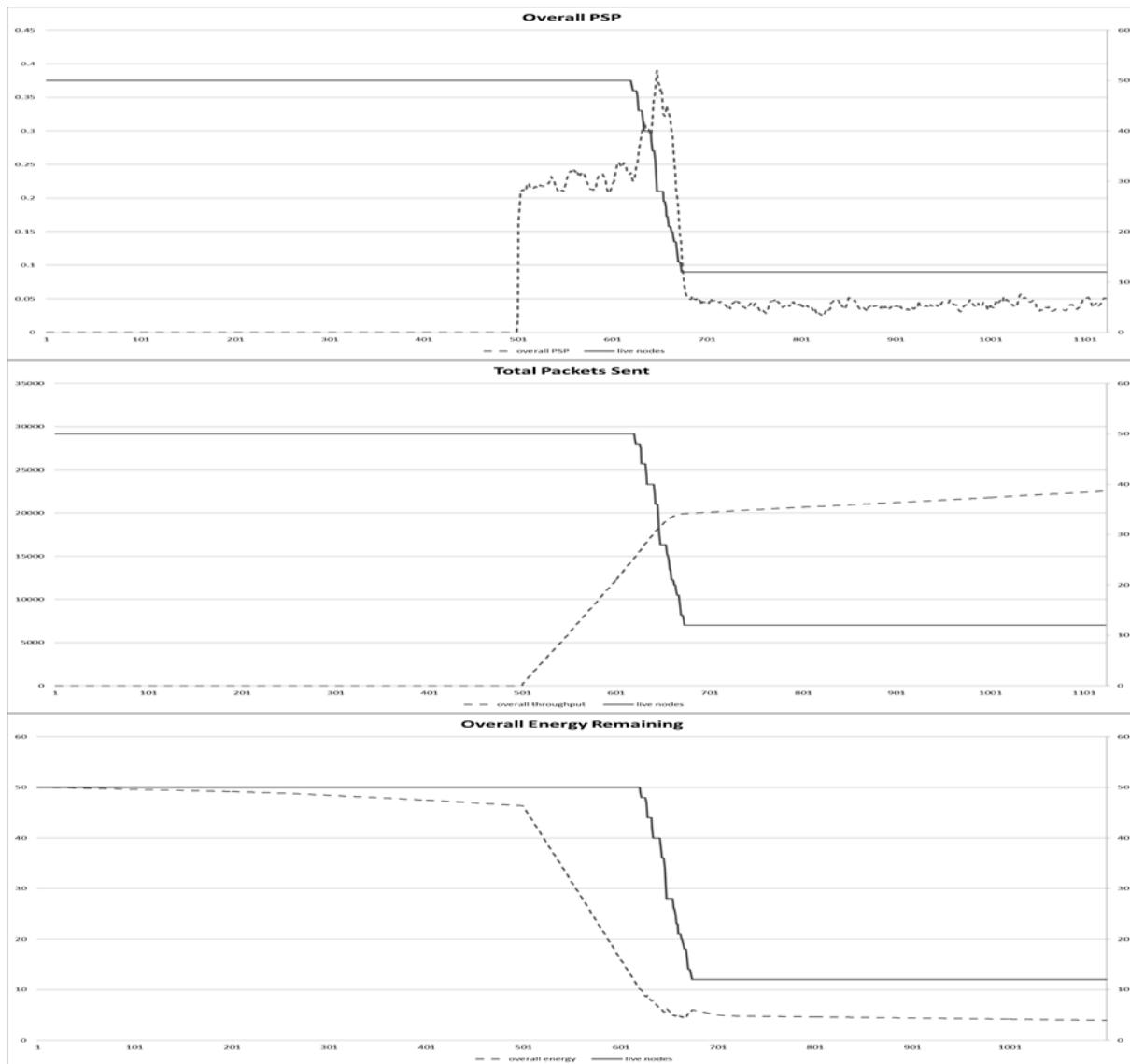


Figure 30 PSP, throughput (packets) and energy (Joules) of *FFA*: high packet arrival rates

Figures 30 to 33 show the PSP, throughput and average energy vs time for a packet arrival rate of 12pps for protocols *FFA*, *TFT*, *OP-const* and *OP-cong* respectively.

We observe that *FFA* has an overall PSP of 0.25 while all nodes are still alive. The first cluster of nodes dies at around 625s. The network continues to function after this with the overall PSP maintaining a similar value before the first cluster of nodes died, but this does not last long. The remaining clusters of nodes die very rapidly after one another resulting in the network reaching its final state at 673s. The overall throughput in *FFA* is 23000 packets.

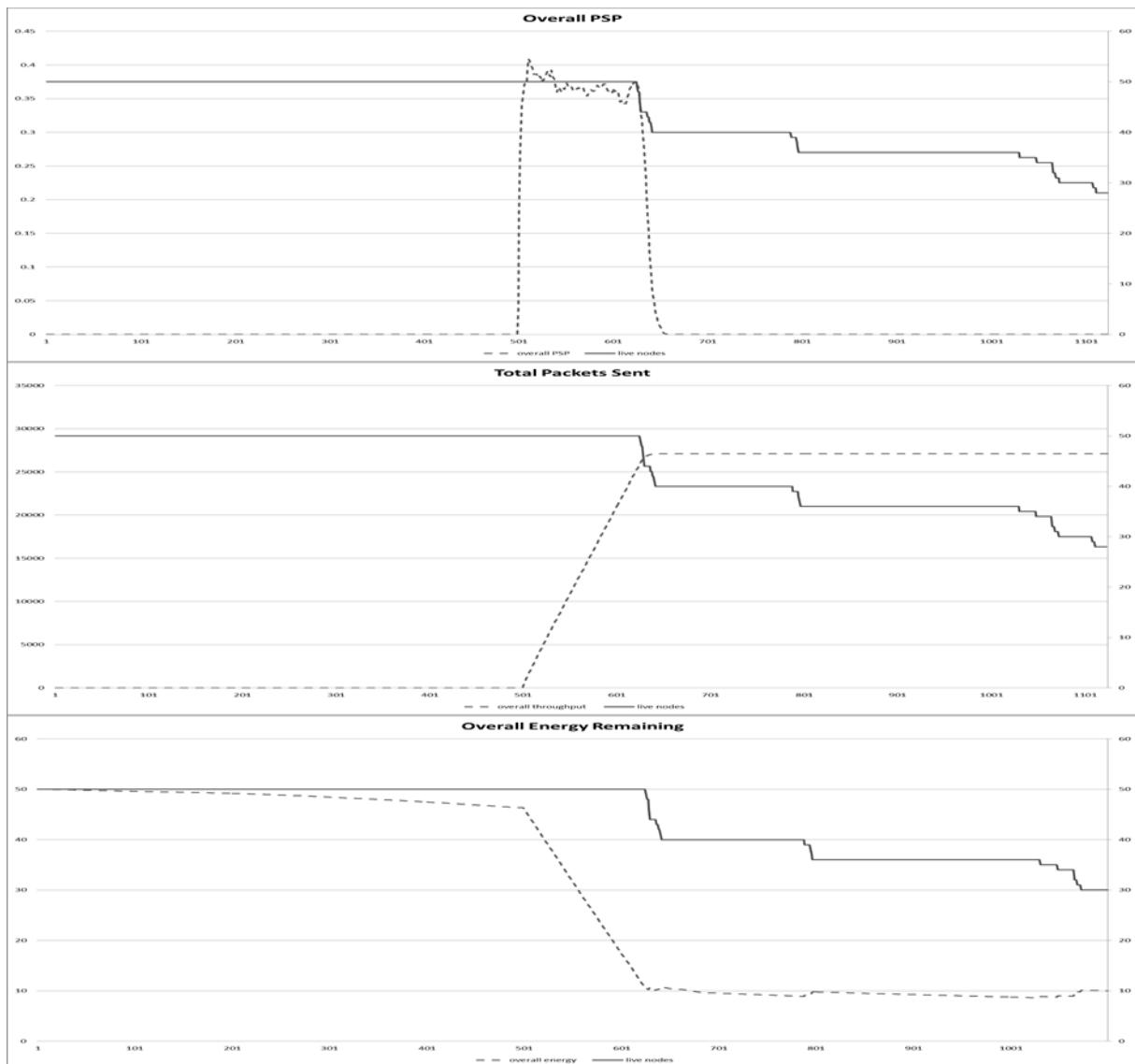


Figure 31 PSP, throughput (packets) and energy (Joules) of *TFT*: high packet arrival rates

In the *TFT* protocol the overall network PSP is better than that of *FFA*. While all nodes are still alive it is around 0.35. Once the first cluster of nodes die in *TFT* the network becomes non-functional immediately. The overall PSP drops to zero even though there are still 40 nodes left with energy. The first cluster of nodes takes a little longer to run out energy than in the *FFA* case. In *TFT* it happens at around 650s. The overall throughput in *TFT* is 28000 packets. This shows a significant improvement over *FFA*.

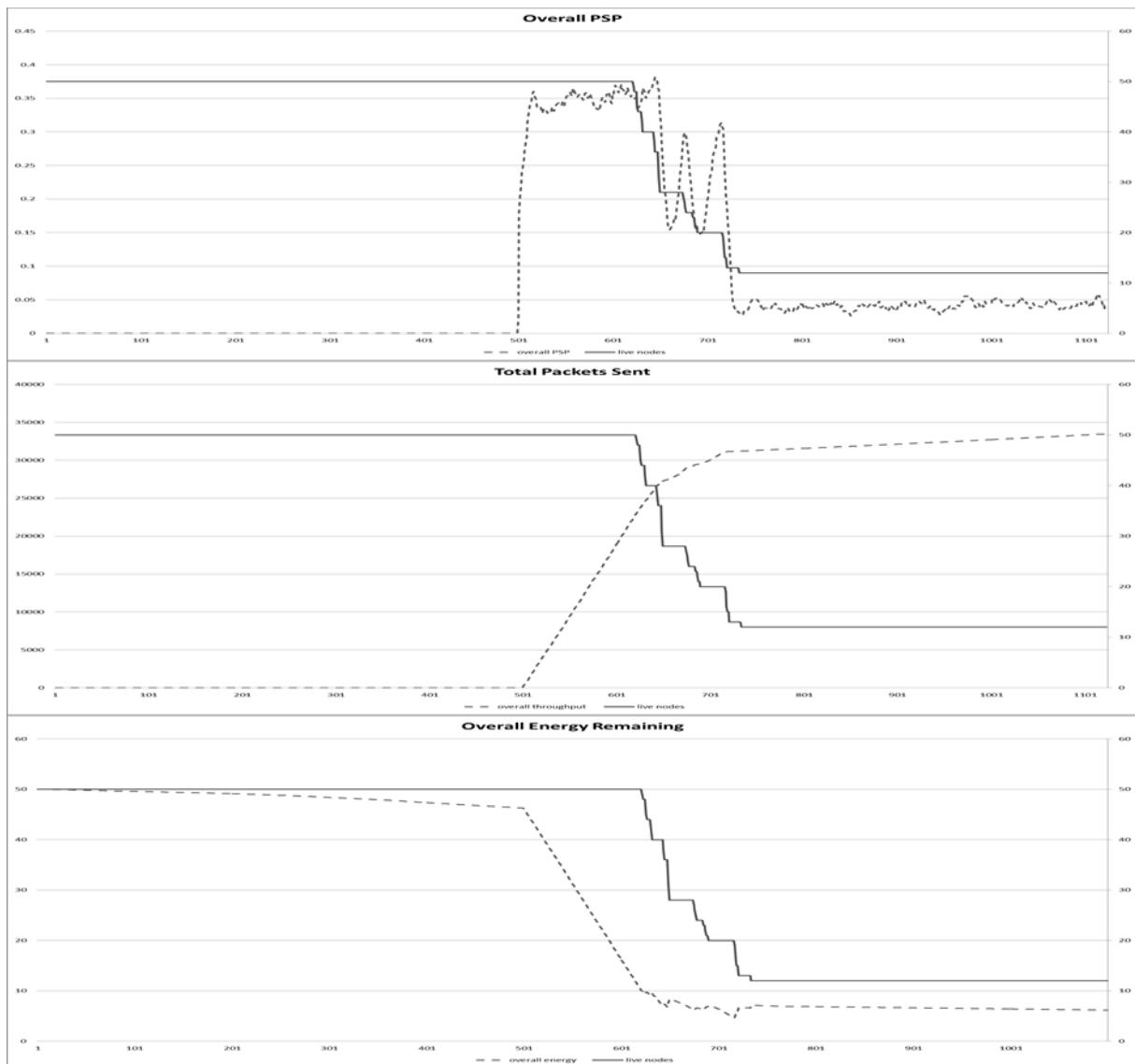


Figure 32 PSP, throughput (packets) and energy (Joules) of *OP-const*: high packet arrival rates

While all nodes are still alive in the *OP-const* simulation, the overall PSP is comparable to that of *TFT*. This shows an enhancement over *FFA*. The overall network PSP is around 0.3.

The first cluster of nodes run out of energy at the same time as *FFA*. This happens at 625s. The remaining nodes continue to operate (albeit at a significantly reduced overall PSP of 0.2) until 720s. This represents an improvement in the network lifetime over *FFA* and *TFT* and a significant increase in the overall throughput. The overall throughput for *OP-const* is 34000 packets.

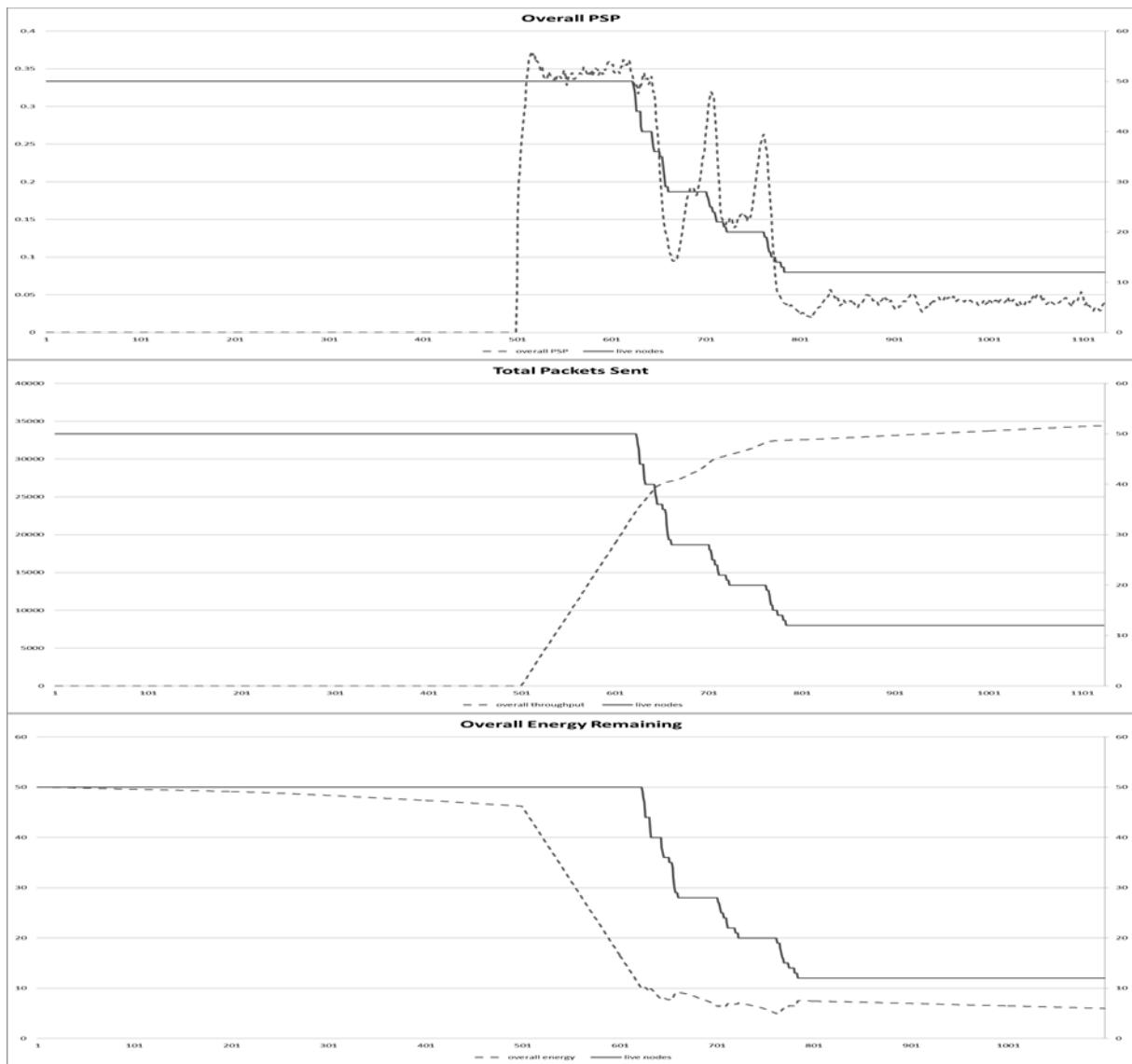


Figure 33 PSP, throughput (packets) and energy (Joules) of *OP-cong*: high packet arrival rates

At a packet arrival rate of 12pps, there is significant congestion at all nodes and the price per service in the *OP-cong* protocol will tend to be close to 1 most of the time. This means that the behaviour of the *OP-cong* protocol approaches that of the *OP-const* protocol. This is clear from comparing figures 31 and 32.

The overall PSP is around 0.35. The first cluster of nodes die at around 625s and the network continues to perform poorly with a PSP of 0.2 until 770s when the last cluster of nodes die and the network enters its final state. This is a slight extension of time over the *OP-const* protocol. This happens because the congestion decreases once the first cluster of nodes die and the *OP-cong* protocol behaviour becomes less restrictive than that of *OP-const*. The overall throughput of *OP-cong* is 34000 packets and also shows a significant improvement over that of *FFA* and *TFT*.

5. Conclusions

This thesis investigates two important questions regarding wireless ad hoc networks. The first aspect that we investigated addressed the topic of stimulating cooperation between nodes. In order for a wireless ad hoc network to function, nodes are required to cooperate and relay packets on behalf of one another. *How can cooperation among individual nodes be managed to improve overall wireless ad hoc network performance?*

The concept of a credit-based scheme was proposed by Buttyan et al [2] and explored further by Crowcroft et al [3] and Gobel et al [4]. This thesis demonstrates that a credit-based scheme can be employed in a simple decentralised manner to stimulate and regulate cooperation between nodes in a wireless ad hoc network. Such a scheme has many variants depending on which node is considered to be paying for the service, what the price of each service should be and how we route packets through the network using information derived from the credit-based economy. This thesis demonstrates that several variants of a credit-based scheme can be implemented in a packet based simulator and that these variants result in the stable operation of the network and improve the overall performance.

The first protocol we considered was *FFA*. *FFA* has no restriction and allows nodes to send packets regardless of the work they have or have not done for the community. In this case the nodes cooperate freely and the network performs well at low packet arrival rates. As the traffic load on the network increases, the performance of *FFA* declines rapidly as resources are not effectively managed to deal with the increased demand.

The second protocol we considered was *TFT*. *TFT* is the most restrictive protocol and demands that a node only be allowed to receive service from the network if it has performed an equivalent amount of service on behalf of the network. At low arrival rates, *TFT* performs poorly compared to *FFA*. This is because the added restrictions due to the *TFT* protocol mean that many packets that could be delivered by the network are dropped. As the traffic load increases on the network the restrictions of the *TFT* protocol begin to have a positive effect on network performance. At packet arrival rates greater than 6pps in our 50 node network, the PSP of *TFT* exceeds that of *FFA*. *TFT* also improves the TAL of the network at higher packet arrival rates by limiting the number of packets that are sent into the network.

FFT and *TFT* represent the two extremes with regards to the cooperation of nodes in wireless ad hoc networks. The credit-based schemes proposed in this thesis are between the unrestrictive *FFA* and the overly restrictive *TFT*. The aim of our experiments was to investigate which credit based protocol showed the best performance over varying packet arrival rates.

OP-const was the first credit-based protocol proposed. In this scheme the originating node is responsible for paying for the delivery of packets and the price per service is constant at one unit per service. We saw that the *OP-const* scheme performed well at low packet arrival rates with a PSP of 0.9 for a packet arrival rate of 2pps. This is not as good as that of *FFA* but shows a marked

improvement over that of *TFT*. As the traffic load on the network increased, so the performance of *OP-const* improved over that of *FFA*. At 8pps and 16pps the PSP of *OP-const* is comparable to that of *TFT*. The *OP-const* protocol also substantially reduced the TAL of the network at higher packet arrival rates when compared to *FFA*. The *OP-const* protocol yielded good performance of *FFA* at low packet arrival rates and achieved the benefits of the *TFT* protocol at high packet arrival rates.

The rest of the credit based protocols were variants of the basic credit based scheme proposed for *OP-const*.

In *OP-cong* the originating node pays for the delivery of packets but the price of a service is dependent on the congestion at a node. *OP-cong* resulted in similar performance to *OP-const* at higher packet arrival rates and it performed better than *OP-const* at low packet arrival rates. At low packet arrival rates the price per service is close to zero because the packet queues at nodes are empty and so the *OP-cong* protocol tends to mimic the behaviour of *FFA*. The slight degradation in network performance that resulted from the *OP-const* protocol is thus eliminated by pricing the service intelligently at each node.

In *OP-cred* the originating node is responsible for paying for the delivery of packets and the pricing scheme is based on the credit available at each node. *OP-cred* showed no improvement over *OPcong* and in fact had a higher latency. The PSP, TRPN and ERPN were all similar to *OP-cong*.

In *OP-energy* the originating node is responsible for paying for the delivery of packets and the pricing scheme is based on the remaining energy at each node. *OP-energy* showed no improvement over *OP-cong*. The PSP, TRPN, TAL and ERPN were all similar to *OP-cong*.

In *OP-const-cred-routing* the originating node is responsible for paying for the delivery of packets and the pricing scheme is constant at one unit per service. The routing of packets is based on the credit available at each node. *OP-const-cred-routing* showed no improvement over that of *OP-cong* and showed a degradation in the TAL. PSP, TRPN and ERPN were all comparable to that of *OP-cong*.

In *OP-const-cong-routing* the originating node is responsible for paying for the delivery of packets and the pricing scheme is constant at one unit per service. The routing of packets is based on the congestion at each node. *OP-const-cong-routing* showed no improvement over that of *OP-cong* and showed a degradation in the TAL. PSP, TRPN and ERPN were all comparable to that of *OP-cong*.

In *FFA-battery-routing* there are no conditions on sending packets (as in *FFA*) but the routing of packets is designed to avoid nodes with low battery energy. *FFA-battery-routing* had very similar results to that of *FFA* except that it caused the overall ERPN of the network to increase as packets took longer routes in order to avoid nodes with low batteries. *FFA-battery-routing* did not improve the performance of the network.

In *DP-const* the destination node is responsible for paying for the delivery of packets. The pricing scheme is constant at one unit per service. We found that the performance of *DP-const* degraded the performance of the network slightly at low packet arrival rates and had no benefit at higher packet arrival rates. The benefits of the origin pays schemes in which packets are stopped from being sent

into the network at the originating node are not apparent in the destination pays schemes. If a packet traverses the network and is dropped at the destination node, there is no value added by the credit-based scheme and as a result no improvement in network performance. This was also found to be the case for *DP-cong* in which the destination node pays and the price of a service is based on the congestion at the servicing node.

Experiment 1 showed that *OP-const* shows improved performance over *TFT* at higher packet arrival rates. It has a PSP in the region of 0.9 at low packet arrival rates, which is not as good as *FFA* but is better than *TFT*. *OP-const* also improves the TAL of the network at higher packet arrival rates.

OP-cong shows an added improvement over *OP-const* in that it has comparable performance for higher packet arrival rates and performance comparable to that of *FFA* at low packet arrival rates. At low packet arrival rates of 2pps, the slight degradation experienced by *OP-const* is no longer apparent in *OP-cong*. This is because *OP-cong* mimics *FFA* for low packet arrival rates because the price of a service is virtually zero due to the lack of congestion.

All the other elaborations of the credit based schemes (*OP-energy*, *OP-cred*, *OP-cong-routing*, *OP-const-cred-routing*, *FFA-battery-routing*, *DP-const* and *DP-cong*) showed no improvement over the performance of *OP-cong*.

Experiment 2 investigated the effect of the redistribution mechanism on the credit based protocols. Experiment 2 showed that for all the credit based protocols, the redistribution mechanism is a critical component of the protocol. Without the redistribution mechanism, many of the credit based protocols (particularly *OP-const* and *OP-cong*) showed degraded performance in terms of PSP, TRPN and TAL.

Experiment 3 investigated the effect of varying the redistribution rate. For *OP-const* and *OP-cong* the network performance is good for a range of values. This indicates that the δ value does not need to specifically tuned for each particular network, but rather that a reasonable δ value can be relied upon to provide good performance for all networks. We selected a value of 0.05 for the remainder of the experiments in this thesis.

Experiment 4 investigated the effect of varying the target credit balance. As the target credit balance increases, so the network tends to behave more like *FFA*. This is because all the nodes have enough credit to send their packets. At low packet arrival rates the network performance improved with increasing target credit balance. There was no point at which the performance began to decline with increasing target credit balance. This is because *FFA* offers the best network performance at low packet arrival rates.

For higher packet arrival rates however, there is a peak in network performance at a target credit balance around 100 credits. Once the target credit balance increases above this value, the performance of the network declines. For *OP-const* and *OP-cong* the optimum target credit balance is 100 credits.

The second aspect explored in this thesis concerns the concept of resource utilisation. Resources are generally limited in wireless ad hoc networks and effectively managing these resources can lead

to improvements in network performance. How can the use of the limited energy and bandwidth resources of wireless ad hoc networks be maximised? Experiment 5 investigated the effect of the credit protocols on the use of resources in the network.

For low packet arrival rates we found that the performance of *FFA*, *OP-const* and *OP-cong* were similar. The credit based schemes did not show an enhancement of the use of resources over *FFA* at low packet arrival rates. This is due to the fact that at low packet arrival rates the credit-based schemes approach the behaviour of *FFA*. *TFT* did extend the time before the first cluster of nodes failed, but this did not add value to the network because the PSP is degraded by the *TFT* restrictions and while all nodes are alive the overall network throughput is limited. Once the first cluster of nodes dies in *TFT*, the network becomes non-functional because nodes are unable to earn the right to send packets if the nodes in the centre of the network are dead. In the credit-based schemes the network remains functional after the first cluster of nodes die because credit is redistributed by means of the redistribution mechanism. The overall network throughput at the end of the simulation was approximately 25000 packets for all schemes.

At medium packet arrival rates the credit-based schemes have a beneficial impact on the network performance. In *OP-const* and *OP-cong* the first cluster of nodes dies at a similar time to that of *FFA*, but the decline in network performance from that point onwards is less severe than in *FFA*. It is also clear that before nodes begin running out of energy in the *OP-const* and *OP-cong* protocols, the performance of the network is better than that of *FFA*. The overall throughput at the end of the simulation is approximately 25000 packets for *FFA* and *TFT* and 29000 for *OP-const* and *OP-cong*.

At high packet arrival rates *TFT* shows improved performance over that of *FFT*. The network remains functional for slightly longer than in the *FFA* case and the PSP of the network is higher while all nodes are alive. At high packet arrival rates the overall throughput of the simulation for *TFT* is 27000 packets and 22000 packets for *FFA*. This demonstrates that as the packet arrival rates increase and the network is placed under more strain, the restrictions of *TFT* begin to show more benefit. *OP-const* and *OP-cong* however, both further improve the network performance at high packet arrival rates.

The beneficial effects of the *TFT* protocol while all nodes are still alive are apparent in that the time before the first cluster of nodes dies is extended and the PSP of the network is higher than in the *FFA* case. The improvement in performance of *OP-const* and *OP-cong* occurs once the first cluster of nodes dies. In the *TFT* protocol, the network becomes non-functional immediately after the first cluster of nodes dies and the remaining energy resources in the peripheral nodes are lost. The edge nodes are unable to send due to the *TFT* restrictions despite having the capability to do so. In *OPconst* and *OP-cong* the network continues to function once the first cluster of nodes dies due to the redistribution mechanism which redistributes credit to the edge nodes and enables them to send packets until they too run out of energy. The overall throughput of *OP-const* and *OP-cong* is approximately 34000 packets representing a significant improvement over that of *TFT*.

We see that at low packet arrival rates the credit-based mechanisms have a negligible positive effect on the network performance by means of maximising the limited battery and bandwidth resources. As

the packet arrival rate increases and the resources of the network become more strained, the positive effect of the credit-based mechanisms increases. At 12 pps the overall throughput of the *FFA* protocol is 22000 packets and the overall throughput of the *OP-const* and *OP-cong* protocols is approximately 34000 packets. This represents an increase of 50% more packets traversing the network when the credit based constraints are used.

The credit-based mechanisms represent an effective means of improving network performance. The credit-based mechanisms improve network PSP, TRPN and TAL mimicking the unrestrained properties of *FFA* at low packet arrival rates and the restrictive properties of *TFT* at higher packet arrival rates. The redistribution mechanism is an essential component of the credit-based mechanism and enables networks to reach a stable state of operation. For the 50 node network we investigated the optimal value for the redistribution rate is 0.05 and the optimal value for the target credit balance is 100. The credit-based mechanisms also show significant improvement to network performance in resource constrained conditions and represent an effective means for maximising limited energy and bandwidth. The effectiveness of the credit-based mechanisms increases as the load on the networks increases.

6. Recommendations for Further research

Wireless ad hoc networks have been an active area of research since the 1990's. A number of important fields have evolved from the field of wireless ad hoc networks including disaster relief networks, sensor networks and vehicular ad hoc networks. The wide variety of wireless ad hoc networks leads to a wide variety of problems and challenges peculiar to each application of wireless ad hoc networks. A number of niche routing protocols have been developed to improve network performance under specific conditions. Examples of these are geographical routing protocols or SrcRR at the MIT Roofnet project [56]. These protocols are designed for specific networks and are tailored to meet specific challenges.

This thesis focuses on the idea of setting up a market based economy using node credits to regulate the packets sent into the network. By using resources fairly and stimulating cooperation amongst nodes we have shown that the performance of the network can be improved. An interesting extension to the current research would be to look at the specific applications of wireless ad hoc networks and investigate the notion of a credit-based mechanism to improve these specific networks performance. The ability of the credit-based mechanism to improve the PSP, TRPN and TAL of real world networks would be an interesting next step.

There are possibilities for improving the performance of the credit based schemes proposed in this thesis by modifying the linear pricing model used to determine service prices at the servicing node. A pricing scheme that causes the price of a service to increase more rapidly with decreasing resources than the linear model employed would intuitively result in a stronger influence from the credit based protocols and thus improve their beneficial effects on network performance.

The notion of change is also an interesting addition to the protocols described in the current research. Each node can keep track of the extra credit received in the packet purse and can send this credit back as change. This enhancement would add greater accuracy to the credit-based mechanisms and would more accurately reflect the use of resources across the network.

7. References

- [1] D. A. Maltz, "Resource Management in Multi-Hop Ad Hoc Networks," 1999.
- [2] Levente Buttyan and Jean-Pierre Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *MONET Journal of Mobile Network*, vol. 8, no. 5, pp. 579 - 592, 2002.
- [3] J Crowcroft, R Gibbens, F Kelly, and S Ostring, "Modelling incentives for collaboration in mobile ad hoc networks," in *Proceedings of the WiOpt*, 2003.
- [4] Johannes Gobel and Anthony E. Krzesinski, "Stimulating Cooperation in Mobile Ad Hoc Networks," in *Proceedings of SATNAC 2009*, 2009.
- [5] Hossain, Teerawat Issariyakul, and Ekram, *Introduction to Network Simulator NS2.*: Springer, 2009.
- [6] R.E Kahn, "The organization of computer resources into a packet radio network," *IEEE Transactions on Communications*, vol. 25, no. 1, pp. 169-178, 1997.
- [7] J. Jubin and J.D.Turnow, "The DARPA Packet Radio Network Protocols," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 21-32, 1997.
- [8] Wolfgang Kiess and Martin Mauve, "A survey on real-world implementations of mobile ad-hoc networks," *Ad Hoc Networks*, vol. 5, no. 3, pp. 324-339 , 2007.
- [9] Humayun Bakht. The history of mobile ad-hoc networks. [Online]. <http://www.computingunplugged.com/issues/issue200508/00001598001.html>
- [10] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [11] Mingyan Liu, Neal Patwari, and Andreas Terzis, "Special Issue on Sensor Network Applications," *PROCEEDINGS OF THE IEEE*, vol. 98, no. 11, pp. 1804 - 1807 , 2010.
- [12] Peter Corke et al., "Environmental Wireless Sensor Networks," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1903 - 1917 , 2010.
- [13] JeongGil Ko et al., "Wireless Sensor Networks for Healthcare," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1947 - 1960 , 2010.

- [14] Yasser Toor, Paul Muhlethaler, Anis Laouiti, and Arnaud de la Fortelle, "Vehicle Ad Hoc Networks: Applications and Related Technical Issues," *IEEE Communications Surveys*, vol. 10, no. 3, pp. 74 - 88 , 2008.
- [15] C. David Wang and James P. Thomson, "Apparatus and method for motion detection and tracking of objects in a region for collision avoidance utilizing a real-time adaptive probabilistic neural network," 5,613,039.
- [16] S. Dornbush and A. Joshi, "StreetSmart traffic: Discovering and disseminating automobile congestion using VANET's," in *IEEE Vehicular Technology Conference*, Dublin, 2007.
- [17] Kevin Collins and Gabriel-Miro Muntean, "A Vehicle Route Management Solution Enabled by Wireless Vehicular Networks," in *IEEE INFORCOM Proceedings*, 2008.
- [18] Marc Bechler, Walter J. Franz, and Lars Wolf, "Mobile Internet Access in FleetNet," in *8th World Congress on Intelligent Transport Systems*, Sydney, 2001.
- [19] <http://www.memsic.com>. [Online]. <http://www.memsic.com/products/wireless-sensor-networks.html>.
- [20] www.microstrain.com. [Online]. <http://www.microstrain.com/wireless-sensors.aspx?gclid=CL3Rpf6XvacCFUSGpAodux95AA>
- [21] www.csiro.au. [Online]. <http://www.csiro.au/news/Super-Sensors-Available-Soon.html>
- [22] A. Arora, "ExScal: Elements of an extreme scale wireless sensor network," in *Int Conf on Embedded and Real-Time Computing Systems and Applications*, 2005.
- [23] T. He et al., "Energy efficient surveillance system using wireless sensor networks," in *Proc. of the Second Int. Conf. on Mobile systems, applications and services*, 2004.
- [24] M. Yarvis et al., "Real-world experiences with an interactive ad hoc sensor network," in *Proc. of the Int. Workshop on Ad Hoc Networking (IWAHN)*, 2002.
- [25] www.melbournewireless.org.au. [Online]. <http://www.melbournewireless.org.au/wiki/?How2GetStarted#hardware>
- [26] <http://wirelessleiden.nl/>. [Online]. <http://wirelessleiden.nl/>
- [27] <http://www.wug.za.net/>. [Online]. <http://www.wug.za.net/>

- [28] <http://pdos.csail.mit.edu/roofnet/doku.php>. [Online]. <http://pdos.csail.mit.edu/roofnet/doku.php>
- [29] http://en.wikipedia.org/wiki/Champaign-Urbana_Community_Wireless_Network. [Online]. http://en.wikipedia.org/wiki/Champaign-Urbana_Community_Wireless_Network
- [30] Yair Amir, Claudiu Danilov, Raluca Musăloiu-Elefteri, and Nilo Rivera, "The SMesh Wireless Mesh Network," *ACM Transactions on Computer Systems (TOCS)*, vol. 28, no. 3, 2010.
- [31] <http://www.computerworld.com>. [Online]. http://www.computerworld.com/s/article/109662/New_Orleans_Wi_Fi_network_now_a_lifeline
- [32] <http://www.businesswire.com>. [Online]. <http://www.businesswire.com/news/home/20040727005484/en/MeshNetworks-Wireless-Provide-Mobile-Broadband-Solution-Innovative>
- [33] <http://web.archive.org>. [Online]. <http://web.archive.org/web/20080214232340/meraki.com/oursolution/hardware/>
- [34] <http://www.belairnetworks.com>. [Online]. http://www.belairnetworks.com/news/press_releases/view.cfm/p_id/99
- [35] Guido R. Hertz et al., "IEEE 802.11S: THE WLAN MESH STANDARD," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 104 - 111 , 2010.
- [36] Sushant Jadhav, Timothy X Brown, Sheetalkumar Doshi, Daniel Henkel, and Roshan George Thekkekunnel, "Lessons Learned Constructing a Wireless Ad Hoc Testbed," in *Proc. of the 1st Workshop on Wireless Network Measurements*, 2005.
- [37] R.S. Gray et al., "Outdoor experimental comparison of four ad hoc routing algorithms," in *Proc. of the 7th ACM Int. Symp. on Modeling, analysis and simulation of wireless and mobile systems (MSWiM)*, 2004.
- [38] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordström, and C. Tschudin, "A large-scale testbed for reproducible Ad Hoc protocol evaluations," in *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2002.
- [39] Regis Vincent, Benoit Morisset, Andrew Agno, Michael Eriksen, and Charlie Ortiz., "CENTIBOTS: LARGE-SCALE AUTONOMOUS ROBOTIC SEARCH AND RESCUE EXPERIMENT," in *2nd International Joint Topical Meeting on Emergency Preparedness & Response and Robotics & Remote Systems*, 2008.

- [40] CK Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 138 - 147 , 2001.
- [41] Charles E Perkins and Pravin. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *SIGCOMM '94 Proceedings of the conference on Communications architectures, protocols and applications*, 1994.
- [42] (2012, November) Wikipedia. [Online]. http://en.wikipedia.org/wiki/Bellman%20%93Ford_algorithm
- [43] David B. Johnson Maltz and David A., "Dynamic Source Routing in Ad HocWireless Networks," *Mobile Computing*, vol. 353, 1996.
- [44] David B Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," in *First Workshop on Mobile Computing Systems and Applications*, 1994.
- [45] David B. Johnson, David A. Maltz, and Josh Broch, "DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," in *Ad Hoc Networking.*, 2001, pp. 139 - 172.
- [46] M. Scott Corson and Vincent D. Park, "A Highly Adaptive Distributed Routing Algorithm for MobileWireless Networks," in *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 1997.
- [47] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva, "A Performance Comparison of Multi-HopWireless Ad Hoc Network Routing Protocols," in *MobiCom '98 Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, 1998.
- [48] C. E. Perkins, "Ad Hoc On Demand Distance Vector Routing," in *Mobile Computing Systems and Applications, Proceedings. WMCSA '99*, 1999.
- [49] S. Singh, M. Woo, and C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," in *Proc. Mobicom 98.*, Dallas, 1998.
- [50] Rahul C. Shah and Jan M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Wireless Communications and Networking Conference, 2002. WCNC2002*, 2002.
- [51] J. Hu and M. Burmester, "Cooperation in Mobile Ad Hoc Networks," in *Guide to Wireless Ad Hoc Networks.*: Springer, 2009, pp. 43 - 57.

- [52] Juan José Jaramillo and R. Srikant, "A game theory based reputation mechanism to incentivize cooperation in wireless ad hoc networks," *Ad Hoc Networks*, vol. 8, no. 4, pp. 416–429, 2010.
- [53] Boudec., S. Buchegger, and J.-Y. Le, "Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes – Fairness in Distributed Ad-hoc NeTworks)," in *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, 2002.
- [54] Johannes Göbel, Anthony Krzesinski, and Michel Mandjes, "Incentive-based control of ad hoc networks: A performance study," *Computer Networks*, vol. 53, no. 14, pp. 2427 - 2443, 2009.
- [55] Kevin Fall, The ns2 Manual (formerlyns Notes and Documentation), 2010.
- [56] P. De, A. Raniwala, S. Sharma, and T.C. Chiueh, "Design Considerations for a Multihop Wireless Network Testbed," *IEEE Communications Magazine*, vol. 43, no. 10, pp. 102-109, 2005.
- [57] <https://support.skype.com/>. [Online]. <https://support.skype.com/en/faq/FA1417/How-much-bandwidth-does-Skype-need>
- [58] M. Takai, J. Martin, and R. Bagrodia, "Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks," in *2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2001.