Power Efficient Communications in Low Power Ad Hoc Radio Networks

Adam Peter Greenhalgh

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of the

University College London

Department of Computer Science

University College London

2011

I, Adam Greenhalgh confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

In this thesis we investigate the feasibility of using information overheard by wireless devices to reduce their overall energy consumption for communications. Specifically we investigate the hypothesis “It is more efficient in terms of energy consumption to constrain the transmission power based upon a combination of received signal strength with a minimally extended MAC, than to utilise an unchanged MAC and full power.”.

We investigate the hypothesis in the context of an ad hoc wireless network comprising of devices that use low power radio systems. We investigate two different low power radio systems, a standard 802.11 system and a custom low power radio device from Philips Research Labs. We examine in detail the energy consumption of the Philips’ low power radio device in its three modes of operation; transmission, reception and idle. From this, we propose a generic framework for power measurement and illustrate the technique with a case study. Specifically, this technique identifies the three modes in a trace of the energy consumption of the low power radio device, and uses this information to accurately extract the consumption figures for the different modes of operation.

Using our measurements and the energy consumption parameters for an 802.11 radio device, we examine in simulation the complex behaviour that emerges from the implementation of an energy-aware system using a simple transmission power control algorithm that exploits overheard MAC-level information to reduce device’s energy consumption. We evaluate this simple algorithm using two radio systems and show that, in spite of the complexity, energy savings can be obtained using a scheme that takes advantage of overheard information.

Acknowledgements

I would like to express my deepest gratitude to Professor Stephen Hailes. This thesis owes much to the constructive supervision and continuous encouragement he has provided. He has been an immense source of inspiration over the years.

Professor Mark Handley has been an inspiration and guidance over the past decade or more, helping me to develope as a researcher through a number of fun projects.

University College London's Department of Computer Science and in particular the Networks Research Group has been home for the past ten years; I would like to thank them collectively for all the help and assistance they have been over that time. Special thanks must go to Manish Lad for reviewing this thesis from start to finish.

This work was supported by the EPSRC through an Industrial Case Award in collaboration with Philips Research Labs of Redhill. I must thank them for their contributions financially and in the form of prototype hardware.

I would like to thank my PhD examiners, Stephen Pink and John Bigham, for their time and effort in reviewing the draft and for making the PhD Viva a pleasent experience.

My parents and sister have given me the chance of an education and have been an invaluable source of support and encouragement. They have probably played a more important role in my life than they think.

I am lucky to share Juliet's life. She has been the greatest source of happiness and has given me our wonderful son Thomas. Her love and support mean the world to me.

And finally, I must thank Professor Laurent Mathy for pressing me to submit this thesis after far too long.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis examines the transmission power optimisation techniques used to minimise the overall energy consumption of an ad hoc network. We determine the energy consumption characteristics of a real low power radio device and then use these results to simulate different transmission power optimisation techniques and to evaluate an ad hoc routing algorithm.

1.1 General setting

Ad hoc networks are networks of autonomous nodes that have wireless connections between each other. Since there is no central point of connectivity such as a base station, such a system can only work if each node is prepared to forward packets destined for other nodes. Regrettably, since nodes move relative to one another, the precise connectivity of any given node is time varying. Put another way, the network topology varies as nodes change location, move out of range of other nodes or fail completely. In addition, either all nodes or a proportion of them are assumed to be self-powered. Consequently, the conservation of energy becomes important not just for the node concerned, but also for the survivability of the system as a whole.

This dynamic behaviour violates a set of assumptions that are made in the fixed network. Thus, for example, in the fixed network, hosts are assumed not to change their point of connection, allowing moderately static routing protocols to be deployed. The issue of node dynamism has to be addressed along with requirements that affect traditional routing protocols such as loop free routing, completeness and stability. There are several proposed solutions to this problem, such as DSR [44], AODV [70] and ZRP [32] to cite but three.

One major advantage of ad hoc networks, particularly from the viewpoint of military and civil defence applications, is the fact that the very dynamism that proves so troublesome in realising such systems, forces them not to be reliant on single points of failure. As a consequence, there is no obvious point of attack or failure that would cause the system to cease operation. We believe that this also makes such systems highly suitable for use in hostile environments.

The original motivation for this work was cast in terms of space exploration by autonomous self-powered robots [10]. An example of a task that might be performed by such robots is the initial surveying of an area intended for colonisation, since the hazards are unknown and it maybe too costly to transport people and their support systems to carry out a surveying task. It was shown by the NASA Pathfinder mission [62] to Mars that remote control of a robot explorer is very limiting, because the round trip time for signals to Mars and back is significant. Using a single large device to carry out a task carries more risk than using a collection of small devices, because failure of a critical component on the large device will cause the mission to fail (e.g. the subsequent failure of the Mars Polar Lander [61]). In contrast, a collection of smaller autonomous devices that are able to cooperate to carry out a task may be more resilient to changes in the environment. Indeed, NASA have suggested precisely this approach for the exploration of asteroids [10].

In tasks such as the exploration of Mars, the devices exploring the surface of the planet have a fixed amount of energy available to them at any one time. However, such devices can also use solar cells to recharge themselves. This recharging is more effective in bright sunlight than in partial shade, so there is a further parameter in our scenario - the position of the node relative to the topography of the surface determines how fast it can regain energy and, consequently, its overall availability.

Work on energy saving has already been performed in the context of terrestrial mobile computing. The work proposed here is orthogonal to this - the techniques we propose to explore are complementary to standard techniques such as disk spin down, processor speed variation, and so forth. Of more direct influence is the work performed in energy efficient radio and protocol design for ad hoc networks. As a consequence, one assumption that we will make is that the system is architected around a low power radio system such as Bluetooth [37] , 802.15.4 (Zigbee) [38] or 802.11 [36]. Given the availability of such systems, this assumption has the advantage that any work we do in simulation will be (partially) capable of validation in the real world.

Finally, it is worth noting that we believe that the lessons we are likely to learn from this work have application outside the rather esoteric application domain on which we based our scenarios above. Hostile environments are not confined to space - they exist in active volcanic environments, in Antarctic conditions, in earthquake zones and, regrettably, in buildings and tunnels in the most tragic of circumstances.

A problem domain that is of particular interest is the use of low power radio devices in disaster and earthquake zones or mine and building collapses where the rescue scene maybe dangerous for the rescuers or where there are an insufficient number of people to search the area quickly and effectively enough to rescue survivors. As has been shown in the World Trade Centre disaster, robotic devices are able to explore areas of a collapsed building that a person is unable to reach. By taking the idea of using robots in collapsed areas to carry out sensing tasks one stage further it would be interesting to make the building itself a sensor by embedding sensors into the fabric of the building, there are already numerous sensors in many building ranging from movement sensors for automatic lighting systems and anti-theft systems to temperature sensors for heating regulation and fire detection, this is not taking into account the numerous other systems that could host low power radio systems such as lights, id badges and even coffee cups. However, imagine if the low power sensors were attached to walls either by embedding them in the wall or by bolting them to it and had the ability to sense their immediate surroundings. The sensors need not have to have the ability to process the data they sense but must be able to store it for a period of time and/or relay it via a radio device. Thus it would be possible to pass by the low power sensors with another device and extract the information that they hold. From this information it could be possible to build a three dimensional map of the building. Now, if the building collapsed, and a robot device were sent to explore the rubble, a three dimensional map of the rubble could be created and the areas big enough to allow people to survive in identified. The problem of exploring a collapsed building cannot be partitioned into grid like zones where one robot has to search a physical area to complete its task because the rubble will create tunnels and dead-ends which obviously do not fit a grid like structure. Instead, a better solution perhaps might be that the robots are distributed across the scene and collectively told to explore the zone. To do this, the robots must communicate with one another to propagate the information that they have discovered from the sensors they have themselves or those that are embedded in the building. Thus, the three dimensional map of the searched area is propagated to those searching.

However, a situation as complex as exploring a collapsed building is too challenging a problem to initially investigate so a similar problem of investigating a planar environment where the energy usage of the system is an important aspect, and, in particular, the energy usage of the communications system as a whole. Where the system is taken to be all the nodes involved in the particular task, which has been assigned to them collectively.

An alternative deployment scenario is the tracking and monitoring of wild animals over a long period of time to help scientists understand the movement and social dynamics of the animals. This deployment scenario has its own particular challenges, it is particularly sensitive to energy usage as there are very limited mechanisms through which the devices can be recharged once they have been deployed, they are deployed for extended periods away from fixed base stations so data must also be collected and stored before being re-transmitted when the opportunity arrises.

1.2 Challenges and motivation

It has been noted previously that nodes have fixed energy resources and that techniques for the maximum effectiveness of this resource are needed. To reduce the overall energy consumption of a device every aspect of the device's operation needs to be optimised; however the focus of this work is the optimisation of a device's transmission power. To carry out the optimisation we need to base it upon some information. We are deliberately not saying what information is required to determine a device's optimal transmission power, since to gather and analyse all the possible information required to generate an optimal transmission power would incurs costs greater than the savings achieved. To carry out a per transmission optimisation would require global information about all the transmissions occurring or to occur in the near future, the propagation ranges of all transmission devices, the location of all devices and the environmental conditions experienced by the system. This list is not exhaustive but gives an indication of the complexity required for a single transmission. If we take this further and consider the scenario in which multiple simultaneous transmissions are occurring at once, it is easy to see that this is a distributed parallel optimisation problem which would be difficult and complex to solve using small low power devices with limited energy and limited computational resources.

Because of the complexity described above our emphasis is upon modifying the transmission power of a device based on information we have observed from our radio environment. This approach is simple, with a minimum of information needing to be passed between devices. Based on the information received, each device makes an independent decision to determine the level at which to transmit. We propose to also use this information to aid the selection of routes in a routing protocol to reduce the energy consumption of the device's communication system. The philosophy we are using here is that if we have learnt information about our environment through having our receiver switched on for another purpose, then we should make full use of that information since we have incurred a cost in receiving and decoding that information. In short *"If you heard it, use it"*.

Firstly, we need to make the case that there are savings to be made by modifying the transmission power of a device. Secondly, we show a multiple device scenario where the path taken affects the energy consumption of the system.

1.3 Overview of the problem

Let us, for the sake of simplicity assume that the relationship between the signal propagation distance and transmission power is given very naively[1] by $P \propto d^n$ (Log-distance Path Loss Model [72] section 3.9.1) where n is dependent upon the environmental conditions. In this power law equation d is the dominant term, small changes in d cause large changes in P, so reducing d leads to a large reduction in P and hence an energy saving which is our ultimate goal. This leads us to the question, "how can we reduce the effective distance between two devices ?". The simple answer is nothing, except move the devices closer together. However if we approach this question from a different angle we can do two different things, firstly we can transmit only as far as necessary to reach the intended device, as described below. This does not reduce the distance between the devices but reduces the distance the signal propagates and hence the required transmission power. The second answer is that we do not have to transmit directly between two devices: if we use relayed communications we can reduce the size of d for each hop of the communication. Whilst $\sum_{i=1}^{n} d_i \geq d$, it is not always true that $\sum_{i=1}^{n} d_i^n \geq d^n$. We are able to make an energy saving if $\sum_{i=1}^{n} d_i^n < d^n$. These two optimisations form the basis of this thesis. In this thesis we examine them in more detail, question their assumptions and elaborate on the complexities involved. Because these two optimisations are the starting point of this thesis, they should not be used without consideration of the side effects of using them, which include the break down in MAC protocol caused by over optimisation leading to the sending of additional packets.

1.3.1 Examples of the optimisations

We begin the exploration of these two optimisations by illustrating them both with examples.

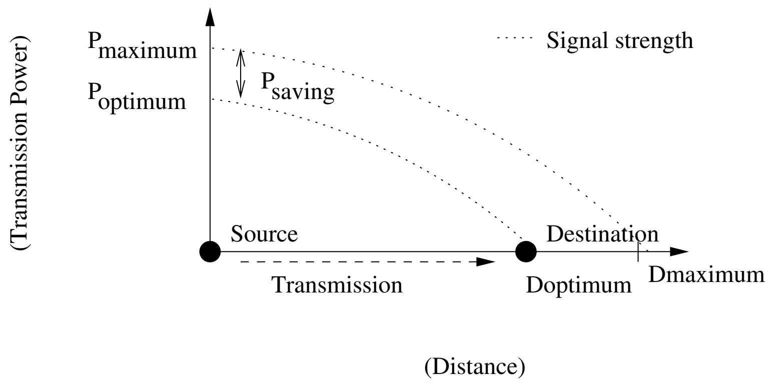


Figure 1.1: *Radio progragation optimisation example*

In figure 1.1 we illustrate the possible savings in terms of transmission power, when trans-

[1] We are only using this naive relationship to illustrate the problem.

mitting only as far as necessary. All radio transmitters have a maximum transmission power, P_{max}. In many devices this is the default transmission power used for communicating with all devices irrespective of their distance from the transmitter. The propagation distance of a signal transmitted at P_{max} is D_{max}. If the distance between two devices is less than D_{max}, then the device has transmitted at a transmission power that is higher than necessary to just reach the target device. So, if the distance between nodes is $D_{optimum}$ then the transmission power required to just reach $D_{optimum}$ is $P_{optimum}$ which results in a saving of $P_{max} - P_{optimum}$ over the default transmission level.

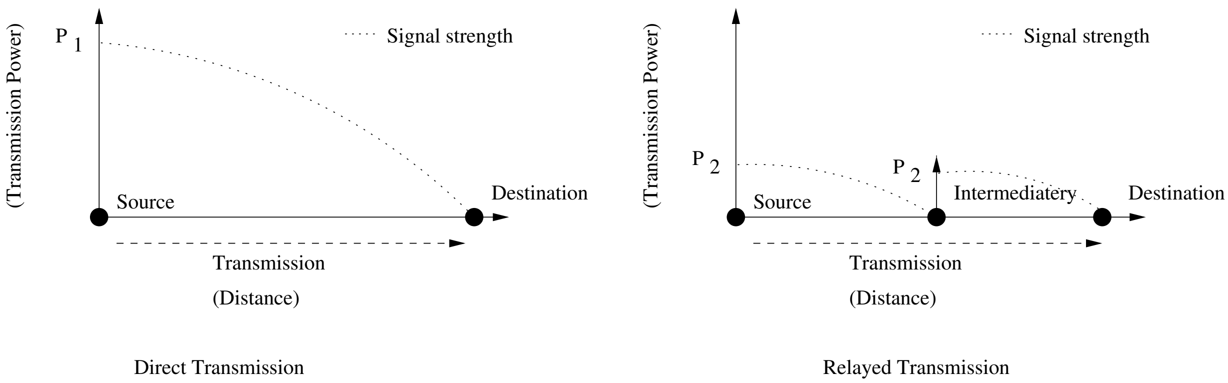


Figure 1.2: *Radio progragation relaying example*

In figure 1.2 we illustrate the possible savings of relaying traffic with the aim of transforming the transmission into smaller lower cost radio transmissions rather than a single long expensive transmission. Figure 1.2, illustrates a transmission between a source and a destination, which are 4 units apart and the environmental conditions cause the path loss exponential to be 4. For a direct (left hand figure) transmission between the source and destination the transmission cost would be 256 units of power, ($P = 4^4$). For an indirect (right hand figure) transmission via an intermediatory device positioned at the midpoint between the source and destination, the transmission cost would be 32 units of power, ($P = 2^4 + 2^4$). The relayed example uses 1/8th of the power of the direct transmission scenario, a significant saving.

The approach taken in figure 1.2, is very simplistic and ignores the cost of reception, the effect of the scheme on the Medium Access Control (MAC) protocol, the Hidden and Exposed terminal problems, the Antenna height/orientation and the issues associated with interference; along with many others not listed here. As well as the fact that the model is a huge simplification of the reality. We address these issues fully in later chapters.

1.3.2 Summary of the challenges and motivation

The two optimisations described above are made more complex by trying to deploy them in an infrastructure-less ad hoc network environment. These optimisations would be simpler if every node had a complete and up-to-date picture of the entire network. However, this is unrealistic. The other extreme is where the nodes know nothing about the network and, to make things even more complex, the network changes with time. The later scenario is the more realistic scenario and it is this that is being addressed. To put it simplistically, the only information a device will know about the network structure is from the information it is sent or overhears. It is obvious that significant savings are possible by using these optimisations, but at what cost do we use them ?

1.4 Hypothesis statement

Taking the challenges and motivations from the previous sections we come to our hypothesis :

"It is more efficient in terms of energy consumption to constrain the transmission power based upon a combination of received signal strength with a minimally extended MAC, than to utilise an unchanged MAC and full power."

The problem domain is a set of wireless nodes with lower power radio systems, that are able to modify their radio transmission power. The environment will be assumed to be planar and uniform. This hypothesis is framed in terms of minimising the energy consumption of a network of mobile ad hoc nodes by taking advantage of the benefits of relaying traffic using short hops rather than a few long hops. Using smaller hops means that the transmission power required to transmit from node to node is reduced.

1.4.1 Objective 1

For simulations to be credible, the simulation parameters should be based upon actual device specifications or observed values. In chapter 3, we aim to satisfy this first objective by measuring the power consumption figures from a real device.

1.4.2 Objective 2

The optimisation of the transmission power has to occur at both a local local and global level for an overall minimum to be achieved. Chapter 4 gives further background to the power control algorithms used. Whilst chapter 5 investigates the local hop by hop optimisation where there is no control coordination.

1.4.3 Objective 3

The third objective is to evaluate the correctness of the basic assumption that using multiple small transmissions is more efficient than a single direct transmission. There will be circumstances where relaying is not the optimum thing to do and this is the third objective. Chapters 5 and 6 aims to address this objective.

1.4.4 Objective 4

The fourth and final objective is to investigate the optimisation of the network's routing and transmission power level based upon information observed from the environment. Chapters 6 and 7 aims to investigate the savings achievable from using observed data and Chapter 8 aims to investigate the routing related optimisation.

1.4.5 Justification of the objectives

To explore the hypothesis we must investigate the area using realistic parameters and models. If the parameters and models are not grounded in reality it would be trivial to prove or disprove the hypothesis according to any agenda we might have. Objective 1, addresses this concern by carrying out actual measurements of real devices. The hypothesis does not state whether it is more efficient to use relayed traffic on an individual flow or on a collection of flows. To remove this ambiguity we first investigate flow by flow optimisations with objective 2, and the group optimisation in objective 3, thus covering the two possible interpretations. Objectives 2 and 3 have to rely upon the devices knowing a-priori, the environment the devices operate in, this is unrealistic and objective 4 aims to address this short coming. Objective 4 aims to have the devices use observed information to optimise the routing, which gives a more global view, and to optimise the hop by hop transmission power.

1.5 My contribution

The main contributions of this work are fourfold. They are :

- The measurement of the energy consumption of a low power radio device, in order to ensure that subsequent simulations are well founded.
- A MAC-based power minimisation technique.
- The interactions of the MAC-based power minimisation technique with TCP.
- An energy-aware routing protocol.

Firstly, a measurement exercise to measure the energy consumption of a real low power device under different conditions was under taken. This work was carried out to provide the simulator with real values for the radio model, to increase the confidence in the simulator's output. The radio device being measured was reported to have a transmission/reception model that is contrary to the conventional model where to transmit a bit is more costly to receive than to receive a bit. This work forms chapter 3.

Secondly, whilst investigating the background to the third component of this work it became obvious that little or no work had been carried out into minimisation of the transmission power between two nodes which dynamically adjusts to the network's state. This scheme takes advantage of the broadcast nature of the radio transmission and uses the data it observes to gain information about the nodes around it. This work forms a component of chapters 4, 5 and 6.

Thirdly, this work investigates the relaying of traffic through short hops rather than large hops or direct transmission. This work is broken down into several sections that progressively work through the problem finally developing and testing a routing protocol. Specific scenarios are examined in chapter 5, where devices are placed and have predefined traffic routed using predefined paths through the network. Later in chapters 6 and 7 we remove the predefined placements and predefined traffic, but maintain the predefined routing, i.e. direct or relayed traffic. Chapter 8 introduces a routing protocol.

1.6 Thesis outline

This thesis is laid out as follows. Chapter 1 introduces the topic and the background before moving on to the challenges and motivation for the hypothesis and objectives. Chapter 1 concludes with a justification of the objectives and outline of the contribution of this thesis. Chapter 2 reviews the current literature in the area. Chapter 3 investigates the transmission power requirements of a real low power radio device through measurements of the device. Chapters 4 through to 7, are a logical analysis of the benefits of using multiple relayed communications in contrast to the direct transmissions, using both an 802.11 and 802.15.4 radio model. These chapters analyse the benefits of using overheard information to optimise transmission power levels with different optimising algorithms. Chapter 8 examines a routing protocol that uses the information and observations of the previous chapter. Finally chapter 9 concludes the thesis summarising the results and detailing the conclusions.

1.7 Note about graphs

Please note throughout this thesis for space reasons the graphs in this thesis are labelled "link max" , "link min" , "Alg A" and "Alg B", these respectively refer to the upper bound algorithm, the approximate lower bound algorithm, experimental algorithm A and experimental algorithm B.

Chapter 2

Literature Review

In this section we aim to give a background to the literature in the field and an understanding of some of the issues.

2.1 Energy Efficient MAC protocols

A number of people have also identified the MAC protocol in a low power wireless system as a potential source of energy saving. In S-MAC[83] they take a wholesale view of changing the entire MAC protocol to eliminate the costly idle listening, collisions, overhearing and control overheads. S-MAC focuses on having three major components: periodic listen and sleep, collision and overhearing avoidance, and message passing. Neighbouring nodes form virtual clusters and use an auto synchronisation mechanism to co-ordinate sleeping and listening phases of the protocol introducing a messaging overhead and latency when passing messages through the network. This leads it to use more energy than an 802.11 MAC under heavy traffic loads. T-MAC [80], is a hybrid algorithm that has synchronous sleep periods with an asynchronous low power listening mechanism, which means that under varying load it outperforms S-MAC whilst maintaining S-MAC's performance under constant load. T-MAC dynamically adapts its listen/sleep duty cycle dynamically through a series of fine-grained timeouts, resulting in the awake time being shortened if the channel is idle. X-MAC[12] in contrast to S-MAC and T-MAC[80] shortens the transmitter preamble and embeds the target MAC address in to a strobed preamble, enabling non target devices to sleep again having only briefly woken to detect the channel. The strobed preamble can easily be interrupted early by the target device reducing the per hop latency. The per hop latency is further reduced by using an adaptive algorithm which changes the device's duty cycle to match the offered load. C-MAC[54] aims to address the shortcomings of the above protocols by supporting low latency and high throughput whilst maintaining a low duty cycle. It achieves this by using an aggressive RTS mechanism rather than a long pre-amble, with nodes waking up periodically to detect the channel for activity. It then uses any-cast rather than unicast transmissions to forward packets towards the first potential forwarder to wake up, converging back to unicast transmissions once it has contacted the final receiver. In a low duty cycle environment CMAC achieves the throughput and latency performance of standard CSMA whilst being more efficient than its competitors.

All the energy efficient MAC protocols presented above aim to completely change the MAC protocol. For the system they are investigating, this is a significant change and would require a long period of standardisation for any such scheme to be adopted. In this thesis we explore whether there is a simpler approach, that would also complement any of the protocols above.

2.2 Energy Saving and Capacity Improvements in Ad Hoc networks

The GPC work by Monks et al. [59] is similar to this work and investigates controlling the transmission power of a device whilst in conjunction using a distributed medium access control mechanism. The work shows that power control improves the network capacity and reduces the energy consumption of the devices. It is acknowledged in the work that the scheme used in GPC requires full knowledge of the channel and network.

In [55], the authors present a practical interference aware metric, Network Allocation Vector Count (NAVC), and show that this can be used as a metric for AODV and transmission power control. In this work they take a different approach to us and analyse the channel's level of interference and use this as a model for the power control and routing metric. It differs from this work in that nodes exchange information about their NAVC values which complicates the system more than ours.

2.3 Low Power Radio

The radio technologies of particular interest to this piece of work are the 802.11 and 802.15 series of IEEE standards. These standards have been focused upon because a number of real world implementations can be used to carry out practical validation of simulations.

2.3.1 IEEE 802

The IEEE 802 [1] set of standards define the Local Area and Metropolitan Area network protocols. The 802 standards are limited to defining layers 1 (Physical) and 2 (Data Link) of the ISO Open Systems Interconnection Model. This framework is set out in [40]. The 802 series of standards also includes a set of standards for wireless communications in particular wireless

[1] The IEEE 802 committee first met in February 1980, hence 802

lans (802.11) and wireless personal area networks (802.15). It is these that are of interest to us.

2.3.2 802.11 Wireless LAN MAC and Physical Layer specification

The two standards 802.11a and 802.11b are both supplements to the 802.11 standard. 802.11 is specified to operate in the 2.4 GHz, unlicensed ISM band and defines both 1Mbps and 2Mbps bandwidths using different modulation technologies. These are:-

- 2Mbps (1 Mbps noisy) Direct Sequence Spread Spectrum (DSSS)
- 1Mbps (2 Mbps noiseless) Frequency Hopping Spread Spectrum (FHSS)

It also operates in the infrared spectrum providing 1Mbps and 2Mbps bandwidths. Both 802.11a and 802.11b both use Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) for the MAC layer with the only modification being to the physical layer. Hence the different data rate communications can co-exist allowing a transition to higher data rates.

2.3.3 802.11a (5 GHz)

The 802.11a standard supplements 802.11 and operates in the 5 GHz band and uses Orthogonal Frequency Division Multiplexing (OFDM) and optionally supporting data rates of 9, 18, 36, 48 and 54 Mbps and mandatory supporting data rates of 6, 12, and 24 Mbps. However the increase in data rates are not without a cost, the range of transmission is reduced. Noise, protocol overheads and interference/error correction reduces the actual data rates available to applications.

2.3.4 802.11b (2.4 GHz)

The 802.11b standard supplements 802.11 and operates in the 2.4 GHz ISM band and uses Direct Sequence Spread Spectrum (DSSS) and can operate at 1, 2, 5.5 and 11 Mbps.

2.3.5 802.15 Wireless Personal Area Networks

Wireless Personal Area networks are designed to provide short range, low data rate communications, effectively removing the need for wires between every day devices such as mobile phones and hands-free kits.

2.3.6 802.15.1 Bluetooth

The Bluetooth consortium has developed Bluetooth [27] and has standardised the protocol as IEEE 802.15.1 [37]. Bluetooth has been developed to provide device interconnection, using low cost components costing $10 or less and provide seamless connectivity.

2.3.7 802.15.4 Zigbee

Zigbee has been developed by the Zigbee working group [6] and has been standardised as IEEE standard 802.15.4. Zigbee uses a master-slave model of network communications and is aimed at home automation type tasks. It has been explicitly designed with the goals of keeping the cost of manufacture low, $1 per radio system or less and whilst also being very low powered for example a device should last between 6 months and 2 years on 2 AA batteries. The master-slave model is designed to reduce the energy costs of slave devices. Zigbee also includes the following features[2]:

- master/slave topology
- automatic network configuration
- dynamic slave device addressing
- power management
- CSMA-CA channel
- service discovery
- data rates ranging from 28 Kbps to250 Kbps.

Zigbee lacks a mechanism to change the transmission power of the radio, it is fixed.

2.4 Simulation techniques

To make ad hoc networks interesting, a large number of mobile nodes are needed. But until recently it was impractical and uneconomic to actually perform large scale experiments, so simulation becomes the method of experimentation for much research. Real small-scale tests are viable and both provide an insight into some of the problems in developing and deploying a working system, and provide a method to validate the simulation. The Monarch Project [29] used such a technique in [56]. However, the Millennium bridge in London which initially suffered from significant oscillations when first constructed has shown that simulation does not provide a perfect representation of reality but rather an approximation that is limited by the information fed into it and the model used.

The simulation systems in use by the ad hoc network community are NS [30], Glomosim [28], the commercial simulator OpNet [41] and many custom simulators and mathematical analysis packages.

[2]Source Zigbee Group Website [6]

2.4.1 NS

NS is an event driven network simulator written using a combination of C++ and TCL (more specifically OTCL). NS was initially written by the Network Group at LBL and is now maintained by ISI/UCB and is distributed under a range of different licenses, but is an open source style project. One of the main advantages of NS is the range of contributed code, yet this is also one of its main disadvantage with contributed code sometimes breaking existing code. The Monarch [29] project at CMU added mobility support to NS. NS now supports five different ad hoc routing protocols in the main source distribution with numerous others being available from other locations.

2.4.2 GlomoSim

Glomosim is an event driven simulator for mobile systems, it is built on top of the Parsec parallel simulation environment. Glomosim is written in C, and provides a set of well defined interfaces to components of the system. Unlike NS, Glomosim is completely compiled it has no scripting language components, which provides for faster simulations. Glomosim supports a slightly smaller number of simulation protocols compared with NS. Experimentations with Glomosim has found that its random number scheme to be insufficiently flexible and the work required to fix these problems would not be out weighed by any benefits derived from using Glomosim. The Glomosim project has ceased, with its work transferring to create the commercial simulator QualNet. QualNet's fixes are not being back ported to Glomosim.

2.4.3 OMNet++

OMNet++ [63] is an extensive open source simulator component based simulator library and framework written in C++. It has a split license model where it is free for academic and non-profit use, but commercial users must obtain a license. OMNet++ is widely used with many extensions and extensive documentation. Unfortunately given the license restrictions and my Industrial Case award, it is unclear whether we would have been required to purchase a license or not, so we did not pursue OMNet++ further.

2.4.4 OpNet

OpNet is a commercial network simulator for which a number of ad hoc protocols have been developed including AODV and ZRP. Since this is a commercial simulator this has not been investigated further. The non-commercial version of the simulator requires all source code developed from it to be released, this is in contradiction to my Industrial Case award contract.

2.4.5 Combined Simulators

Work by Wei Ye et al. at USC and ISI [84] focused on the combining of two simulators NS [30] and Arena [1] to provide a simulation environment for a set of robots to explore and evaluate. The Arena simulator is designed to simulate the operation of the robot in terms of its movement and its sensors, and provide reasonable approximations to the operation of real systems. NS provides the simulation of the network component of the system. Socket style communications are used to link the two simulators so movement in Arena is relayed to NS, updating the radio propagation models to account for the movement. The work showed that it was feasible to combine these two simulators. However the work did not examine energy usage of the simulated system, but equally did not preclude it.

2.5 Power Control

To provide true low power devices energy saving needs to be carried out at all levels from applications to hardware. Although processor speeds are increasing at roughly Moore's law, battery lifespan is growing much more slowly; at roughly 20% every ten years. Various techniques can be employed to reduce energy consumption, some of which can be carried out independently; others require coordination of various components.

2.5.1 Disk spin down

To keep a hard disk spinning can be expensive if it is "idle". However, there is also a cost associated with spinning up a stationary hard disk to its operating speed. So a trade off needs to be made; is it more efficient to keep a hard disk spinning "idle" or to spin down the hard disk and to spin it back up again when it is required, suffering additional latency in disk access ? In fact, there is a minimum idle time for the disk, for this to be more efficient. Since the hard disk is a slave device to the operating system and the applications it is difficult for the hard disk to predict when it should spin down and when it should keep spinning. If the operating system and the applications are power aware then disk access can be scheduled to provide optimal usage of the energy.

2.5.2 Processor stepping

Each processor cycle uses energy, so the higher the clock rate of a processor the faster it uses energy. If a processor is "idling" waiting for work to do it is wasting power. It can be more efficient to reduce the clock rate of the processor and take a longer period of time to carry out computations, thus leaving the processor to idle for less time. Intel's mobile Pentium III processor use processor stepping technologies. But reducing clock speed is not the most efficient thing to do, since if the clock speed is linearly reduced, then the energy is linearly reduced, but execution time is linearly increased. This only provides a benefit if the idle time of the processor is decreased. A more effective solution is to reduce the voltage to reduce the clock speed at which the processor runs, since reducing the voltage by a factor of n decreases the energy usage by n^2 [33].

2.5.3 Load balancing

Load balancing is a technique in which tasks are shared out between a collection of devices, with the idea that the overall performance of the system is maximised according to some metric. This metric maybe the the response time of the application, in the case of battery powered nodes, the amount of power consumed, or some combination of metrics. To reduce the energy consumption of a mobile device, it may be more efficient to transfer some of the processing task that it would perform to a more powerful host. The transferring of the task is not without cost, and the savings in transferring the task to another host must outweigh the costs involved in the transfer.

2.5.4 Communications

Work by Robin Kravets, Karsten Schwan and Ken Calvert [50] examines power aware communications for mobile computers, looking at the entire picture rather than a specific a subset of the problem such as power usage of the radio device, CPU power consumption in generating the traffic. However, ad hoc communication strategies are explicitly not investigated in this piece of work. But the strategy taken in this work to evaluate the overall system rather than a specific subset is one we wish to follow. Earlier work by Robin Kravets and P. Krishnan [51] proposed a new protocol for controlling when nodes are active. A node, acting as a master, defines when the base station, acting as a slave, is able to transmit information to it. This allows the mobile station to suspend its radio device to conserve power. However, as the work points out, leaving the device suspended for too long causes significant latency in communications, whilst suspending for too short a period leads to energy wastage. Hence it is proposed that the protocol be adaptive to the communications patterns of the system. But taking this a step further and exposing the MAC layer, layer 1 of the OSI model, to Application layer, layer 7, breaks the concept of layering but may be necessary if power aware applications are to be developed.

2.6 Ad hoc routing protocols

Since the 1970s wireless networks have grown in popularity. Ad hoc networks are networks of autonomous nodes that have wireless connections between each other. These connections can created and destroyed, changing the network topology as nodes change location, move out of range of other nodes or fail completely. Ad hoc networks pose an additional set of problems to those encountered in traditional fixed networks or wireless cellular networks. Dynamically forming the communications infrastructure from mobile devices is the source of these complications. One way of thinking about this is to imagine the problems caused by continually moving and changing the router you use to get from your local subnet to the rest of the world. How would packets get to or from you ? This type of question has to be addressed along with requirements that affect traditional routing protocols such as loop free routing, completeness and stability. The proposed solutions to this problem have focused on developing ad hoc routing protocols such as DSDV [69], DSR [44] and AODV [70] to cite three. Activity within the IETF on ad hoc networking is co-ordinated by the MANET group [3].

The general goal in ad hoc networks is to provide connectivity between hosts with no fixed infrastructure. Unless no path exists between sender and receiver, i.e. the network is partitioned, then it is always possible to get a message from one host to another through aggressive flooding of the network. Ad hoc routing protocols are trying to achieve this aim more efficiently. However, it is feasible that in very dynamic ad hoc networks that the only solution is to flood the network.

Given the postulated environment, it is clear that ad hoc routing protocols significantly underpin the foundations of the proposed work. Traditionally, ad hoc routing protocols have either reacted to changes in network topology as they occur or pro-actively disseminate topological information. Newer protocols use clustering techniques or have an awareness of power consumption or position or are hybrids of other ad hoc routing protocol classes. Each protocol has a different set of goals and priorities above and beyond those of providing a routing service in a dynamic, and thus demanding, environment.

In an ad hoc network each node can forward packets destined for other nodes. Nodes maybe be mobile and have fixed power levels which adds extra constraints to the network that are not present in fixed networks. Mobility of the network infrastructure may cause the network topology to change rapidly during the lifetime of the network, thus routing protocols for ad hoc networks must be able to adapt to the changing topologies.

In [68, page 17] Perkins discusses the policies nodes might adopt when transmitting, receiving and forwarding packets. A node that is energy constrained might take a different approach to participation in a global system to a node that is not energy constrained. Leaving aside security considerations and configuration issues, nodes in a fixed network that are powered are able to acts as routers for other nodes at little cost to themselves in terms of processing power [3].

[3]Assuming a moderate amount of packet forwarding. Although in an interrupt driven system, high levels of network activity can cause latency in the system, which is much higher than the processing requirements of the forwarded traffic would suggest.

However in an energy constrained ad hoc network, the additional constraint of a limited amount of energy requires nodes to be more cautious about what tasks they carry out themselves or on behalf of others. Tasks such as packet forwarding are expensive since they require reception, processing and transmission of packets. It can not be assumed that using the radio to transmit and receive packets is significantly more expensive than leaving a radio to idle. It is shown, respectively, in [77], [48] and [13] that the ratios for energy consumption for the different states idle:receive:transmit are 1:1.05:1.4, 1:2:2.5 and 1:1.2:1.7 (AT&T 802.11 PCMCIA card). Although these figures slightly differ, it does show that a radio in the idle state uses at least 50% of the energy of the receive state and at least 40% of the energy of the energy of the transmit state, hence it is not realistic to ignore the idle state. It is costly for a node to act as a forwarder for other nodes, since to remain in a listening state is expensive. In [82] it is shown that the energy consumption of the four ad hoc routing protocols AODV [70], DSR [44], DSDV [69] and TORA [4] are very similar when taking into account the idle energy usage of the nodes of the under the model proposed by [77]. This leads to approaches such as SPAN [13] and GAF [82] where the radios are turned off, rather than being left to idle, and awoken at intervals to listen for transmissions. Switching the radio off for periods of time causes an increase in network latency but increases the lifetime of the system. Other issues involved in ad hoc routing are set out by David B. Johnson in [43] and [45].

Since Elizabeth Royer and Chai-Keong Toh wrote “A Review of Current Routing Protocols for ad hoc Mobile Wireless Networks “ [75] in 1999, ad hoc networks have made significant progress. Many new classes of protocol have been developed, expanding the two main classes considered in [75], namely Source driven and Table driven protocols, to a whole collection of more specific classes. These classes are Hybrid Protocols, Geographically Aware Protocols, Clustering Protocols, Locally Repairing Protocol and Energy Efficient Protocols. The categorisation of routing protocols in [75] placed a clear distinction between Source driven and Table driven protocols. With the additional classes mentioned above, the distinction between protocols is not so clear. Protocols have properties of one or more classes or ad hoc protocol. For example ZRP [32] is a Hybrid protocol and has features of both Source and Table driven protocols. A comparison was carried out between AODV and DSR in [19]. In particular this survey shows that the two protocols studied show that in an environment with 100 nodes, 40 traffic sources and using a uniform traffic pattern that more control traffic is generated than actual traffic. Caching Strategies for ad hoc protocols are described in [35] but these mainly focus on the effects of different caching strategies when used with DSR.

2.6.1 Table driven routing protocols

Table driven or *pro-active* ad hoc routing protocols were amongst the first ad hoc routing protocols. There are two types of table driven protocol: link state and distance vector.

In link state protocols each node maintains a view of the network topology with a cost associated with each link. Periodically the node broadcasts the costs of its outgoing links, which ensures that the other nodes have a consistent view of the network. The updates cause nodes to apply a shortest path algorithm to discover its routes to all the nodes in the network. Since nodes broadcast updates at different times it is feasible that inconsistent information maybe present in some routing tables and this can lead to short lived routing loops. Link state protocols suffer from the problem that in a highly dynamic environment the amount of control traffic is significant, if all nodes are to achieve an identical view of the network. Recent protocols STAR [68, Chapter 4] and some cluster based algorithm are designed to operate in a situation where not all nodes have a consistent view of the network.

In distant vector protocols each node contains in its routing table a metric for each node, often the number of hops to the destination. Periodically each node broadcasts a copy of its routing table to its neighbours. If nodes send stale information to each other, then distance vector protocols can be slow to converge. However they do use less memory per node than link state protocols and have more localised updates. But it is still possible for an update to the distance metric for to a node to propagate throughout the entire network. Distance vector protocols also suffer from a count to infinity problem that can be partially addressed by a combination of techniques such as Poison Reverse or Split-Horizon. DSDV is an example of a distance vector based ad hoc routing protocol. The main limitations of distance vector ad hoc routing protocols is the periodic transmission of routing information that makes the network control traffic proportional to time rather than actual network traffic. In a highly dynamic network, distance vector protocols are too slow to adapt and converge once a change has occurred. In very highly dynamic networks they are unlikely to converge at all. CGSR and WRP are also distance vector protocols.

2.6.1.1 Destination Sequenced Distance Vector

Destination Sequenced Distance Vector, DSDV, was described in [69] and [68] section 3.3. This is one of the first ad hoc routing protocols and is basically an adaptation of the Bellman Ford algorithm [46]. It was developed by Perkins et al. in 1994 and has been superseded by other ad hoc routing protocols, including AODV also by Perkins [67]. Each node maintains a list of all other nodes in the network along with a next hop to them, the number of hops to the destination and a sequence number. The sequence number is used to distinguish stale routes from fresh routes. Routing table updates are periodically broadcast throughout the network to ensure consistency. To minimise the effect of these broadcasts there are two different types of broadcast, a full update and a partial update.

One problem with DSDV is that it broadcasts routing table updates throughout the network, many of these changes may never be used by the hosts receiving them and hence are wasteful of bandwidth. Since DSDV is not source routed then the complete route to the destination is not required, so it is more efficient to locally correct a route failure as is the case in cluster based routing protocols.

2.6.1.2 The Wireless Routing Protocol

The Wireless Routing Protocol, WRP, is a table based routing protocol [60], where nodes inform their neighbours of updates messages, which must be acknowledged. A constant level of network activity must be maintained by the node either through it sending data or via *hello* messages. The interesting component of WRP is that it ensures loop free behaviour by nodes communicating the distance to the destination and the second to last hop of each route. Each node is then forced to perform consistency checks of predecessor information reported by all its neighbours. This produces a faster link convergence upon node failure and ultimately ensures a loop free situation. WRP suffers from the same problems that all other table driven protocols suffer from.

2.6.2 Source driven protocols

Source initiated on-demand or *reactive* routing takes a different approach to table driven routing; routes are only created when a route to a destination is required. The source initiates a route discovery process, when the process returns an optimal route is chosen from the set of routes found. This route is established and maintained by a route maintenance process. The route is maintained until it is no-longer required or the destination is inaccessible, via every possible path.

Source driven protocols suffer from a scaling problem. If the average length of a path between two nodes is proportional to the diameter of the network, in hops, then, as the network gets wider (larger), the average path length gets longer. The number of bits transmitted due to the routing protocol increases, since each data packet contains a list of the nodes that the packet must traverse in order to reach its destination. It is feasible that source routing could break down altogether if the number of bytes of routing protocol information is greater than the maximum packet size. This problem is soluble but it makes source routing complex.

In very large highly dynamic networks the source route establishment process may be too expensive if the route is outdated before it is used sufficiently often to make the discovery process overhead a small fraction of the overall cost of the transmission. In this situation, directed flooding could produce improved performance.

Source routing techniques come into their own when used in small to medium sized networks where the mobility is moderate; indeed, this is a stated assumption of DSR. Source routing has a lower network overhead than broadcast-based routing techniques in this type of environment. This is due to nodes taking part only maintaining routing information for the communications they are involved in. Source routing protocols are less prone to suffer from routing loops and the slow convergence of routing tables found in table driven protocols. The information used to produce a source route is fresh because it has been generated on demand.

In very slowly changing networks, the overhead of having source routes in each packet reduces the throughput of the network, making source routed messages more expensive than table routed packets.

Examples of source instated routing protocols are DSR [44], AODV [70], TORA [4] and ABR [79].

2.6.2.1 Dynamic Source Routing

Dynamic Source Routing, DSR was developed by the Monarch project [29] and is a dynamic source routed protocol that carries out on-demand routing. DSR was initially set out in [44] and then standardised in RFC 4728 [42]. DSR is a purely reactive routing protocol that uses source routing to discover routes to the destination. In DSR, a node wanting to sends a packet to another node first sends a *Route Request* packet that is received by all nodes within radio range. If the node knows a route to the destination or is the destination it replies to the originating host with a *Route Reply* packet containing the route to the destination, otherwise it appends itself to the list of hosts traversed by the *Route Request* packet and then retransmits the packet to its neighbours. DSR incorporates a number of techniques to remove route looping etc. In DSR nodes also cache learnt routes in an attempt to reduce the amount of routing related traffic in the network. A follow up paper by Yih-Chun Hu and David B. Johnson [35] discusses the caching strategy used in DSR and explores the use of other strategies. The use of source based routing rather than table based routing imposes an additional overhead on the network, since every packet carries a complete list of the hosts it has traversed. In a power aware environment the additional cost of transmitting the these extra bits of information may be costly. An additionally limitation is as the network grows the size of the control messages and the source routing component of network traffic increases providing DSR with a scaling problem.

In general DSR is well suited to the ad hoc network environment (for which it was designed), the use of on-demand routing means that the level of traffic on the network is a function of the load on the network, the caching strategy used and the size of the network, rather than as a function of time as is in table driven protocols such as DSDV [69]. DSR is not well suited to slowly changing or stationary networks since the overhead placed on the network by source routing is much higher than by table based routing where only deltas to the routing table were sent, as in BGP. However the use of caching and parameters tuned for slow moving networks could reduce this unsuitability in DSR. One of the assumptions made when designing DSR is that it would be used in ad hoc networks where the speed at which hosts host move is "moderate" compared with the packet transmission latency and the wireless range of transmission. This assumption targets DSR to provide routing capabilities for ad hoc nodes that do not change too fast or too slow. In rapidly changing networks the use of caching could adversely effect the performance of a network since the cached route maybe consistently incorrect resulting in packets being incorrectly routed and a *Route Error* packet being sent.

One fundamental flaw in DSR is its lack of Multicast support natively in the routing protocol, as is the case AODV.

The experimental section of [44] only examined the performance of DSR in a relatively small environment using a relatively small number of nodes, 24 or less. The scaling properties of DSR to a larger environment and a larger number of nodes were not shown and this is an area of work which should be explored. Another area of experimentation that is lacking is a comparison of DSR against another ad hoc routing protocol such as DSDV, as AODV did not exist in 1996.

2.6.2.2 Ad hoc On Demand Distance Vector

Ad hoc On Demand Distance Vector, AODV was initially set out in [70] and has been standardised in RFC 3561, [67]. AODV is an on demand ad hoc routing protocol that provides both unicast and multicast routing. In contrast to DSR, AODV does not use source routing but rather dynamically creates routing entries in intermediate nodes between the source and destination. ADOV adopts a similar approach to DSR in that the source wanting to send information initiates a Route Request, *RREQ*, which is broadcast throughout the ad hoc network until it reaches a node, that maybe the destination itself, which has a route to the destination. This node then propagates back a Route Reply, *RREP* to the source. The traversal of the network by the *RREQ* and *RREP* packets is the mechanism used to establish routing entries in the intermediate tables. Various mechanisms are used to ensure that routing loops do not occur and that only a single path through the ad hoc network is established.

The disadvantages of establishing routing table entries in the intermediate nodes is the entries need to be maintained and expired if they become unused. With source based routing the source of the route is aware of when a route is no-longer required and all that is required is to locally drop the entry, but this is not possible for protocols such as AODV where the intermediate nodes have no knowledge of the usefulness of the route. Techniques such as Route Table Management and Path Maintenance are required to address this issue. To ensure the consistency of a path, nodes must inform other members of the path when they physically move. Movement of the source node initiates a new discovery process, where as movement of the intermediate nodes on the path cause special *RREP* messages to be sent too affected sources. Optionally, *Hello* messages can be sent; the sending of *Hello* messages reduces some of the efficiently AODV gains by being an on-demand protocol. ADOV includes a mechanism for repairing links that are unexpectedly broken. The upstream node propagates an unsolicited *RREP* with a new sequence number and a hop count of infinity. This *RREP* message is propagated to all active nodes. The source, upon receiving the *RREP* message may restart the discovery process if the route is still required. Forcing the source to re-initiate the discovery process ensures that the route is still required and is loop free, however it is wasteful of resources since it maybe possible for the upstream node which sent the *RREP* message to initiate a local repair of the routing table. This approach is taken in cluster based solutions where each cluster is responsible for the routing through itself, and is compatible with the intra domain routing used by BGP.

AODV discovers its immediate neighbours either via observation of the neighbours broadcasts or via *Hello* messages that have a limited local scope. The use of local *Hello* messages ensures that only nodes with bi-directional links are considered to be neighbours. The inclusion of *Hello* messages in AODV is to remove requirement that the underlying MAC protocol has neighbour discovery and maintenance features. The *Hello* messages provide an additional overhead that may be unnecessary; this is acknowledge by the authors and is being investigated to see if it is necessary.

The experimental section of the AODV [70] examines the performance of AODV in a much larger range of scenarios than the DSR work [44], showing the performance of the protocol with a larger range of hosts, between 50 and 1000 hosts. However no experimental comparison was made between ADOV and other ad hoc routing protocols, such as DSDV and DSR. The comparison of ad hoc routing protocols is examined in [11] and [18].

2.6.2.3 Temporally-Ordered Routing Algorithm

Temporally-Ordered Routing Algorithm, TORA, was first proposed by Vincent Park and Scott Corson in [64] and its applicability to mobile tactical networks is set out in [65] is defined IETF Draft [4], version 04. TORA is designed to be a highly adaptive source initiated ad hoc routing protocol using link reversal techniques to provide multiple routes for any given source/destination pair. A key concept in TORA is that when a topological change occurs the impact of the resulting control messages should be local in scope, rather than having global scope as in DSR and AODV.

One major disadvantage of TORA is the need for the nodes to have synchronised clocks; a suggested mechanism for clock synchronisation is the use of a GPS time signal. However in some environments, such as on Mars or underwater, it is not possible to have a GPS signal and hence another reliable time synchronisation mechanism is required. Time is a component of the *height* metric used during creation and maintenances phases. The height metric is used to establish a directed acyclic graph routed at the destination and terminated at the source. The *heights* are such that information always flows *down* towards the destination from the source. In [75] it is pointed out that TORA does have the possibility to create oscillations when partitions are being detected concurrently and new routes being built dynamically upon one another, however this is temporary and the routing tables will eventually converge. TORA does not natively support multicast routing.

2.6.2.4 Associativity-Based Routing

Associativity-Based Routing, ABR [79] takes a different approach to ad hoc routing than other protocols. The approach taken by ABR is to use the most stable route from a source to a destination. Stability is determined by each node periodically broadcasting a beacon, each of its neighbours holds a counter for each of its neighbours and increments the counter for each tick, beacon. The counter is resent each time the node moves out of range. Broadcast discovery is used to discover routes. The returned routes to the source have all the addresses of the nodes in the route and their associativity ticks for their upstream neighbour. The best route is chosen by examining the ticks along the path, and the number of hops is used in case of a tie. Using associative ticks ensures that the most stable route is chosen, which is the fundamental objective of ABR.

The use of beaconing and the broadcast discovery imposes an overhead on the network, which could be avoided by sending beacon messages only when the node is not broadcasting other packets.

2.6.2.5 Signal Stability Routing

Signal Stability Routing, SSR [20] is an on-demand routing protocol that uses the signal strength and the stability of nodes as a routing metric. The protocol is designed to favour routes with a stronger signal. SSR uses periodic beacons to determine signal strength and that it is loop free. However the processing of packets in SSR is slightly convoluted and in general other ad hoc protocols are better designed than SSR, however the concept of using signal strength as a metric is good.

2.6.3 Hybrid routing protocols

Hybrid routing takes the best of pro-active and reactive routing and combines the two. In hybrid routing each node maintains a zone around itself where it carries out pro-active routing. The assumption is that inside a zone the network traffic is higher than outside the zone. So, if network control traffic is proportional to actual traffic, it is less efficient than having control traffic proportional to time, i.e. pro-active routing. However where traffic is lighter, i.e. outside the zone, then network control traffic proportional to network traffic is more efficient than having than control traffic proportional to time. Thus traffic inside the zone is routed pro-actively and traffic outside the zone is routed reactively. To make hybrid routing efficient, the size of the zone must be able to dynamically adjust to changes in traffic patterns.

Hybrid protocols are susceptible to the problems of both table driven and source driven protocols if the zones are configured badly. Conversely, if configured correctly, many of the problems due to table and source driven protocols are reduced and the advantages exploited. Combining the two routing concepts is not without an extra management cost, and in ZRP this comes in the form of an additional protocol.

2.6.3.1 Zone Routing Protocol

Zone Routing Protocol, ZRP is defined in the IETF draft [32] and has been developed by Z. Hass and M. Pearlman of Cornell University [31].

ZRP is a hybrid routing protocol that groups nodes into zones. Each node pro-actively maintains a routing table for the nodes within its zone. Outside the zone reactive routing is used. In ZRP the size of the zone can be adjusted dynamically to adjust to the conditions. Hence the degree to which ZRP is pro-active or reactive is dependent upon the size of the zone the larger the zone, the more pro-active ZRP becomes and vice-versa.

Again ZRP lacks the ability to tune the radio device's transmission power, if knowledge of the actual distance between neighbours were known and the radio tuned appropriately, then routing within the zone could be improved further.

2.6.4 Cluster-based routing protocols

The use of clustering and hierarchical approaches to routing are valuable methods of breaking the problem down into manageable components for which tasks or sub-tasks can be allocated. The following set of work provide a good starting point for this approach, [52], [18], [58], [7], [66] and [71]. Although only CGSR is described here.

The intent of clustering is to group nodes into geographically close collections, which form peer-to-peer links for communication between nodes in the cluster. Each cluster has a gateway node(s) through which information destined for outside the cluster must pass. In cluster based networks, routing is defined as occurring via clusters rather than via nodes. Each cluster is responsible for routing packets through itself.

Clustering solutions do not specifically address the energy consumption issues. The use of cluster gateways can impose additional overheads, when a node in a cluster is in a different cluster to a near-by node with which it wishes to communicate, the communication must occur via the cluster gateways, rather than directly. Additionally clusters place an additional management overhead on the network.

However the positive side of clustering solutions is that they provide in path fault tolerance for end-to-end routing. For example, in a source based routing protocol, if a node moves, all the routes passing through that node must be recomputed by the source(s); in a cluster based protocol, only the path through the cluster must be recomputed.

2.6.4.1 Clusterhead Gateway Switch Routing

Clusterhead Gateway Switch Routing, CGSR is a cluster based routing protocol initially proposed in [14]. CGSR is based on DSDV, which has be modified to pass all messages not destined for inside the local cluster to the appropriate cluster gateway node, via the cluster head node. Each cluster has one or more cluster gateway nodes, a cluster gateway node is a node which is in range of one or more cluster head nodes, from one or more different clusters. Each cluster has a single cluster head node that is determined using a distributed algorithm, CGSR uses the Least Cluster Change algorithm, LCC, to reduce the amount of time nodes spend electing cluster heads. In LCC the cluster head node only changes when two cluster heads come into contact with each other.

Because CGSR uses DSDV as the underlying routing protocol it is prone to all the inefficiencies associated with it. DSDV is used to periodically broadcast a cluster member table to all nodes in the network, where each node is associated with its appropriate cluster head node. Additionally a routing table is maintained so that nodes know what the next hop to a destination is. When a node receives a packet for a destination it first looks up in the cluster member table the cluster head node associated with the packet and then looks up the in the routing table the next hop to the selected cluster head node.

2.6.5 Power based routing protocols

One approach to reducing the energy consumption of ad hoc networks is to switch off the radio for a period of time and then switch it back on again. The following two pieces of work investigate this effect and demonstrate that the life span of the network can thereby be extended, [81] and [13]. The work, GAF, by Ya Xu et al. at USC/ISI, combines switching off the radio device with an adaptive fidelity algorithm for nodes in close proximity to each other. This approach could be taken further by having N-levels of power rather than binary power. Combining this with adaptive fidelity could provide additional energy savings.

The SPAN work of MIT is an energy-efficient algorithm to decide whether a node should form part of a backbone of ad hoc nodes that provides network connectivity. The nodes decide whether to join the backbone or not using a distributed algorithm. The algorithm used in SPAN is a possible candidate for deciding whether to modify the transmission power or to relay traffic via other nodes.

The work by Qun Li et al of Dartmouth [53] describes a routing protocol which aims to increase the lifetime of the entire system. This work is of interest to our proposed work although it does not take into account the ability for radio devices to modify their transmission power. However, one of the interesting aspects of the work is that the decision process is localised to clusters with each cluster being given the task of routing messages through themselves using the most efficient route.

The general problem with power aware routing protocols is that they often assume that the radio devices are either on or off, rather than having a variable transmission power. Some 802.11b chip-sets allow the modification of the transmission power, but Zigbee [6] does not, since there are no applications that require it, a catch 22 situation.

If low power energy devices are to be successful then every aspect of their operation needs to be designed with minimising energy consumption as an objective, ad hoc routing protocols are no exception. In addition to PARO and Leach, power-aware localised routing in wireless networks is describe in [78] and [34].

2.6.5.1 Power-aware Routing Optimization

Power-aware Routing Optimization, PARO is defined in [25], also see [22] for related work. PARO is described further in [23] and focuses on energy efficiency in personal area networks, such as Bluetooth [27]. PARO takes a similar approach to our work proposed, but limits its scope to a small area where nodes can be in direct radio contact with each other rather than having global routing as we propose. PARO modifies the transmission power to better utilise energy resources. PARO uses a pricing policy based approach where each node determines the price it would charge to forward a packet. For example nodes with non-rechargeable batteries will price their forwarding of a packet higher than a node with a rechargeable battery, or a node with a fixed power supply. The authors acknowledge that PARO lacks "wide area" support and state that this is part of the future work.

2.6.5.2 Low-Energy Adaptive Clustering Hierarchy

Low-Energy Adaptive Clustering Hierarchy, LEACH is primarily interested in providing a clustering based solution to the problem of low power ad hoc networking rather than adjusting the transmission power of the device. The approach taken by the LEACH work is of interest and could compliment modification of the transmission power of the transmitter. However the LEACH work is a modified MAC protocol upon which is built a routing protocol that uses a clustering algorithm. The cluster head node is randomly rotated amongst the membership of the cluster thus spreading the energy usage of the system across the cluster. The random cluster head allocation algorithm is biased to favour nodes with more remaining energy. Each cluster head schedules the non-cluster head nodes to switch their radios off during all non-transmission periods to increase the lifetime of the device.

2.6.6 Position based routing protocols

Position-aware routing protocols use geographic information to aid routing. Having geographic information can reduce the amount of wasted network traffic by steering traffic towards its destination. The use of geographic information can be used to reduce the impact of flooding algorithms, which have been shown to be wasteful of network bandwidth. The idea behind geographic routing protocols is that if you know roughly where a node is it is pointless looking for it, flooding, where you know it isn't, as is the case with other ad hoc routing protocols.

There are many position aware protocols but the main criticism of them is that they are reliant upon GPS data to provide positional information. However, it is possible to build relative topology maps by measuring round trip times to devices. GPRS is one example of a position aware routing protocol but it relies on GPS data which currently makes it impossible to use in an environment such as Mars. An alternative solution must be found.

Other than the additional cost of maintaining and obtaining the positional information of nodes in the network, geographic based protocols have few drawbacks other than the those already associated with the routing protocol family they use for the underlying structure. The main advantage of geographic routing protocols is that they direct traffic to only those relevant portions of the network.

2.6.7 Greedy Perimeter Stateless Routing

Greedy Perimeter Stateless Routing, GPSR is explained in [47] and is a geographically aware routing algorithm. GPSR uses two techniques for forwarding packets, whenever possible it forwards packets in a *greedy* manor, otherwise it uses *perimeter* forwarding. Greedy forwarding in GPSR is based upon forwarding to the packet to which ever of your neighbours is closest to the destination, repeating the process until the packet reaches its destination. Node positional information is periodically broadcast to neighbours, this is pro-active routing but its effect is minimised by piggy backing the positional information on the back of other data packets. Perimeter forwarding is used to route around radio voids, when a radio void is encountered packets are routed around the edge using a right-hand rule based approach.

2.6.7.1 Location Aware Routing

Location Aware Routing, LAR, is described in [49]. It is an on demand routing protocol similar to DSR, but uses geographic information obtained via a GPS system to restrict control traffic flooding to an area.

2.6.7.2 Geography-informed Adaptive Fidelity

Geography-informed Adaptive Fidelity, GAF adds an extra layer of protocol on top of an existing ad hoc routing protocol and uses the extra layer of protocol to provide information about the energy status of node and is defined in [81]. The system requires nodes to know about there location, via GPS for example, although work is being carried out into only using radio based location techniques. The GAF system splits the network area up into a grid. In a grid square there maybe any number of nodes only one of them has an active receiver / transmitter, the others are in a sleeping pattern. Each sleeping node periodically wakes up and may swap to being the active node. This technique causes a drop of 40% to 60% in the energy usage of the system [82]. Although GAF imposes an extra overhead on the network it sensibly separates functionality from existing routing protocols. In this way the routing protocol used can be chosen based on the network conditions. GAF and LAR are able to operate seamlessly together.

2.6.7.3 Fisheye State Routing

Fisheye State Routing, FSR is defined in [2] and is a modified link state algorithm where each node maintains a topology map of the network. Where it differs from other protocols is that it does not flood the network each time a topology change is detected. Periodically each node exchanges the changes it has noticed with its neighbours only. The entries in the tables with the smaller sequence number are exchanged with those with larger sequence numbers, this table exchange process is very similar to the table exchange process of DSDV.

Chapter 3

The energy consumption of a low power radio device.

In this chapter we investigate the energy consumption of a low power radio device and obtain P_{tx}, P_{rx}, and P_{idle} for the device. We identify these modes of operation from the traces of the energy consumption of the device by modifying the device's firmware to include and exclude certain events. The low power radio device investigated was implemented using a custom development board developed by Philips Research Labs (Redhill) for low power radio applications. Among its uses was the early prototyping of the IEEE 802.15.4 protocol at 900MHz. Whilst this is a custom development device with the firmware available to us, the measurement techniques described here are applicable to generic commercial radios and the radio parameters are representative of a low power radio device. We are specifically looking at a different radio system to 802.11 , to see whether the 802.11 parameters, as measured by Feenay et al. in [21], are generalisable as a generic radio model.

When simulating low power radio devices for use in ad hoc networks, three key parameters are used to calculate the energy consumption of the device: the energy consumption when transmitting, P_{tx}, the energy consumption when receiving data, P_{rx}, and the energy consumption when the device is idle (i.e. when not transmitting or receiving data), P_{idle}.

Simulation of ad hoc networks is the approach taken by many when researching ad hoc networks; however, simulations are only as accurate as the models and parameters used in them. This chapter aims to improve the energy consumption model of the simulator by measuring the important parameters of an actual low power radio device, and in so doing answer objective 1 of the hypothesis.

3.1 Related work

Related work has been carried out by [21], [77], [50] and [51], but this work has focused on the 802.11b family of devices. None of these pieces of work has had access to the device firmware and none have been able to carry out an event analysis of the energy consumption traces as we do here. The techniques used here to calculate the energy consumption of different stages of the operation are similar to those used by Feeney et al. [21]; however we take these techniques a stage further by disabling specific sections of the firmware to aid identification.

3.2 Experimental setup

Given the nature of the device, and low power radio devices in general, it is infeasible to measure the power consumption of specific components. It is, however, simple to measure the overall power consumption of the device. The overall power consumption of the device is , $P = I_{cc}V_{cc}$, where Vcc is the voltage across inputs to the circuit, [1] and I_{cc} is the current flowing through the device, shown in figure 3.2. I_{cc} can be measured by measuring the voltage, V_r, across a resistor, R, placed in line with the supply V_{cc}, $I_{cc} = V_r/R$. As the device performs different functions, computation, transmission, reception, etc. I_{cc} varies.

Using a digital storage oscilloscope to measure and store V_r, the power consumption of a specific event starting at time t_0 and finishing at time t_1 can be measured by applying equation 3.1 .

$$P_{t_0,t_1} = \frac{V_{cc}}{R}\sum_{t=t_0}^{t_1} V_{r_t} \tag{3.1}$$

Equation 3.1 assumes that both V_{cc} and R are constant with respect to time throughout the experiment. Figure 3.2 shows the experimental configuration.

Three versions of the device firmware were developed; a version that sources packets (a source), a version that receives packets (a sink) and a version that receives packets and retransmits them (a relay). The source device can be configured to generate packets of 9, 39, 69 or 99 bytes continually at a rate of 5 packets per second, with its transmitter circuitry either enabled or disabled. The devices are used in two different scenarios: individually and in pairs. Each experiment was repeated 5 times and the results averaged. Where specific events need to be identified in each experiment, such as the start of a transmission, these are manually identified for each iteration of an experiment.

[1] The positive voltage input is called the Common Collector (cc) and V_{cc} is measure between the common collector and ground (GND)

3.3 Case study specific setup

A custom development board developed by Philips Research Labs (Redhill) for low power radio applications was modified to remove all software layers except the Medium Access Control layer (MAC) and Physical layers to enable the power consumption of only the radio interface software and hardware to be measured. The development board has an 8-bit Mitsubishi 3807 [16], a serial interface [57], a 512Kbit eprom [76] and a RFM TR1001L radio module [74] . The MAC collision avoidance scheme in use is CDMA/CA. The digital storage oscilloscope used here is a HP Model 5450C [5].

The digital storage oscilloscope was used to measure and store V_r, the power consumption of a specific event starting at time t_0 and finishing at time t_1 can be measured by applying equation 3.1 .

Figure 3.1: *Philips Mitsubishi 3807 development board*

3.3.1 Firmware

When the experiment was initially designed, the effect of sending, receiving, and relaying packets upon the energy usage of a particular device was unknown. In order to investigate this problem space, three different versions of the firmware were developed: a source, a sink, and a relay. The firmware was designed to run unaided, so, for each different configuration of the experiment, a separate version of the firmware was needed. The device has no other input/output mechanism other than the RS232 serial interface. To eliminate the power requirements of the serial interface, this was disabled after development.

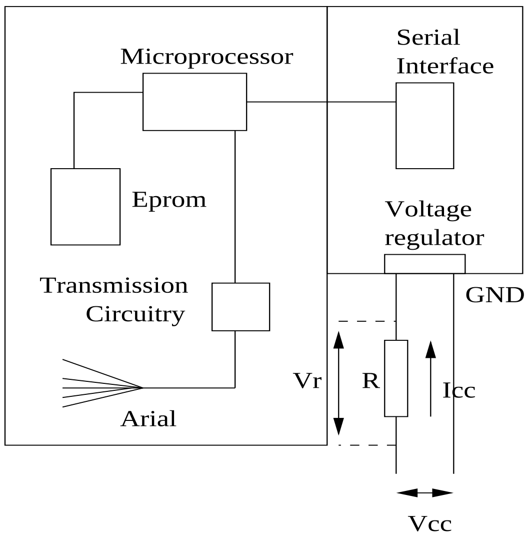


Figure 3.2: *Experimental Setup*

3.3.2 Packet format

The packet format used in the testing is loosely IP/UDP-like (figure 3.3). The packets are continuously sent by a source at a fixed interval, specified in the packet[2], with a fixed data payload. The minimum sized packet is 9 bytes, and the maximum is 255 bytes.

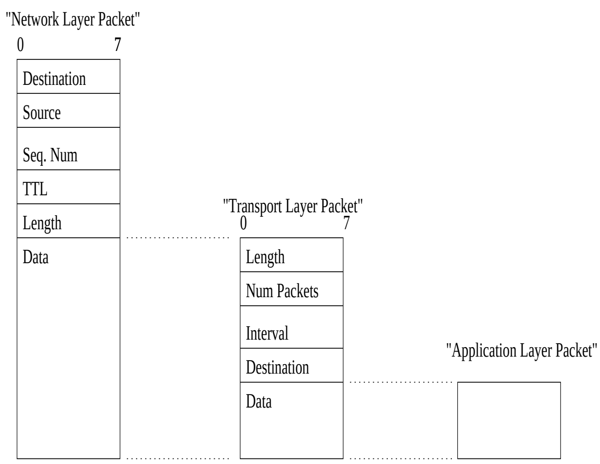


Figure 3.3: *CBR packet format*

3.3.3 Scope configuration

The HP (model 5450C) oscilloscope used in these experiments can store up to 8192 different points about a trigger point. The trigger point is determined by a specific change in the variable being measured; typically, this constitutes a large increase or decrease in the variable. In this experiment, the trigger was set to be when V_r rapidly increased, either when the device transitioned from “tx off” to “tx on” or from “rx off” to “rx on” (figure 3.4). If no trigger event is

[2]During the experiments the packets were sent at a rate of 5 per second.

present, the oscilloscope can be set to trigger periodically.

3.4 Implementation problems and solutions

In this section, we detail some of the tools and techniques used during the development process. The development board firmware was programmed from a Windows PC via the serial line using a custom program. The development board has limited input and output functionality, all input is via the serial device or the radio device and all output is via the serial device, the radio device or the 4 bi-colour LEDS. To aid debugging during development, a mixture of turning on the LEDs to represent different events and echoing of characters to the serial line was used to trace the program through its execution sequence. We did not have access to the full debugging system. The use of the serial line extended beyond this to include controlling of device level functions such as 'start transmission', 'stop transmission', and 'send packet "X"'. The serial device interface allows debugging information to be sent from the device to the PC attached via the serial line. Packets received by the device are sent to the PC along with various other statistics from the attached device.

3.5 Generic technique

Firstly, we aim to find the quiescent energy consumption costs of the sink, relay and source devices in their different phases of operation (transmitting data, awaiting or receiving data and idle). Following on from the isolated tests of the devices, we test the interaction of the devices. Specifically, we look at the interactions between a source and a relay, and a source with a sink or a relay. Because sinks do not transmit packets, they cause no interactions with sources, hence we do not need to measure the effect of a sink on a source. None of the devices uses periodic beacons to do MAC level signalling or channel control. This is not necessarily the case in general for low power radio devices. The firmware used has the ability not to turn the transmitter circuitry on, enabling the costs of transmission to be separated from the costs of computation. To reduce levels of experimental error, each experiment was repeated 5 times and the results averaged.

3.5.1 Identification of regions of the trace

From analysing different energy consumption traces of P and from an understanding of the execution sequence of the firmware it is feasible to identify specific events in the voltage trace. Figure 3.4 shows the entire P trace for a transmission from a source, showing the key events. From the source code, the order of execution for the transmission of a single packet is to switch off the receiver (rx_off), turn on the transmitter (tx_on) and send the data. Then turn off the

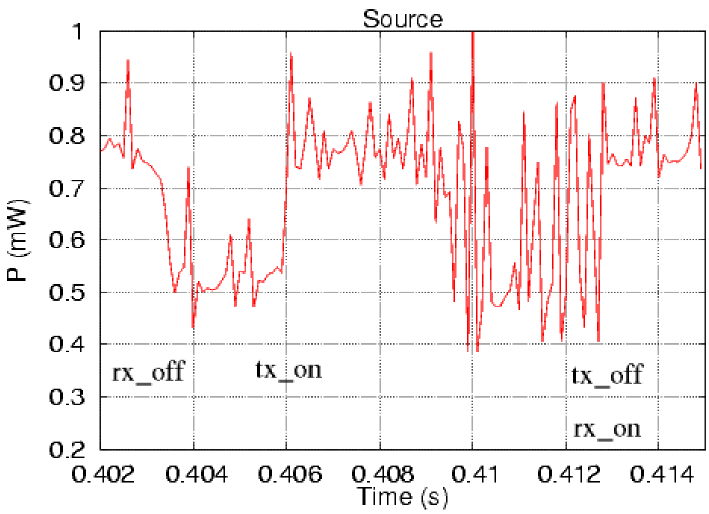


Figure 3.4: *P of a source with its transmitter enabled - marked up to show key events*

transmitter (tx_off) and turn back on the receiver (rx_on). The device polls for the start of reception of a packet or for an event to occur. These events are observable in the power trace.

3.6 Single device experiments

Here we aim to find the quiescent energy consumption costs of the sink, relay and source devices in their different phases of operation (transmitting data, awaiting or receiving data and idle). The energy consumption of the source device in its different phases of operation (transmitting data, awaiting data, idle) is identified by comparing a source with its transmitter enabled with a source with its transmitter disabled. Additionally in this set of experiments we aim to investigate whether the length of the data packet has any effect on the energy consumption during transmission of a packet by sending different sized packets (9, 39, 69 and 99 bytes[3]) and comparing the effect on sources with their transmitter enabled and disabled.

3.6.1 Sink and Relay

A sink and relay device were separately run in the absence of any other device and P measured, figure 3.5 and 3.6.

The variation in P is a result of the device's control loop, the overall power consumption of a sink and the relay remains constant. Table 3.1 shows the average energy consumption of a receiver and a relay over the full period of the experiment.

The exact cause of any individual peak in the energy consumption of receiver, figure 3.5, is not known. We can but speculate that the large periodically repeating peaks for example at 0.412s is caused by the system oscillator. The oscillator for these devices is operating at

[3]The packet sizes chosen are regular increments of 30 bytes from the minimum size packet of 9 bytes.

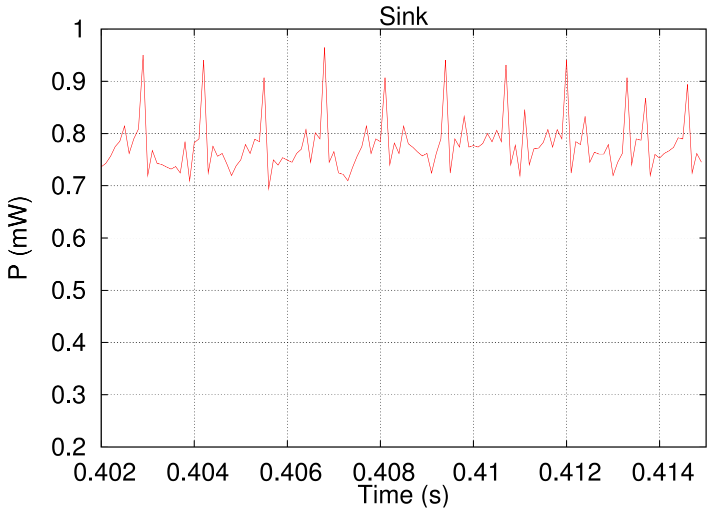


Figure 3.5: *P (mW) consumption for a sink*

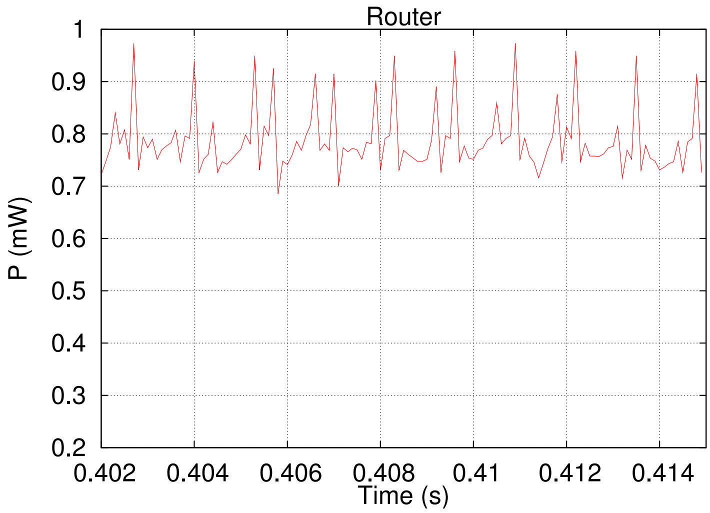


Figure 3.6: *P (mW) consumption for a relay*

7.3728MHz, which would give a peak-to-peak distance of 1.37^{-4} seconds which is approximately the peak-to-peak distance seen in figure 3.5. The source of the additional large peaks seen in the relay trace, figure 3.6, are not known.

Device type	Idle (mW)	Rx (mW)	Tx (mW)
Sink	NA	0.78	NA
Relay	NA	0.79	NA

Table 3.1: *Source and Relay power consumption (mW)*

3.6.2 Source

The source firmware allows for the transmitter circuitry to be disabled, isolating the costs due to the transmission. This can be seen in figures 3.7 (in which the transmitter is disabled) and

3.8 (in which the transmitter is enabled). By comparing figures 3.7 and 3.8, the transmission can be isolated: it occurs between 0.406s and 0.413s in figure 3.8. Before each transmission, the receiver circuitry is switched off and the device transitions to the idle state. Having isolated the transmission, the idle phase is from 0.403s to 0.406s, in figure 3.7.

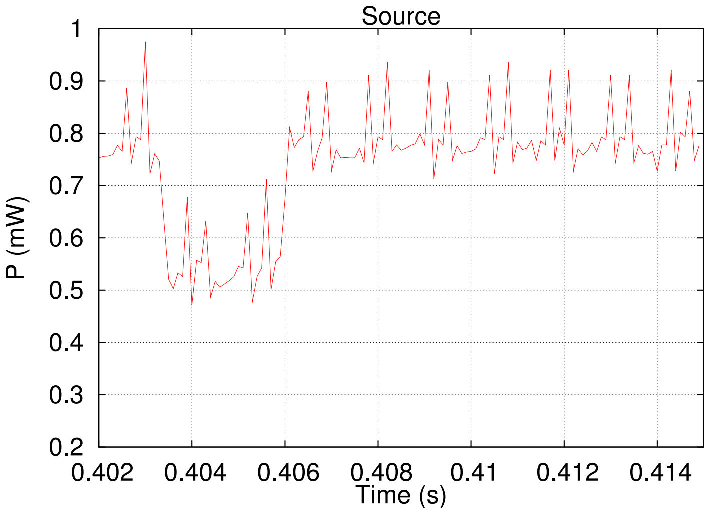


Figure 3.7: *P of a source with its transmitter disabled - 4 byte data packet, 9 bytes total*

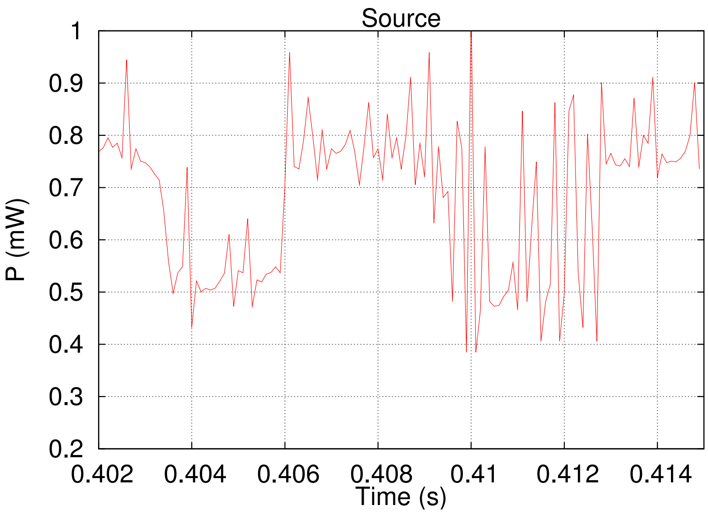


Figure 3.8: *P of a source with its transmitter enabled - 4 byte data packet, 9 bytes total*

Table 3.2 shows the costs of the device in each of its different phases of operation; idle, receiving, and transmitting.

Transmitter	Enabled			Disabled		
Bytes	Idle (mW)	Rx (mW)	Tx (mW)	Idle (mW)	Rx (mW)	Tx (mW)
09	0.22	0.78	0.7	NA	0.78	NA
39	0.21	0.77	0.68	NA	0.79	NA
69	0.23	0.79	0.72	NA	0.78	NA
99	0.24	0.81	0.75	NA	0.79	NA

Table 3.2: *Source consumption (mW)*

3.6.3 Summary of results

The cost of reception was found to be 0.79 mW (SD 0.01[4]), transmission was found to be 0.73 mW (SD 0.03) and the cost of being idle was found to be 0.23 mW (SD 0.01). Inspection of the data implied that their might be a relationship between the length of the transmission and the energy consumption, unfortunately with only 20 samples there is insufficient data to be statistically significant. This is an acknowledged design decision to limit the number of repetitions of each experiment to 5.

3.7 Two device experiments

The previous section reported isolated tests of the devices; in this section, we test the interaction of the devices. Specifically, we look at the interactions between a source and a relay, and a source with a sink or a relay. Because sinks do not transmit packets they cause no interactions with sources, hence we do not need to measure the effect of a sink on a source. This is not necessarily the case in general for low power radio devices.

3.8 Source and Relay

In this experiment a source is sending data to a relay device that retransmits whatever it hears. The experiment is carried out four times using different packets sizes (9, 39, 69, and 99 bytes). Tables 3.3 and 3.4 respectively show the energy consumption of source when sending packets to a relay and the energy consumption of a relay when receiving traffic from a source and relaying.

Packet size (bytes)	Idle (mW)	Rx (mW)	Tx (mW)
9	0.22	0.77	0.69
39	0.21	0.77	0.68
69	0.22	0.77	0.71
99	0.22	0.78	0.71

Table 3.3: *Source consumption in source to relay (mW)*

[4]Calculating the Standard Deviation over such a small number of iterations could be unreliable, but we give it here as an indication of what was observed.

Packet size (bytes)	Idle (mW)	Rx (mW)	Tx (mW)
9	0.23	0.79	0.71
39	0.22	0.79	0.71
69	0.22	0.78	0.74
99	0.23	0.78	0.74

Table 3.4: *Relay consumption in source to relay (mW)*

Comparing the cost of reception for the two device experiments with the cost of reception for the single device experiment, table 3.1, shows that the energy consumption of the source devices and relay devices are approximately the same whether there is a transmitting source or not. The energy consumption of a relay device is, on average, higher than that of a source device in both scenarios.

Transmission costs are approximately 90% of the cost of reception. This is interesting because this result is different to the assumptions of many people who expect transmission costs to be higher than reception costs. We briefly discuss the implications of this in section 3.12.

3.9 Source and Sink

One of the challenges faced whilst conducting this set of experiments was in measuring the effect of a transmission upon a receiving device. It was found that no identifiable variation in V_r in either the amplitude or frequency domains is observable to be used as a trigger for the oscilloscope. The oscilloscope's built-in Fast Fourier Transform routine failed to identify a frequency change. A possible reason for this is that the cost of decoding a "signal" is constant whether a signal is present or not.

As it was not feasible to detect when packets start being received and when they stopped being received it was decided to investigate the effect of distance between the source and the sink. Three different distances, 0.1m, 1.0m and 10.0m were tried with a source sending a 99 byte packet. The average cost of reception was calculated over a randomly sampled period, the results are show in table 3.5

Distance between devices (m)	Idle (mW)	Rx (mW)	Tx (mW)
Sink(0.1m)	NA	0.76	NA
Sink(1.0m)	NA	0.77	NA
Sink(10.0m)	NA	0.78	NA

Table 3.5: *Source and Relay power consumption (mW)*

The cost of reception does not seem to be dependent upon distance, which is as expected;

however, distance would affect the bit error rate at the destination, causing more packets to be dropped. Comparing table 3.5 with 3.1 the cost of reception is not influenced by whether there is a transmission or not.

3.10 Summary of results

The cost of reception was found to be 0.78 mW (SD 0.01), transmission was found to be 0.71 mW (SD 0.02) and the cost of being idle was found to be 0.22 mW (SD 0.01). Comparison with the single device experiments does not indicate any influence from the presence of additional devices on the energy consumption of the devices. This is not the case in general, but the firmware used here has no flow control in the MAC layer, or any higher layer. Introducing an 802.11-style CTS/RTS flow control scheme would introduce inter-device dependencies leading to costs related to the presence of other devices.

3.11 Limitations and sources of error

No thought was given to the effect of collisions between packets upon the energy costs; this is something that was outside the scope of this experiment and is complex to measure because it was not possible to determine the start of packet reception from the energy traces.

The known sources of error include the value of the resistor, which was rated with a 2% tolerance, and the identification of the key points within the voltage V_r trace. If the points are identified to an accuracy of ± 2 points on either side, this leads to a 0.002% inaccuracy for a data size of 8192 points. The unknown sources of error include the voltage drift of V_{cc} from recorded value at the start of the experiment, any device variation, and any errors associated with the oscilloscope. The effect of the unknown sources of error is thought to be minimal and unlikely to affect the results, whilst the effect of the known sources of error is thought to have a minimal affect on the results, given a maximum observed SD of 0.03.

3.12 Conclusions

Experiment	Feeney [21]		UCL
Mode	2Mbps	11 Mbps	19.2Kbps
Idle	843.72 mW	739.44 mW	0.22mW
Receive	966.96 mW	900.60 mW	0.78mW
Transmit	1327.20 mW	1346.16 mW	0.71mW

Table 3.6: *Comparison of 802.11 and a low power radio device energy consumption figures*

In this chapter we have addressed objective one by investigating a real low power radio system. We have gathered real energy consumption parameters for use in large-scale simulation and observed two key findings. Firstly, the use of 802.11 based parameters as a model for all low power networks is not realistic, table 3.6, showing that the energy consumption of the low power radio device measured here is three orders of magnitude lower than the 802.11 devices measured in the Feeney study [21].

Secondly, the cost of reception is higher than the cost of transmission. One possible explanation for this is that the cost of the signal processing to decode the radio signal is more complex than encoding of the signal for transmission. However, the important observation here is that the typical assumption that $P_{tx} > P_{rx} > P_{idle}$ does *not* hold for this low power radio system; rather $P_{rx} > P_{tx} > P_{idle}$.

Having addressed objective 1 of the hypothesis, we now have the measurements required to feed into objectives 2, 3 and ultimately objective 4 of the hypothesis, which are addressed in the following chapters.

Chapter 4

An introduction to the control algorithms

In this chapter, we will introduce the transmission power control algorithms used throughout the following chapters by introducing the background and theory to the algorithms; the metrics used to evaluate the algorithms; and the algorithms themselves. We then follow this with a short example. Finally we discuss the issue of the idle radio receivers in the experimentation.

In view of the fact that mobile systems rely for their operation on battery power, it is obvious that the longer that power can be made to last, the longer operation can continue. In ad hoc systems, the problem is significantly worse than for infrastructure systems; the failure of a device affects not only that device, but also all those who would route packets through it. Battery technology is advancing, but not at a rate comparable with Moore's law. Consequently, conservation of energy within batteries is a vitally important aspect of the practical operation of mobile and ad hoc systems.

This observation is not new; indeed, there are various recent attempts to take power consumption into account in a range of protocols at different levels (e.g. [24]). In this thesis we show that with even simpler algorithms than those presented in [24], hop-by-hop energy minimisation exhibits complex behaviour. As an example of such a simple system exhibiting complex behaviour we adopt a simple transmission power control algorithm that exploits overheard MAC-level information to reduce a device's energy consumption. We illustrate this with a number of experiments in this chapter and the following ones, in which we examine the effects of two variables: transmission power as means of affecting the number of hops in a path, and the path itself. We show in this chapter and later ones, that with a simple algorithm that some energy savings are feasible by taking advantage of observed information.

To achieve the minimum possible energy consumption on an end to end basis, a minimisation of the energy costs at every stage of the transmission is required. Different paths through the network will have a different energy cost associated with them. Equally different transmission schemes on a hop-by-hop basis have different energy costs associated with them.

4.1 Background

The energy contained within the battery of a mobile device is finite and is generally not particularly large with respect to typical rates of consumption for mobile devices. Consequently all approaches that can reduce the rate of energy consumption have a positive effect on the overall utility of that device. Ad hoc networks are particularly sensitive to the failure of a single battery in the centre of the network, which reduces the ability of the network to forward packets or could cause the network to partition.

One of the major consumers of energy in simple mobile devices is the radio. In line with our aim of examining complexity by taking a simple view, we explore a simple transmission power control algorithm operating at the MAC level that exploits information overheard by devices to reduce the overall rate of energy consumption whilst retaining appropriate throughput.

In reality, the situation will be more complex than this, with multiple interacting power-saving mechanisms: switching off the CPU or reducing its clock rate; causing processes to migrate to better powered nodes; spinning down disks; and so forth.

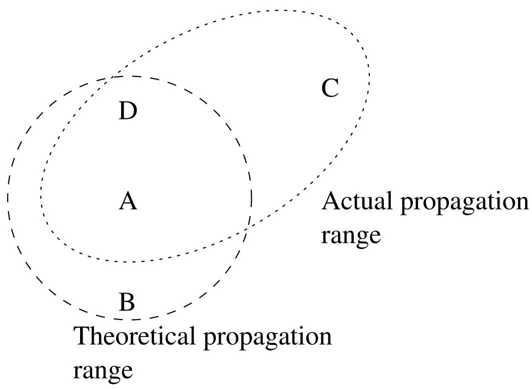


Figure 4.1: *Theoretical and actual radio propagation ranges*

Wireless networks have all the same problems as wired networks; for example, both schemes suffer from packet collisions when two or more devices are attempting to access the same physical medium at the same time. Medium Access Control, MAC, protocols and Ethernet switches, in fixed networks, go a long way towards mitigating the packet collision problem. The properties of radio signal propagation means that no guarantees can be given that the MAC layer signalling will be heard by all devices within the transmission range of the source, and not heard by devices outside its transmission range. Figure 4.1 illustrates this problem, device A is transmitting, the circular area is the theoretical signal propagation range and the oval is its actual propagation range[1]. If A's transmission propagated according to the theoretical prop-

[1]The actual propagation range shown here is only for illustration.

agation model, devices B and D would hear the transmission whilst device C would not. In contrast if A's transmission propagated according to the actual propagation model, devices C and D would hear the transmission and device B would not. The radio signal propagation range varies with the environmental conditions and hence the connectivity can vary with time.

This chapter makes a number of assumptions. Firstly, radio transmissions are broadcast transmissions, so all devices within the transmission range of the transmitting device hear the transmission. Secondly, the radio device is able to transmit at different transmission power levels [2], and thirdly, it is able to determine the strength of a received signal. Fourthly we assume that the antennas used are vertical and omni directional. Fifthly we assume that the path loss between any receiver and transmitter is symmetric. Finally we assume that any mobility the nodes are stationary or have minor relative variations when moving as a cohort.

To be able to transmit at different transmission powers on a per packet basis requires the radio device to be able to adjust its transmission power in the inter packet gap, or the transmission must be delayed until the adjustment has been made. The output power of the RFM TR1001 [74] is proportional to the current on the TXMOD pin, so this provides a mechanism to carry out dynamic power control. Likewise, the Intersil Prism (802.11) chipset is able to adjust its power from minimum to maximum in less than a microsecond, which is sufficiently fast for current 802.11 transmissions. The Atheros 802.11 chips specifically allow the dynamic setting of the transmission power on a per packet basis.

4.1.1 Wireless networking and power consumption

In considering the following examples, it is necessary to bear in mind that there are physical limits to the ability of wireless networks to support communication. These limits include:

- Channel availability
- Channel capacity
- Channel noise
- Signal propagation range
- Device energy resources

The driving factor for this thesis is to optimise the usage of the limited energy resources many mobile devices have. If we examine this from a networking perspective, the following items influence the energy consumption of each device.

[2] The following are examples of chipsets that support this: RFM TR1001, Intersil Prism (802.11) and Atheros 802.11.

- Amount of data sent
- Interference, signal/noise ratio (SNR)
- Signalling
- Idle time
- Transmission power
- Route over which data is sent
- Transport protocol used
- Retransmission strategy

Energy consumption is proportional to the amount of data sent plus a constant. In turn, the amount of data sent is largely determined by the applications and the protocols they use; however, it is also influenced by the underlying signalling schemes. An example of this is the trade-off between proactive and reactive routing protocols, where the energy consumed by reactive routing is proportional to load and that consumed by proactive routing is proportional to time [11].

The idle time of a device in the system is dependent up on three factors: the amount of data sent by other devices *to* this one; the amount of data sent *through* this device; and the amount of data *overheard* by this device but not intended for it. When idle, a device may wish to power down various components, including the radio. Unfortunately, given the unpredictability of traffic and the difficulty of global coordination, this is difficult to accomplish without introducing additional end-to-end latency.

Being unable to modify either the amount of data sent or the idle periods, the only remaining options for power reduction are :

- adjustment of the transmission power
- modification of the route the data takes through the network
- modification of the signalling used

It is feasible to measure the signal to noise ratio of the channel and to adjust the transmission power accordingly. In this chapter, we nominally focus on the first of these. However, in reality, transmission power cannot be considered in isolation from the MAC layer signalling or the routing. For example, if a particular region is known to have a low signal to noise ratio we could route packets around this area. This violates the strict layering model applied to many network architectures, and leaves us questioning, along with many others, whether the layering model is still the best approach in all situations.

4.1.2 Radio propagation models[3]

Wireless channels are extremely difficult to analyse, radio signal propagation is affected by factors including obstacles, environmental conditions, the frequency range in use, the antenna and the reflection from objects. The combination of all these effects makes the modelling of the channel between two devices complex and difficult. Thus, most models focus on calculating an average propagation model.

The simplest channel model is a standard power law equation, where $p \propto d^n$ where p is the transmission power, d is the signal propagation distance, and n is the path loss exponent constant based on the environmental conditions. The Log-distance Path Loss model [73] (page 138), equation 4.1, is the formal definition of $p \propto d^n$, it assumes that the large scale path loss is a function of distance to a path loss exponent, n.

$$PL(d) \propto (\frac{d}{d_0})^n \tag{4.1}$$

or

$$PL(dB) = PL(d_0) + 10nlog(\frac{d}{d_0}) \tag{4.2}$$

Where d_0 is a close-in reference distance. In the log-distance path loss model, n is dependent upon the environmental conditions, a variety of exponents for different conditions are shown in table 4.1.

Environment	*Path Loss Exponent, n*
Free space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
In building line-of-sight	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factories	2 to 3

Table 4.1: *Path Loss Exponents for Different Environment ([73], Table 4.2)*

A more complex algorithm is the Free Space Propagation model described in [73] page 107; however, it only considers the effect of scattering upon the signal propagation. This prop-

[3]A significant portion of this section on radio propagation models has been extracted from Wireless Communications Principles and Practice, 2nd edition, by Theodore S. Rappaport[73].

agation model is used when there is an unobstructed path between the transmitter and the receiver, for example in satellite systems or when using microwave links this is given by the Friis equation:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{4.3}$$

Where $P_r(d)$ is the signal strength at the receiver d metres away from the transmitter that transmitted at a signal strength P_t. The antenna gain is G_r and G_t for the receiver and transmitter respectively. The wavelength in metres is λ and system loss factor is $L(\geq 1)$, and is not related to the signal propagation. The gain of antenna, G is related to the antenna's effective aperture, A_e, and is given by :

$$G = \frac{4\pi A_e}{\lambda^2} \tag{4.4}$$

The signal attenuation between the transmitter and the receiver in dB is given by the path loss equation :

$$PL(dB) = 10log\frac{P_t}{P_r} = -10log[\frac{G_t G_r \lambda^2}{(4\pi)^2 d^2}] \tag{4.5}$$

The Friis Free Space Propagation equation is a only a good representation of P_r for values of d in the far field of the transmitting antenna; for further details please refer to [73] (page 107). The Free Space Propagation model does not take into account the radio signal reflecting off the ground and the effect of signal scattering when calculating the propagation distance of the signal. The Two-ray Ground model [73] (page 120) introduces this additional factor, by including the height in metres of the receiver, h_r, and transmitter, h_t, from the ground, is shown by :

$$P_r = P_t G_t G_r \frac{h_t^2 h_r^2}{d^4} \tag{4.6}$$

In the Two-ray Ground model the signal attenuation between the receiver and transmitter is given in dB by :

$$\overline{PL}(db) = 40logd - (10logG_t + 10logG_r + 2logh_t + 20log_r) \tag{4.7}$$

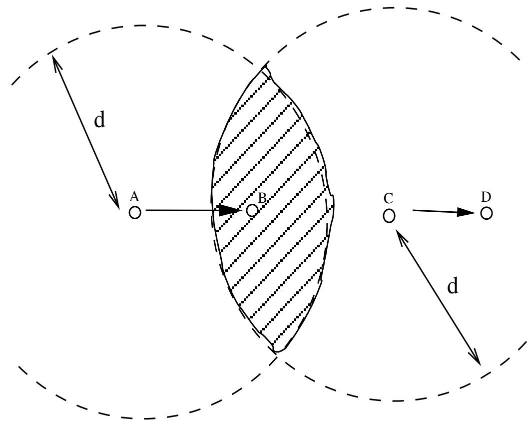
The Log-distance Path Loss model, equation 4.2, does not consider that the radio signal maybe different at different nearby locations due to interference from objects in the environment. Studies have shown that the value of the signal strength at any particular location is random and distributed Log-normally (normal in db) about a mean distance dependent value [17], [9]. These observations are encapsulated in the Log-normal Shadowing [73] (139) model, equation 4.8. In equation 4.8, X_θ, is a zero mean Guassian distributed random variable (in dB).

$$\overline{PL}(d)[dBm] = \overline{PL}(d) + X_\theta = \overline{PL}(d_0) + 10nlog(\frac{d}{d_0}) + X_\theta \tag{4.8}$$

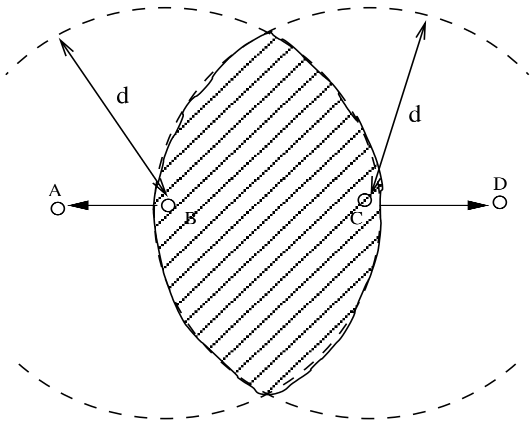
The simulator we use in this thesis is the NS2 simulator [30] and employs the Log-normal Shadowing and Two-ray Ground Propagation models, alternative transmission models including the Stanford University Interim (SUI) models [8] could have been used instead, however we chose to use the models already implemented in the simulator.

4.1.3 MAC layer signalling

Wireless networks suffer packet collisions when two or more devices are attempting to access the same physical medium at the same time in precisely the same way as do wired networks. Medium Access Control (MAC) protocols go a long way towards mitigating the packet collision problem. However, the properties of radio signal propagation means that no guarantees can be given that MAC layer signalling will be heard by all devices within the transmission range of the source, nor that it will not be heard by devices outside its transmission range. The lack of these guarantees leads to the hidden terminal, figure 4.2(a), and exposed terminal, figure 4.2(b), problems.



(a) hidden



(b) exposed

Figure 4.2: *Hidden/Exposed terminal problem*

The hidden terminal problem occurs when node A senses that the channel is idle and transmits to B; however, the channel at B is busy since node C is transmitting to node D. The exposed terminal problem occurs when node B wants to send a packet from node A but backs

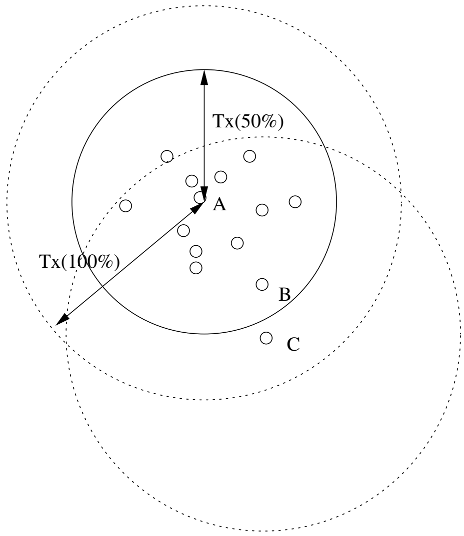


Figure 4.3: *Hidden terminal example*

off because it senses the transmission from node C to node D. However, the signal strength of the transmission from node C to node D is too weak to cause interference with the transmission from node B to node A.

We explore the issues associated with the hidden and exposed terminal problems and transmission power control through a couple of examples. In this first example, we consider the arrangement of devices in figure 4.3. We reduce A's transmission power so that it only just reaches its neighbours where B is the furthest current neighbour, thus saving energy. Reducing the transmission power reduces the propagation distance of a signal, the two quantities being approximately related by the formula $P \propto d^n$ (Log-distance Path Loss Model).

If C now arrives within the maximum transmission range of A but outside its reduced range and C listens to the channel, it will believe it to be free, since it is unable to hear A's communications. Consequently, when it transmits, its packets may well collide with those from A and require retransmission reducing the capacity of the system both directly and as a result of interference caused by this transmission. This is simply the hidden terminal problem which has been exacerbated by the reduction in transmission power.

The issue is complicated by the exposed terminal which transmits at full power, reducing the overall capacity of the network since the savings achieved by A's reduction are to some extent negated by the higher transmission level from C.

In terms of energy, a reduction in transmission power can actually lead to an increase in the overall energy consumed in order to transmit packets correctly from point to point. On the other hand, in cases where the transmissions do not collide, either because C is sufficiently far from A that this cannot happen, or because the transmissions are not simultaneous, then there is

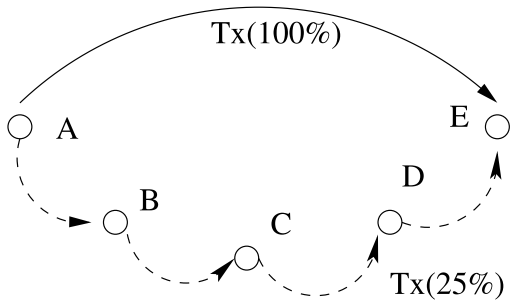


Figure 4.4: *Relay example*

both a gain in the overall transmission capacity and a reduction in energy consumption. These considerations become important in environments in which there are a large number of devices all within a small number of transmission hops of each other. It is particularly important if flows are being funnelled from a large area into a small area, inducing congestion as the flows converge[15].

4.1.4 Relaying

As we reduce the distance a signal propagates, we must increase the number of devices that relay the message, which leads to an increase in the latency of the signal and an increase in the amount of control traffic required. So, if we consider the two levels of power transmission for device A in figure 4.3, one at full power and the other a transmission at half power. Before device C arrives, the reduction device A achieves by transmitting at a reduced power is $(P_{tx_{100\%}} - P_{tx_{50\%}}) + P_{rx}$, if the system is otherwise idle[4]. This leads to the notion that multiple small hops are better, i.e. smaller transmission power per hop, smaller number of overheard receptions per hop, but more hops.

Consider figure 4.4, which has two paths between devices A and E. The first is a multi-hop path using at each stage the least expensive hop in terms of its transmission power, whilst the second is a direct path that is available if we transmit at a higher power level.

However, this is simplistic, and we will illustrate this with reference to figure 4.4. Direct transmission from device A to device E, costs $P_{tx100\%}$ and the reception costs are $4P_{rx}$ since devices B, C, D and E all hear device A's transmission and must decode it to determine whether it is for them. On the other hand, consider the case where we transmit from device A to device E using multiple hops via devices B, C and D. Assuming that $P = K.d^4$ then the power required to reach distance $\frac{d}{4}$, $P_{tx_{25\%}} = K.\frac{1}{4}^4.P_{tx_{100\%}} = \frac{K.P_{tx_{100\%}}}{256}$. With a moderate constant of proportionality, this looks very promising. However, each transmission by a device is overheard

[4] $P_{m\%}$ is the power required to transmit a distance $\frac{m}{100} * d$ where d is the maximum distance transmitted according to $P_{100\%} = K.d^n$. Thus, $P_{25\%} \propto \frac{d^n}{4^n}$.

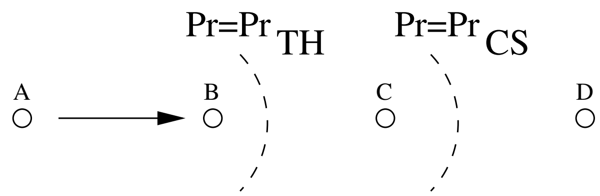


Figure 4.5: *Transmission ranges*

by its neighbours; for example devices B and D, hear device C's transmission, so the total reception cost is $7P_{rx}$.

If we extend this example such that each device's neighbour's neighbour can also hear a transmission, then reception costs increase to $12P_{rx}$, so direct transmission becomes more favourable.

It is worthwhile noting at this point that the costs of reception are not negligible in comparison to transmission costs. In order to receive a packet, one must have the receiver switched on for substantially longer than the time taken to transmit the packet, since one does not know when the packet will actually be sent. Secondly, in low power radio scenarios, the power amplification costs are reduced, but demodulation costs remain the same. In fact, for low power radio systems, power consumption can be dominated by reception costs [26].

So there is a trade off here, reducing the transmission power and adopt a multi-hop approach increases the summed costs of reception, which, by example, counters the hypothesis that if we reduce the transmission power so as only to be able to reach the next node en-route to the destination, we will necessarily save energy. On the other hand, if we transmit over the maximum distance whenever we can, we reduce the channel capacity but decrease the end to end transmission time.

In actuality, the above analysis is still an over-simplification, and the cost of reception is higher than described. This is because a device may receive a packet in three possible ways, as shown in figure 4.5. Firstly, the packet is received at a power level greater than the receiver threshold, Pr_{TH}, is correctly decoded, and consumes $Cr_k(n)$. Secondly, the packet is received at a power level less than Pr_{TH} but greater than the carrier sense threshold, Pr_{CS}. The packet's carrier signal is detected but the decoding of the packet is unsuccessful; however, it still consumes $Cr_k(n)$. Thirdly, no carrier signal is detected and the radio remains idle whilst consuming $Ci_k(n)$.

Channel availability is also important because it has a bearing on the ability of a device to accomplish its tasks. The Log-distance Path Loss model defines the physical area a transmission affects. Transmissions by other devices at the same time in this region could collide with one another and be corrupted. So the larger the transmission area the fewer simultaneous transmissions that can occur.

So, to summarise, even if we ignore MAC level effects, there is still a trade-off to be made; if one reduces the transmission power and adopts a multi-hop approach, one increases the summed cost of reception and increases the latency. If, on the other hand, one transmits to reach the destination directly wherever possible, one increases transmission costs and reduces the channel capacity. Finally, it is worth noting that the above analysis, although showing significant complexity, still does not take account of further complicating factors such as channel SNR and device motion, both of which significantly influence the system's behaviour.

4.2 Metrics

Before considering the algorithms, we introduce a new measure, against which they will be judged, cost per useful bit, C_{UB}, which is a measure of how much energy is used by a system as a whole in successfully transmitting a message between a source and a destination. It provides a better measurement than energy used per bit or the overall energy consumption of the system, since it directly measures the average amount of energy expended in transmitting successfully, taking account of signalling and retransmission overhead.

Throughput is also an important metric, indicating whether packets are able to traverse the network without loss, which is an indication of the reliability of the overall system. The volume of data able to pass through the network in a certain time period is a function of the end-to-end delay and the throughput of the system. The end-to-end delay is affected by the amount of relaying a packet experiences and the delays due to the signalling system.

Since the MAC protocol used is unchanged, the number of dropped packets is an indication of how well the power control mechanisms function. Comparing the number of dropped packets for the new control schemes against the default scheme, the effect of the proposed power control schemes can be ascertained.

The proposed power control schemes reduce the distance a packet will propagate. If CTS and RTS messages do not propagate far enough to reach the appropriate devices then the MAC scheme can breakdown. An outcome of this breakdown is an increase in the number of dropped packets due to collisions, indicating that the number of dropped packets due to collisions is an important metric.

Packets are queued in devices awaiting the completion of the ARP resolution protocol to determine the hardware address of the next device to which to forward the packet. If queue of packets in the devices exceeds a limit, packets are dropped. Packets that are dropped because the interface queue is too long indicates that the ARP resolution protocol is not being successful or that the channel is busy and the device is unable to get a free slot to transmit in. An increase in the number of packets being dropped by the power control schemes compared with the default scheme indicates the power control schemes have exceeded the bounds of the possible optimisation.

Other metrics, not being investigated here, include the energy variation of the devices across the entire system: in particular system death, first death, and network partitioning. First death is the time at which the first device in the system fails because it has run out of energy. System death occurs when there is only one device left. Network partitioning is when there is no longer a route between all devices in the network. The different node failure patterns show how the network load is spread across the network by the different schemes.

4.3 Algorithms

The algorithms we present here illustrate the problems described previously by exploiting the nature of broadcast communications such that each device learns about their own neighbours by observing their communications. If the communication is not destined for a given device that hears it, that device's resources have been used in receiving the packet needlessly. However, benefit could be obtained by taking advantage of this passively overheard information to improve the efficiency of future communications to that device.

We did not have a specific layer 2 implementation for the Philips radio system so we have adapted the basic 802.11 model. To maintain simplicity, we have decided to leave the 802.11 protocol in the simulator unchanged save for the addition of an 8-bit field, which is a linear quantisation of the transmission power used to send the packet. It is assumed that each device has the same transmission characteristics and is able to detect the received power level at which a signal is received. By using the two different sets of values for the 11 Mbps 802.11 model and the Philips model presented in table 3.6 we can see how the four algorithms perform for the two different radio models. In the remainder of this thesis in the simulations we refer to these as 802.11 and Philips respectively.

We present four alternative methods for determining the transmission power to be used for a particular transmission: two naïve approaches (per link maximum and an approximate lower bound per link minimum) that act as the upper and an approximate lower bounds and two algorithms (method A and method B) that solely use observed information to determine the transmission power. Methods A and B take advantage of devices overhearing the transmissions of their neighbours. For each overheard transmission from a neighbour we feed the observed information into the *guess* algorithm. The *guess* algorithm takes the overheard transmission's received signal strength, an 8-bit representation of the transmission power from the packet header and the receiver's reception signal strength threshold to calculate an appropriate value at which to transmit back to the source of the transmission; this value is stored in a table. Methods A and B implement the *guess* function differently, as described in appendix A. When a packet is to be transmitted, the required transmission power to reach the destination is looked up in the table. If the packet's next hop is not listed in the table or the packet is for the broadcast address, then the maximum transmission power is used. If a transmission to a destination is unsuccessful, the stored entry for the destination is increased by a delta; methods A and B use different deltas as described in appendix A. In all schemes, broadcast packets are transmitted at full power. The approximate per link lower bound algorithm is coded to calculate the minimum transmission power to just reach the destination using information from the simulator not normally available to the device. Each algorithm has three functions.

- *received_packet* calculates the power to reply to a device given the power a packet, *p*, has been received at. It stores the calculated power and uses it for future communications to the source of *p*.
- *send_packet* looks up the previously stored transmission value for a particular host and sets the transmission power level for the packet to be sent. The default is 255, full power, if the destination is unknown.
- *drop_packet* adjusts the stored transmission power for a destination, if the packet for the destination or using it as a next hop is dropped.

A key thing to note is that it is not realistic for algorithm A or B to find the minimum transmission power first time every time, but rather intended that reductions will occur in an iterative manner. The iterative process to select the transmission power for a specific neighbour device selects from all the packets overheard from the neighbour device and uses these to estimate the transmission power. The iterative process is aided by the nature of the MAC protocol and the high number of transmissions required to transfer data between two nodes. For example, a single application level packet transmission generates the MAC level transmission shown in figure 4.6. The numbers next to the arrows illustrate the quantised transmission power at transmission, $[n]$, and reception, $\{n\}$. The advantages of using overheard information to supplement direct communications are detailed in section 5.3.

4.3.1 MAC layer modifications

A number of approaches could have been taken to encoding the 8-bit representation of the transmission power into a packet or packets. An early requirement was that the optimisation process occurs across all the packets sent and that the observed data could be gathered from any packet. The first of these requirements requires us to modify the MAC layer since this is the lowest common denominator for all transmissions. The second requirement limits the use of any high level protocol that is connection orientated, such as an oracle-like approach.

If we accept the assumptions that only an 8-bit value is required to describe the transmission power and that the data has to be carried in all packets, then we need to consider strategies for modifying the MAC layer or replacing it. We considered a wholesale replacement of the MAC layer too invasive a change and, for the reasons of simplicity, opted for a minor change to the existing MAC. This leaves three options.

- Extending the current header by 8-bits.
- Squeezing the 8-bits into any reserved bits in the current header.
- Appending the 8-bits to the end of the packet.

We have adopted the first of these three options as it is by far the most straightforward to implement and the least likely for the bits to be misinterpreted. The 802.11e[39] standard that added enhancements for Quality of Service to 802.11 has taken the same approach in 2005. There are insufficient bits (6 bits) available in the default header to adopt the second approach, so this was not pursued. Further work would be required to see whether 6 bits would be sufficient or not. Appending 8-bits to the end of the data packet was also ruled out as it breaks the traditional packet construction methodology and could be a source of misinterpretation.

4.3.2 Minimum Transmission Power

There is a theoretical minimum transmission power on a per link basis for a successful communication between any two nodes; no energy saving technique can achieve better results than this. We need to clarify what we mean by the minimum transmission power. The minimum transmission power could mean the transmission which incurs the least global energy consumption for a transmission. Determining a global minimum is feasible but requires computational resources far exceeding those available. Alternatively the minimum transmission power could be a per link minimum, where only sufficient energy is used to transmit between the source and the destination. It is this latter, link minimum that we are to use as the basis of the lower bound algorithm.

The strength of the signal at the destination influences the probability that the signal is not corrupted by noise or an alternative transmissions. If the theoretical minimum transmission power is too low the signal is unreliable, requiring the transmission power to be increased to increase the reliability. We adopt an approximate minimum per link transmission power to be the lower bound per link for reliable transmission power. To evaluate any energy saving technique, the new technique needs to be measured the against the approximate per link minimum transmission power. Equally the per link maximum transmission power technique should form an upper bound on the transmission costs. Other metrics are used to evaluate the success of an energy saving technique include the number of dropped packets and the number of retried packets.

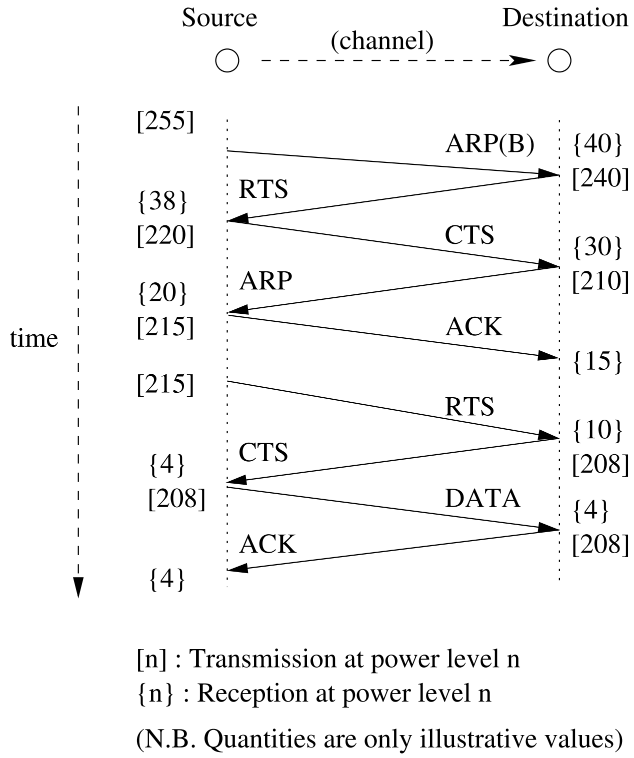


Figure 4.6: *An 802.11 MAC transmission of a single packet*

Here we return to figure 4.6 which illustrates the transmission of a single data packet at the MAC layer. The MAC packets have been annotated with the power control information in figure 4.6. We use this figure to explain the generic technique used to tune the transmission power. As described previously, all transmission power levels are represented in the MAC packet by an 8bit[5] field which is a linear quantisation[6] of the transmission power used. The transmission of the data packet begins with the source device sending an ARP broadcast, ARP(B), to obtain the address of the destination. The ARP is a broadcast and it is sent at the maximum transmission value, 255. Signal attenuation due to path loss means the packet is received at the destination at a level of 40. The destination, knowing the strength of the source's transmission, calculates it

[5]The 802.11 specification does not have 8 free or unallocated bits, the specification allows for new types to be defined which would be a feasible solution.

[6]A linear quantisation is used here, however a exponential quantisation might better reflect reality.

needs to reply at a lower transmission power and transmits the RTS message at a level of 240. The source receives this RTS at a level of 38 and calculates any reply to the destination should be at a level of 220. This process is repeated until a predefined minimum value of reception is reached; in this example a level of 4 has been chosen. At this point, the two devices continue to use the transmission value of 208 to communicate with each other until they detect a change or a packet is dropped. The minimum level and the amount by which the transmission level is adjusted for each iteration is based up on the particular algorithm used. If a third device were to overhear the exchange between these two devices, it would base future transmissions to either of these devices upon the information it had observed, and the application of its local algorithm to the stated transmission powers and observed received powers.

A special case is the transmission of packets that are sent to a broadcast address, for example the transmission labelled ARP(B) in figure 4.6. Broadcast packets are not used to calculate the minimum transmission power and are only transmitted at the maximum transmission power. However, feasible schemes exist to reduce the transmission power of broadcast packets. One such scheme is to transmit at the maximum transmission value required to reach all a node's neighbours. An alternative scheme would be to maintain a list of the ARP packets overheard and store the MAC/IP address mapping, hence reducing the need to send an ARP broadcast.

4.4 Simple two node example

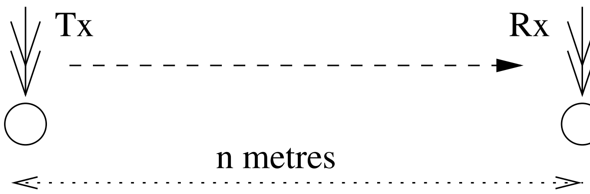
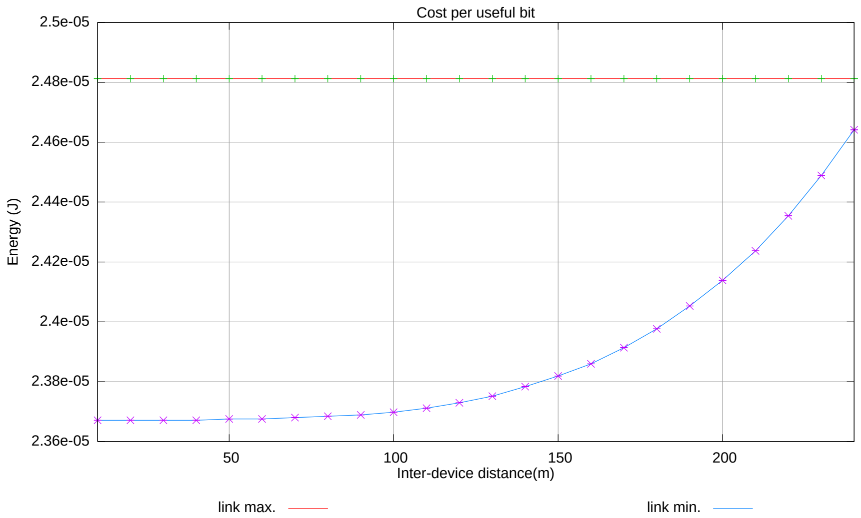


Figure 4.7: *Two nodes*

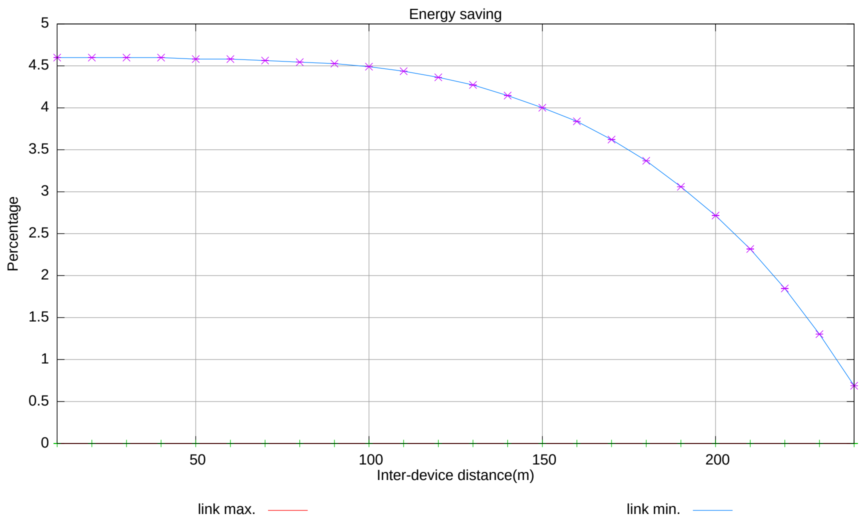
Two nodes are arranged as in Figure 4.7 with the distance between the nodes, n meters, being increased incrementally from 10m to 240m, in 10m increments. Figure 4.8(a) is an approximately ideal solution to the per link minimum transmission power problem. It is only approximate because the method used to determine the minimum uses an iterative approach that induces an error of 0.4%[7]. Figure 4.8(a) shows the energy consumption per useful bit, C_{UB}, using a transmission of 10 packets, from one node to another using the lower bound and maximum per link power algorithms.

The maximum and approximate lower bound per link power curves shown in figure 4.8(b), form an upper and lower bound of the energy consumption for reliable transmissions. It is

[7]The quantisation of the space into 255ths leads to this error $1/255 = 0.4\%$.



(a) C_{UB} (J)



(b) Energy saving (%)

Figure 4.8: *Ideal C_{UB}/Energy saving vs. distance curve*

important to note that it is possible to exceed the upper bound and fall below the lower bound; this is discussed fully in section 4.4.1 but we illustrate this briefly here. It is possible for an algorithm to exceed the upper energy bound by transmitting more packets than were used in the calculation of the maximum. More packets are sent typically as a result of the retransmission of packets or taking a route through the network that requires more hops. If fewer packets are sent, or packets are sent at a transmission power level that is not guaranteed to result in delivery, then it is possible to fall below the lower bound. The complexity of this scenario can be increased if reducing the transmission power causes the hidden terminal problem to occur then it is possible to transmit fewer packets which reduces the overall energy consumption costs.

Figure 4.8(b) shows the percentage energy saving achieved by the approximate per link lower bound power algorithm when compared with the per link maximum power algorithm. The energy saving is approximately 4.7% with the nodes separated by 10m, and drops to approximately 0.7% at 240m. The energy saving follows a power decay curve which is approximately modelled[8] by $1.42^{-11} * x^4 + 0.016$.

The energy consumption of the overall transmission has three components; the cost of transmission, the cost of reception and the cost of being idle. We ignore the idle cost and focus on the cost of reception, figure 4.9(a) and the cost of transmission, figure 4.9(b). Figure 4.9(a) shows that the cost of reception is constant irrespective of the distance between the nodes or the power control algorithm used to vary the transmission power. This is the expected result given the observations in chapter 3. If we compare the cost of reception with the cost of transmission, we find that $P_t > P_r$ for the per link maximum power algorithm using the 802.11 parameters. However, the cost of transmission is proportional to the distance between devices. In figure 4.9(a), the cost of the maximum power transmission is constant; however the cost of the approximate per link lower bound algorithm varies with the inter device distance according to a power law.

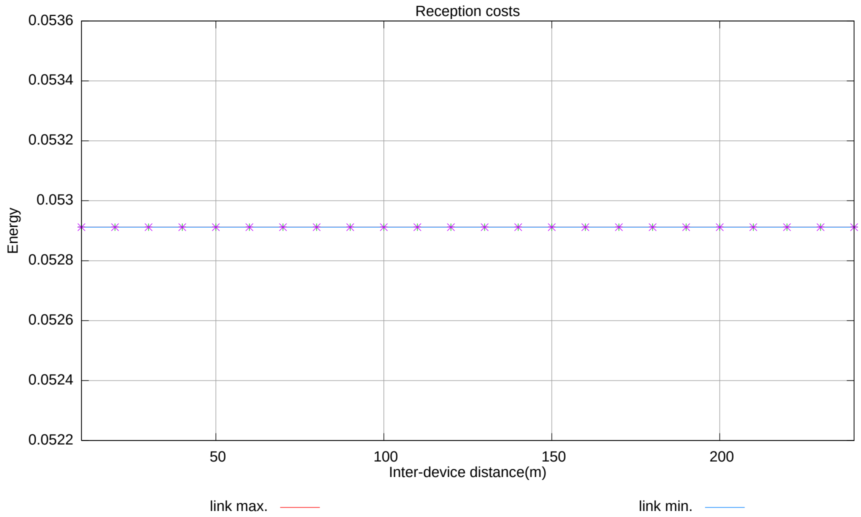
4.4.1 Discussion of the per link lower bound transmission power algorithm

Here we revisit the approximate per link lower bound transmission algorithm using the simple scenario of a single transmission between two devices. The algorithm exploits the simulator's knowledge of where all the other devices are located. The 8 bit field in the MAC header limits the number of possible power levels to 256 different values, which limits the number of possible values to try. Upon receiving a packet, the receiving device uses a function, Pr(), to calculate the signal strength of the received signal after it has propagated, using the Two-ray ground model, from the source. If the signal strength is above a certain level, it is said to have been correctly received. The closest approximate minimisation is achieved by the sender iteratively increasing the transmission power and calling[9] Pr() as if it were the receiver until the packet would be correctly received at its destination; this value is then used as the power at which to transmit.

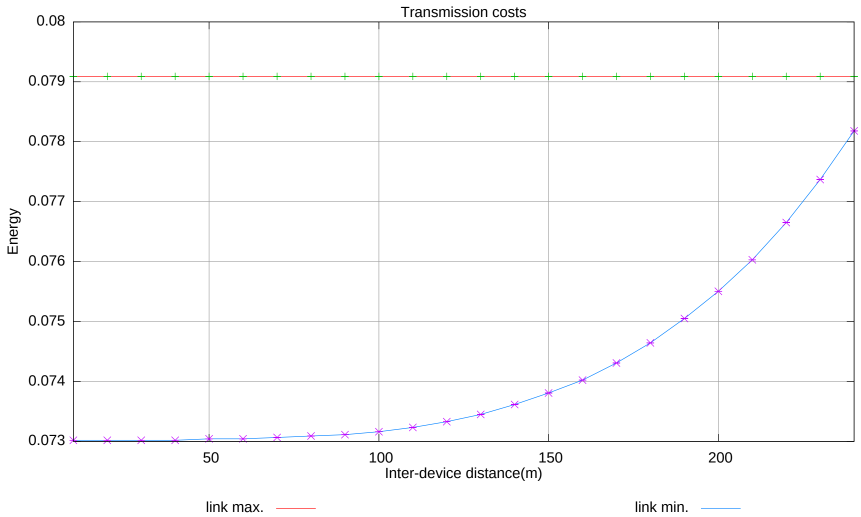
A consequence of reducing the transmission power too much is a degradation in the CTS/RTS mechanism, leading to an increase in the number of collisions and an increase in the number of packets dropped because the channel is "apparently" busy. When a device sends a CTS message it is expecting a RTS message in reply, if no RTS message is received, it is assumed that no device is within range, that the channel is busy, or that the RTS packet was corrupted. Alternatively, the CTS packet sent may not reach the device that would originate

[8]This model has been found by using a least squares fit.

[9]Pr() is the routine that calculates the energy level at which a signal is received at its destination.



(a) Reception



(b) Transmission

Figure 4.9: *Ideal energy consumption of transmission and reception*

the RTS packet. Reducing the transmission power of either the CTS or the RTS packet can result in packets not reaching their intended destinations and/or increase the likelihood of packet corruption.

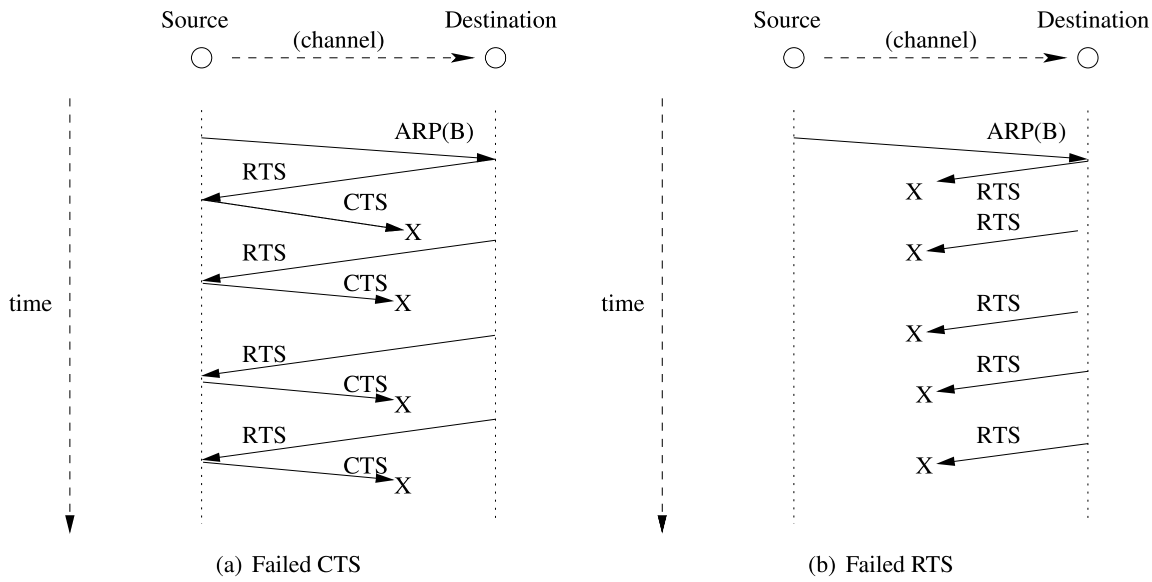


Figure 4.10: *Failed RTS/CTS transmission*

If either the RTS packet fails to react the destination, figure 4.10(b), or after the RTS packet was successfully received, and the resulting CTS fails to reach the node that sent the original RTS as shown in figure 4.10(a). Either failure scenario causes the node originating the RTS packet to retransmit the RTS after a timeout period, which introduces jitter and delay. A number of failed attempts to establish the RTS/CTS handshake leads the sending device to drop the packet it was trying to send due to the channel being busy.

4.5 Idle receiver issues

Throughout this thesis we make the simplification that we set the cost of the radio being idle[10] to zero, such that we are able to concentrate on the per byte costs of transmission and reception. We acknowledge that this will influence the results by hiding the idle receiver costs. The counter point to this and the position we have taken, is that where you have an experiment or set of experiments that have a packet based duration rather than a time based duration it is very difficult to draw any valid comparisons when the time based duration of one experiment is different to another. This is because the when the durations are different the idle costs for each radio are different and can significantly distort and influence the experiment. There are a number of reasons why two otherwise identical transmission scenarios will result in different traffic patterns. In the context of this thesis, a change in the transmission power for a specific packet can caused the MAC layer to behave differently which in turn influences the operation of the application layer distorting the comparison of the two otherwise identical schemes. This

[10]The radio is said to be idle when it is listening to the channel and not receiving or transmissing bytes.

is particularly true of schemes such as TCP.

When making this assumption we need to be particularly aware of what this means. It means that for all time *t* the figures are reduced by $\sum_0^t P_i$. Since we mainly use energy as a comparison between identical traffic scenarios which have different traffic patterns this cost is cancelled out for all packets whilst both radios are under taking the same tasks. When it differs the difference is either $P_t - P_i$ extra or $P_r - P_i$ extra. For example if we consider a scenario where the transmission power is reduced such that a device no longer hears the transmission then the difference is $P_r - P_i$ when comparing with the case where it would have heard it. Using the figures from table 5.1, there is a saving of 89% for the 802.11 model and 51% for the Philips model for this short instant of time.

4.6 Conclusions

In this section we have shown the underpinning arguments behind this thesis by presenting two simple algorithms that aim to reduce the overall energy costs by tuning the transmission power of each node to its neighbours. We have shown that modifying the transmission power can enhance and reduce the hidden and exposed terminal problems and how these manifest themselves as failures in the MAC protocol.

Chapter 5

Relayed and direct packet transmission

In this section, we aim to show that a simple algorithm as described in section 4.3 can achieve energy savings in this complex environment. We initially examine the algorithm's effects upon a single packet traversing a line of five devices using both direct and relayed packet transmissions. We then increase the number of packets being transmitted and take an in-depth look at how the energy costs break down between the reception and transmission phases. In section 5.3, we stated that savings were achievable by taking advantage of overheard transmissions, and hence we examine these next. In each of these experiments, we use both the 802.11 and Philips radio parameters from table 5.1 in comparing the algorithms.

The two radio systems are designed to operate over fundamentally different ranges so we use ranges of distances from 1m to 45m for the Philip's prototype system and 1m to 250m for the 802.11 system. In scenarios where we are looking at both single and multi-hop transmission at the same time we will scale the distances accordingly.

5.1 Simulation models and parameters

To make ad hoc networks interesting, a large number of mobile nodes are needed. But for practical and economic reasons it is infeasible actually to perform large scale experiments, so

Parameter	*802.11*	*Philips*
Frequency (Hz)	$914 * 10^6$	$868.55 * 10^6$
Pt (W)	1.35	$7.1 * 10^{-4}$
Pr (W)	0.90	$7.8 * 10^{-4}$
Pi (W)	0.74	$2.2 * 10^{-4}$
CPThresh (W)	10.0	6.0
CSThresh (W)	$1.559 * 10^{-11}$	$7.943 * 10^{-11}$
RXThresh (W)	$3.652 * 10^{-10}$	$3.162 * 10^{-10}$
Bandwidth (bps)	$11 * 10^6$	$19.2 * 10^3$

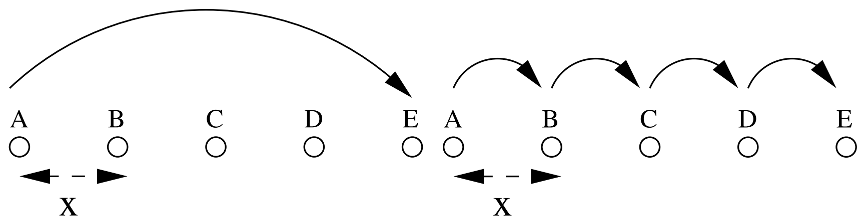
Table 5.1: *A comparison of the parameters of 802.11 and Philips radio parameters.*

simulation becomes the method of experimentation. Simulation does not provide a perfect representation of reality but rather an approximation that is limited by the information fed into it and the model used. Thus, real small-scale tests both provide an insight into some of the problems in developing and deploying a working system, and provide a method to validate the simulation. The Monarch Project [29] used such a technique in [56].

We modified NS (version 2.27) to include power control functions, static routing, and the ability to extract information from all traffic overheard by a device. The CMU extensions to NS were used to provide radio propagation models, ad hoc routing protocols, energy models and mobility models. For the radio model, we use the two-ray ground model and either the 802.11 transmission parameters with the energy consumption figures taken from [21] or we use the power measurement figures (P_t, P_r, P_i) measured by us using a low-power prototype radio developed by Philips. The non-measured values were taken from the device's component's specification sheets. In these experiments we have ignored the idle energy cost of the devices as we are primarily interested in extracting the information about the energy costs of transmission and reception. The actual power consumption figures used are shown in table 5.1.

5.2 Relayed vs. direct transmission

In section 4.1.4 we discuss the energy usage implications of using multiple short hops verses fewer long hops to relay traffic. Here we examine these implications using the four power control algorithms described in section 4.3.



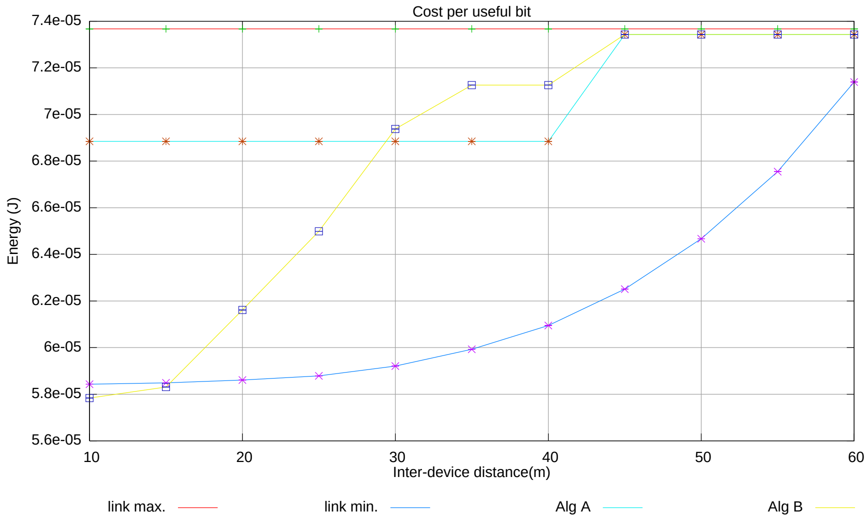
(a) direct transmission (b) relayed transmission

Figure 5.1: *Transmission*

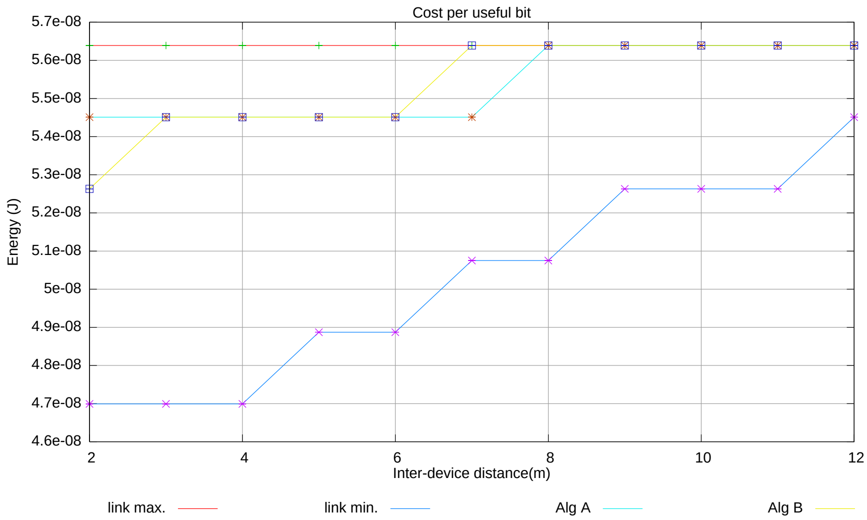
The straight line arrangement of devices shown in figure 5.1 is the simplest arrangement of devices against which we can evaluate the power control algorithms. This simplistic model is not entirely infeasible in real scenarios since it maps well to road or corridor-like environments.

5.2.1 Single packet - Direct

In this experiment we continue the theme of simplicity and send a single data packet directly from device A to device E, figure 5.1(a). Sending a single data packet simplifies the analysis of the experiment and removes the probability that two data packets will collide. Whilst it is feasible that two MAC layer packets may collide, the nature of the MAC protocol makes this improbable in this scenario, and data packets cannot collide. To obtain the characterisation of the energy consumption across a range of distances the inter-device distance x is increased from 5 to 60m in 5m increments for the 802.11 radio model and from 1m to 12m in increments of 1m for the Philips radio model. The total distance between A and E is $4x$m. At each increment, the experiment was replicated 5 times using different random seeds.



(a) 802.11

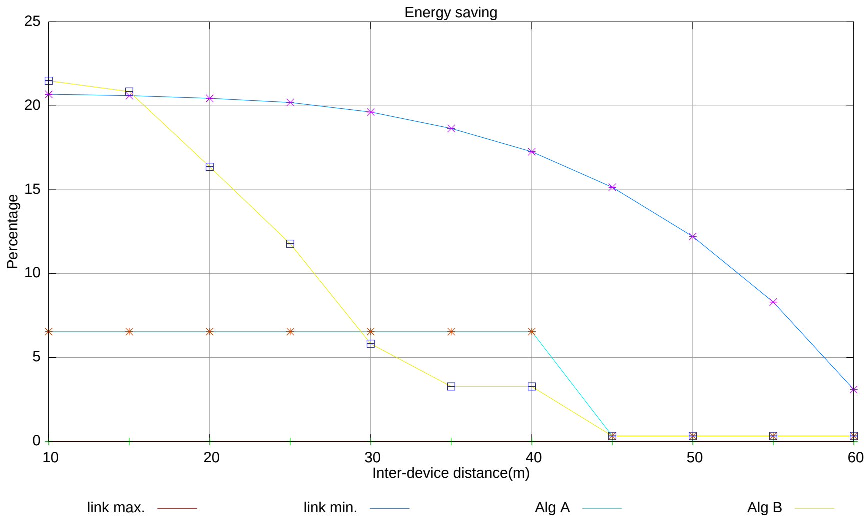


(b) Philips

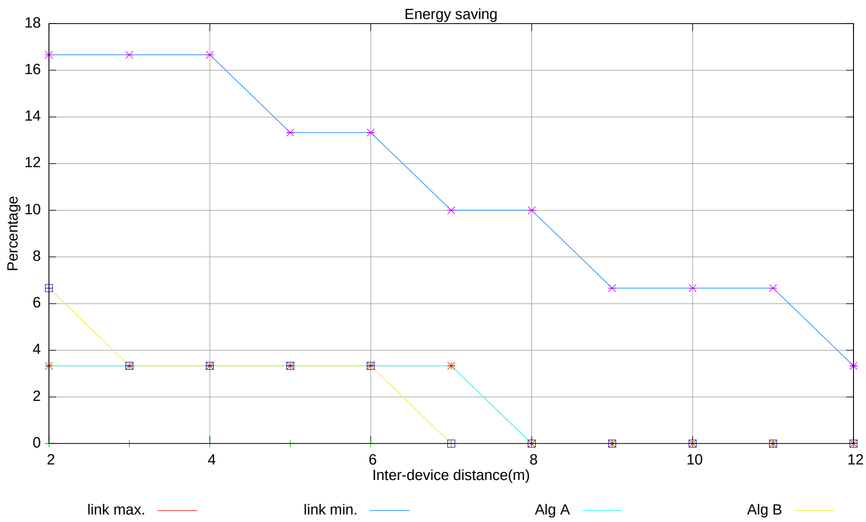
Figure 5.2: *C_{UB} of straight line of 5 nodes, 1 packet - direct*

Figures 5.2(a) and 5.2(b) show the C_{UB} for a single packet transmitted directly from device A to device E, figure 5.1(a), using the 802.11 and Philips radio parameters respectively. For the 802.11 radio model, the approximate lower bound per link algorithm forms a smoothly curving lower bound on the energy required to transmit a data packet between device A and device

E. It approximately follows a fourth order power curve. The smoothness of the lower bound algorithm's power curve and the flat profile of the maximum per link algorithm are indicative of the algorithms functioning correctly and that no packets are being dropped. Packet events such as collisions, retransmission or dropped packets would affect the C_{UB} versus distance graph, figure 5.2(a), by introducing peaks or troughs. The Philips radio model's lower bound curve shows incremental steps, since the resolution of each distance increment is large compared to the total distance. Manual analysis of the trace files for both radio models also showed them to be functioning correctly.



(a) 802.11



(b) Philips

Figure 5.3: *Energy saving* $(\%)$ *of straight line of 5 nodes, 1 packet - direct*

Using the 802.11 radio model, the approximate per link lower bound algorithm achieves a peak energy saving, figure 5.3(a), of 21% (it decreases to just under 4%), when compared against the maximum per link power algorithm (see figure 5.2(a)).

Algorithm A in contrast achieves only a 7% saving for inter-device distances of up to 40m, at which point it reverts to performing as well as the maximum per link algorithm. Algorithm A uses a single stage optimisation algorithm. The profile of the single stage algorithm is observable in figure 5.2(a) as the single step change in the C_{UB} versus distance graph at 40m.

Algorithm B uses four thresholds for the received power level to determine the change in the resultant TX power at which to reply. Crudely, the higher the threshold the more aggressive the cut in TX power. Algorithm B acts more aggressively than algorithm A and even achieves a lower energy usage than the approximate lower bound per link algorithm. The approximate per link lower bound algorithm sets a lower bound on its minimum transmission power to minimise packet loss, Algorithm B is not subject to this, see appendix A. The characterisation of Algorithm B's energy profile can be seen in figure 5.2(a).

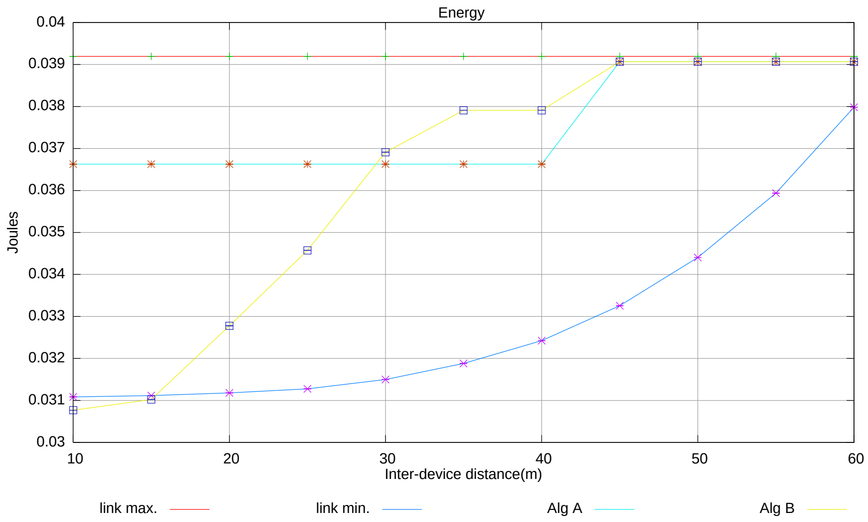
The C_{UB} is approximately three orders of magnitude less for the Philips radio model, figure 5.2(b), than for the 802.11 radio model, figure 5.2(a). Since the distance between devices is small, there is no distinctly noticeable power law effect for the C_{UB}. The cause of the lack of observed power law behaviour is likely to be due to the receiver sensitivity, the signal to noise ratio and the radio propagation model treating near distances differently to far away distances. These differences are reflected in the percentage energy savings achieved by the Philips model, figure 5.3(b).

5.2.2 Relaying

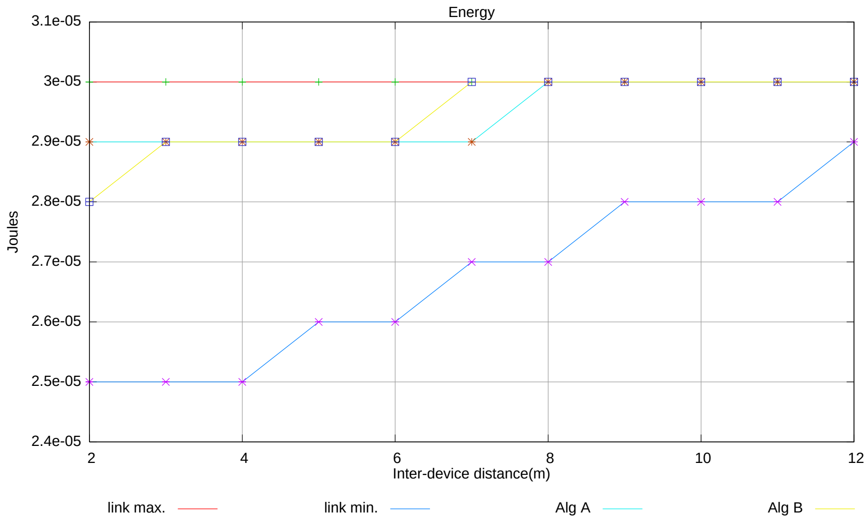
Previously, we examined the characteristics of direct transmission between the two end devices in figure 5.1(a), here we extend this to characterise the effect of relaying one data packet from device A to device E via devices B, C, and D, as shown in figure 5.1(b). The characterisation of the energy consumption of the different power control schemes in this multi-hop environment is shown in figure 5.5(a) for the 802.11 model and figure 5.5(b) for the Philips model.

Ignoring interference effects, comparing figure 5.2(a) and figure 5.5(a) for the 802.11 model, it can be seen that the constant energy consumption of the maximum per link power algorithm and the smooth increase with distance of the approximate per link lower bound algorithm are not present throughout the whole range of distances in figure 5.5(a). Since both the approximate lower bound and maximum per link algorithms exhibit the same behaviour, the cause of the steps in the graphs is not algorithm related but is instead caused by devices not overhearing transmissions from devices two or more hops away. As the distance increases, the number of overheard packets decreases, decreasing the overall energy consumption costs.

The relaying of a packet from one device to another increases the complexity of the en-



(a) 802.11

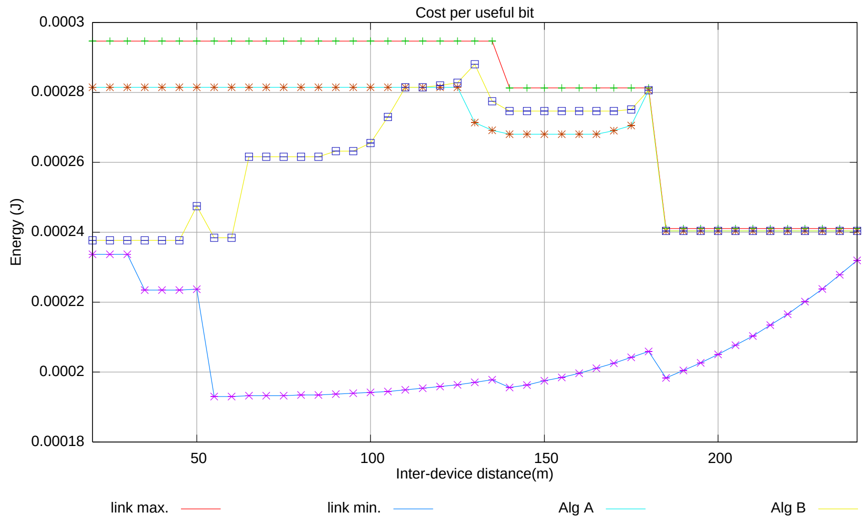


(b) Philips

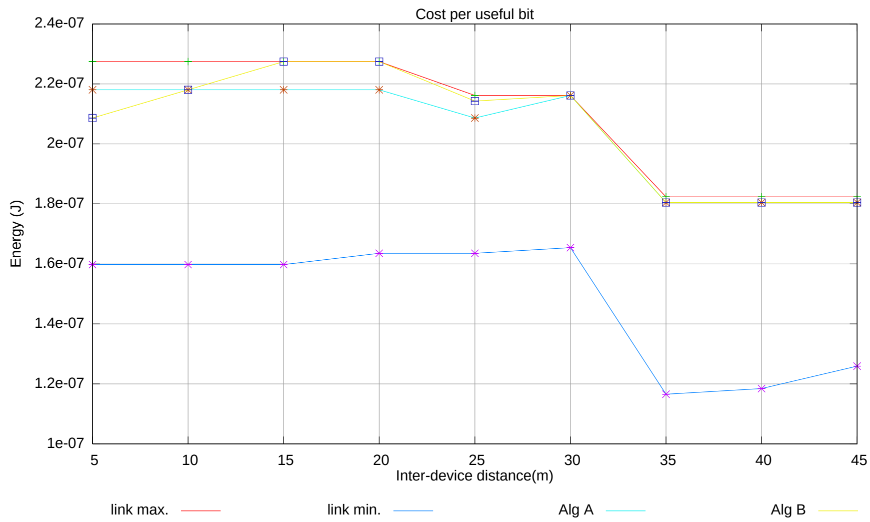
Figure 5.4: *Energy usage* (J) *of straight line of 5 nodes, 1 packet - direct*

ergy profile and this can be seen in the graph of the energy usage, figure 5.5(a). The cost of transmission, figure 5.6(a), increases with distance according to a fourth order power curve for the approximate per link lower bound algorithm. In contrast, the cost of reception for the approximate per link lower bound algorithm decreases with distance as each transmission is received by fewer devices, figure 5.7(a). This can be seen directly in figure 5.5(a), where the four step changes correspond to an incremental reduction in the number of devices not hearing each transmission as the inter-device distance increases. These step changes when combined with the energy consumption due to transmission characterise the energy usage, figure 5.5(a).

Using the Philips model in the multi-hop scenario, the drop in energy consumption at distances greater than or equal to 35m is due to the approximate per link lower bound algorithm



(a) 802.11

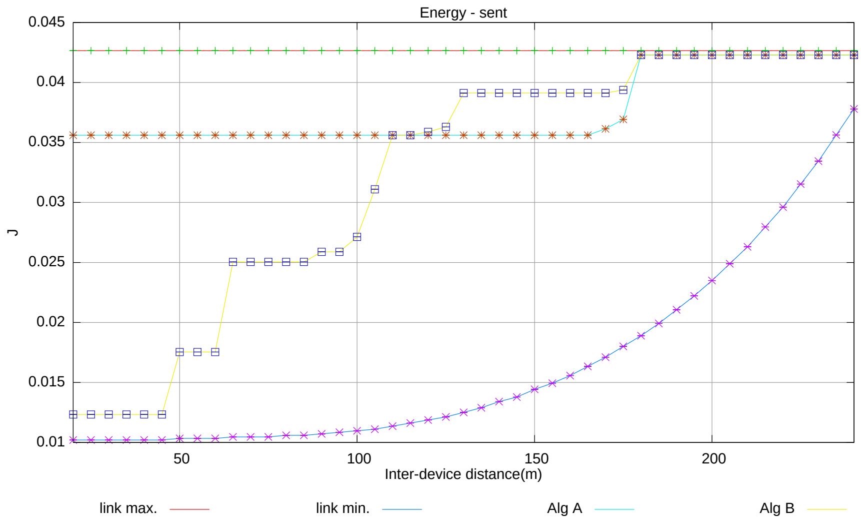


(b) Philips

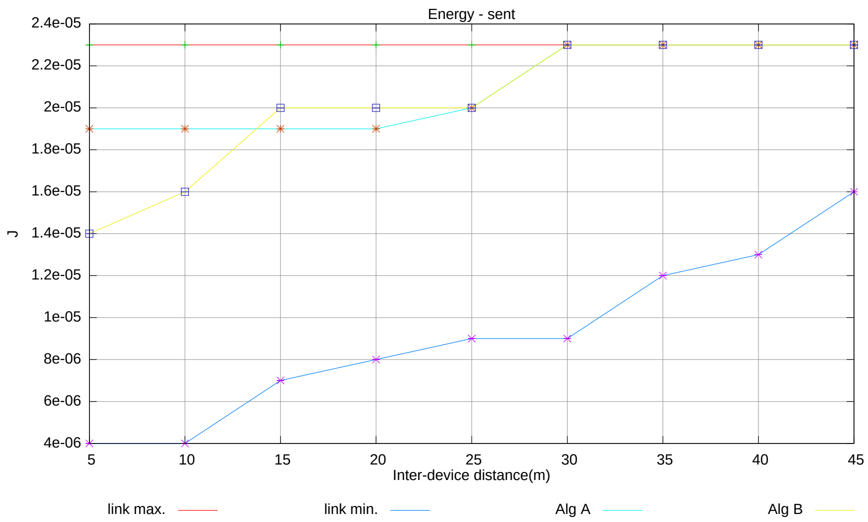
Figure 5.5: *Energy usage J of straight line of 5 devices, 1 packet - relayed*

underestimating the transmission power level required and the transmission failing. The same stepping that can be seen in figure 5.5(b) was observed in figure 5.5(a) can also be seen, and is caused by a reduction in the number of devices that can hear a transmission as the distance between devices is increased.

The multi-hop transmissions maintain a saving of approximately 16% for an inter-device distance of between 10 and 60m, whereas, over the same distance, the single hop saving drops from 16% to 3%. The total cost of transmission of multi-hop transmissions are approximately four times the cost of direct transmission, even though cost of each transmission is less, since the cost of reception is still the same and occurs three times more in the multi-hop scenario. We examine this further in section 5.2.3.



(a) 802.11

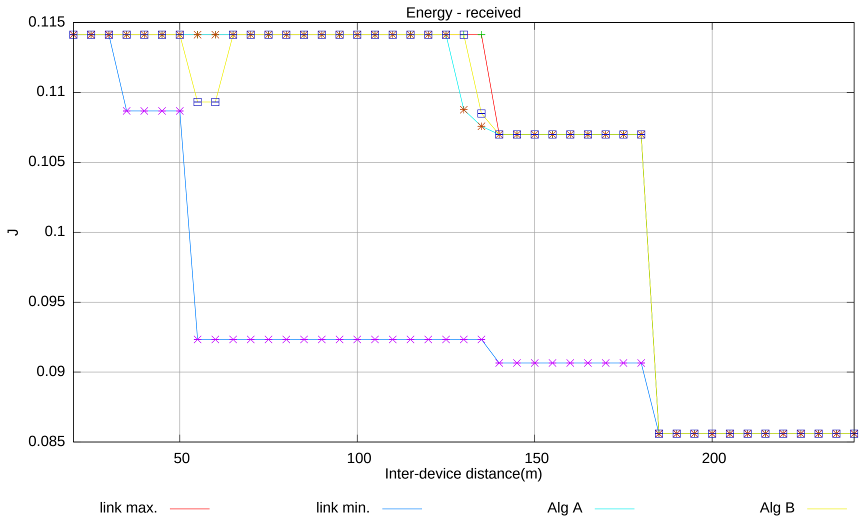


(b) Philips

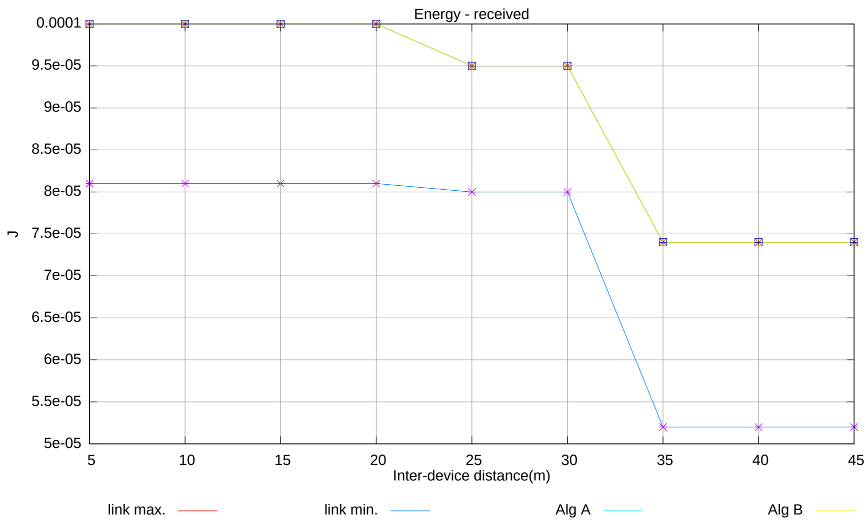
Figure 5.6: *E_{tx} of straight line of 5 devices, 1 packet - relayed*

5.2.3 Cost breakdown

In section 4.1.1, it was stated that the overall cost of reception dominates the total cost of communication. In the first part of this experiment, we explore this by sending a single data packet (and its related MAC control packets) from device A to device E, using relayed and direct transmission. We have chosen two distinct distances between the devices, 30m and 60m, with the experiments at each distance repeated 5 times with different random number seeds. In the left hand side of table 5.2, the standard relationship of the carrier sense threshold being less than the correct reception threshold, i.e. $Pr_{CS} < Pr_{TH}$, holds. Earlier in section 4.1.4, we discussed the different ways in which a device could receive a packet. The second part



(a) 802.11

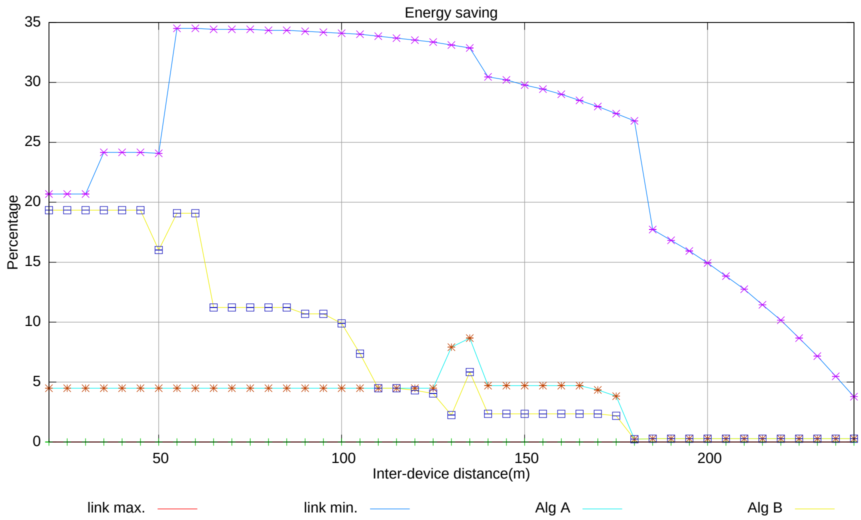


(b) Philips

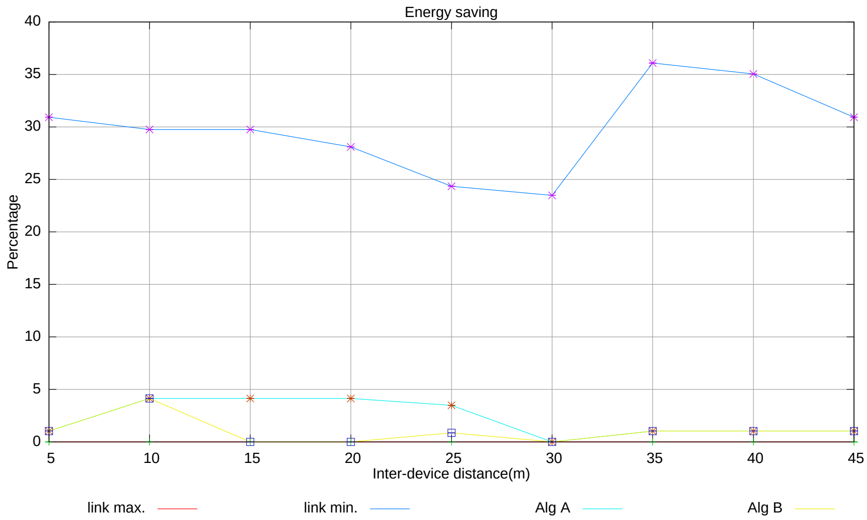
Figure 5.7: *E_{rx} of straight line of 5 devices, 1 packet - relayed*

of this experiment investigates the effect setting the carrier sense threshold as being less than the correct reception threshold. In this region, the device consumes energy as if it is correctly receiving a packet but the packet is corrupted and hence we set $Pr_{CS} = Pr_{TH}$.

Table 5.2 shows the breakdown of transmission and reception costs for the transmission at different inter-device distances. As the number of devices along the transmission path increases, the overall cost is dominated by the summed reception costs (e.g. 114.1 J) rather than the summed transmission costs (e.g. 42.7 J), as can be seen in the table. The table illustrates the dominance of the cost of reception associated with the number of devices that can overhear the transmission. From table 5.2, it can be deduced that each transmission is overheard by all devices, whether the transmission power has been reduced to a per link lower bound or whether



(a) 802.11



(b) Philips

Figure 5.8: *Energy saving % of straight line of 5 devices, 1 packet - relayed*

it is still being set at a per link maximum. The cost of reception is magnified when traffic is relayed, making relaying less attractive.

Table 5.2 shows that only transmitting as far as is necessary is beneficial in terms of cost of transmission but does not influence the received power usage, as table 5.2 does not show a reduction in the cost of reception between approximate per link lower bound and maximum power algorithms, as would be expected.

Earlier in section 4.5 we discussed the issues associated with setting P_i to be zero. In this section it is something we must consider as it influences the results here. For example in table 5.2, P_{rx} (max) assumes that the intermediate nodes have turned of their receivers which is equivalent to P_i being zero.

		$Pr_{CS} < Pr_{TH}$				$Pr_{CS} = Pr_{TH}$			
		30m		60m		30m		60m	
		Relay	Direct	Relay	Direct	Relay	Direct	Relay	Direct
P(tx)	Max	42.7	10.7	42.7	10.7	42.7	10.7	42.7	10.7
	Min	17.5	4.7	17.6	9.7	17.5	4.7	17.6	9.7
	A	35.6	8.1	35.6	10.5	35.6	8.1	35.6	10.5
	B	12.3	8.4	17.5	10.5	12.3	8.4	17.5	10.5
P(rx)	Max	114.1	28.5	114.1	28.5	114.1	28.5	114.1	28.5
	Min	114.1	28.5	114.1	28.5	114.1	28.5	114.1	28.5
	A	114.1	28.5	*114.1*	28.5	114.1	28.5	*108.5*	28.5
	B	*114.1*	28.5	*109.3*	28.5	*108.8*	28.5	*96.3*	28.5
P	Max	156.8	39.2	156.8	39.2	156.8	39.2	156.8	39.2
	Min	131.6	33.2	131.7	38.3	131.6	33.2	131.7	38.3
	A	149.7	36.6	*149.7*	39.1	149.7	36.6	*144.1*	39.1
	B	*126.5*	36.9	*126.9*	39.1	*121.1*	36.9	*113.8*	39.1

Table 5.2: *Breakdown of energy (mW) consumption*

The right hand columns in table 5.2 shows the effects of setting Pr_{CS} to be Pr_{TH} and repeating the experiment. The differences between the left hand and right hand sides of table 5.2 are shown in italics, it can be seen that increasing the threshold at which transmissions are received decreases the associated energy costs. This is a possible source of additional energy savings if the carrier sense threshold and the correct reception threshold could be brought closer together. Even with this additional saving, it is more expensive to relay traffic through multiple devices than to communicate as far as possible in a single transmission, if the intermediate receivers are idle and off or have very low P_i values.

5.2.4 When to transmit

It has been shown that only transmitting as far as is necessary is beneficial both in terms of the cost of transmission and in terms of received power usage because devices outside the transmission range do not incur the receptions costs. This is only beneficial in terms of energy whilst the summed cost of reception of all the devices is greater than the summed cost of the devices being idle, as set out in the inequality equation 5.1.

$$\sum_{k=1}^{N-1} Cr_k(n) > \sum_{k=1}^{N_s} Ci_k(n) \tag{5.1}$$

Being able to reduce the reception costs is important because the right hand side of table 5.2 shows that reception costs for both direct and relayed traffic are at least triple those of transmission, and increase as the device density increases. It is obvious from table 5.2 that Cr

dominates Ct and hence devices should take advantage of overheard information.

5.3 Overheard information

A key optimisation of the algorithm is for a device to be able to learn about other devices by observing their transmissions. Consider three devices arranged in an equilateral triangle, figure 5.9, where the arrows show the transmissions of 1 packet or 10 packets between the devices at times, t_0 and t_1. The distance between the devices, x metres, is increased from 10 to 120 metres, in increments of 5 m. For the sake of brevity we only use the 802.11 radio model here.

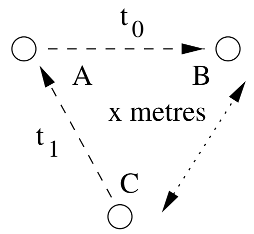


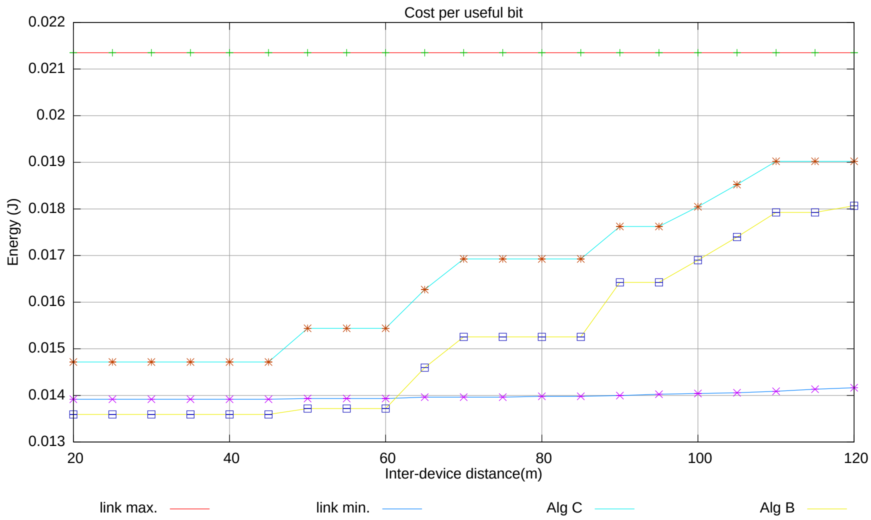
Figure 5.9: *Triangle arrangement of device*

The transmission from device A to device B at time t_0 is overheard by device C. Device C stores its "guess" as to the level at which it should communicate in future with devices A and B based on what it has observed. We use the standard approximate per link lower bound and per link maximum algorithms to give upper and lower bounds on the results. To demonstrate the benefits of using overheard information, we compare algorithm B with algorithm C where algorithm C is a version of algorithm B with the capability to take advantage of overheard information disabled, except when information is sent directly to the device. In this example, device C is able to take advantage of the traffic between devices A and B when algorithm B is used, but not when algorithm C is used.

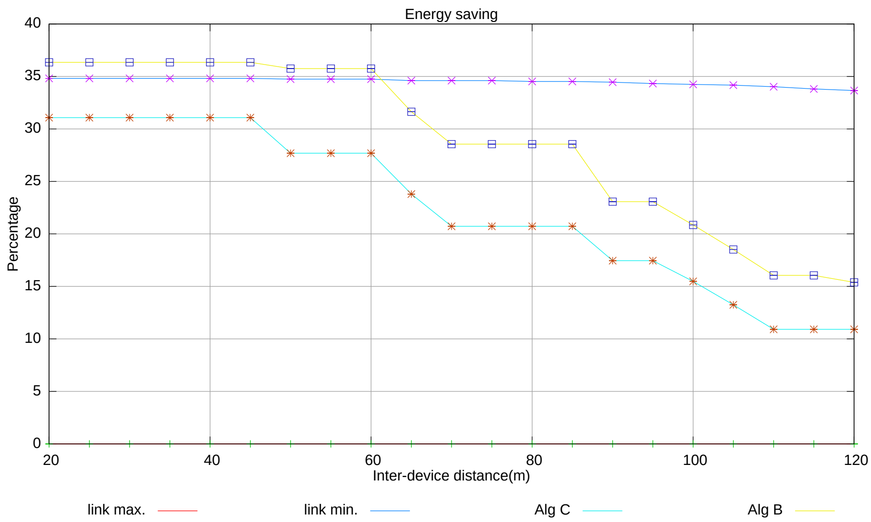
5.3.1 Results of using overheard information

Initially, we examine the effect of the transmissions being made up of a single packet sent from node A to node B, and later node C to node A. Figure 5.10(a) illustrates the energy consumption, C_{UB}, and figure 5.10(b) the energy saving of algorithms B, C and the approximate per link lower bound when compared with the per link maximum algorithm. As expected, and seen previously, the C_{UB} increases and the percentage energy saving decreases as the distance increases. The savings achived by taking advantage of the overheard information is approximately 5% of the total energy usage for the whole transmission when a data single packet is transmitted.

We extend the experiment to examine the effect of transmissions being made up of ten data packets being sent from device A to device B, and later from device C to device A. Figure



(a) C_{UB}

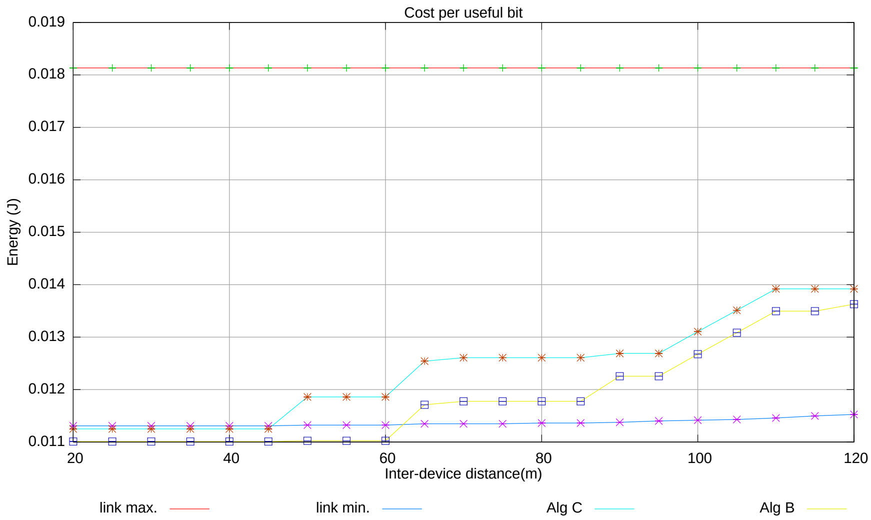


(b) Energy saving $\%$

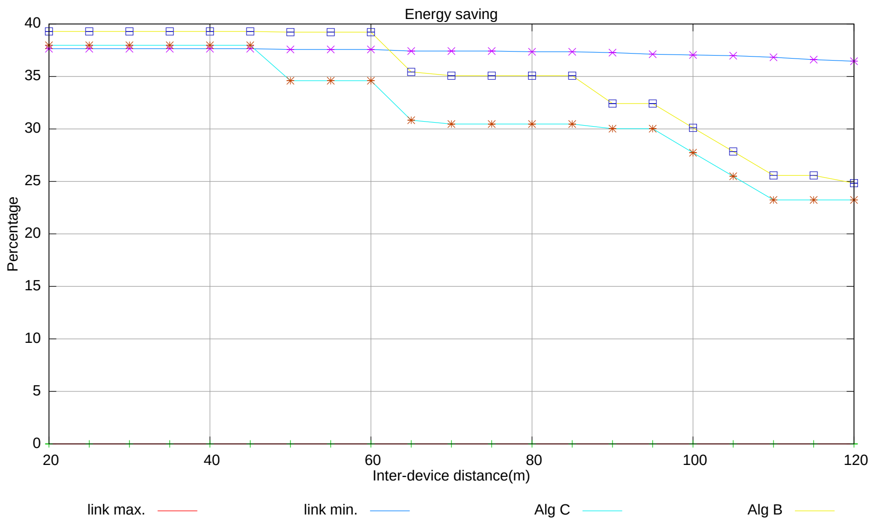
Figure 5.10: *Triangle arrangement of devices, 1 packet*

5.11(a) illustrates the energy consumption of algorithms B, C the approximate per link lower bound and the per link maximum. As expected, and seen previously, the C_{UB} increases as the distance increases. Comparing algorithms B and C the saving due to the overheard information is approximately 5%. The use of overheard information allows the algorithm to reduce its convergence period, this results in the energy saving. The saving is more pronouced as the number of neighbours increases.

Increasing the number of data packets sent from one to ten increases the overall energy saving achieved, figure 5.11(a) illustrates the C_{UB} and figure 5.11(b) the energy saving of algorithms B, C and the approximate per link lower bound when compared with the per link maximum algorithm. The saving due to the overheard information remains approximately 5%,



(a) C_{UB}



(b) Energy saving %

Figure 5.11: *Triangle arrangement of devices, 10 packets*

but the overall saving has increased by between 3 and 4 percentage points.

The increase in the percentage energy saving due to the overheard information is caused by the dominance of the algorithm's convergence period by the algorithms stable period, as the number of packets being sent increases. The use of overheard information allows the algorithm to reduce its convergence period, resulting in the saving.

5.3.2 Longer duration transmissions

In this experiment we extend the previous experiment that was limited to sending a single packet from the source to the destination, to sending two streams of one hundred packets and one thousand packets, with an inter packet interval of 0.1 seconds from the source to the destination. The single packet experiment gave an informative insight into the characteristics of the different power control algorithms. Increasing the number of packets sent exercises the multistage optimisation algorithms of algorithms A and B. As the number of packets a device observes increases, so the algorithms are able to adjust and tune their transmission power closer to the required level. This was evident when comparing graphs 5.2(a) and 5.5(a) where it can be seen that algorithm B in the multi-hop scenario more closely follows the approximate per link lower bound than in the single hop scenario where each node only observes approximately 25% of the packets.

5.3.2.1 Direct Transmissions

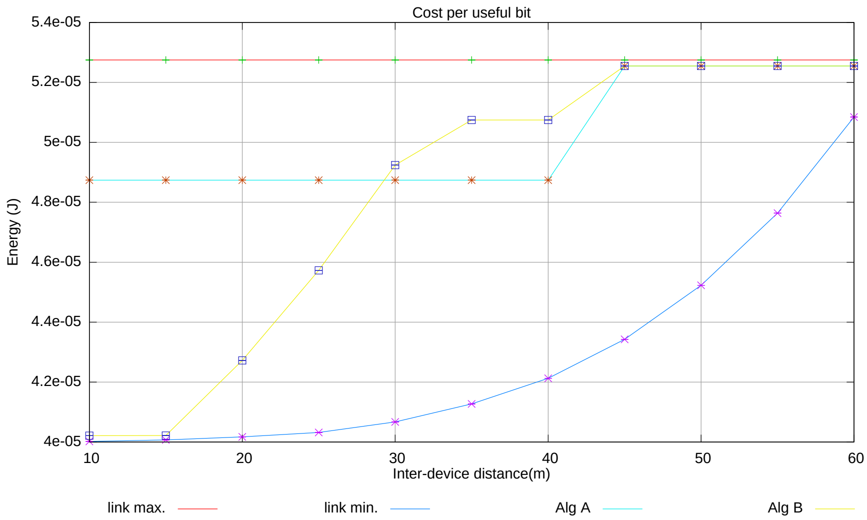
We repeat the experiment in section 5.2.1 but increase the number of packets from one to one hundred directly sent between the end devices.

In figure 5.12(a) the same relationship observed in figure 5.2(a) is also present between the different algorithms here. However the C_{UB} is lower as the overheads of the ARP mechanisms and other protocol mechanisms are averaged across a larger number of data packets. The extended transmission induces the same response in the Philips, figure 5.12(b), model as is in the 802.11 model. The experiments here are unable to confirm the claim that increasing the number of packets allows the algorithms to tune closer to the required level. As we will show later, the savings achieved in the C_{UB} by averaging the cost of the broadcast packets over a larger number of data packets masks any savings due to the increased number of tuning steps.

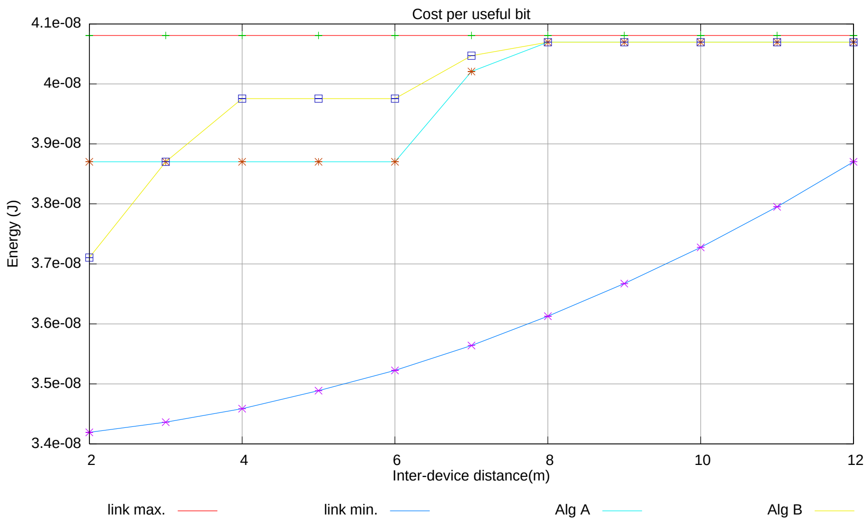
As the number of packets being sent increases, the graphs become smoother as the change between the increments become less marked. In the single packet example, the smooth change in energy usage between the distance increments is small for a single packet but is observable for 100 packets. Both the Philips and 802.11 models show an increased level of energy saving when the number of packets is increased.

If we increase the number of packets sent still further from one hundred to one thousand, figures 5.14(a) and 5.14(b), we would expect to (and do) see the observed trend in the decrease C_{UB} to continue.

For brevity we are just considering the 802.11 model, figure 5.15(b) shows the percentage decrease in C_{UB} for 100 and 1000 packets compared with the C_{UB} for a single packet. This is the expected result because the overheads associated with broadcast traffic, which is transmitted at full power, forms a smaller proportion of the C_{UB} average as the number of data packets sent increases. This is shown in equation 5.2, where n_d is the number of data packets, n_b is the number of broadcast packets, C_b is the cost per broadcast packet and C_d is the cost per data packet.



(a) 802.11

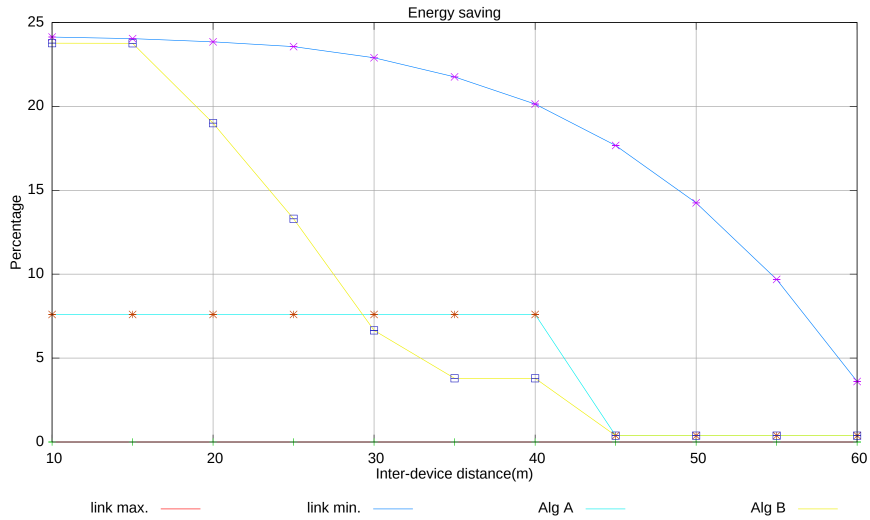


(b) Philips

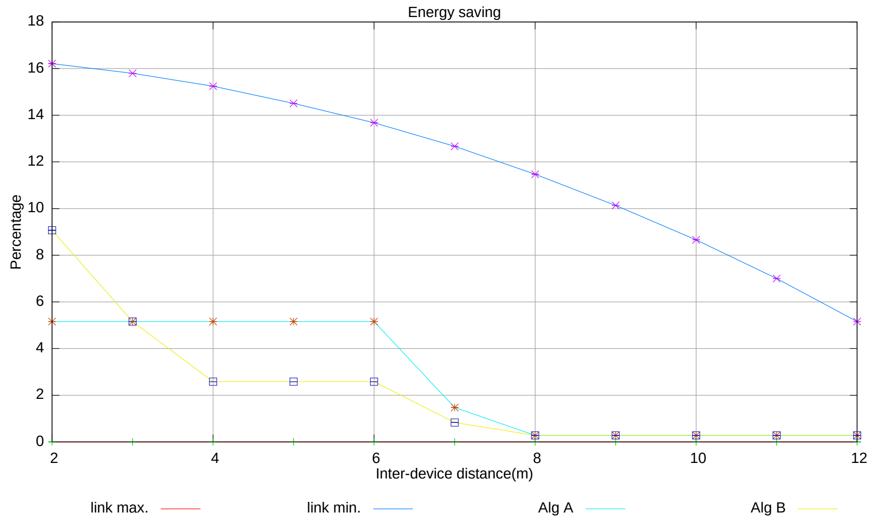
Figure 5.12: *C_{UB} of straight line of 5 nodes, 100 packets*

$$\begin{aligned} C_{UB} &= \frac{n_b C_b + n_d C_d}{n_d} \\ & C_d + \frac{n_b C_b}{n_d} \\ as\ n_b C_b &\approx const\ (and\ small) \\ C_{UB} &\rightarrow C_d\ as\ n_d \rightarrow \infty \end{aligned} \tag{5.2}$$

The C_{UB} percentage change in figure 5.15(b) tracks the fourth order power curve that relates energy consumption to distance, because as the saving $(C_b - C_d)$ tends to zero as the



(a) 802.11



(b) Philips

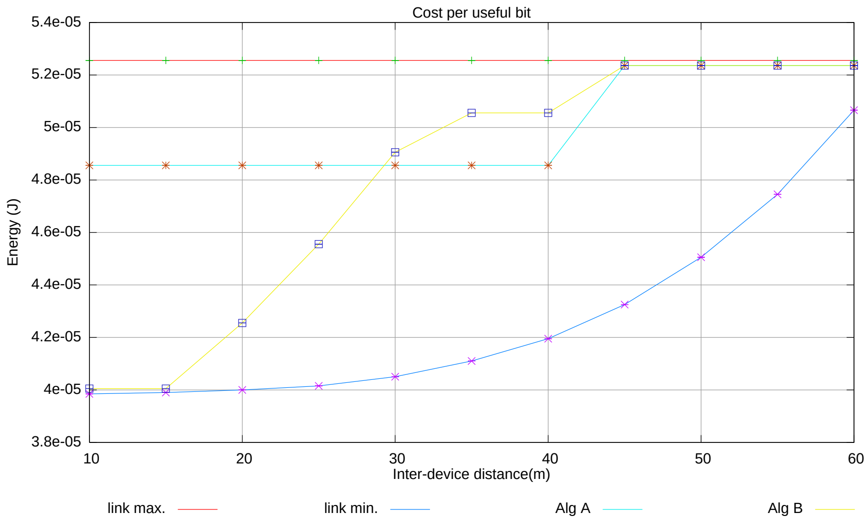
Figure 5.13: *Energy saving % of straight line of 5 nodes, 100 packets*

inter-device distance increases to the point where data packets are being send at the maximum transmission power. However the additional increase from 100 to 1000 in the number of packets sent only reduces the C_{UB} by between 0.21 and 0.29 % , as shown in figure 5.15(b).

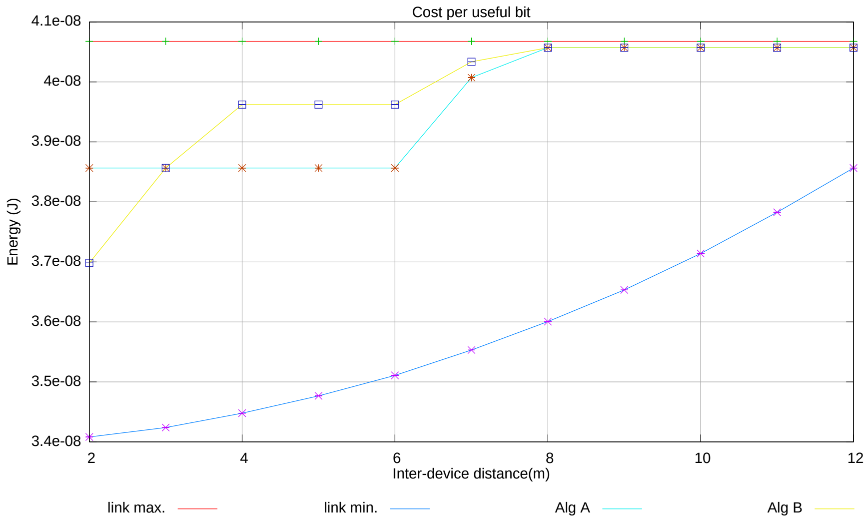
5.3.2.2 Relayed Transmissions

As with the direct transmission experiment above we repeat the earlier experiment from section 5.2.2, but we extend the transmission from one packet to one hundred packets, which are relayed from along the line of devices.

The same trend seen in the direct transmission scenario above repeats itself here. If we compare figures 5.16(a) and 5.5(a), we observe that the profile of the different C_{UB} algorithms



(a) 802.11



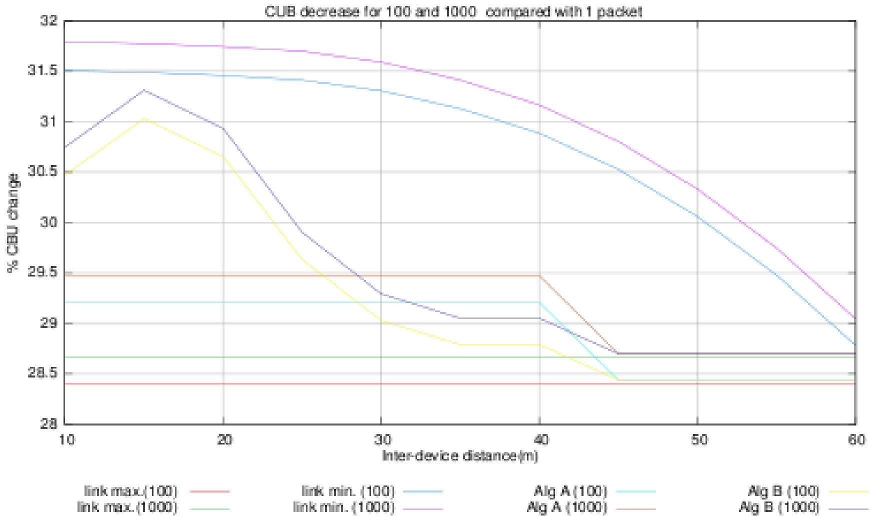
(b) Philips

Figure 5.14: *C_{UB} of straight line of 5 nodes, 1000 packets*

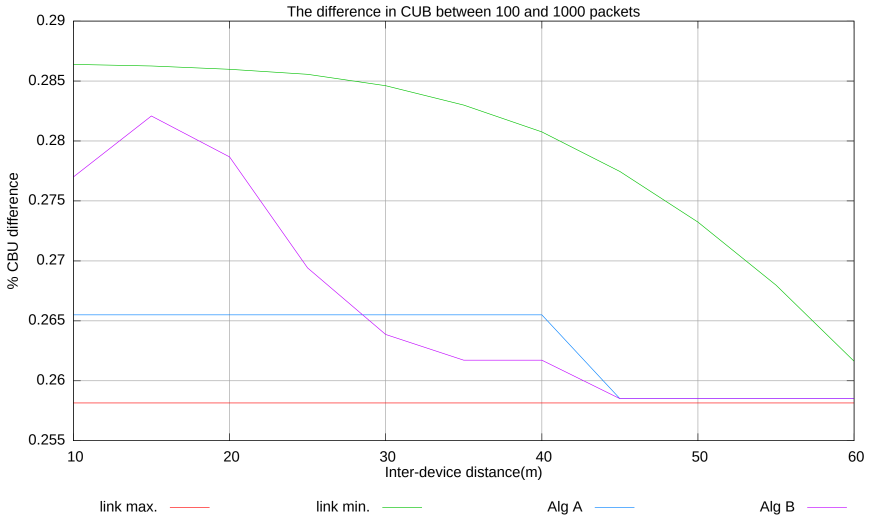
is the same in both graphs. As we saw in the direct packet scenario increasing the number of packets sent decreases the C_{UB} for the same reasons as laid out there. A result of this is a reduction in the magnitude of the change between events, for example the change in the approximate per link power lower bound algorithm at 180m. As the C_{UB} decreases as the energy saving compared with the default increases, this trend was again observed in the direct transmission example.

5.4 Hidden and Exposed terminal problems

Earlier in section 4.1.3, we discussed the hidden and exposed terminal problem , in the following two experiments we investigate this further. In both experiments a stream of one hundred 1500



(a) 100 and 1000 packets compared with 1 packet



(b) 100 packets compared with 1000 packets

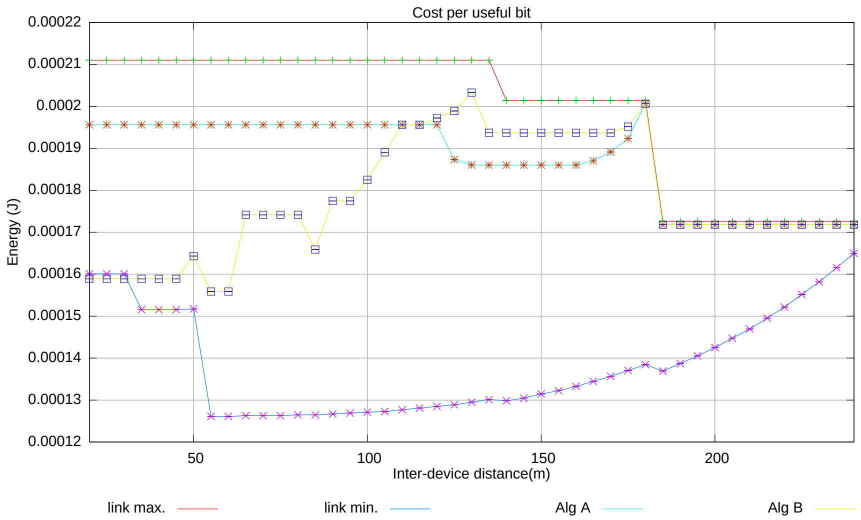
Figure 5.15: *802.11 % change of C_{UB} in a straight line of 5 nodes.*

byte packets are sent from C to D. For the hidden terminal scenario, an identical stream of packets is sent at the same from A to B. For the exposed terminal scenario, an identical stream is sent at the same time from B to A. The distance between the nodes is progressively increased from 10m in increments of 5m to 120m, with each increment being repeated 5 times. For simplicity and brevity we are only considering the 802.11 radio model.

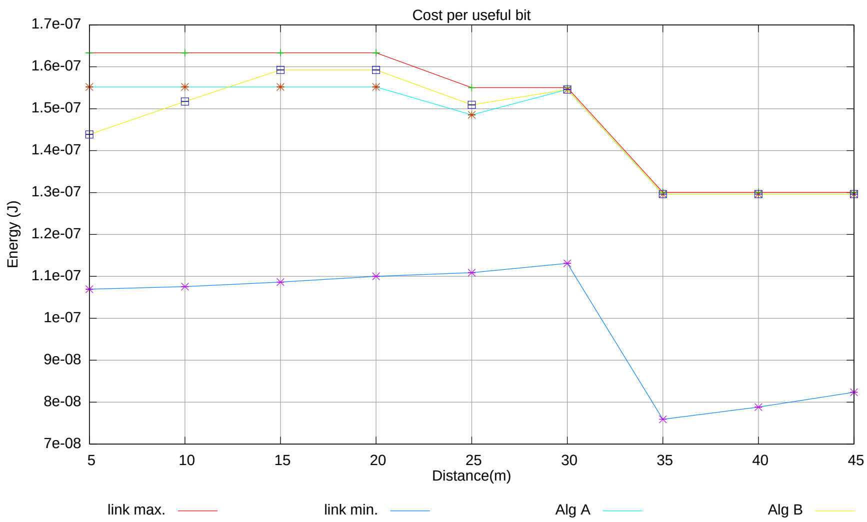
5.4.1 Exposed terminal

In this section, we specifically examine the effect of the power control algorithms upon the exposed terminal scenario.

Both algorithms A and B perform better than the maximum per link algorithm. If we



(a) 802.11

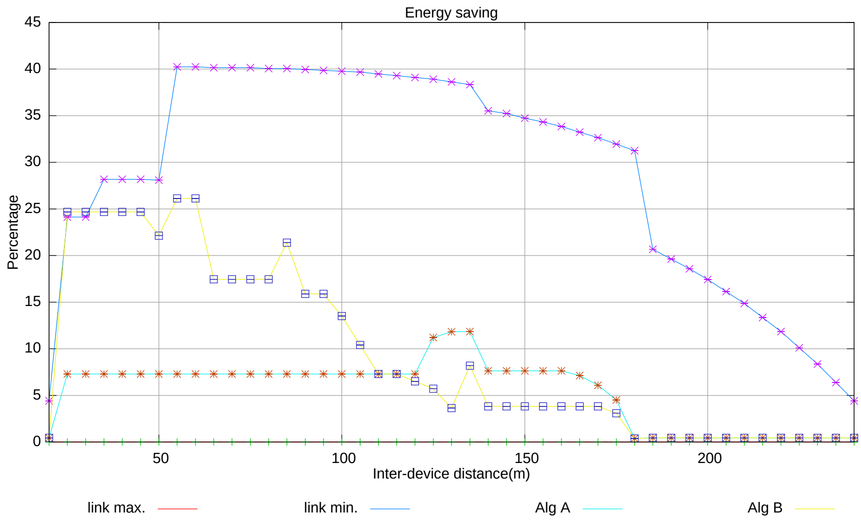


(b) Philips

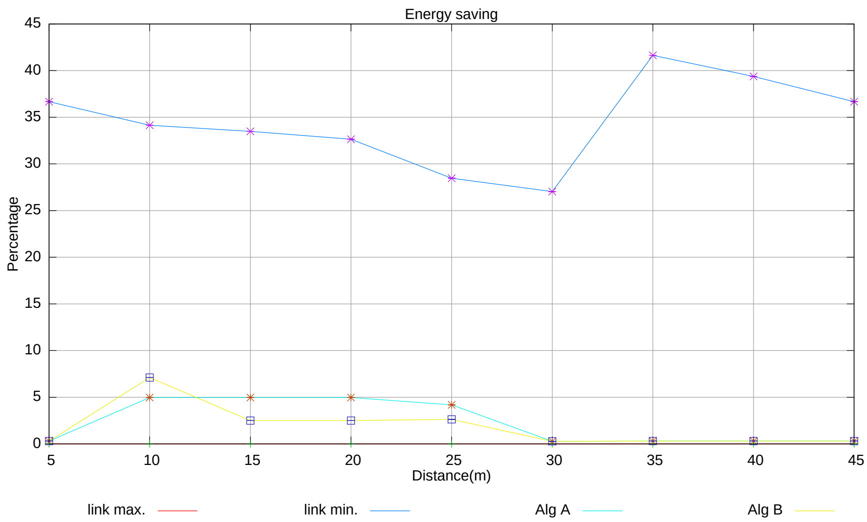
Figure 5.16: *C_{UB} of straight line of 5 nodes, 100 packets*

consider only the number of dropped packets the approximate per link lower bound algorithm exhibits worse performance than the other three algorithms. Comparing the total number of packets dropped, figure 5.19(a) with the number of these packets dropped due to collisions, figure 5.19(b), it is evident that packet collisions are almost the entire source of dropped packets for the approximate per link lower bound algorithm. Earlier in section 4.4.1 we discussed the effects of reducing the transmission power too far. By setting the transmission power too low using the approximate per link lower bound algorithm it has made the exposed terminal scenario worse.

We observe here that Algorithm B has the lowest number of packets dropped due to collisions, as shown in figure 5.19(b). The number of packets dropped due to collisions directly



(a) 802.11

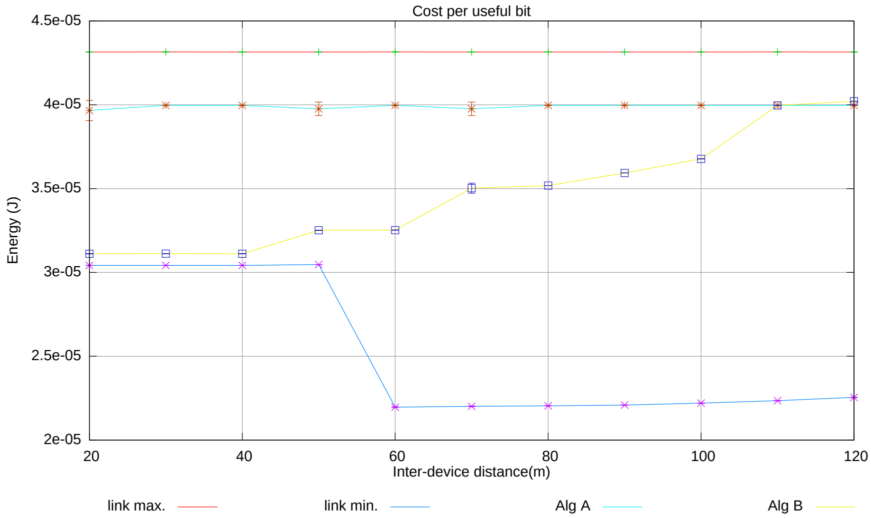


(b) Philips

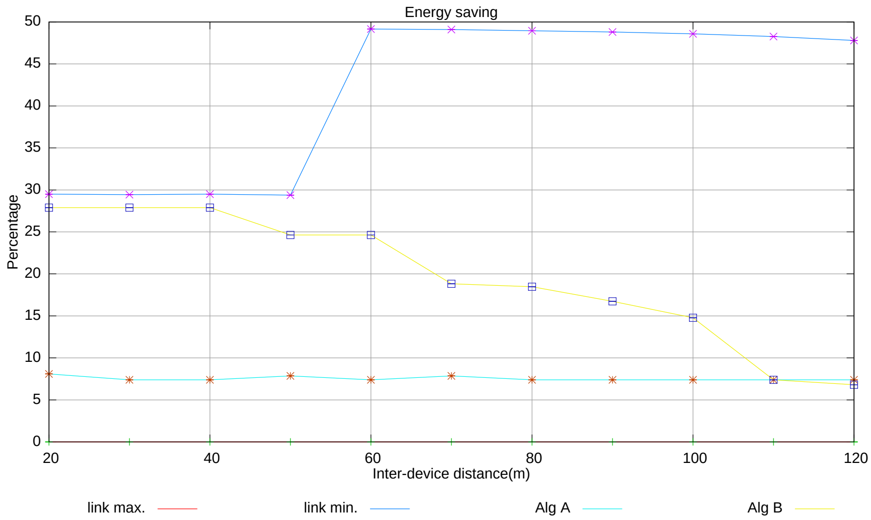
Figure 5.17: *Energy saving % of straight line of 5 nodes, 100 packets*

influences the overall number of dropped packets and the number of retries. Reducing the number of dropped packets increases the overall energy saving of the system by reducing the number of retransmissions needed.

As the distance between devices increases beyond 50m the number of dropped packets increases, and hence the number of retried packets also increases as the exposed terminal problem is made worse by the increased separation of the nodes. Overall, reducing the transmission power increases the effect of the exposed terminal problem. Algorithm B performs better than the maximum algorithm by having fewer dropped packets and the same throughput.



(a) C_{UB}



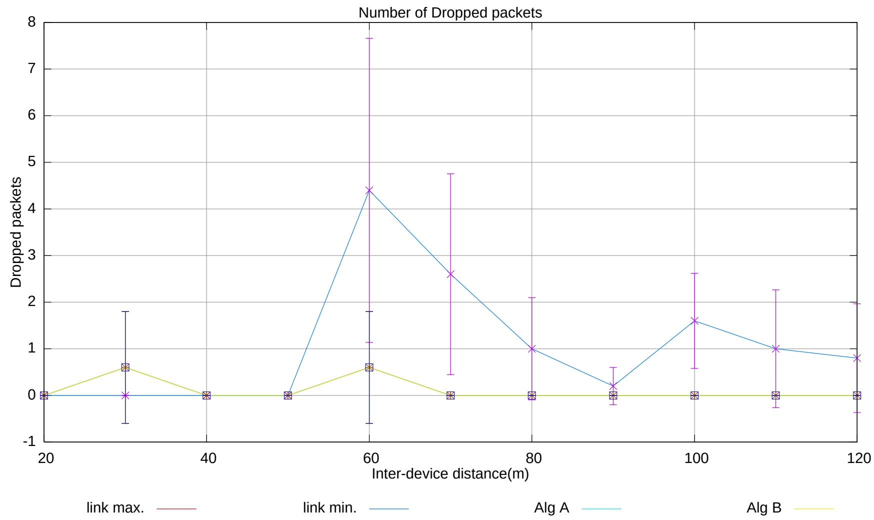
(b) Energy saving (%)

Figure 5.18: *Exposed terminal scenario*

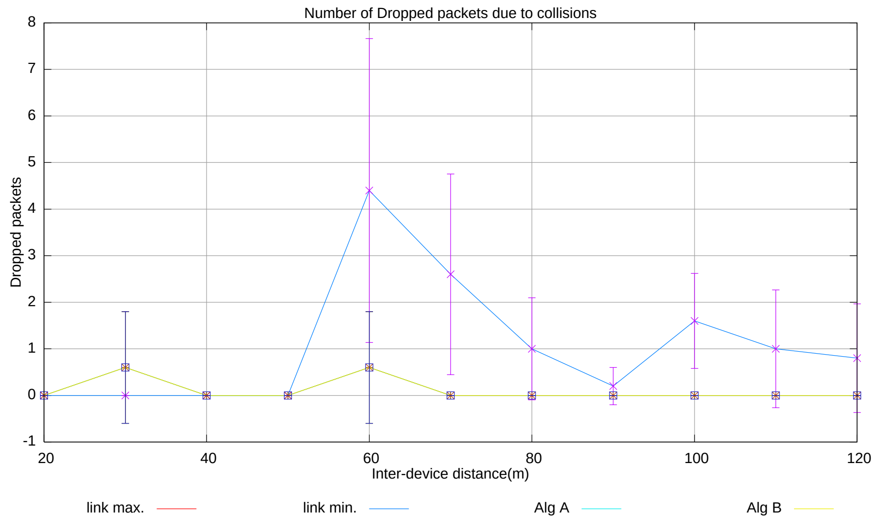
5.4.2 Hidden terminal

In the previous section we examined the effect of the different algorithms upon the exposed terminal scenario; here, we investigate the hidden terminal scenario under the same conditions.

As observed previously in the exposed terminal scenario, the approximate per link lower bound algorithm causes more packets to be dropped than either of the other three algorithms. In contrast to the exposed terminal scenario the number of packets dropped by the approximate per link lower bound algorithm in the hidden terminal scenario is half the number dropped in the exposed terminal scenario. As we saw in the exposed terminal scenario, the cause of dropped packets is almost entirely due to packet collisions caused by the breakdown in the



(a) dropped



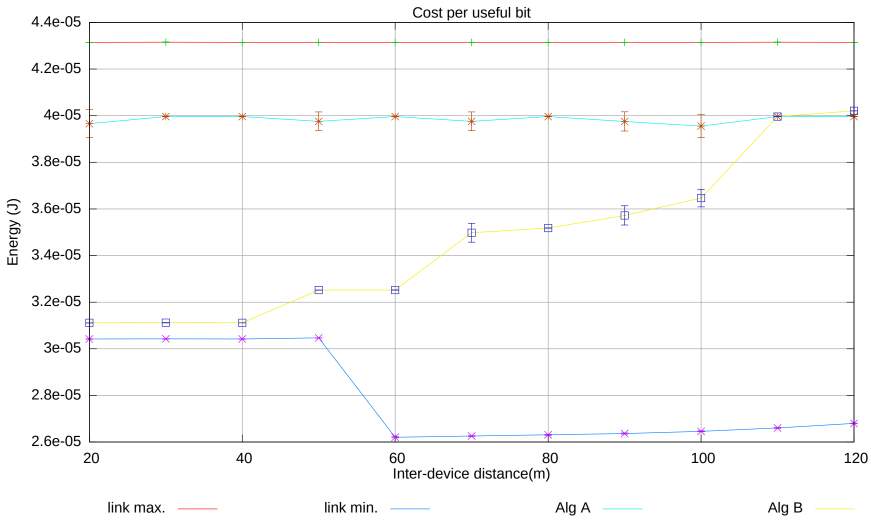
(b) collisions

Figure 5.19: *Total dropped packets and packets dropped due collisions in an exposed terminal scenario*

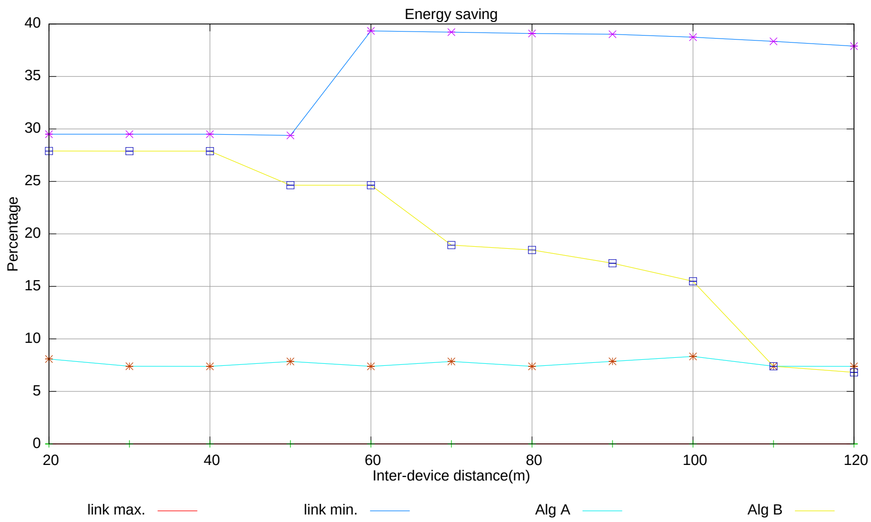
MAC protocol.

Algorithms A and B both perform better than either the approximate lower bound or maximum per link power algorithms with no additional losses whilst providing lower energy consumption costs than the maximum per link power control algorithm. The approximate per link lower bound algorithm provides a trade off between a much lower per packet energy usage model and an increase in the number of dropped packets. In a later section we replace the unacknowledged UDP traffic used here with acknowledged TCP traffic, for which the higher levels of loss can be compensated for by the transport layer.

The energy efficiency, figure 5.20(a), of the different algorithms follows the same trend as seen in the previous experiments, figures 5.12(a) and 5.12(b).



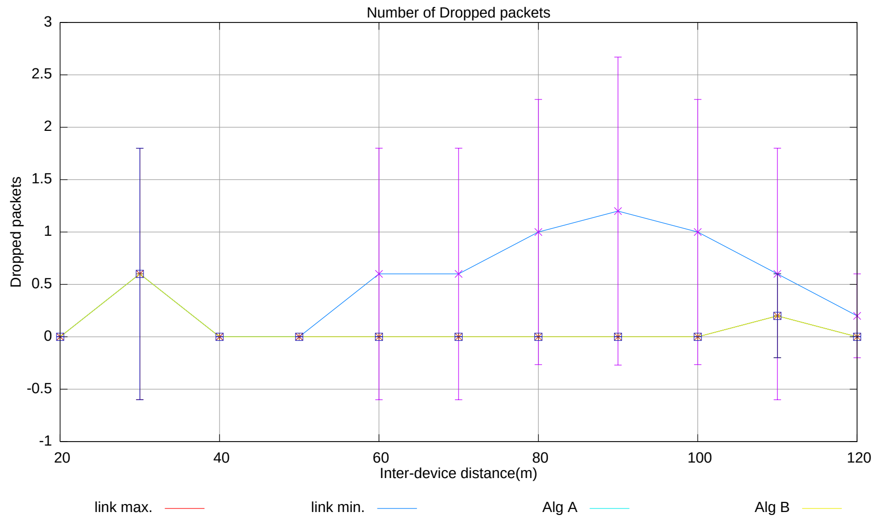
(a) C_{UB}



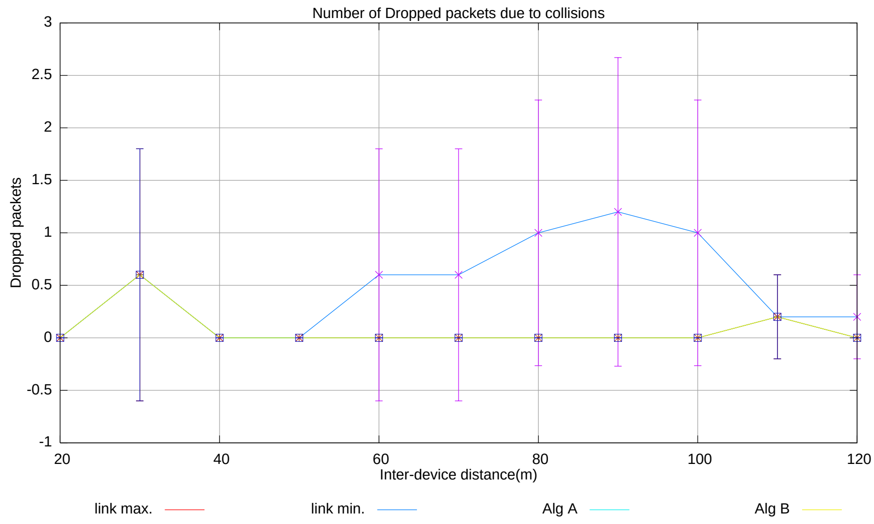
(b) Energy saving (%)

Figure 5.20: *Hidden terminal scenario*

As observed in the exposed terminal scenario, Algorithm B has the lowest number of dropped packets due to collision, figure 5.21(b). This directly reduces the overall number of dropped packets and hence the number of retries, thereby reducing the overall number of packets sent and the total energy consumption. The approximate per link lower bound and maximum per link algorithms perform least well in terms of the number of dropped packets due to collisions, compared with algorithm A or B. The exposed terminal problem is exacerbated by the signal propagating further than necessary; hence one would expect the approximate per link lower bound algorithm to perform better than either algorithm A or B in this scenario. However, it is worth noting that the approximate per link lower bound algorithm has the highest average number of application packets received and the lowest energy consumption.



(a) dropped



(b) collisions

Figure 5.21: *Total dropped packets and packets dropped due collisions in an hidden terminal scenario*

Overall, reducing the transmission power too far increases the effect of the hidden and exposed terminal problems as has been shown by the poor performance of the approximate per link lower bound algorithm in both the hidden and exposed terminal scenarios. Neither algorithm A or B is perfect: each transmits at a level slightly above the lower bound, which reduces the influence of the hidden and exposed terminal problems whilst providing an overall energy saving.

5.5 Conclusion

In this chapter we examined the energy costs associated with the transmission of packets in a simple scenario and have shown that the complex behaviour is observable even when using a simple power control scheme such as the one presented here. We have shown that as the number of packets observed by a device increases the savings of the algorithm increase until the algorithm has iterated through to a minimum value. We have shown that the energy costs associated with reception dominate those associated with transmission as the number of nodes decoding the signal increases. Thus, reducing the transmission power so fewer nodes overhear the transmission is beneficial to the overall power consumption. If the receiver of the radio needs to be on to enable routing of packets then we have shown that savings are achievable by taking advantage of overheard information. The limitation of the work presented here is that the scenarios are very artificial; we address this in the next chapter where we generalise the scenario further.

In section 4.5 of the previous chapter we discussed zeroing of the receiver idle power in the simulations; this will affect the results here. The affect here is of limited consequence because the simulations are not greatly different in length in comparison to each other due to there being a very limited number of dropped packets. However the overal message remains the same and is unchanged by the consideration of idle costs.

Chapter 6

Tuning of the transmission power of a low power radio system

In the previous chapter we showed that energy savings could be achieved by simple algorithms that tune the transmission power of the radio device based upon overheard information. The experiments in the previous chapter were intentionally very artificial as the aim was to explore the problem in a simple scenario and observe the underlying dynamics of the algorithms. To do this we considered a linear arrangement of devices and examined the flow of packets between them. This simple scenario is too artificial and ignores the majority of scenarios, so we aim to widen the scenarios here to include small groups of devices, at first in a regular arrangement and later randomly arranged.

The previous linear arrangement of five devices meant that each device could have up to a maximum of four neighbouring nodes including both one and two hop neighbours. Increasing the number of neighbours a device possesses, has both positive and negative side effects on the transmission of packets using the simple power control algorithms presented earlier. On the positive side we showed in section 5.3 that making use of overheard information to aid the learning process results in an additional 5% saving for a triangular arrangement of three devices. As the number of neighbours increases, so the number of devices overhearing a transmission also increases, this enables the overheard information to be exploited resulting in an energy saving. However, as the number of neighbours increases, so the probability of packet collision also increases. The MAC protocol regulates transmissions with the aim of reducing the number of collisions in the network. However, if the MAC protocol's control packets are adversely affected by the power control protocol, then the number of collisions would increase.

6.1 Small group of devices

In this section, we aim to generalise the scenario by evaluating the different transmission power control mechanisms with devices placed randomly in a square area. However, before we investigate a general system, we need to gain an understanding of what happens when we group a number of nodes in a small area scenario. As a starting point we start out by investigating 49 devices arranged in a 7 x 7 square, figure 6.1. We increase the inter-device distance x from 40m to 230m in steps of 10m for the 802.11 model and from 15m to 45m in steps of 2m for the Philips model. This range of inter-device distances ensures that there is always a path between all pairs of devices. When sending packets along the diagonal, the inter-device gap is $x\sqrt(2)$. The routing for this set of experiments is pre-computed for each distance increment using the shortest path algorithm. A pseudocode implementation of this is provided in the appendix C.

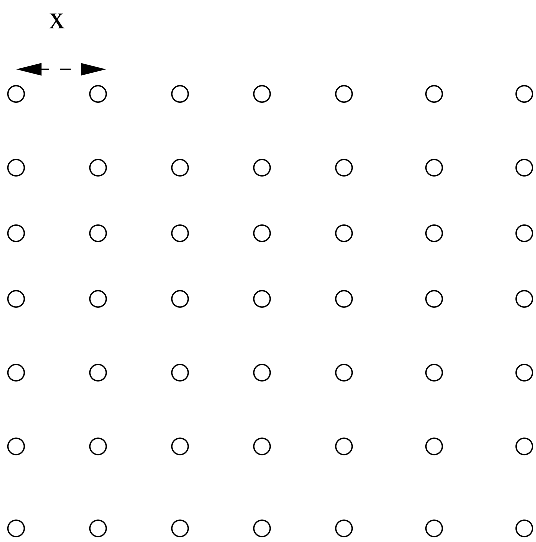
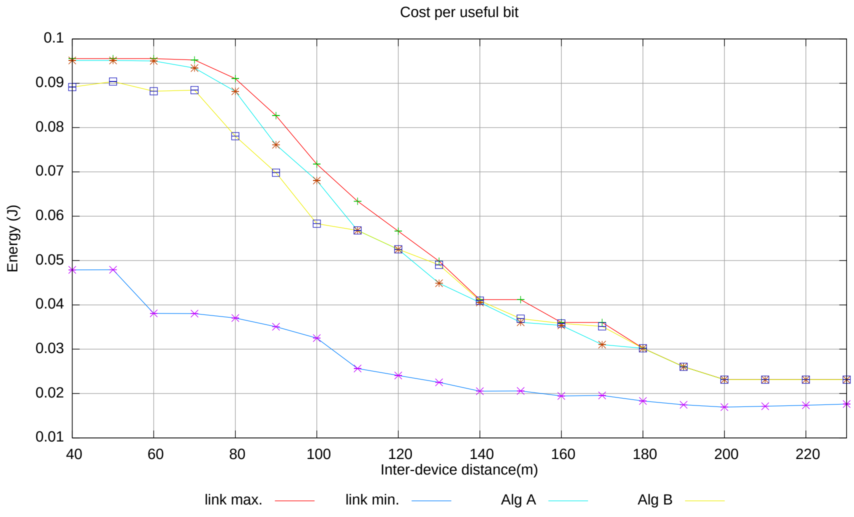


Figure 6.1: *7 x 7 square arrangement of devices*

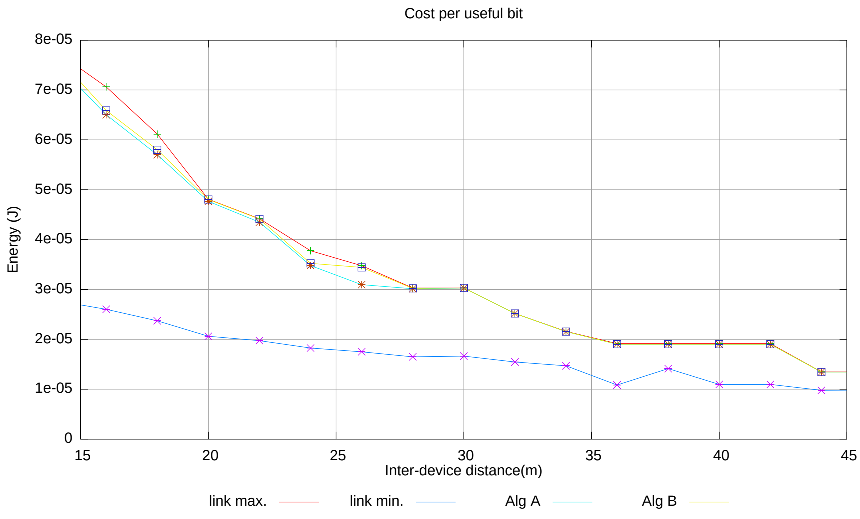
For the first experiment, we propagate a single UDP packet from the top left node to the bottom right node of the experiment depicted in figure 6.1.

Figures 6.3 and 6.4 show that both models follow the earlier observed trend that the C_{UB} decreases as the inter-device distance increases. Breaking down the energy consumption into the costs associated with transmission, figure 6.3(a),6.4(a), and reception, figure 6.3(b),6.4(b), it is obvious that the overall cost is being dominated by the cost of reception. The curvature of the transmission power graphs, figures 6.3(c),6.4(c) for both the 802.11 and the Philips radio models exhibit the same curvature and properties as was seen in the previous chapter, figure 5.6

.

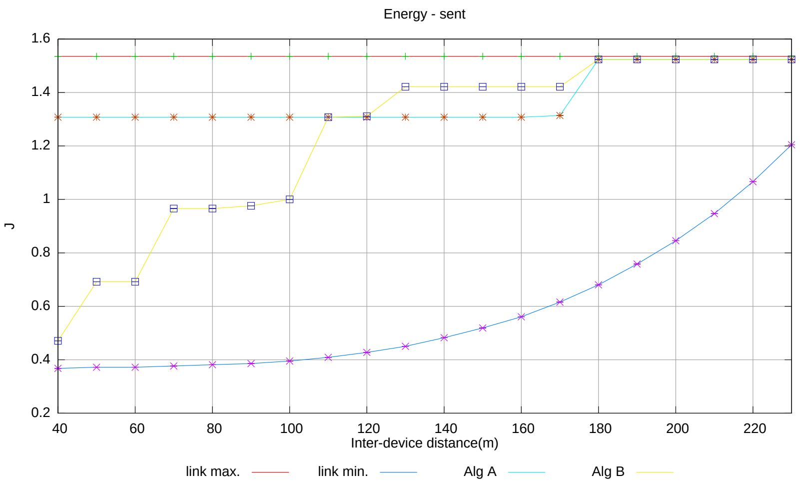


(a) 802.11

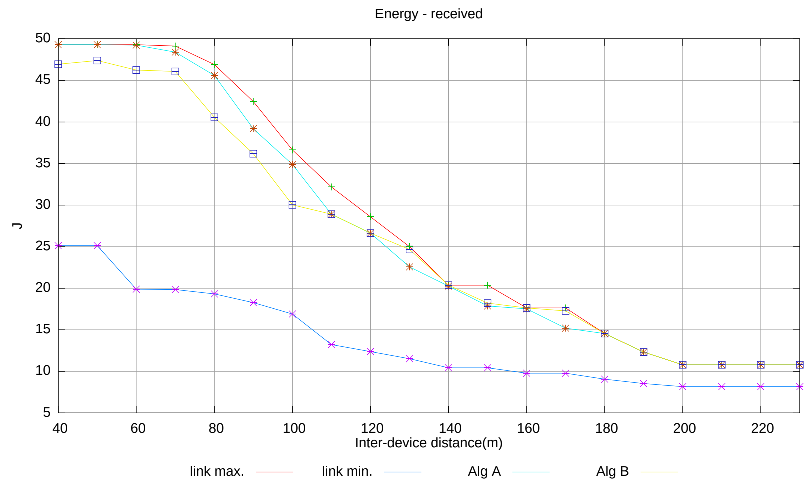


(b) Philips

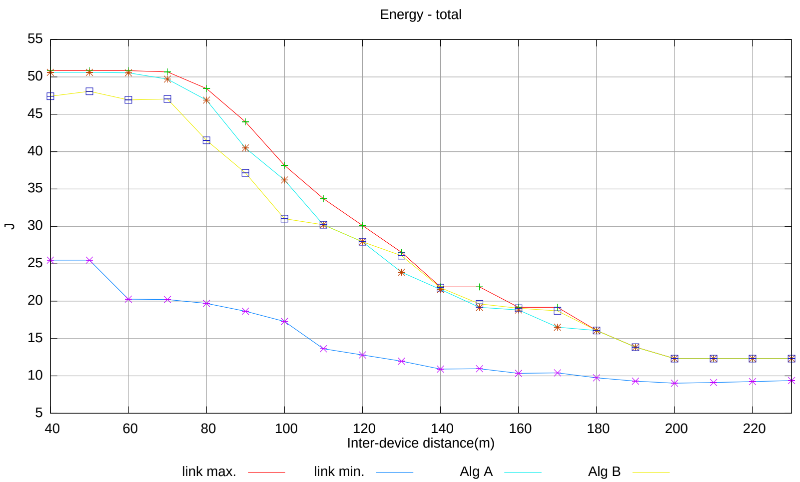
Figure 6.2: C_{UB} *of a single packet in a regular small area scenario (UDP)*



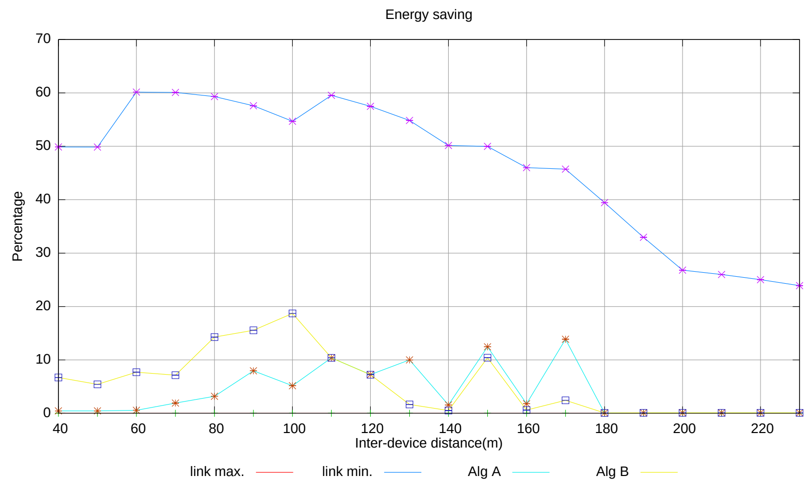
(a) E_{tx}



(b) E_{rx}

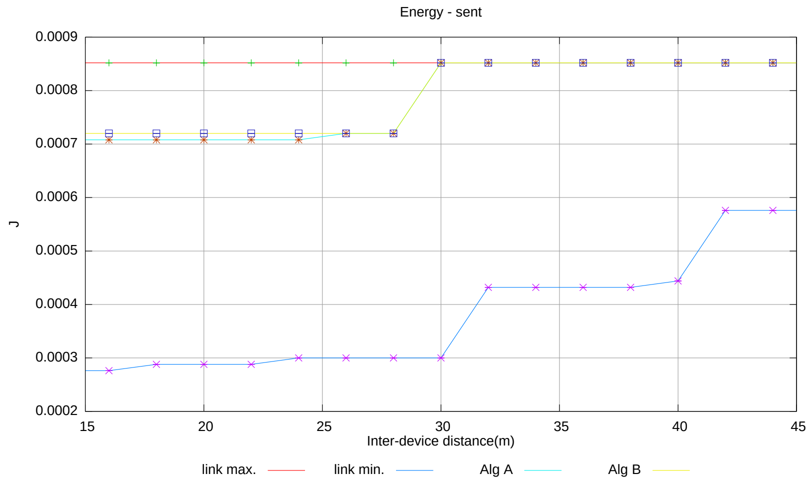


(c) E_{total}

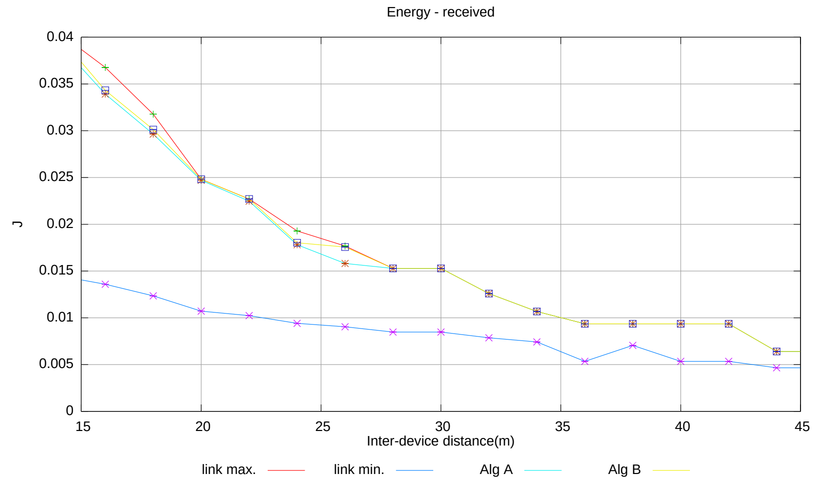


(d) Energy saving %

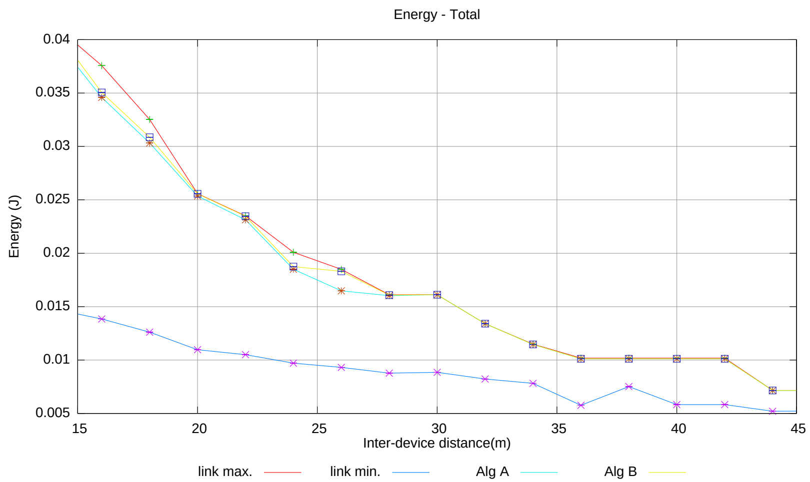
Figure 6.3: *Energy breakdown of straight line of 5 802.11 devices, 1 packet - relay*



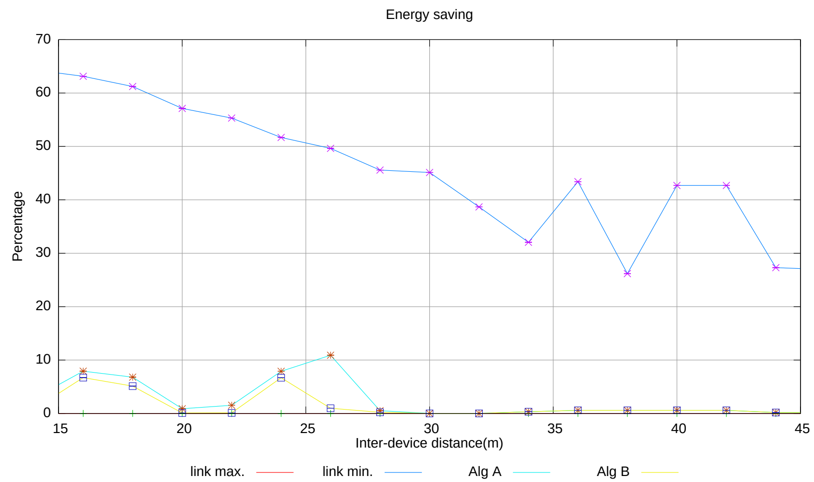
(a) E_{tx}



(b) E_{rx}



(c) E_{total}



(d) Energy saving %

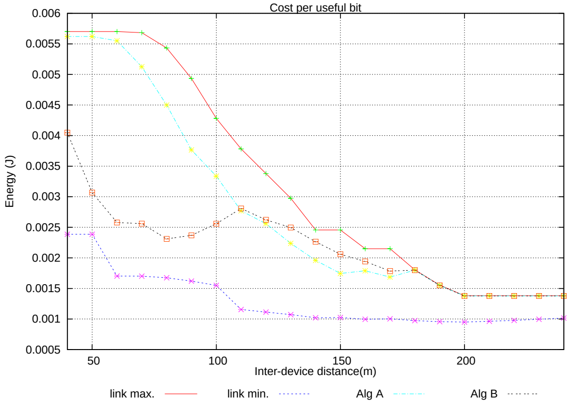
Figure 6.4: *Energy breakdown of straight line of 5 Philips devices, 1 packet - relay*

Examining the cost of reception, figures 6.3(b),6.4(b), it can be observed in a static network that as the density of hosts in the network decreases[1] the cost of reception also decreases. This results in the energy saving, figures 6.3(d),6.4(d), from the approximate lower bound per-link power control algorithm being significantly greater than those of either algorithm A or algorithm B. The approximate lower bound per-link power control algorithm minimises the number of devices overhearing a particular transmission by minimising the distance it transmits each packet and thus the producing an increased energy saving.

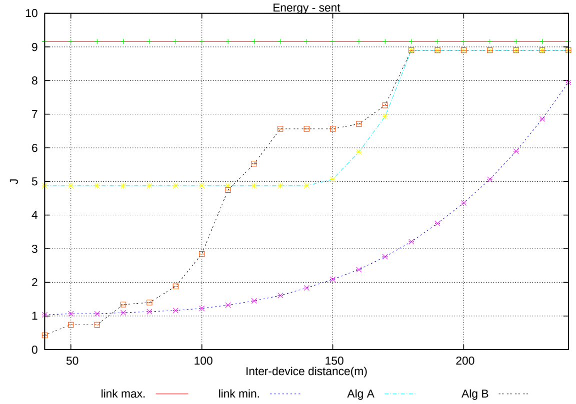
6.2 One hundred packets in a small group scenario

In the previous chapter experiments showed that both algorithm A and algorithm B need to observe multiple packets before they are able meaningfully to reduce the transmission power for a specific node. To investigate this in a small area scenario, we extend the transmission to transmit one hundred packets instead of one packet. The stream of UDP packets again flows from the top left hand corner to the bottom right hand corner device in figure 6.1. This stream of packets will examine the interactions between packets in the same flow and the tuning of the different algorithms. The inter packet interval is set as 0.1 seconds.

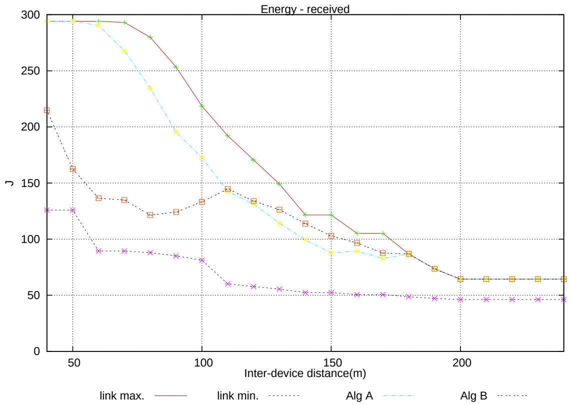
[1]The network density decreases as the inter-device distance increases.



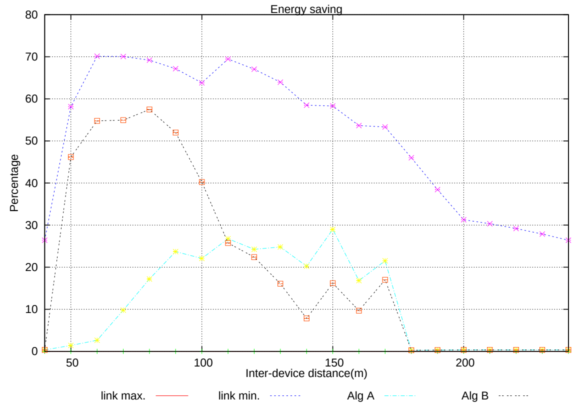
(a) C_{UB}



(b) E_{tx}

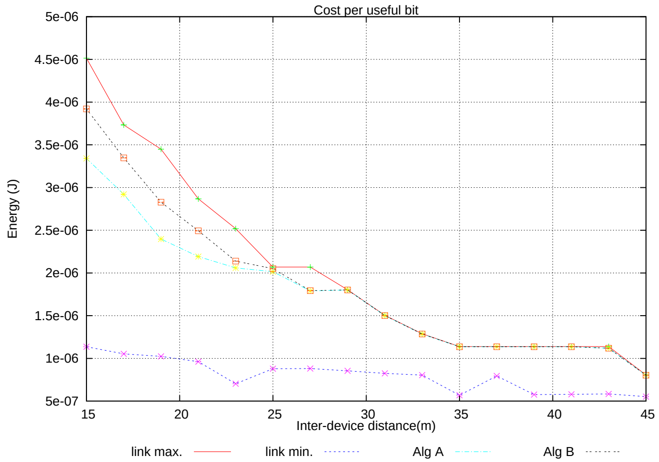


(c) E_{rx}

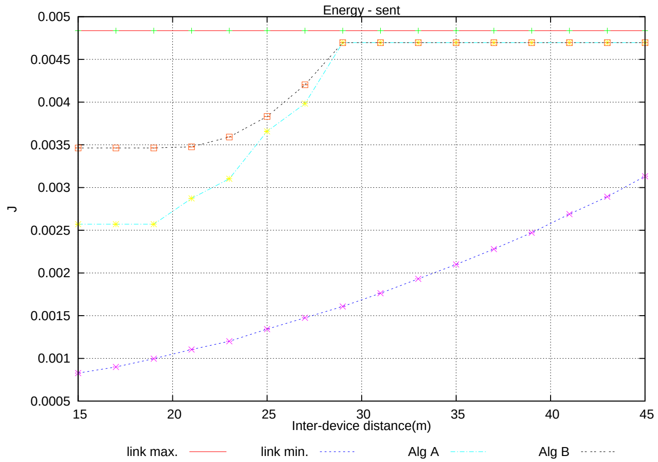


(d) Energy saving %

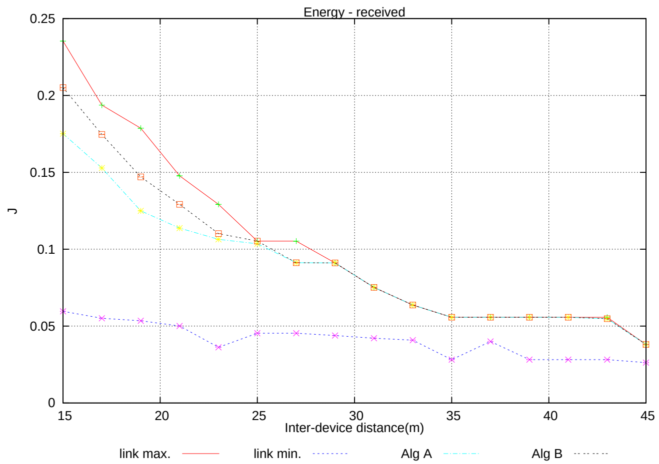
Figure 6.5: *100 packets in a regular small area scenario (802.11,UDP)*



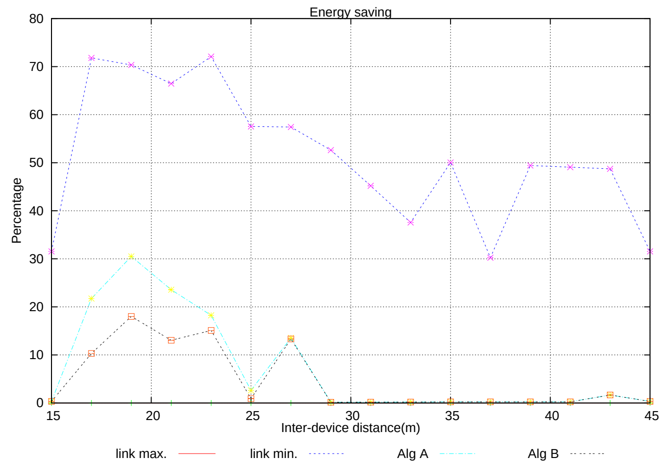
(a) C_{UB}



(b) $E_{tx}1$



(c) E_{rx}



(d) Energy saving %

Figure 6.6: *100 packets in a regular small area scenario (Philips,UDP)*

In earlier experiments in the previous chapter (section 5.3.2), we showed that, as the number of packets sent increased, the C_{UB} decreased as the overheads due to the broadcast and protocol traffic are distributed over an increased number of data packets.

If we compare the results here, figures 6.5 and 6.6, against those from figures 5.16(a),5.16(b) in the previous chapter, where we sent 100 packets along a line of five devices, we notice that the C_{UB} in figures 5.16(a),5.16(b) increased as the distance between devices increased, in contrast to figures 6.5(a),6.6(a) here, where the C_{UB} decreases as the inter-device distance increases. This was also the case for figure 6.2(a),6.2(b), but it is more emphasised here, showing the increased dominance of the cost of reception, figure 6.5(c),6.6(c), when compared with the cost of transmission, figures 6.5(b),6.6(b). The dominance is due to the increased number of neighbours.

As we saw in the previous chapter, as algorithms A and B observe more packets, their C_{UB} moves from being close to the maximum per link algorithm to being nearer the approximate lower bound. This change can be observed by comparing figures 6.2, 6.5 and 6.6.

From these graphs, we have observed that, as the density of the hosts in the network decreases, so the cost of reception decreases. This is not a surprising result because, as the density decreases, so the number of devices overhearing a particular transmission is also reduced. Whilst this reduces the dominance of the of the cost of reception compared with the cost of transmission it still remains at an approximate ratio of 10:1 for the approximate per-link lower bound power algorithm in this particular example.

The approximate per-link lower bound power algorithm as shown in the previous chapter transmits at a lower power level, thus fewer devices overhear its transmission, resulting in the increased energy savings shown in figure 6.3(d),6.4(d). In the straight line example in chapter 5, figure 5.8, we showed that savings of up to 35% were achievable, here we observe savings of up to 60% for the minimum per link algorithm and more modest savings for algorithms A and B. Neither algorithm A nor algorithm B perform nearly as well as the minimum per link power algorithm. This behaviour was also observed in the straight line examples where only a single packet is sent, preventing the learning algorithms from having many packets from which to learn.

But why the sudden change in the feasible savings? In the earlier straight line example, we were considering nodes only in a one dimensional space. If we think in terms of a single transmission in the following two scenarios :

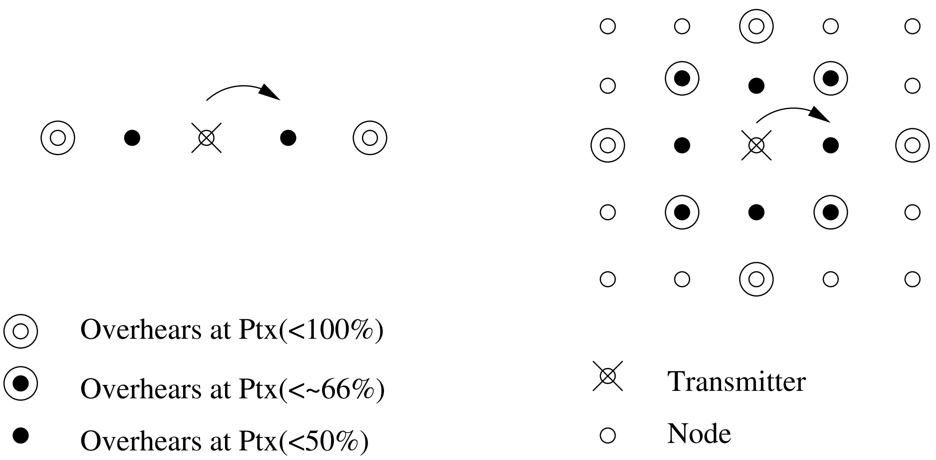


Figure 6.7: *Cost of transmission*

If we consider the straight line case, if the sending device tunes its radio so as to only transmit as far as needed only the recipients and one neighbour hear the transmission, $E = E_{tx_{min}} + 2E_{rx}$. If we consider the cost of the transmission in the small area scenario, for the same transmission now the recipient and 3 other neighbours now hear the transmission, so the costs now rise to $E = E_{tx_{min}} + 4E_{rx}$. However if we now compare against a transmission at full power for the straight line case, $E = E_{tx_{max}} + 4E_{rx}$, and for the small area case $E = E_{tx_{max}} + 12E_{rx}$. It is obvious from these equations that the cost of reception is the dominant energy consumer. If we crudely set $E_{tx_{min}}$ to be 1 (unit cost), $E_{tx_{\sqrt{2}}}$ to be 1.5, $E_{tx_{max}}$ and E_{rx} to be 2, the total unit costs are shown in table 6.1.

Scenario	Straight line (saving %)	Small Area (saving %)
Tx_{max}	10	28
$Tx_{\sqrt{2}}$	5.5 (45%)	17.5 (37.5%)
Tx_{min}	5 (50%)	9 (68%)

Table 6.1: *Unit Costs*

If we consider the scenario in which we make an error in the minimum power calculation and tramsit at a level slightly too high so the transmission carries a distance of $\sqrt{2}$, $P_{tx_{66\%}}$. In the straight-line example, no further devices observe transmission so the cost is $E = E_{tx_{\sqrt{2}}} + 2E_{rx}$. However, in the small area scenario, the costs rise to $E = E_{tx_{\sqrt{2}}} + 8E_{rx}$. The additional cost for the small area example is $(E_{tx_{\sqrt{2}}} - E_{tx_{min}}) + 4E_{rx}$ compared with $E_{tx_{\sqrt{2}}} - E_{tx_{min}}$ for the straight line. The savings achieved drops from (assuming $E_{tx_{\sqrt{2}}} = 1.5$) 68% to 37.5% due to the error. From this, we can observe that, as the density increases, so the possible achievable savings increase, but at the cost of a decreased tolerance in the accuracy of the setting of P_{tx}. Both algorithms A and B exhibit large inaccuracies when they only observe a small number of packets, leading to a lower than expected energy saving.

In the previous experiment, neither algorithm A nor algorithm B showed an energy saving nearing that of the approximate per-link lower bound power algorithm. In this experiment where we increase the number of packets being transmitted, there has been, as expected, an increase in the observed energy savings, figures 6.5(d),6.6(d), because the algorithms are able to adjust further towards the minimum.

6.3 Competing flows of traffic

We further introduce an additional seven packet flows into the experiment. All flows are shown in figure 6.8, with t_x indicating their start time. In this experiment, we are not concerned with the congestion caused by completing flows but instead we are examining the convergence of the power control algorithms under the influence of more traffic. The traffic flows are of one hundred packets and are sequentially started at intervals of 10 seconds, t_{10}, t_{20} etc, such that no two flows are concurrently operating. We note and acknowledge that all the packet flows do flow through the centre of the square and that these nodes will experience the most congestion but they will equally observe most packets and will be able to tune more accurately towards a minimum value.

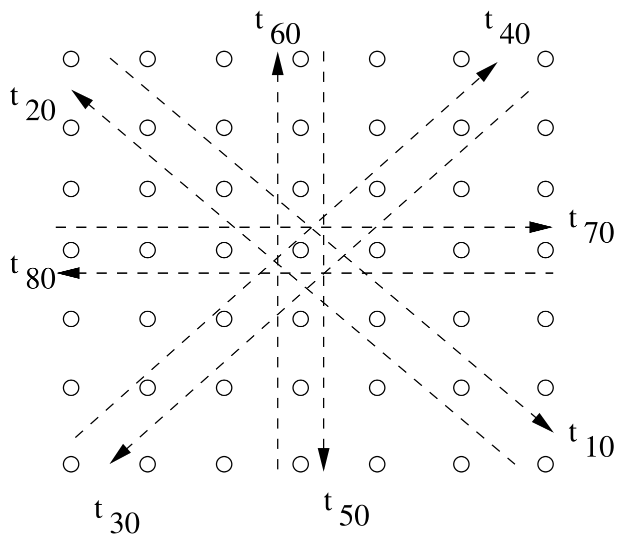
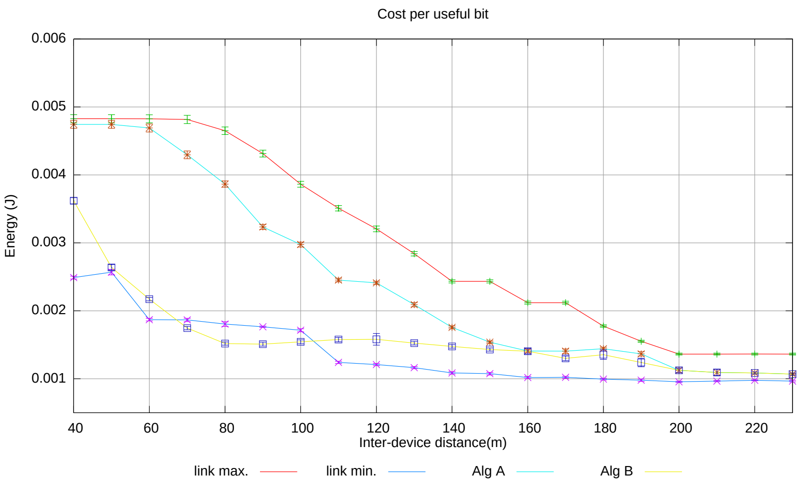
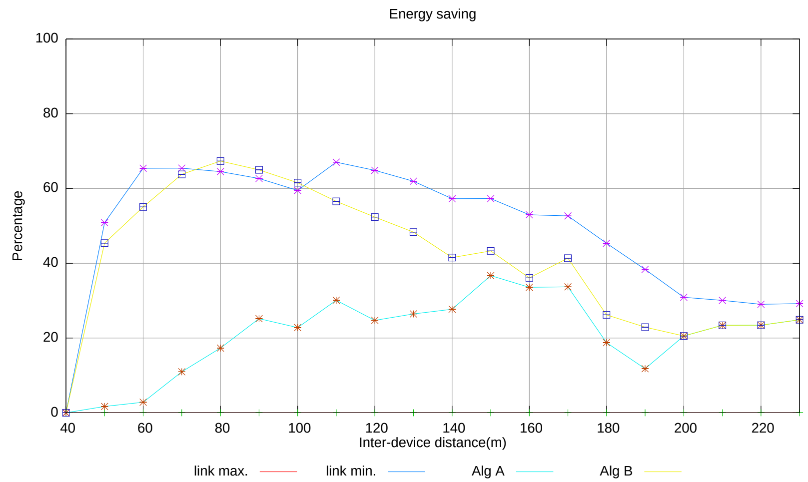


Figure 6.8: *Traffic in a 7 x 7 square arrangement of devices*

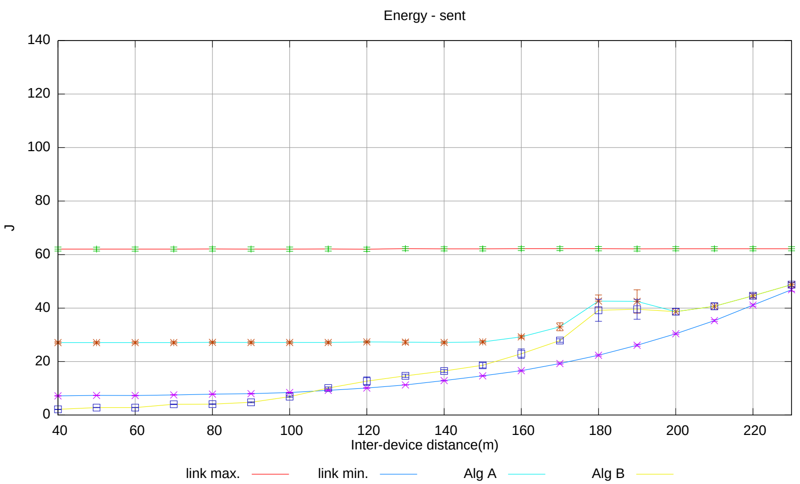
We saw in the previous experiment that, as we increase the number of packets, algorithms A and B are better able to tune their transmission parameters towards a minimum level. Increasing the number of packets by a factor of 8 should increase this saving further; however, there is an upper limit to these savings as was shown in section 5.3.2.1.



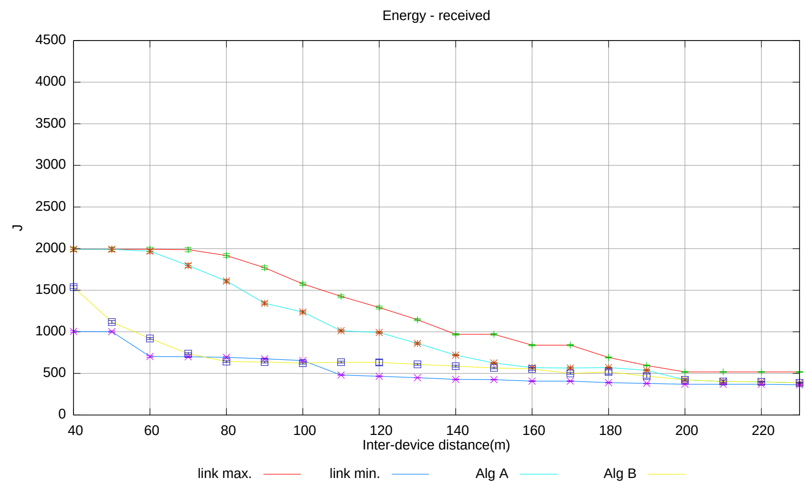
(a) C_{UB}



(b) Energy saving %

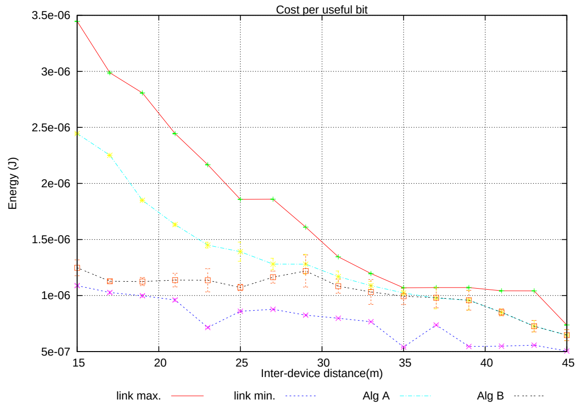


(c) E_{tx}

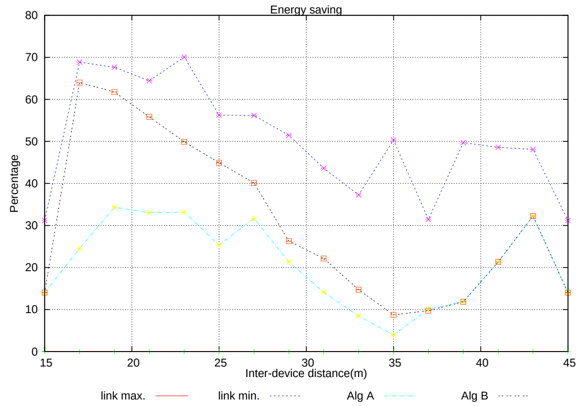


(d) E_{rx}

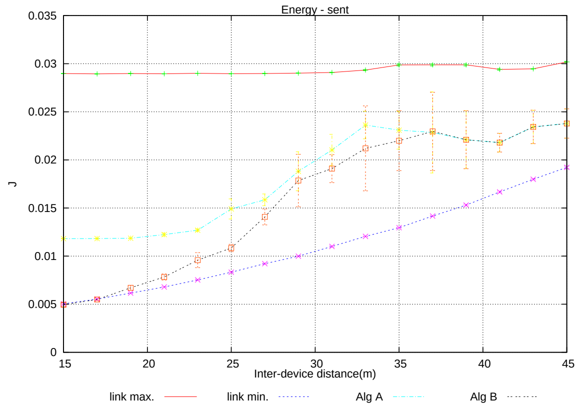
Figure 6.9: *800 packets in a regular small area scenario (802.11,UDP)*



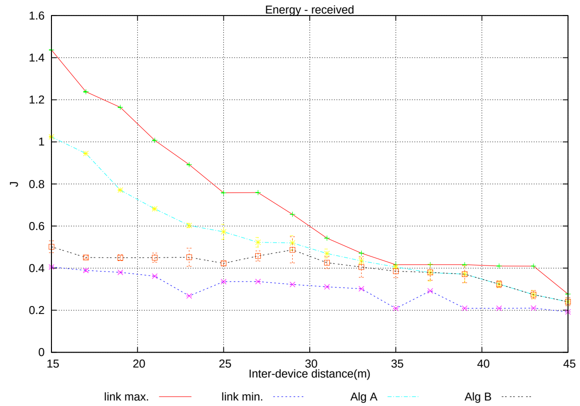
(a) C_{UB}



(b) Energy saving %



(c) E_{tx}



(d) E_{rx}

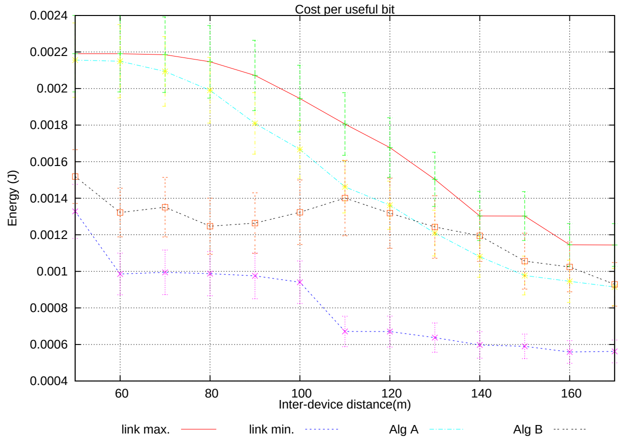
Figure 6.10: *800 packets in a regular small area scenario (Philips,UDP)*

For the 802.11 radio model, increasing the number of flows and packets further reduces the difference in the C_{UB} between the approximate per-link lower bound power algorithm and the other algorithms, in particularly algorithm B, but also, to a lesser extent, algorithm A. If we compare this experiment against the previous experiment where only 100 packets flowed through the network, we can see by comparing figures 6.5(a) and 6.9(a) the relative and absolute reduction in the C_{UB}. The cause of the absolute reduction is due to an increased number of data packets being observed as both the maximum and minimum algorithms have experienced approximately the same reduction. The relative change observed here between algorithms A and B and the minimum and maximum algorithms is due to the additional number of packets observed and the further savings achieved.

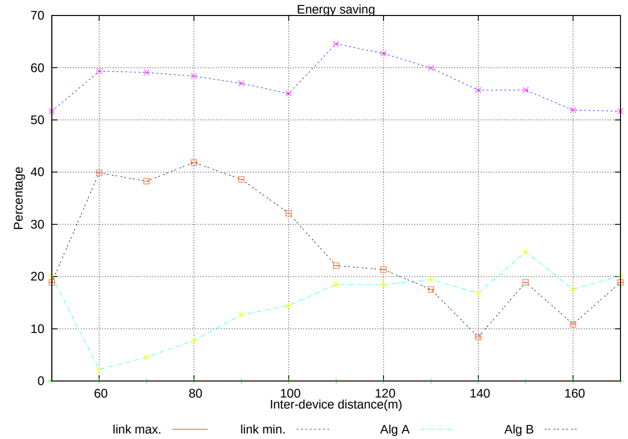
6.4 Random flows, structured layout

Using fixed flows of traffic that flow along "regular" paths is an ideal scenario to ensure good performance from the transmission algorithms. Thus, in this experiment we use random traffic flows, where we would expect to see a reduction in the relative percentage energy saving achieved by the different algorithms whilst following the same overall profile. In other words, the savings should get worse as the scenario becomes more realistic.

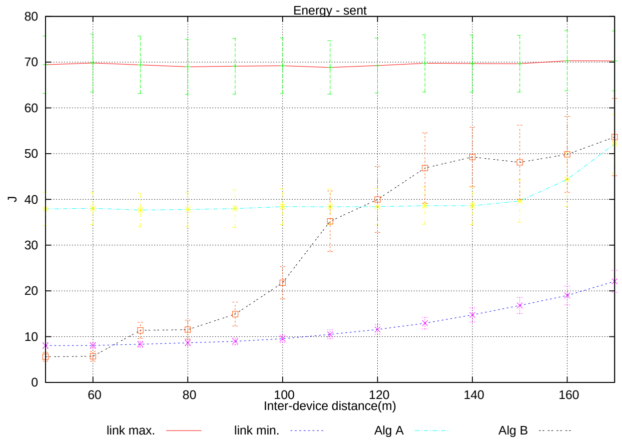
To exercise the algorithms further, we maintain the same fixed arrangement of devices, figure 6.1, but replace the structured flows with 20 random flows of 100 packets. The random flows have random source and destination nodes which have been selected uniformly at random from the set of 64 devices, where the source and destination nodes are not the same. The inter device distance, x, is increased from 40m to 170m in 10m increments for the 802.11 model and from 15m to 45m in 2m increments for the Philips model. The inter packet gap is distributed uniformly at random between 0 and 0.1 seconds. Each randomly selected set of flows forms a traffic scenario; with five different traffic scenarios for both the 802.11 and the Philips models being randomly generated and used at each distance increment.



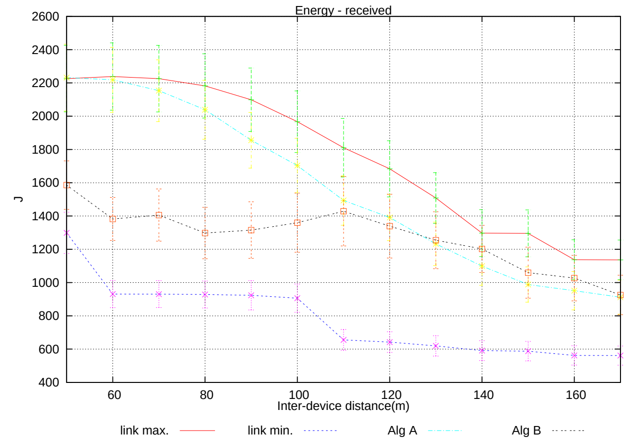
(a) C_{UB}



(b) Energy saving %

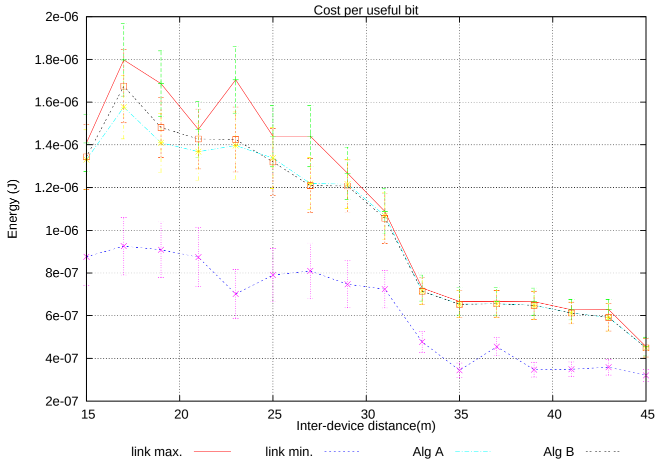


(c) E_{tx}

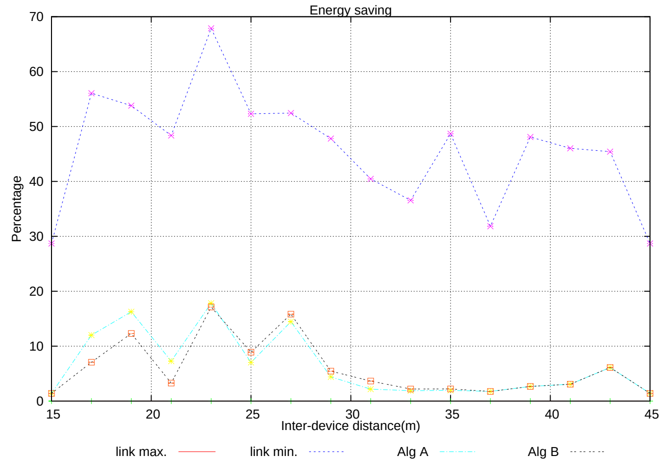


(d) E_{rx}

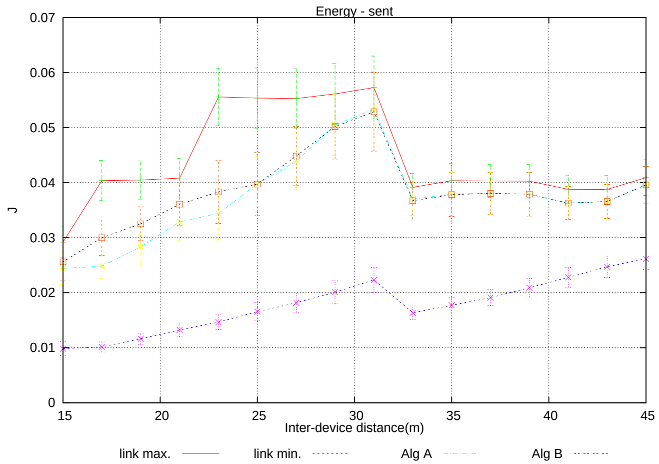
Figure 6.11: *Energy breakdown of 20 random flows in a regular small area scenario (802.11,UDP)*



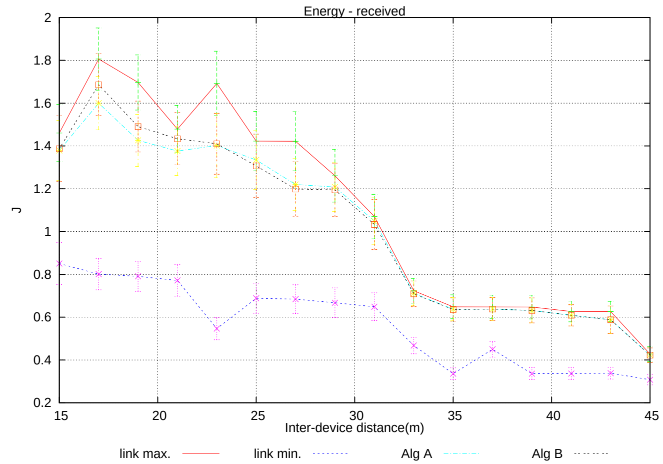
(a) C_{UB}



(b) Energy saving $\%$



(c) E_{tx}



(d) E_{rx}

Figure 6.12: *Energy breakdown of 20 random flows in a regular small area scenario (Philips,UDP)*

Comparing figures 6.9(a) and 6.11(a) which show the C_{UB} for the 802.11 radio model for the 800 packet scenario and this scenario of 20 random flows in a structured layout of devices, we observe the same trend with the C_{UB} decreasing as the inter-device gap, x, increases. As expected we observe that the C_{UB} is lower in this scenario than in the 800 packet example earlier, approximately 50% of the 800 packet example. This same reduction in C_{UB} was noted earlier in chapter 5, due to the higher number of packets flowing. The most notable change here is the relative increase in both algorithms A and B. As expected algorithm B is performing worse in this scenario than was observed earlier as the average path length has reduced.

If we compare the $\%$ energy saving graphs for this experiment and the previous earlier 800 packet one, figures 6.11(b),6.12(b) and 6.9(b),6.10(b), an approximately drop of 20% is observable as the traffic model changes. The energy consumption for packet transmission and reception for the 802.11 model, figures 6.11(c) and 6.11(d) respectively, are in line with those seen earlier. However closer comparison for figures I.2(a) and 6.11(c) we can observe that algorithms A and B have moved away from closely tracking the approximate per-link lower bound power algorithm. We showed earlier that as the number of packets a device observes, so the more able the algorithm is to able to tune towards the minimum per link algorithm. As the average path length reduces, which is the case here, so the number of packets observed on average across the experiment reduces. It is this overall reduction that causes the algorithms to be less able to closely track the minimum per link algorithm.

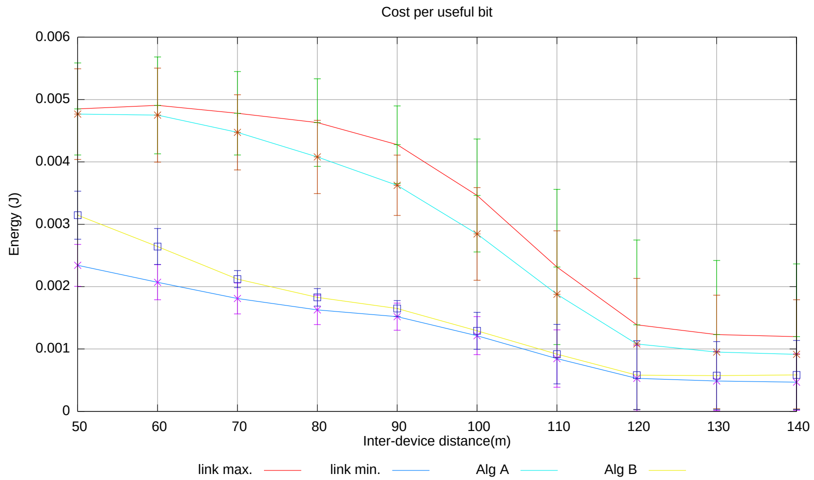
We have observed the same trend as before, the C_{UB}, figures 6.11(a),6.12(a), drops as more packets are observed and the density of the network overall decreases because of the overall dominance of the cost of reception, figures 6.11(d),6.12(d). If we look at figures D.5(a),D.6(a),figures D.5(b),D.6(b), which shows the number of sent and received application level packets for each radio model we see that for both models, the approximate per-link lower bound power algorithm is most negatively affected by packet loss. To better understand the cause we have to look at the MAC layer statistics, in figures **??**,**??** we observe that the approximate per-link lower bound power algorithm is experiencing significantly more packet collisions than any of the other algorithms indicating that the minimum algorithm is causing the hidden terminal problem to be over emphasised. We can see that the less accurate algorithms, A and B, are less prone to this and that the packet collisions in general are the cause of the vast majority of the packet losses.

We can see that there is a fine line to be drawn between over optimising the power algorithm's efficiency and achieving reliable packet delivery. In fact, we see that the learning algorithms are both increasing the overall energy saving, figures 6.11(b),6.12(b), whilst only suffering a limited amount of addition packet loss compared with the maximum power algorithm. Is the same true of a scenario in which the devices are arranged randomly?

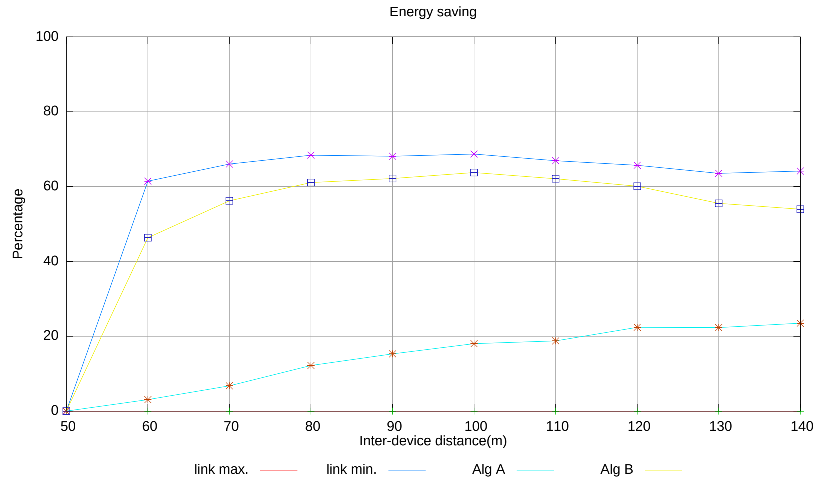
6.5 Random flows in a random area

In all previous experiments, we have had some degree of regularity in the placement of devices and/or sources and destinations of data flows. In this experiment, we remove these limitations and randomly place nodes in a square area of *x* by *x* meters.

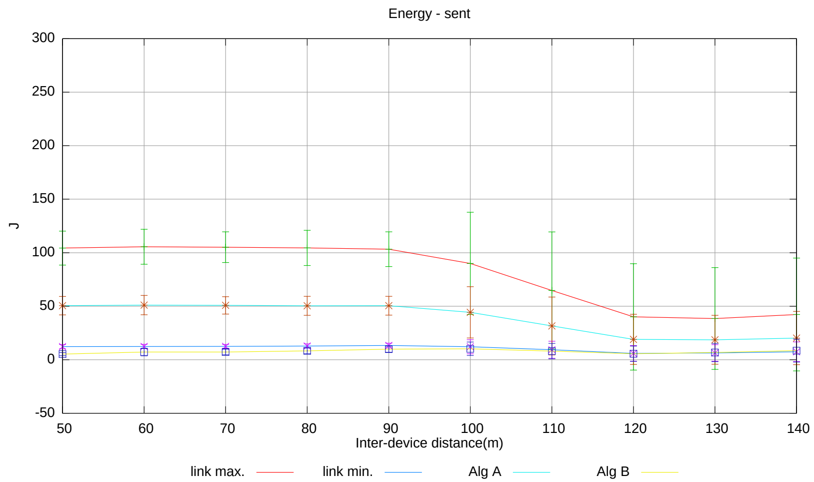
To maintain parity with the previous experiments we calculate *x* to be, $x = (\sqrt{n} - 1) * d_i$, where d_i is the distance increment for each experiment. For the 802.11 model d_i, is incremented from 50m to 170m in 10m steps, for the Philips model we increment d_i from 15m to 45m in 2m steps. Each node is randomly placed in the square area at a point (x_n, y_n) where x_n and y_n are independently determined. We continue to use the same model for the 20 random flows as was used in the previous experiment.



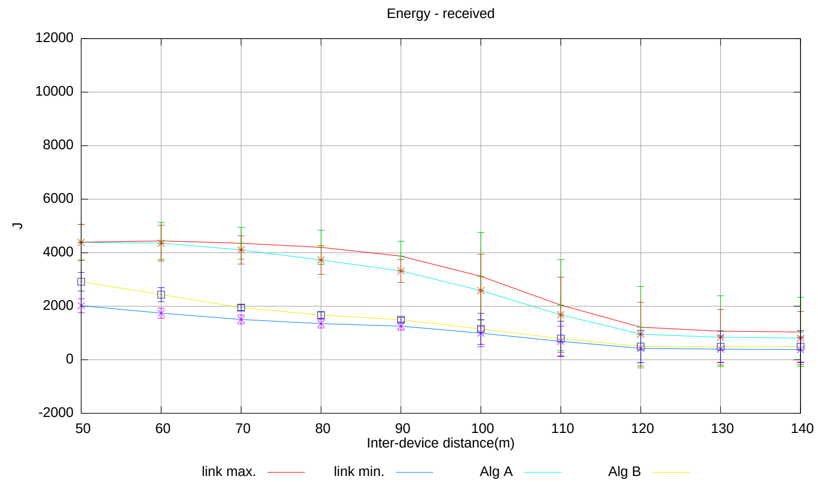
(a) C_{UB}



(b) Energy saving %

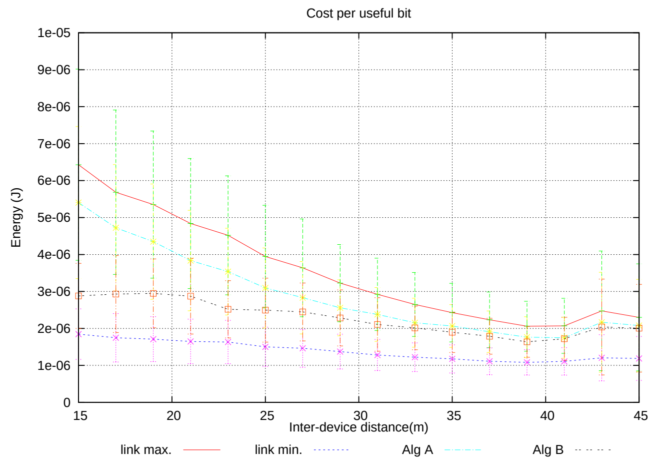


(c) E_{tx}

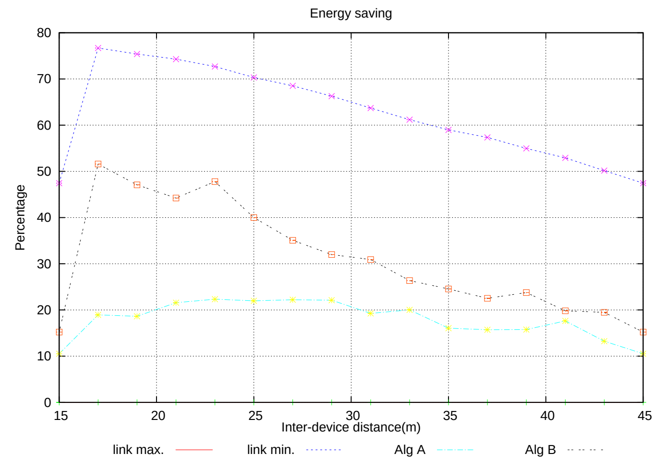


(d) E_{rx}

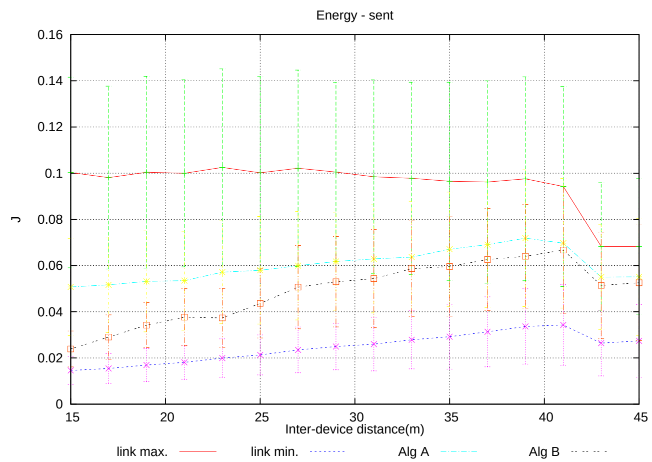
Figure 6.13: *20 random flows in a random small area scenario (802.11,UDP)*



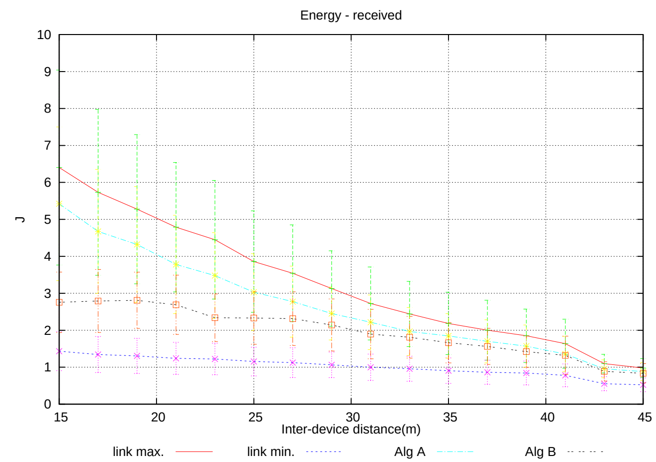
(a) C_{UB}



(b) Energy saving %

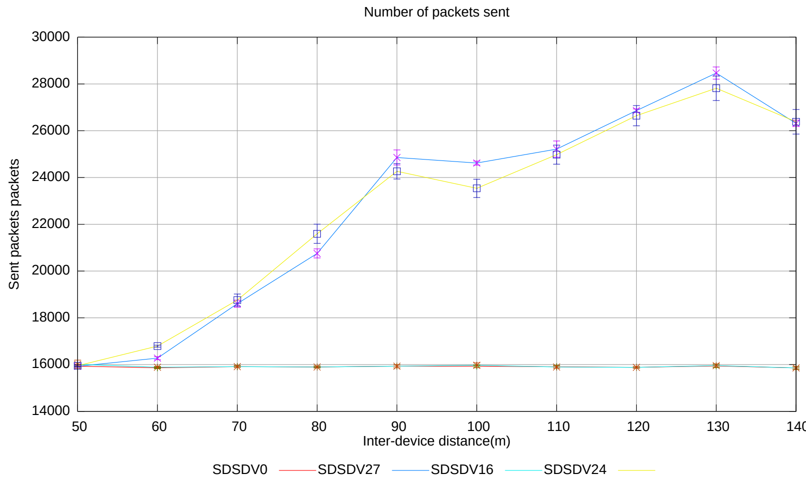


(c) E_{tx}

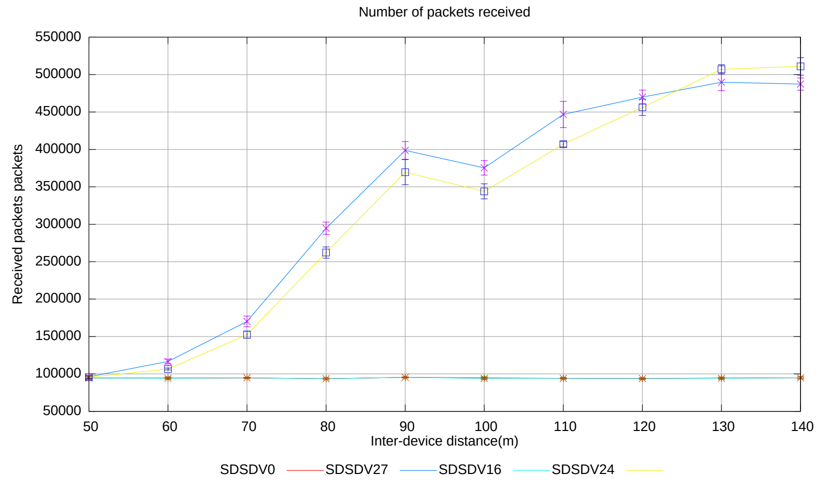


(d) E_{rx}

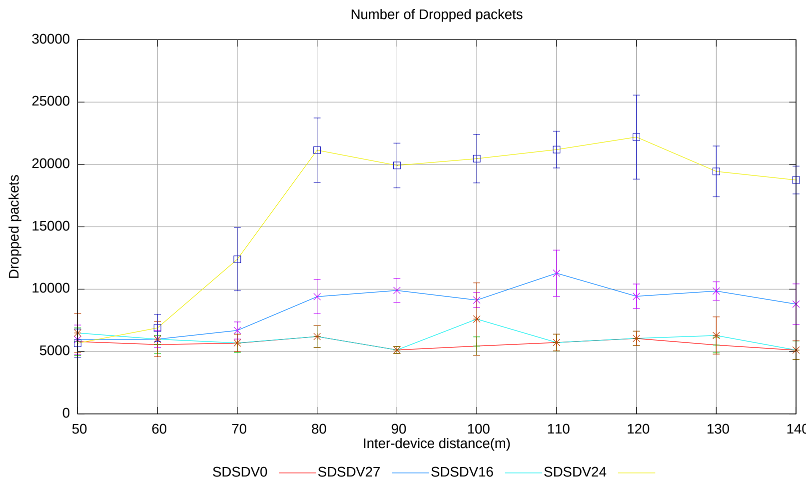
Figure 6.14: *20 random flows in a random small area scenario (Philips,UDP)*



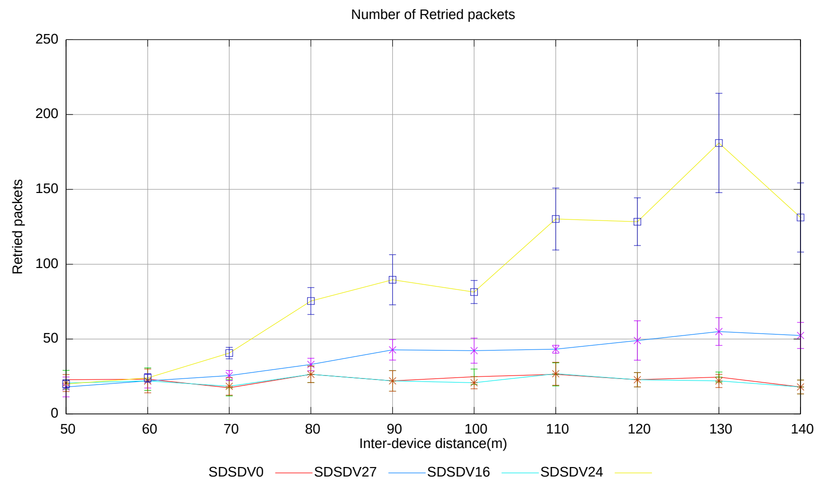
(a) Sent - 802.11



(b) Received - 802.11

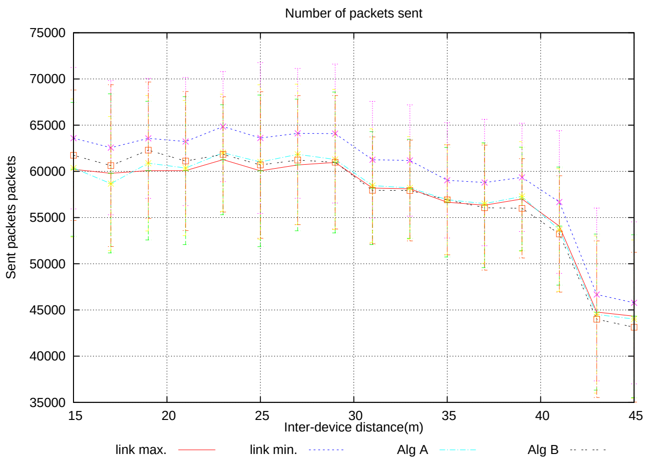


(c) Dropped - 802.11

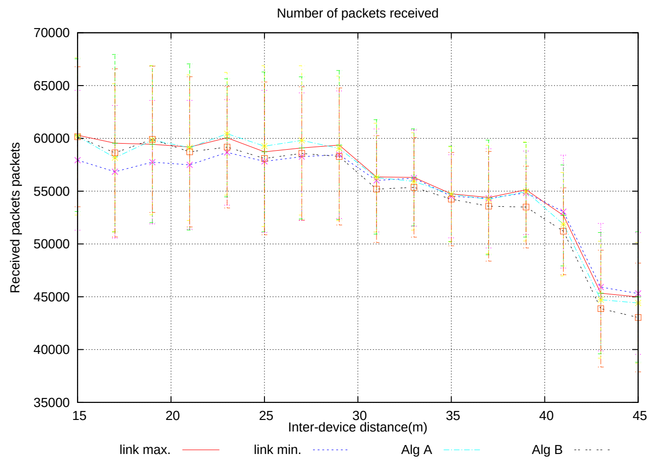


(d) Retried - 802.11

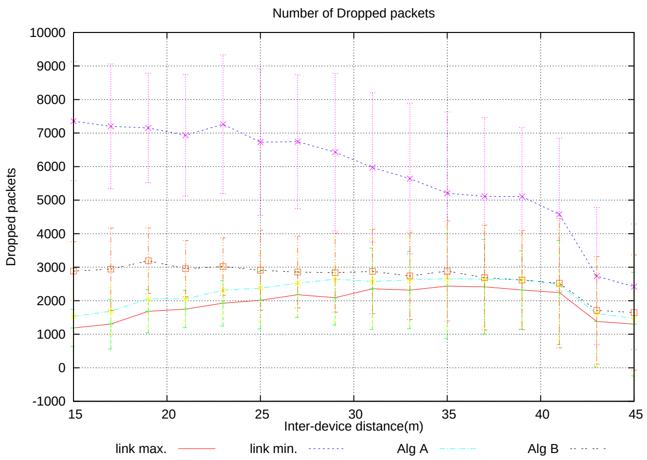
Figure 6.15: *Number MAC level packet events of 20 random flows in a random small area scenario (802.11,UDP)*



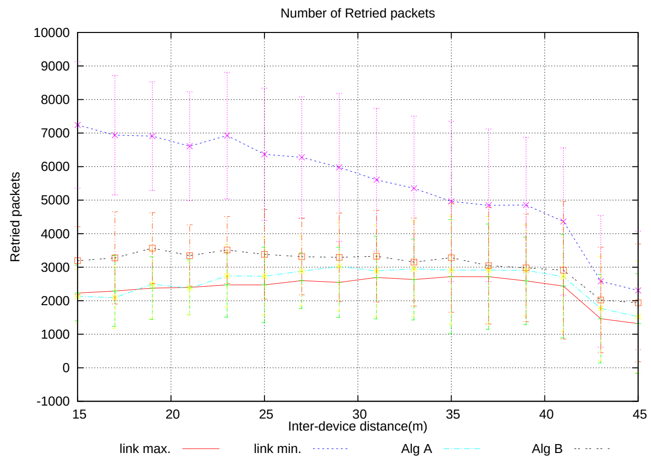
(a) Sent - Philips



(b) Received - Philips



(c) Dropped - Philips



(d) Retried

Figure 6.16: *Number MAC level packet events of 20 random flows in a random small area scenario (Philips,UDP)*

Transitioning from a structured layout of devices in the previous experiment to one in which the devices are randomly distributed in the same area causes a rise in the C_{UB}, figures 6.13(a),6.16(a), across all the algorithms by approximately 3x times. As this affects the maximum power algorithm in the same way as all the other algorithms, we can attribute the rise in cost to the change in the physical node locations rather than the algorithms. Removing the structured layout of the system has re-enforced the conclusions of the previous experiment that concluded that the over optimising is counter-productive and that having less tuned power control optimisation algorithms is beneficial in terms of packet throughput, figures E.1(a),E.1(c) and that you can achieve energy savings similar to those of the approximate per-link lower bound power algorithm but with the reliability of the maximum power algorithm, figures 6.13(b),6.14(b).

6.6 Conclusion

In this chapter we have extended the earlier analyses introducing firstly a regular 2-dimensional array of devices and then a random arrangement of devices and observed two key findings. Firstly, as the system density increases so the cost of reception becomes the dominant factor suggesting only transmitting as far as necessary is hugely beneficial because you can trade P_i for P_{rx} for those nodes that are not over hearing the transmission. Whilst we have set P_i to zero in these experiments a saving is still achievable since in table 5.1 we show that P_i is 89% of P_{rx} for the 802.11 model and 51% for the Philips model. Secondly, we show that over optimising a system can lead to instabilities and that less finely tuned algorithms can often perform better in more realistic scenarios and we see that here with algorithms A and B performing nearly as well as the lower bound in terms of energy usage yet in terms of reliability they are more in line with the upper bound.

Chapter 7

The influence of TCP on power saving

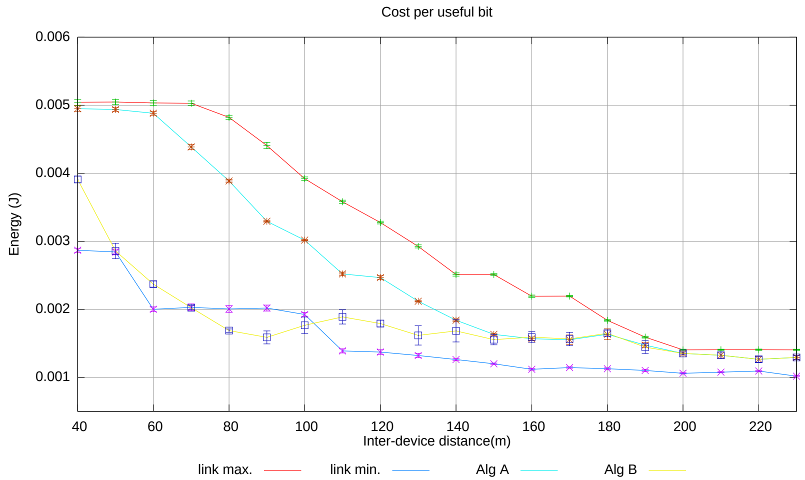
One possible solution to an unreliable lower layer is to introduce reliability at an upper layer. In this section we examine whether TCP provides the additional reliability desired at a cost that is reasonable. In earlier experiments we used only UDP based traffic flows, TCP adds reliability and congestion control at the cost of additional packets. Reliability will be measure in terms of the number of application level packets successfully delivered, whilst the cost will be measured in terms of the C_{UB}.

Additional packets have been shown in our earlier experiments to increase the ability of algorithms A and B to effectively tune towards the lower bound. We have observed that adjusting the transmission power in some circumstances can lead to an increase in the number of packets dropped at the MAC layer. Using a reliable transport layer is one approach to shielding the application layer from unreliable lower layers.

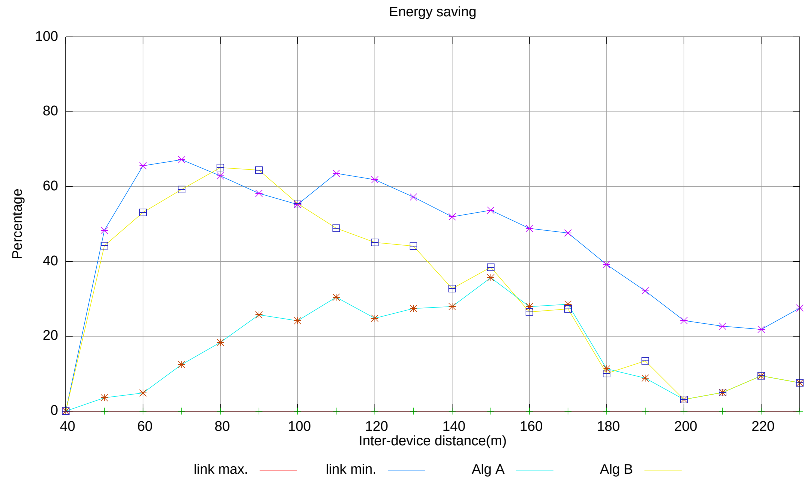
As per the previous UDP experiments we will slowly examine TCP under different scenarios. Since TCP is a stream-based protocol, it is not reasonable to start with only sending a single packet as we did in the UDP scheme. Many of the lessons learned in the previous section still apply to TCP so we limit ourselves to just looking at two scenarios. To understand the changes TCP introduces to the system we compare the experiments where we run eight competing flows over the standard 7 by 7 matrix of nodes, figure 6.1, and 20 random flows in an area with randomly distributed nodes, with their UDP counter parts from the previous chapter. To simplify the analysis, we limit ourselves to the 802.11 model. We use the NS default TCP implementation (Tahoe) for testing because we only need the generic TCP concepts and not specific metrics or features. Unfortunately to make the simulations tractable in this more complex environment we have had to reduce the range of distances over which the simulations are run. For the structured 7 x 7 arrangement of devices, the inter-device distances are increased in 10 m steps from 40m to 230m. For the random small area example from section 6.5, x is increased from 50m to 140m in 10 m increments. At each distance increment each algorithm is run for five iterations using different random seeds.

7.1 Eight non-competing TCP flows

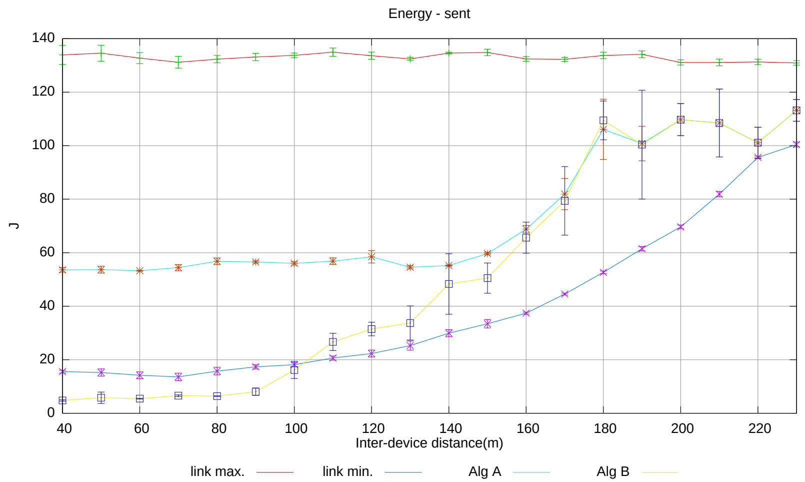
We saw in the UDP scenario that adding an additional 7 packet flows into the experiment, figure 6.8, decreased the C_{UB}, figures 6.9 and 6.10, when compared with the single flow experiment, figures 6.5 and 6.6. In this experiment we would expect the C_{UB} to be reduced further, algorithms A and B to align themselves closer with the approximate lower bound per link power algorithm and for TCP to compensate for any lost packets through re-transmission. This will be at the cost of an increase in the total energy consumption. The traffic flows are of 100 hundred packets and are sequentially started at intervals of 10 seconds.



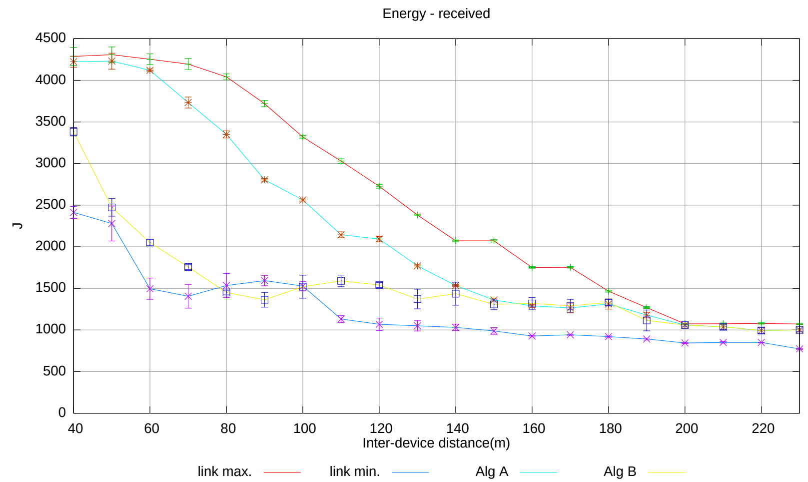
(a) C_{UB} - TCP



(b) Energy saving % - TCP

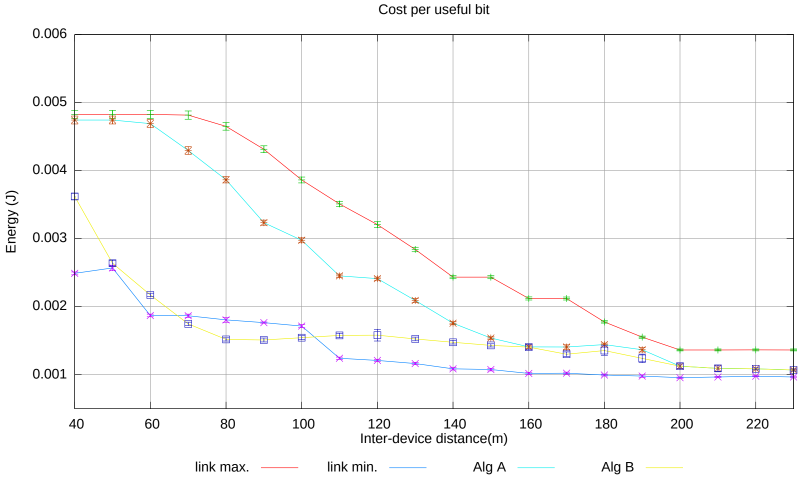


(c) E_{tx} - TCP

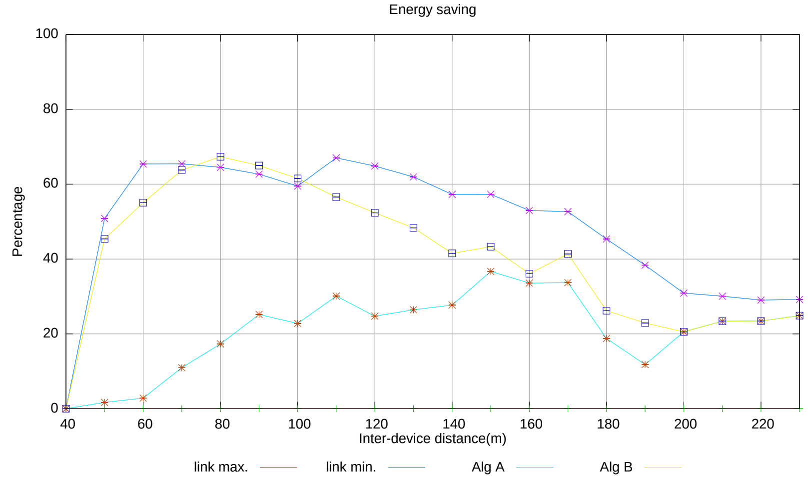


(d) E_{rx} - TCP

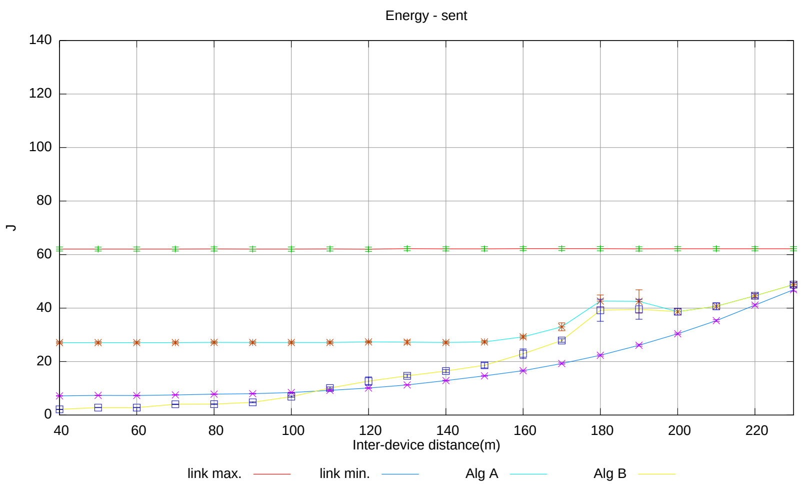
Figure 7.1: *TCP energy usage of 800 packets in a regular small area scenario*



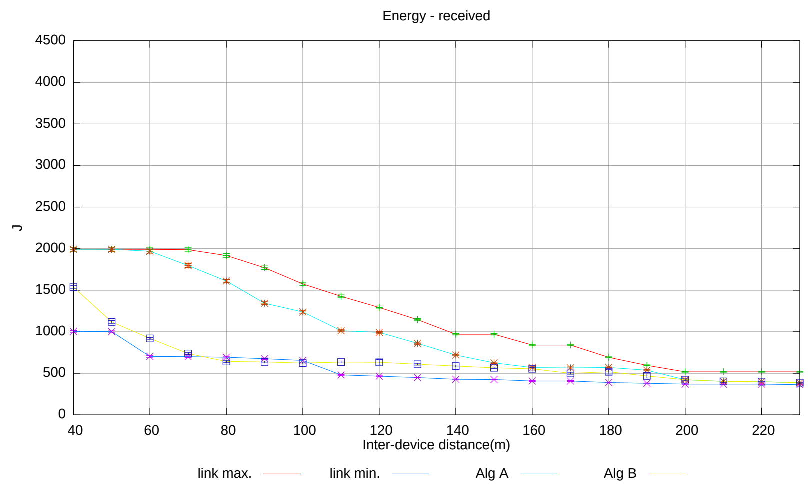
(a) C_{UB} - UDP



(b) Energy saving % - UDP



(c) E_{tx} - UDP



(d) E_{rx} - UDP

Figure 7.2: *UDP energy usage of 800 packets in a regular small area scenario*

TCP has two key features; reliable ordered delivery and congestion avoidance, achieving both of these has the net result that TCP introduces more packets into the network than a corresponding UDP transmission. This decreases the ratio of data traffic successfully received by the application to the combination of control traffic and lost traffic, and this what we can see when comparing the TCP and UDP graphs in figures 7.2, 6.9. We see that both the E_{rx} and E_{tx} are both increased by approximately 100 %, as is the C_{UB} but to a much lesser extent. The slight increase in the C_{UB} despite the 100 % increase in the overall energy usage levels is a result of more application packets reaching the destination, bring down what otherwise would be a significant increase in the C_{UB}.

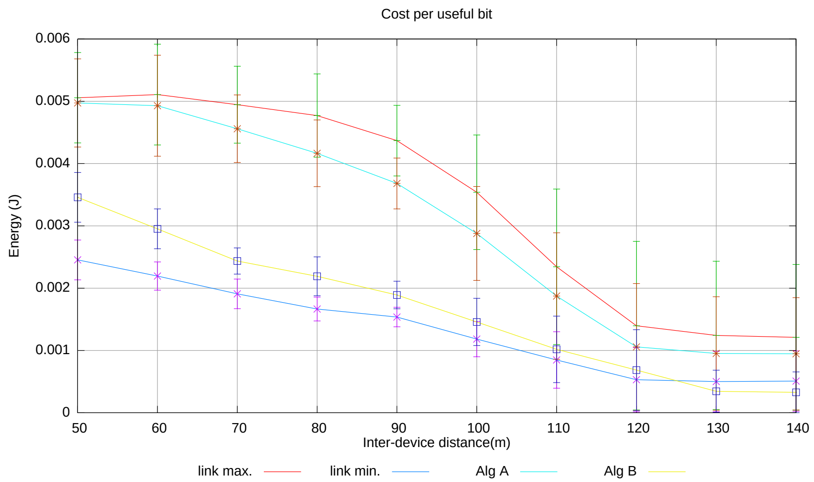
The increase in traffic caused by TCP can clearly be seen when comparing the number of packets sent and received at the application layer, figure F.2 (appendix). In all the algorithms, including the unmodified maximum, the number of packet sent and received has at least doubled. The approximate lower bound algorithm exhibits an even greater increase than the other algorithms. We re-iterate our argument from before that this is a result of over optimisation and that the less highly optimised alternatives are performing nearly as well as the maximum power algorithm whilst showing a saving in terms of energy consumption. These application level packet events are borne out in the MAC layer statistics too, so we don't show these for the sake of brevity.

When we consider the causes of packet drops at the MAC layer, figures F.3 and F.4 (appendix), we see a marked increase for TCP in the number of packets dropped due to collisions for all the algorithms.

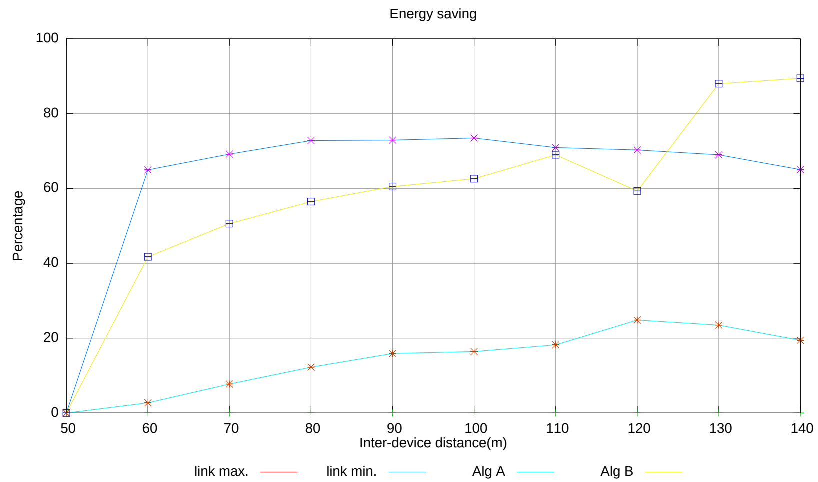
This leads to the conclusion that reliability and congestion control semantics are viable in a low power radio system, but do come at an increased absolute energy cost which is important when a mobile device has a limited amount of energy. However, the reliability brings about only a slight increase per correctly delivered application level packet.

7.2 Twenty random TCP flows in a random area

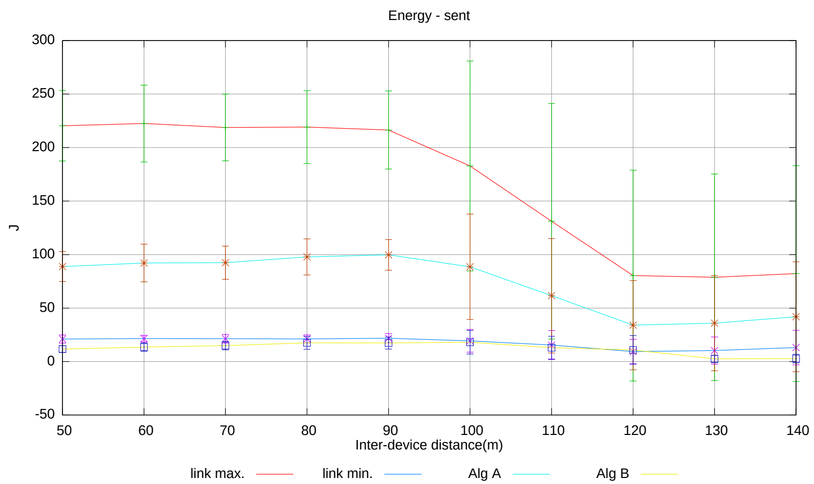
We repeat the earlier experiment from section 6.5, this time with TCP and compare it with the previous UDP version with the axes of the graph scaled to the TCP settings.



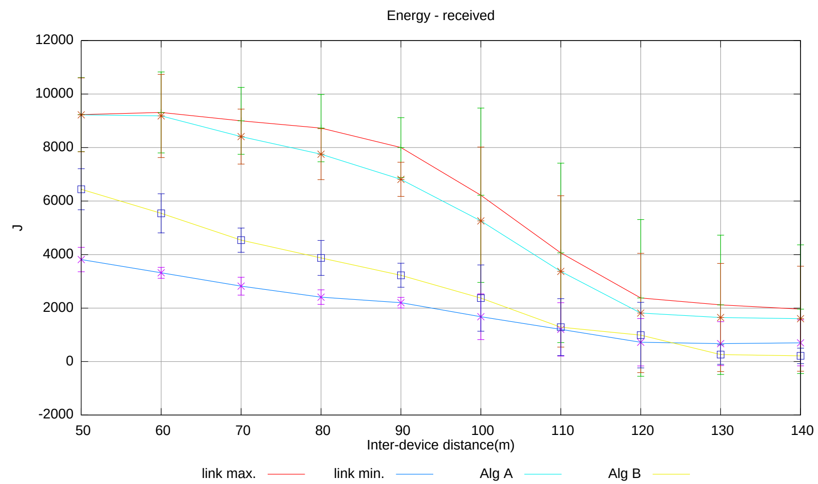
(a) C_{UB}



(b) Energy saving %

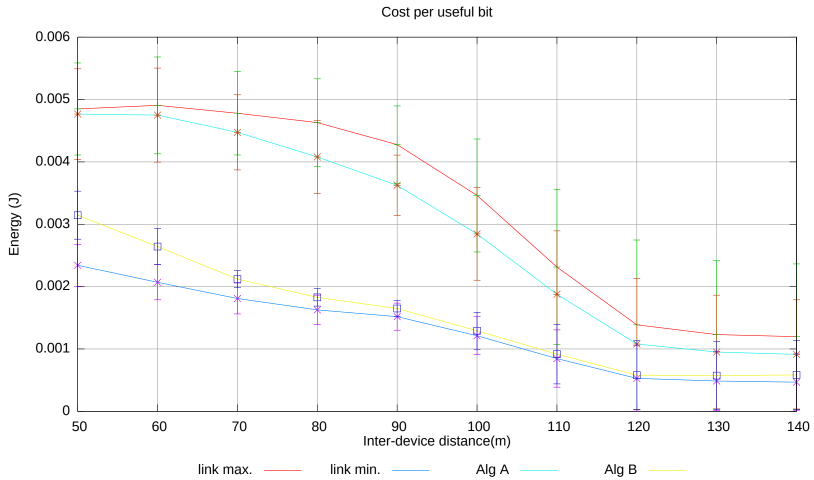


(c) E_{tx}

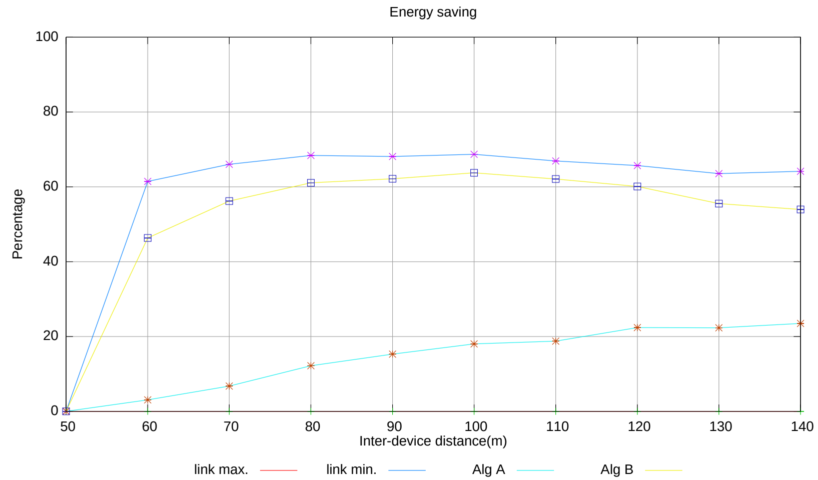


(d) E_{rx}

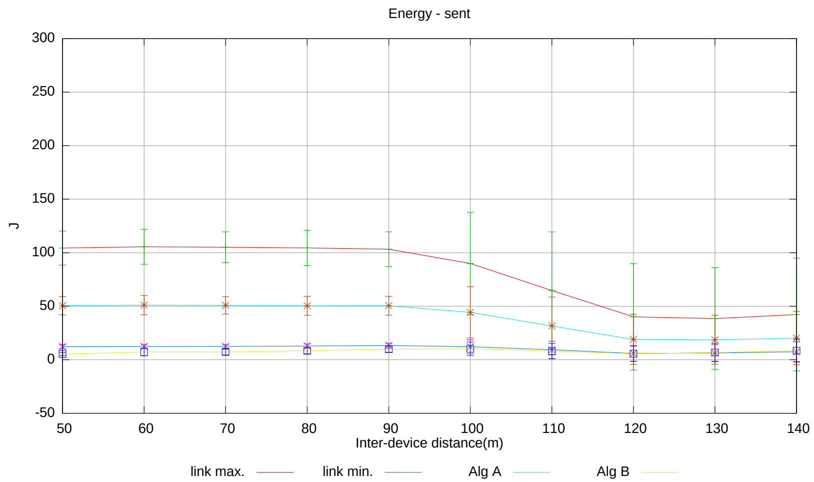
Figure 7.3: *Energy usage for 20 TCP flows in a random small area scenario*



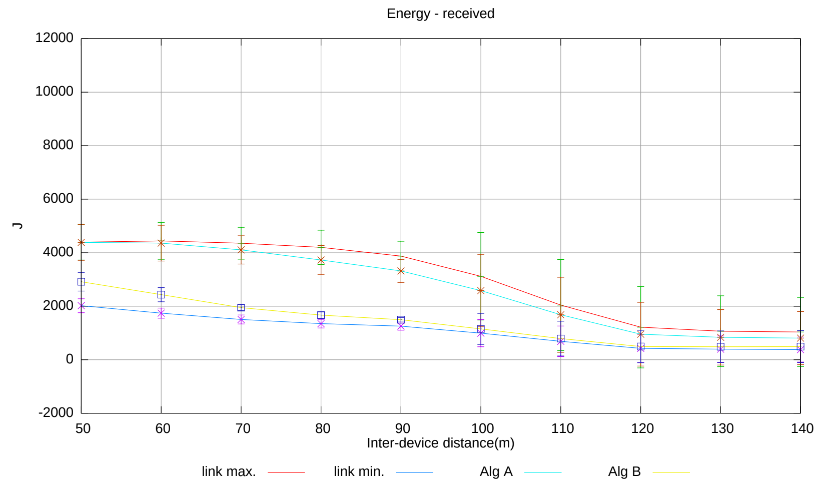
(a) C_{UB}



(b) Energy saving %



(c) E_{tx}



(d) E_{rx}

Figure 7.4: *Energy usage for 20 UDP flows in a random small area scenario*

If we consider the previous section and the earlier UDP experiments, it does not come as a surprise that we observe the same trends that TCP in a random area with random traffic is able to benefit from overheard information for only a small increase in the C_{UB}, but at the cost of an overall doubling of the total energy used. This can be seen in the graphs presented in figure **??**, the application and MAC layer statistics for this experiment can be seen in appendix G.

7.3 Conclusion

We have shown that the use of TCP comes as a double edged sword; increased application reliability and congestion control comes at a minor increase in the C_{UB}, but also at a very significant doubling of the overall energy usage of the system due to the extra packets sent in comparison to UDP. It is therefore a system level design issue as to whether to use TCP or UDP for packet transmissions, there is no cut and dry answer for all systems.

Chapter 8

Routing Protocol

The previous chapter examined the effects of using observed information to modify the transmission power of devices on a hop by hop basis. In this chapter, we replace the previously used static routing with a varient of the DSDV routing protocol that supplements the normal routing protocol messages with overheard information.

Combining routing information and transmission power control information is a powerful combination because it is feasible to adjust the power transmission level based upon a larger set of knowledge. For example consider the scenario in figure 8.1. Device A is unable to transmit directly to device E; it must route via another device, in this scenario either B, C or D.

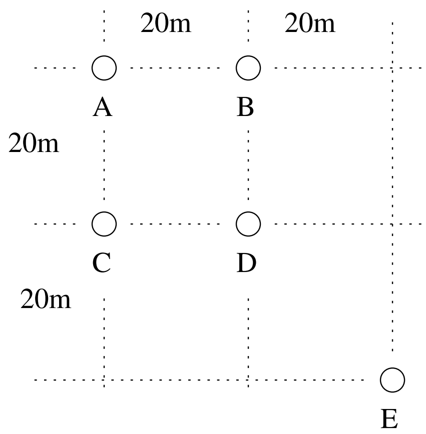


Figure 8.1: *Routing Scenario*

The scenario here is a simple example that is illustrative of the complexity of more general situations. Each device has the following radio characteristics :

- Transmission range is 50m (maximum)
- (P_i) Idle energy cost per message period[1] : 0.54 units

[1] The ratio of the different energy consumption figures is taken from [21]

- (P_r) Reception energy cost per message period : 0.67 units
- (P_t) Transmission energy cost per message period at 50 m : 1 unit
- Propagation model is $P = d^4$ and gives the P_{tx} at the following distances as :
  - 20m : 0.026 units
  - 28.3m : 0.103 units
  - 44.7m : 0.639 units

So the total cost of the transmission of a packet from A to E, via the three paths, are as follows:

Path (1) : $A \rightarrow D \rightarrow E$
$P_{ADE} = P_{t_{AD}} + P_{r_{C,B,D}} + P_{t_{D,E}} + P_{r_{A,B,C,E}} + P_{i_E}$
$P_{ADE} = 0.103 + 3 * 0.67 + 0.103 + 4 * 0.67 + 0.54$
$P_{ADE} = 5.436$ units $(P_t = 0.206, P_r = 4.69, P_i = 0.54)$
Path (2) : $A \rightarrow C \rightarrow E$
$P_{ACE} = P_{t_{AC}} + P_{r_{C,B}} + P_{t_{C,E}} + P_{r_{A,B,C,E}} + P_{i_{D,E}}$
$P_{ACE} = 0.026 + 2 * 0.67 + 0.0.639 + 4 * 0.67 + 2 * 0.54$
$P_{ACE} = 5.765$ units $(P_t = 0.665, P_r = 4.02, P_i = 1.08)$
Path (3) : $A \rightarrow C \rightarrow D \rightarrow E$
$P_{ACDE} = P_{t_{AC}} + P_{r_{C,B}} + P_{i_{D,E}} + P_{t_{C,D}} + P_{r_{A,C,D}} + P_{i_{B,E}} + P_{t_{DE}} + P_{r_{A,C,B,E}}$
$P_{ACDE} = 0.026 + 2 * 0.67 + 2 * 0.54 + 0.026 + 3 * 0.67 + 2 * 0.54 + 0.103 + 4 * 0.67$
$P_{ACDE} = 8.345$ units $(P_t = 0.155, P_r = 6.03, P_i = 2.16)$

If we consider the overall energy consumption of the different routing scenarios it is obvious that the longer path, $A \rightarrow C \rightarrow D \rightarrow E$ is least efficient. The picture is less clear with the remaining two paths. Whilst Path 1 uses less energy than Path 2, if we assume that all devices in the scheme would be idle anyway and it is the variation from the idle state that is important now Path 1 uses less energy than Path 2. This is important because, whilst it would seem from the analysis presented in section 4.1.1 that multiple hops, favouring Path 3, would be beneficial but since the cost of reception that dominates the overall cost of a packet traversing the network. This leads to the situation in which we need to minimise the number of hops between a source and a destination but we need to consider the affect of a transmission on the area the transmission affects. The area a transmission effects is $\pi(\sqrt[v]{P_t})^2$, assuming a circular transmission range[2], where v is the Path Loss Exponent [72]. All devices in this area hear the transmission and hence incur the cost P_r. Logically, as the node density increases around a device transmitting, the term $\sum P_r$ dominates P, the total power consumed during a transmission. Thus, reducing the P_t may reduce the number of neighbours and hence would reduce $\sum P_r$. A situation will exist in which taking a path with more hops that are closer together will reduce $\sum P_r$ even though the number of hops is higher, because the total number of nodes overhearing the transmission is lower.

An additional issue is the utilisation of the channel, in general each transmission occupies the radio channel within its transmission range for the period of the transmission, any transmission by another device within the same area will cause the transmission to be corrupted, i.e. a collision occurs, MAC protocols are designed to eliminate this problem through a series of handshake mechanisms. Each time the packet is relayed and transmitted by another device, the channel is occupied within the transmission range of the relaying device. As the number of hops in the transmission path increases so the total channel occupancy for the transmission also increases, equation 8.3. To explain further where equation 8.3 comes from if we assume that each radio transmission affects a circular area.

$$\pi.r^2 \tag{8.1}$$

If the radius of the circle is given by the transmission power then.

$$r = \sqrt[v]{P_t} \tag{8.2}$$

Where v is the path loss exponent. If a transmission is of N steps and at each step the transmission occupies an area for the duration of the transmission, t. The entire transmission occupies the channel :

$$C_{occ} = \sum_{n=0}^{n=N} t_n.\pi(\sqrt[v]{P_{t_n}})^2 \tag{8.3}$$

Where n is the current hop, N is the total number of hops, P_{t_n} is the transmission power P_t at hop n, C_{occ} is a measure of the area and time the channel is occupied for a packet transmission. Whilst C_{occ} is not a physical measurement it provides a qualitative measure to compare different transmission paths and transmission power levels. C_{occ} is the time the channel is

[2] Radio transmission ranges are irregular and constantly fluctuate with environmental conditions, antenna etc. Without knowing the exact conditions it is impossible to make any other assumption, other than make no assumption.

occupied for a packet transmission.

So a routing protocol needs to consider firstly the route between a source and a destination and primarily to choose the route with the least number of hops between the source and destination, hence reducing $\sum P_r$ and C_{occ}. If there is more than one route with the same number of hops minimising the route with the minimum $\sum P_t$ is the preferred route, since this reduces $\sum P_r$, C_{occ} and $\sum P_t$. However, through an area of high node density then increasing the number of small hops could be beneficial.

It would be a computationally expensive problem to calculate the routes and transmission powers to minimise the global energy consumption of the network given arbitrary packet flows. To do this you would need to know the location of all the devices, the range of their transmissions given a certain transmission power, and the traffic patterns of all traffic. Whilst it is feasible to do this globally, it is infeasible to provide each device with this information such that they could make the decisions to achieve a global minimum, since the propagation of the information would form a significant portion of the network's traffic such as to make any savings achieved by using the information void. If we introduced the extra consideration of movement to the scenario, the problem would become more complex still.

What we propose here is to supplement the routing protocol with the information a device can learn by passively listening to the channel. Listening to the channel will provide a device with information about the location of its neighbours in terms of radio range and hence the density of devices, the volume of traffic passing by the device. We have chosen to supplement DSDV with this information, we call our modified version, Supplemented DSDV, SDSDV. DSDV is a table based proactive routing protocol that periodically broadcasts routing information to its neighbours. These broadcasts are used to update internal state information held in its neighbours' internal routing tables.

We have chosen to use DSDV because of its simplicity. We could have chosen to use routing protocols such as AODV, DSR or ZRP but the additional complexity of these protocols would obfuscate the information we are interested in, namely the affect of supplementing routing tables with information about a device's surroundings.

We examine two sets of scenarios, firstly the effect of supplementing the routing tables in contrived scenarios where we investigate the affect of node density and a second scenario where we examine the new protocol in a random placement scenario.

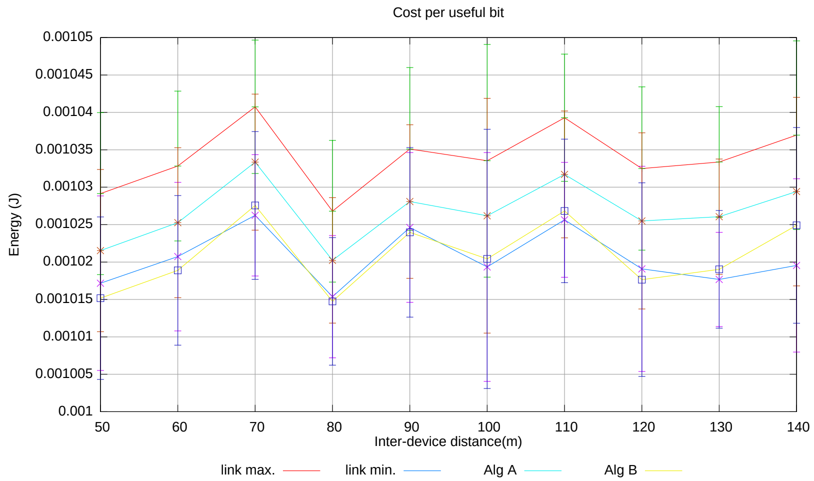
8.1 Random flows in a structured area

In this first scenario we compare SDSDV with DSDV using 20 random flows of 100 packets between randomly selected source and destination nodes taken from the now familiar 7 by 7 arrangement of 49 devices. If we examine the energy characteristics of the two routing protocols, as shown in figure H.1, it is clear from both the C_{UB} and the percentage energy saving graphs that SDSDV performs worse than standard DSDV and that using the per node transmission power as the link weight metric is the wrong approach. We see that for both the upper and lower bound algorithms, where the per node transmission power is stable, that both algorithms have approximately the same profiles for energy transmission. However algorithms A and B where we have introduced a level of dynamism to the routing metric weights through the use of the changing per node transmission power we see they are both performing worse in SDSDV than DSDV.

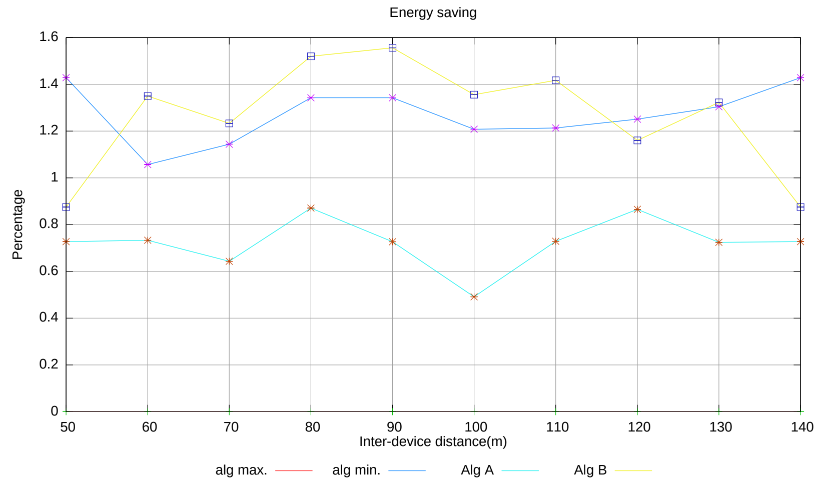
8.2 Random flows in a random area

In this experiment we remove the structured layout of devices from the experiment and replacement with a random arrangement of devices, see section 6.5 for details of the placement.

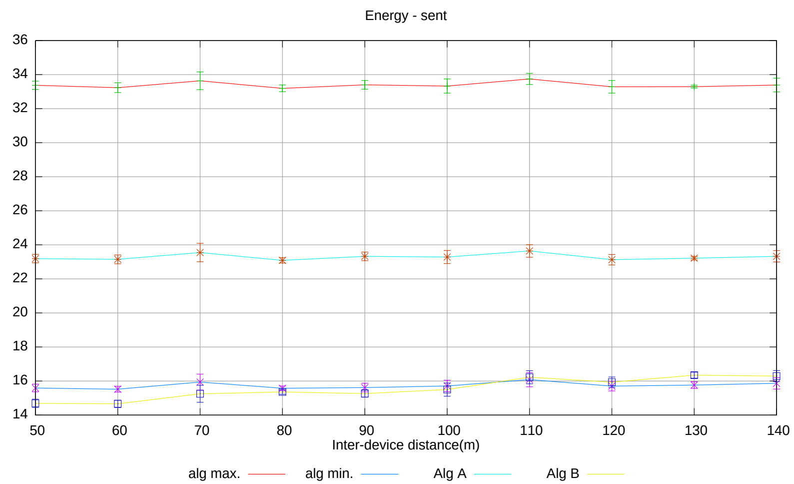
It should not be unexpected to find that removing the structured layout does not improve the performance of SDSDV when compared to DSDV and that neither algorithms A nor B are performing as well as the lower or upper bound algorithms and this can be clearly seen in figure I.1. This result re-enforces the result from the previous section.



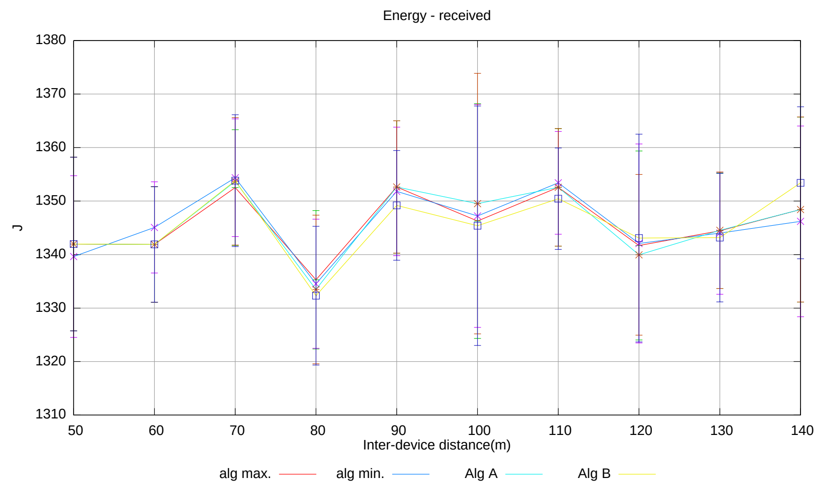
(a) C_{UB} - DSDV



(b) Energy saving $\%$ - DSDV



(c) E_x - DSDV



(d) E_{rx} - DSDV

Figure 8.2: *Power control with DSDV*

However if we now just consider DSDV as we have shown that SDSDV is not the correct approach when the link weight metric changes dynamically. In figure 8.2(d) we see that both algorithms A and B are able to achieve a modest saving compared with the upper bound and in the case of Algorithm B, the lower bound too. The broadcast nature of DSDV means that the power control scheme we have presented is not effective because it does not act upon broadcast traffic which is utilised by DSDV to transmit routes. The small differences observed between the different algorithms results from the algorithms working on the data packets.

8.3 Conclusion

In this chapter we have shown that SDSDV is not an effective mechanism for coupling the learned transmission power based information we exploit with a routing protocol. Using a link weight that is highly dynamic such as the calculated value for the transmission power is a receipt for instability when coupled with a dynamic routing system. Dampening the transmission power control transitions may lead to more stable system but that would be further work.

Using SDSDV we have highlighted the need to match your power control scheme with your routing protocol to ensure they are compatible. For example DSDV, and hence SDSDV, is reliant upon broadcast traffic which is explicitly not power controlled in our scheme.

Chapter 9

Conclusion

In this thesis we have explored the feasibility of implementing a simple decentralised transmission power control scheme for low power ad hoc networks. We ask the question that if the transmission power required to transmit a distance x is approximately given by $P \approx x^n$, what can we do to reduce x given that it is the dominant factor[1]? We present two solutions to reducing x. Firstly, reduce x such that you transmit only as far as necessary rather than at full power, thereby reducing the transmission power used. Secondly, if it is possible to relay the transmission, we can break down into smaller pieces since $x^n > \sum_1^j x_j^n$ where $x = \sum_1^j x_j$. Neither of these schemes is as simple as it seems because we ignore the cost of reception and the cost of the radio being on but idle. We have explored these schemes in the context of the following limitations.

- The devices in the network are considered to be stationary, or have minor variations in their relative positions when moving as a cohort.
- The path loss is symmetric between any pair of devices.
- The antennas are vertical and omnidirectional.

The first scheme, leads to the question: how is the reduction in x calculated? We solved this by introducing a simple quantisation of the transmitted power of the packet and included it in the packet header as an 8 bit value. Exploiting the radio's ability to determine the received power strength for each transmission, it is possible to guess a reduced (or increased) transmission power at which to transmit back to the originating device. This has the advantage of not needing to know the actual distance between the devices and takes into account the channel conditions in a completely decentralised way. We developed two simple algorithms using different sets of threshold values to calculate the transmission values. We examined both of these algorithms

[1] n is the Path Loss Exponent and is based on the environmental conditions.

along with the two that calculate the upper and an approximate lower bound under a variety of scenarios. We have shown that using such a simple power control scheme such as this is feasible and can produce savings in both the simple case and the more complex cases that we have been evaluating. We have observed that as the complexity of the scenarios has increased, algorithm B has tended towards the approximate lower bound algorithm whilst maintaining the reliability of the upper bound algorithm. This demonstrates that using over tuned algorithms, such as the approximate lower bound algorithm, may result in system instability and that using a more generic algorithm might result in greater savings overall. This generalisation was true for TCP as well as UDP, but whilst TCP brought increased reliability and congestion control, it came at the cost of a doubling of overall the energy whilst only increasing slightly increasing the C_{UB}. This leaves the system architect with a tussle to trade off reliability and congestion control for overall energy usage. There is no generic resolution of this tussle and it must be resolved on a per system basis with consideration given to the power resources available.

A note of caution must be raised here, since we have made two assumptions about the underlying system that can and do influence the radio model's behavior. Firstly, we have assumed that the antennas are all vertical and omni-directional. Directed or steerable antennas could be used to focus the signal where it would be most beneficial, reducing the possibility of packet collisions. Secondly, we have assumed that the channel's behaviour is symmetric and this is present as an assumption in the learning algorithm's design. Its not obvious what influence non-symmetric channel conditions would have on the results, and this would form further work.

We briefly explored the impact of combining the power control algorithms with a routing protocol by modifying the routing protocol DSDV. We replaced the link weight with the transmission power level required to transmit between the neighbours, thus creating Supplemented-DSDV (SDSDV). Two issues arose out of this investigation. Firstly, DSDV's dependence upon broadcast packet's limited the effectiveness of the power control schemes. Secondly, coupling a dynamic variable such as the per link power transmission value with a dynamic routing protocol lead to instability and inefficiently. The conclusion we draw from this inter-layer dependency is that we have to be carefully matched to ensure the system is stable.

The second scheme, is limited by the cost of reception because even though the cost of transmission has been reduced by using relaying, the summed cost of reception becomes the dominant factor. This leads to the conclusion that the transmission power should be reduced in all but the most dense networks such that it just reaches its destination in a single hop thus reducing the summed cost of reception.

In this thesis we briefly investigated routing and did not investigate the effect of node movement upon the transmission power control algorithms. Further work is required to understand the implications of these scenarios upon the savings achieved by the transmission power control algorithms. Whilst we have shown these schemes are effective in the static case and make a promising addition to the MAC protocol, the dynamism of node movement coupled with routing needs to be investigated.

Throughout this thesis it is the cost of reception that has been the dominant factor. Thus if you have received the packet, you should attempt to make use of the data and this conclusion applies across all low power radio systems. In summary, we have shown the hypothesis, *It is more efficient in terms of energy consumption to constrain the transmission power based upon a combination of received signal strength with a minimally extended MAC, than to utilise an unchanged MAC and full power*, to generally hold in the static low power scenarios we have explored. In essence *if you heard it, use it !*

Appendix A

Power control algorithms API

A.1 Methods API

We implemented an API and the following methods in C++ in the NS simulator and we outline the algorithms in pseudo-code.

Three events trigger the algorithms, they are; receiving a packet, sending a packet and detecting a dropped packet. Upon receiving any packet, not necessarily intended for the host, the *Guess* function is called to estimate what transmission power level would be required to transmit back to the originating host. This estimated value is stored in a table keyed by the originating host's MAC address. When a device transmits to another device, it calls the *Lookup* function to lookup in the table created by the *Guess* function the estimated transmission level required to reach the destination. If the MAC layer detects a packet is dropped it uses the *Dropped Packet* function to increase the estimated transmission level required to reach the destination. The algorithm's API is as follows:

- void Dropped_Packet(Source_MAC src)
- int Lookup_Power(Packet p)
- int Guess(Receiver_signal_level rsl, Transmission_signal_level tsl, Receiver_threshold rt)

Appendix B

The algorithms

Below we describe and outline the pseudo-code for the four power control algorithms used in this chapter. Where appropriate we include any tuning parameters used with the algorithm.

B.0.1 Naïve method - per link maximum

This is the standard approach taken by wireless devices. It is a worst case scenario in terms of the magnitude of each transmission. Each packet is transmitted at full power, 255 in the quantised representation.

B.0.2 Naïve method - approximate per link lower bound

The algorithm approximates to a lower bound for the energy consumption of the system by calculating an approximate lower bound for each link. The transmission power required to transmit only as far as the next hop is calculated by using a version of the Shadowing model that provides an inverse reception power function, given the receiver's threshold level, the channel conditions and the distance between the devices returns the required transmission power level. The transmission power is quantised to the nearest $1/255^{th}$ above the required level, to enable it to be represented in the packet. This introduces an error of up to $1/255^{th}$.

B.0.3 Algorithm A

Algorithm A utilises a single level of estimation of the power to transmit back to a source. This algorithm will converge slowly upon a finally value.

```
C1 = 20
C0 = 1
CF = 4

T1 = 0.000000001

received_packet( Packet p ):
  if ((p.recieved_level - RECEIVED_THRESHOLD) > T1)
    change = C1
  else
    change = C0
```

```
  source_power = lookup_power ( p.source )

  if ((source_power - change) < TX_MIN )
     store_power( p.source , TX_MIN )
  else
     store_power( p.source , source_power - change )

send_packet( Packet p ):
  power = lookup_power ( p.destination )
  set_power ( p , power )

drop_packet( Packet p ):
  power = get_power ( p )

  if ( (power + CF) > TX_MAX )
    store_power ( p.destination , TX_MAX )
  else
    store_power ( p.destination , power + CF )
```

B.0.4 Algorithm B

Algorithm B utilises a four levels of estimation of the power to transmit back to a source. This algorithm will converge quickly upon a finally value, but might over shoot more often than algorithm A.

```
C4 = 100
C3 = 50
C2 = 20
C1 = 10
C0 = 1
CF = 4

T4 = 0.00000005
T3 = 0.00000001
T2 = 0.000000005
T1 = 0.000000001

received_packet( Packet p ):
  if ((p.recieved_level - RECEIVED_THRESHOLD) > T4)
    change = C4
  else if ((p.recieved_level - RECEIVED_THRESHOLD) > T3)
    change = C3
  else if ((p.recieved_level - RECEIVED_THRESHOLD) > T2)
    change = C2
  else if ((p.recieved_level - RECEIVED_THRESHOLD) > T1)
    change = C1
  else
    change = C0

  source_power = lookup_power ( p.source )

  if ((source_power - change) < TX_MIN )
     store_power( p.source , TX_MIN )
  else
     store_power( p.source , source_power - change )
```

```
send_packet( Packet p ):
  power = lookup_power ( p.destination )
  set_power ( p , power )

drop_packet( Packet p ):
  power = get_power ( p )

  if ( (power + CF) > TX_MAX )
    store_power ( p.destination , TX_MAX )
  else
    store_power ( p.destination , power + CF )
```

Appendix C

Shortest Path Routing Algorithm

Where A is a temporary matrix, C is the matrix of link costs, P is the next hop matrix and n is the matrix dimension.

```
def shortest(A,C,P,n):
        for i in range(0,n):
                for j in range(0,n):
                        A[i,j] = C[i,j]
                        P[i,j] = -1
        for i in range(0,n):
                A[i,i] = 0
        for k in range(0,n):
                for i in range(0,n):
                        for j in range(0,n):
                                if ((A[i,k]+A[k,j]) < A[i,j]):
                                        A[i,j] = A[i,k] + A[k,j]
                                        P[i,j] = k
```

The shortest path algorithm gives the last hop before the destination so to convert this to providing the next hop routing table we apply the following.

```
def next_hop(path,i,j):
        nh = path[i,j]
        new_dst = j
        while ( nh != -1 and nh != new_dst):
                new_dst = nh
                nh = path[i,new_dst]
        return new_dst
```

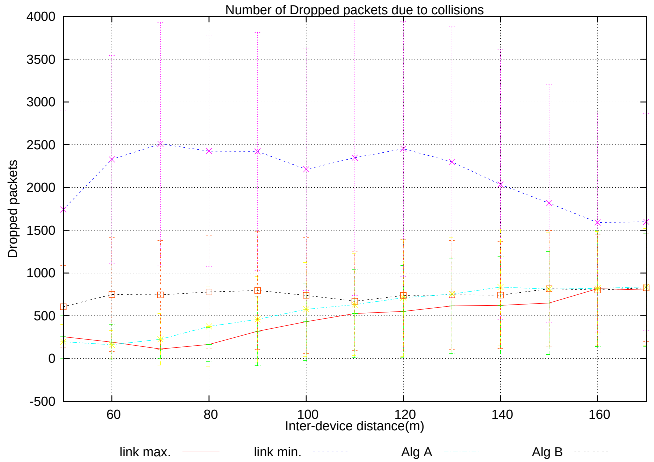
Putting it all together, we generate the routing table for all pairs of nodes as follows.

```
def write_short_route(p,n,f):
        o_matrix = zeros((n,n))
        power_matrix = zeros((n,n))
        path = zeros((n,n))

        for i in range(0,n):
                for j in range(0,n):

                        dist = distance(p[i,0],p[i,1],p[j,0],p[j,1])
                        if dist < 241:
```

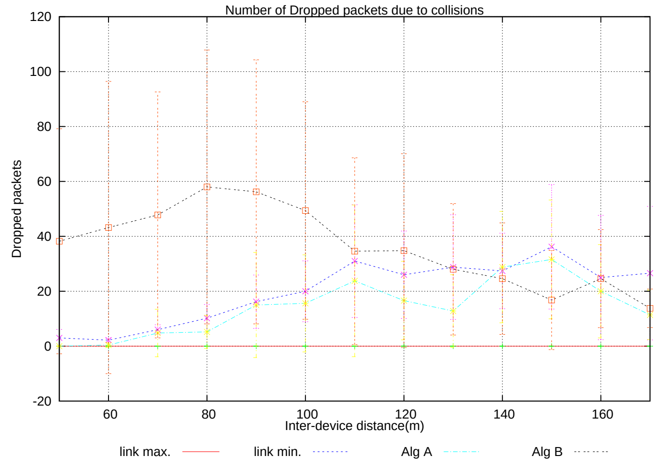
```
                                power_matrix[i,j] = dist ^ 4
                        else:
                                power_matrix[i,j] = Infinity

        shortest(o_matrix,power_matrix,path,n)

        for i in range(0,n):
                for j in range(0,n):
                        if i == j :
                                print "i j j"
                        else :
                                if ( path[i,j] != -1 ):
                                        print  "i j next_hop(path,i,j)"
                                else:
                                        print "i j j"
```

Appendix D

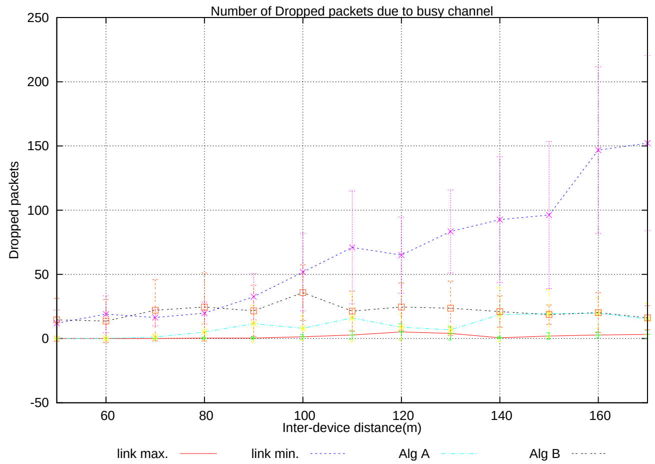
Graphs for the random flows in a regular small area scenario



(a) Collided

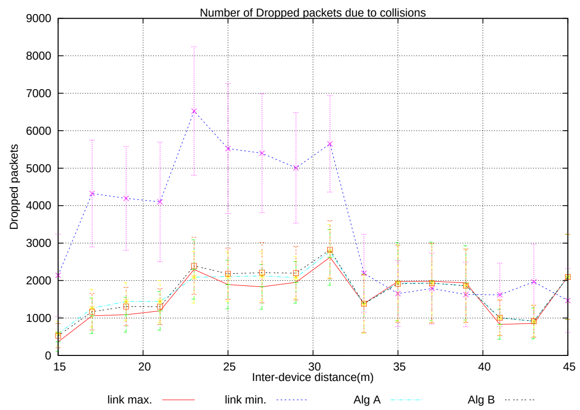


(b) Duplicated

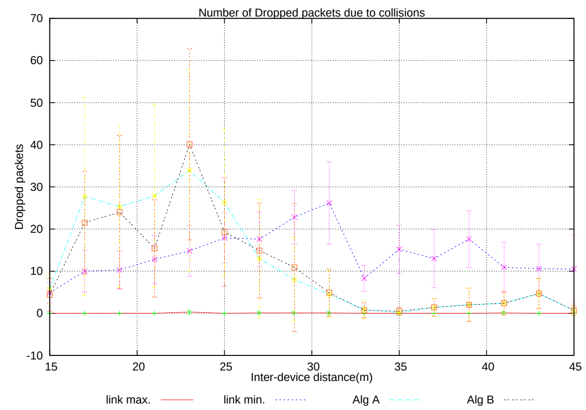


(c) Busy

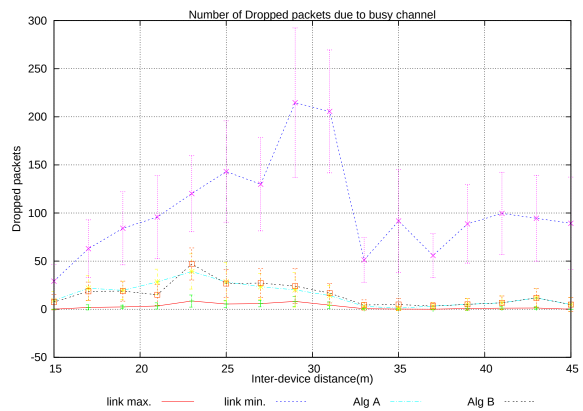
Figure D.1: *Number MAC Packet Events of 20 random flows in a regular small area scenario (802.11,UDP)*



(a) Collided

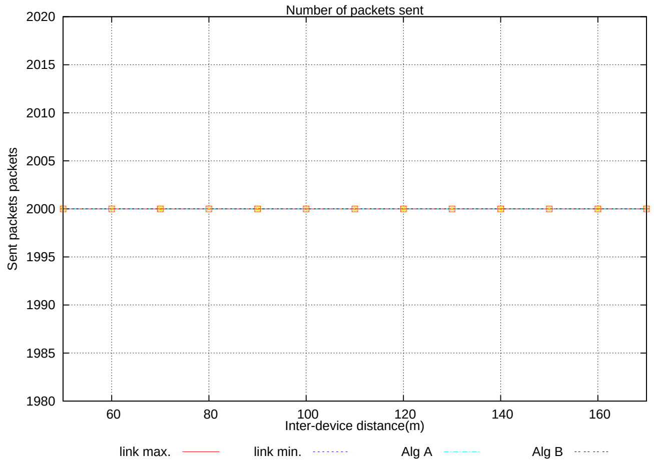


(b) Duplicated

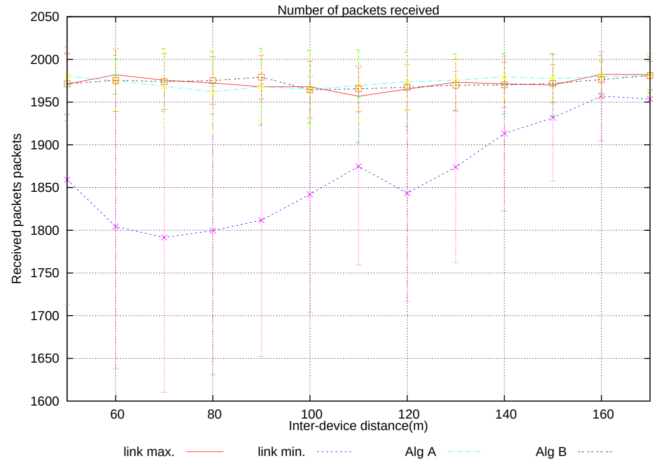


(c) Busy

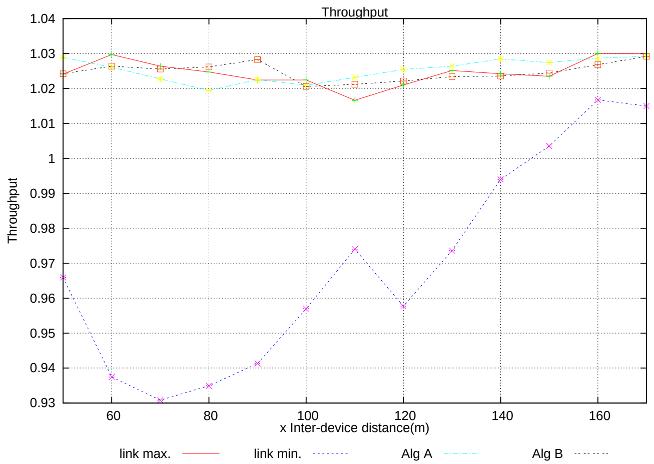
Figure D.2: *Number MAC Packet Events of 20 random flows in a regular small area scenario (Philips,UDP)*



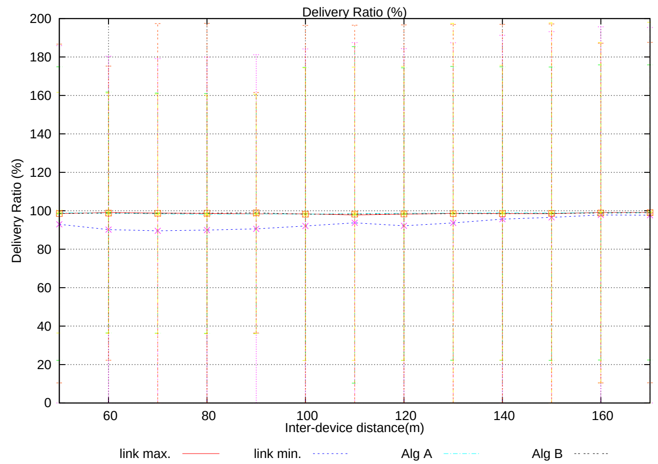
(a) Sent



(b) Received

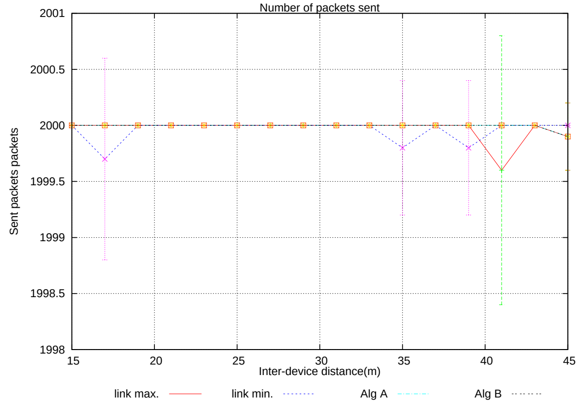


(c) Throughput

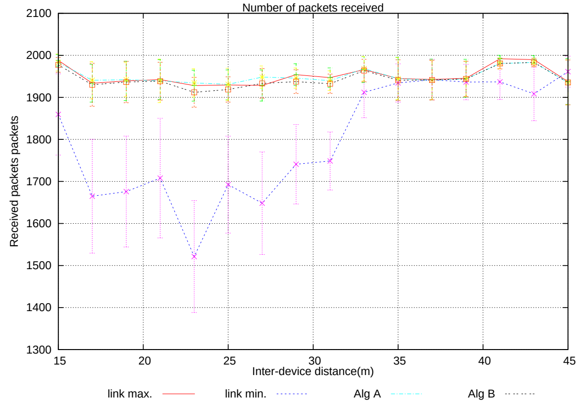


(d) Delivery Ratio

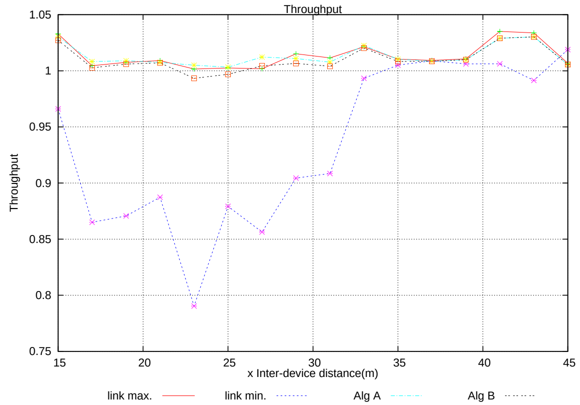
Figure D.3: *Number Application level packet events of 20 random flows in a regular small area scenario (802.11,UDP)*



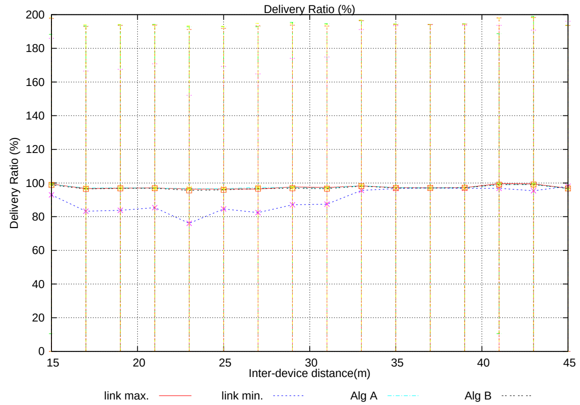
(a) Sent



(b) Received

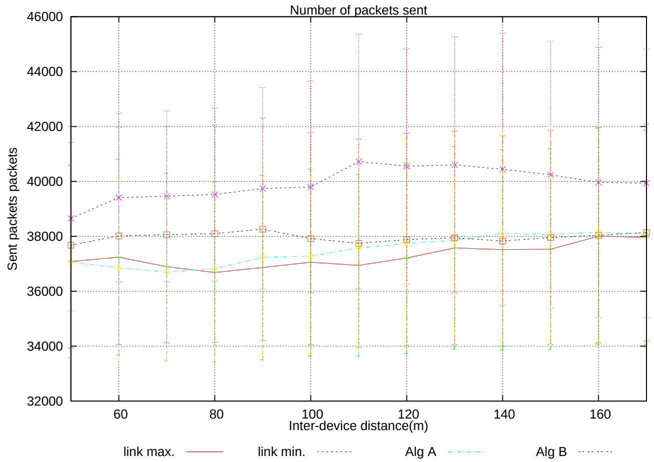


(c) Throughput

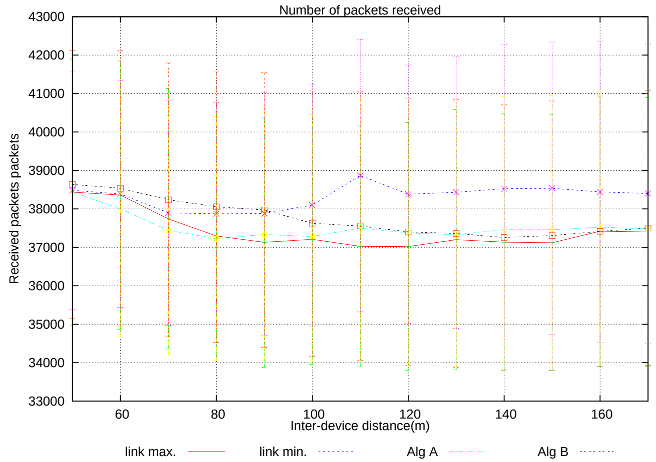


(d) Delivery Ratio

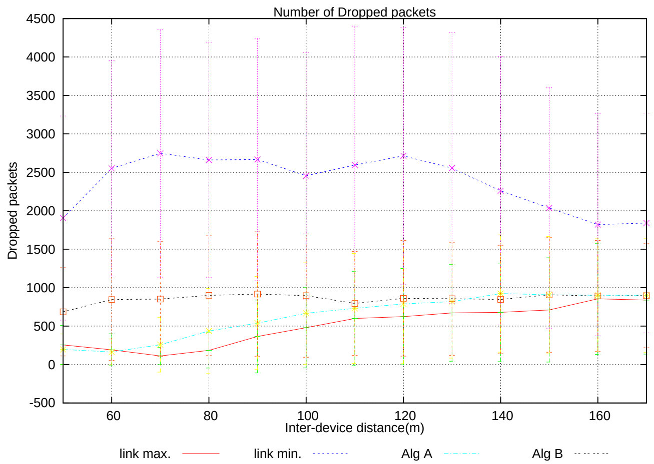
Figure D.4: *Number Application level packet events of 20 random flows in a regular small area scenario (Philips,UDP)*



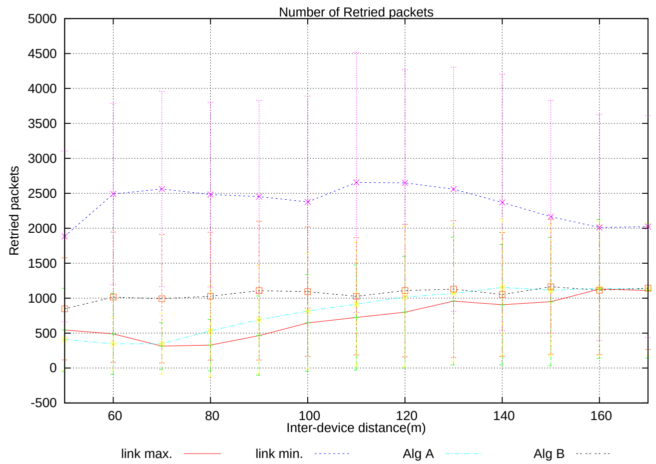
(a) Sent



(b) Received

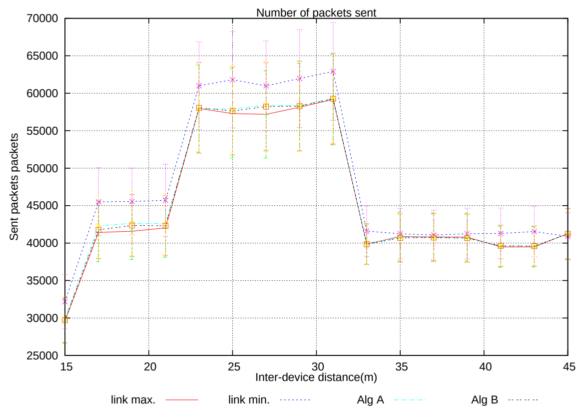


(c) Dropped

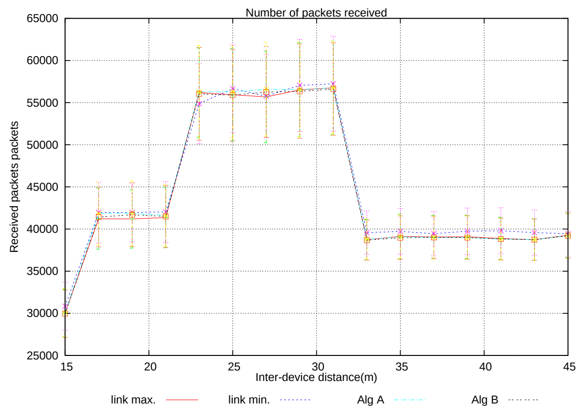


(d) Retried

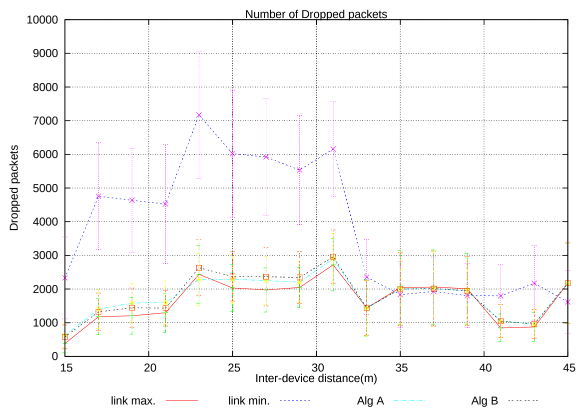
Figure D.5: *Number MAC level packet events of 20 random flows in a regular small area scenario (802.11,UDP)*



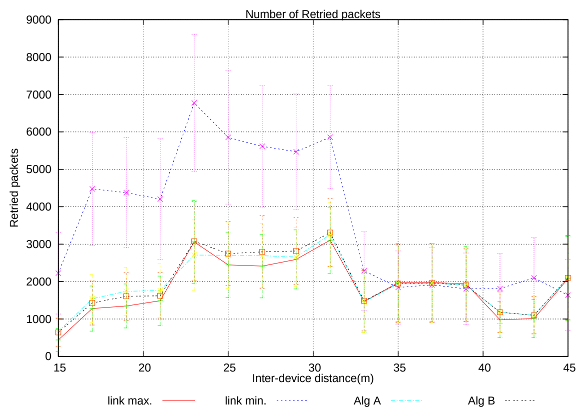
(a) Sent



(b) Received



(c) Dropped

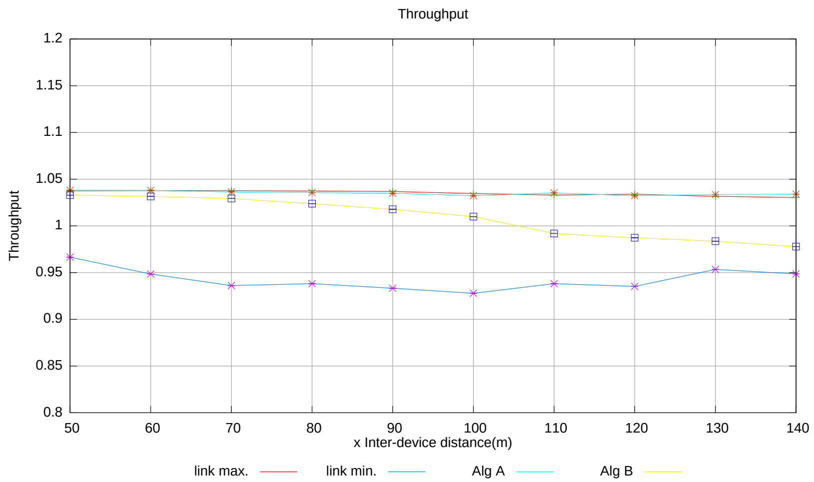


(d) Retried

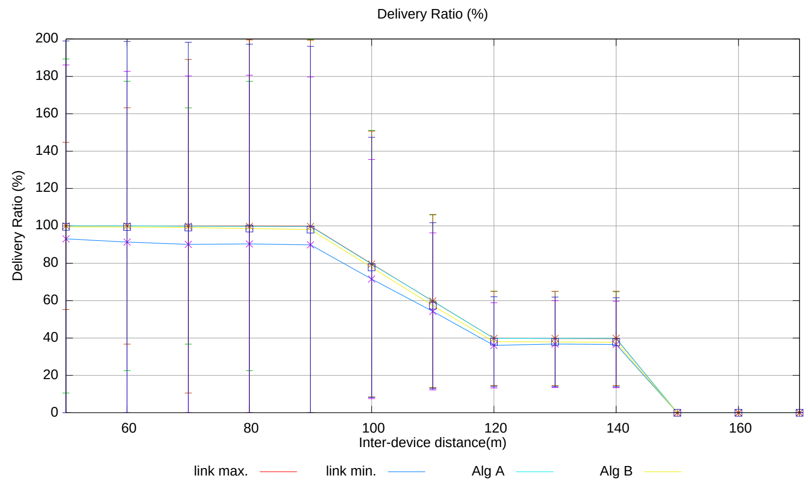
Figure D.6: *Number MAC level packet events of 20 random flows in a regular small area scenario (Philips,UDP)*

Appendix E

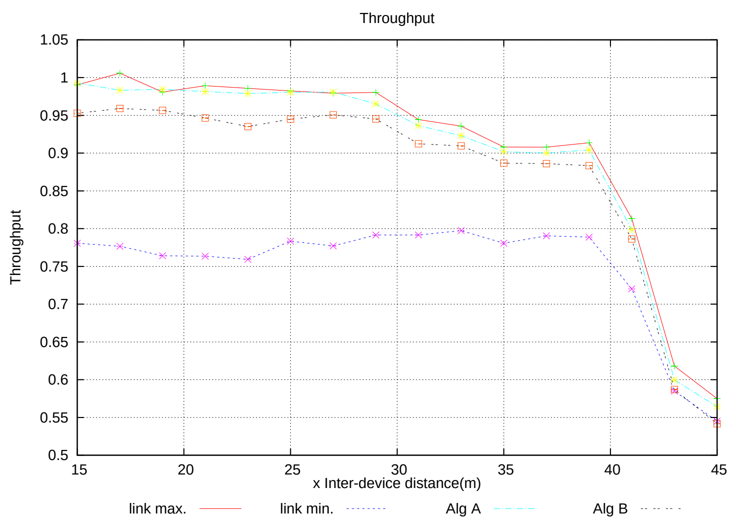
Graphs for the random flows in a random small area scenario



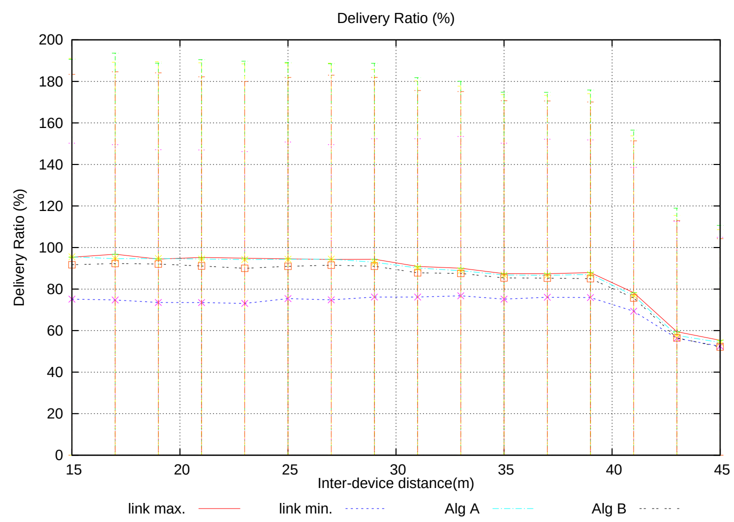
(a) Throughput - 802.11



(b) Delivery Ratio - 802.11

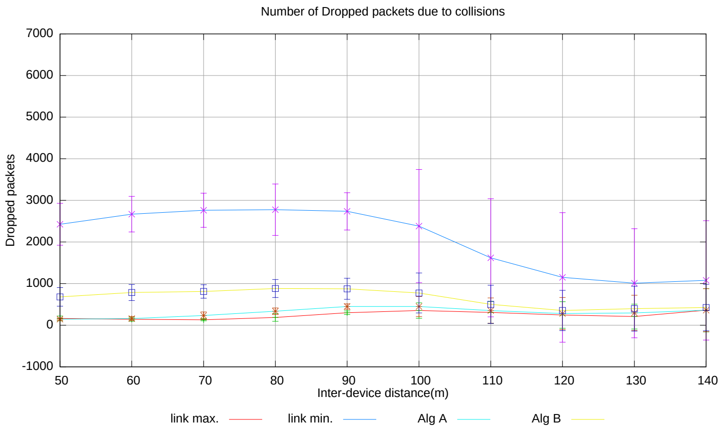


(c) Throughput - Philips

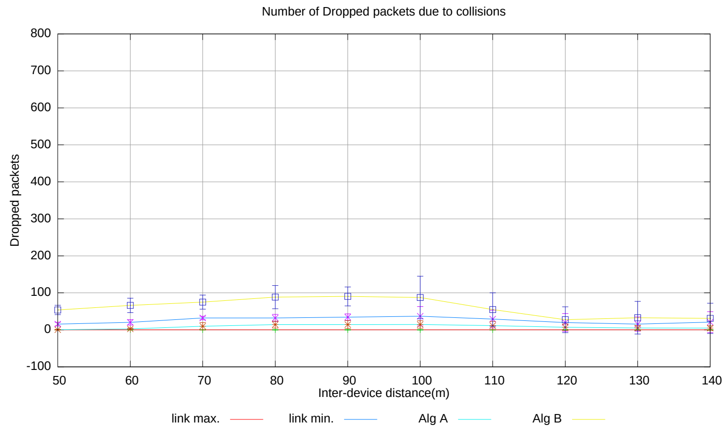


(d) Delivery Ratio - Philips

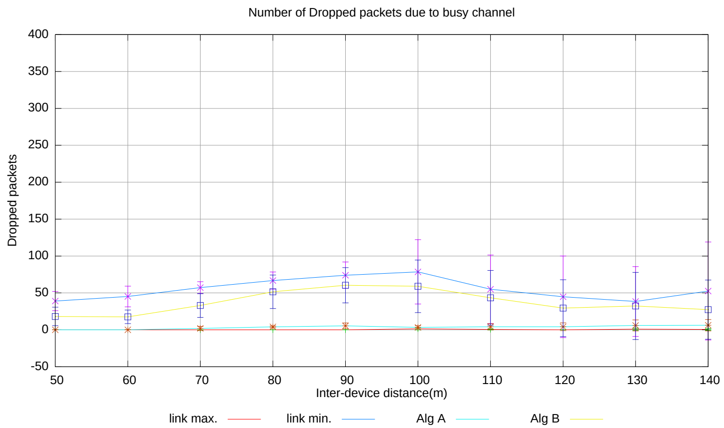
Figure E.1: *Delivery Ratio and Throughput for 20 random flows in a random small area scenario (UDP)*



(a) Collided

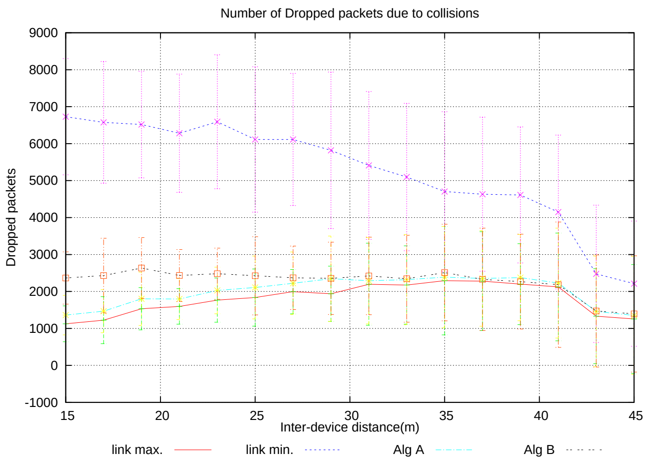


(b) Duplicated

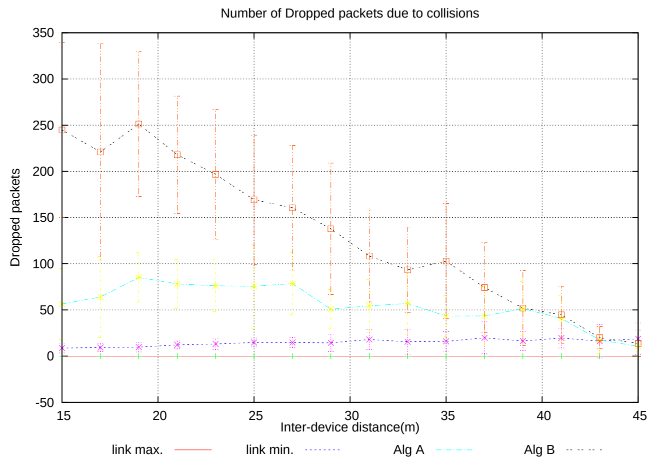


(c) Busy

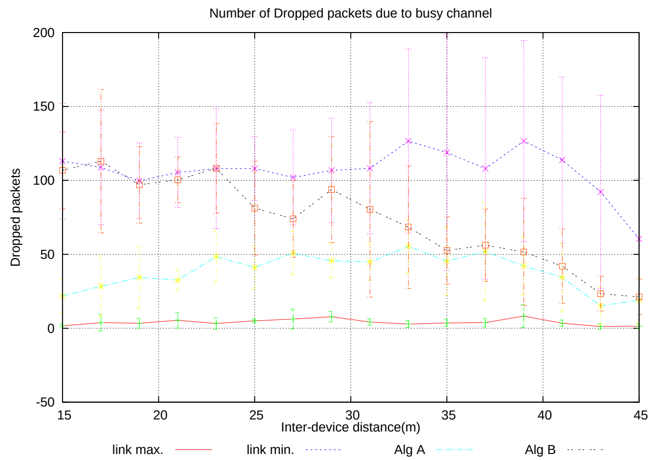
Figure E.2: *Number MAC Packet events of 20 random flows in a random small area scenario (802.11,UDP)*



(a) Collided

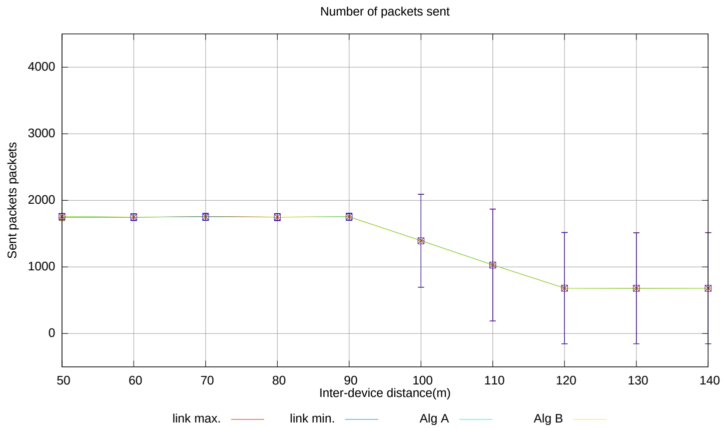


(b) Duplicated

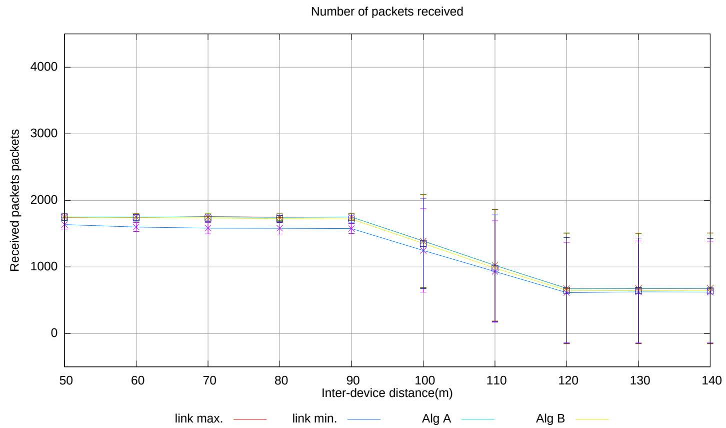


(c) Busy

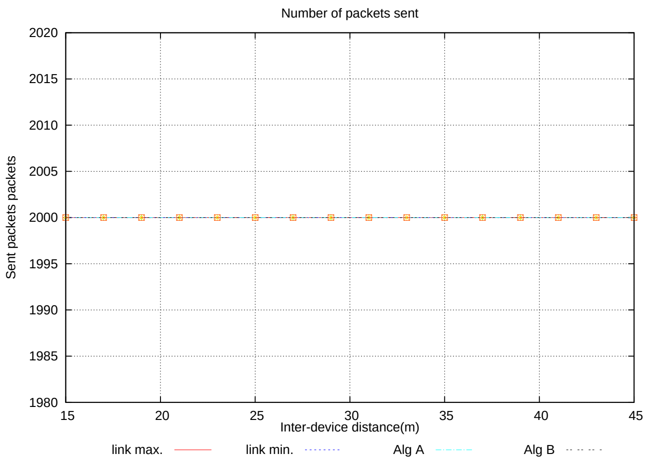
Figure E.3: *Number MAC Packet events of 20 random flows in a random small area scenario (Philips,UDP)*



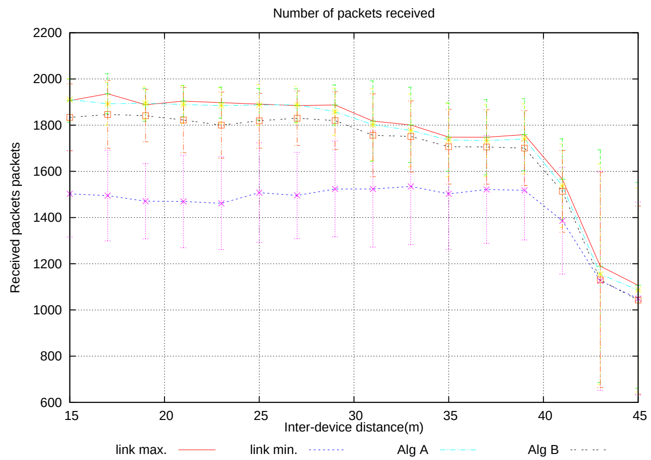
(a) Sent - 802.11



(b) Received - 802.11



(c) Sent - Philips

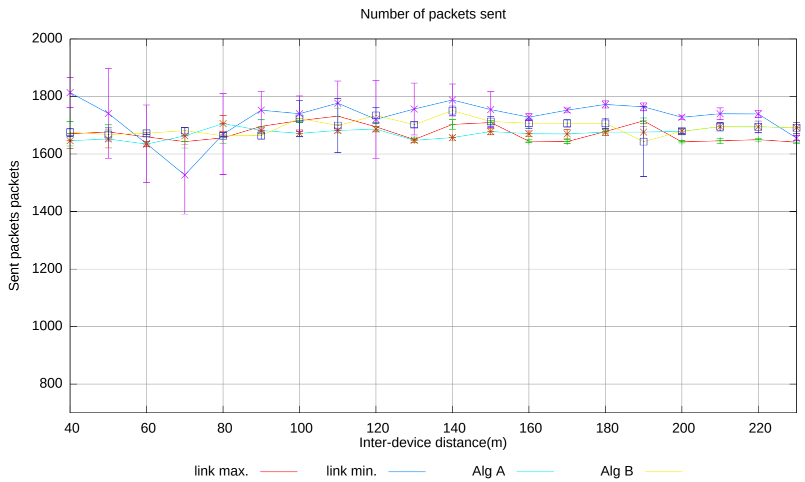


(d) Received - Philips

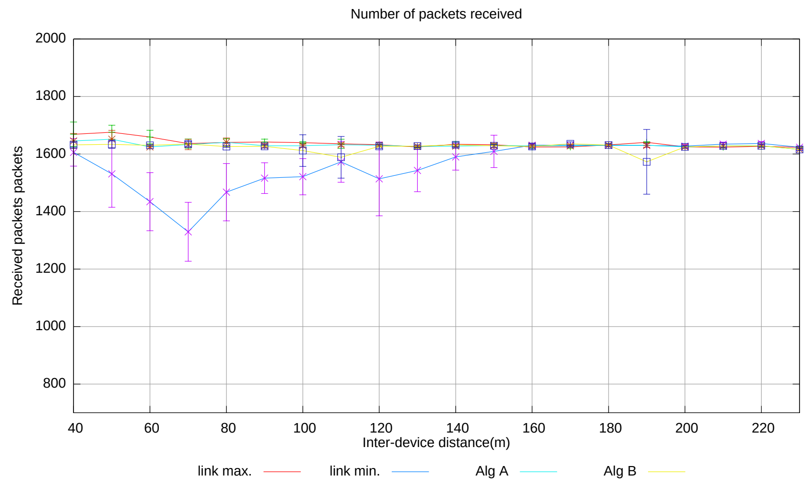
Figure E.4: *Number Application level packet events of 20 random flows in a random small area scenario (UDP)*

Appendix F

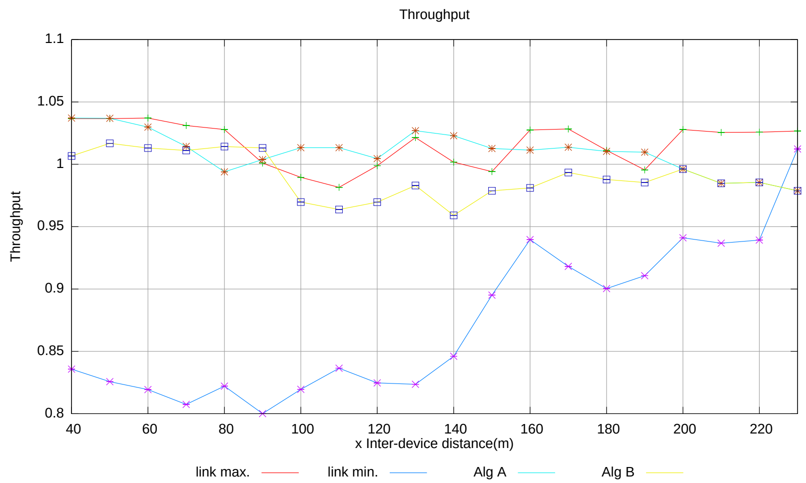
Graphs for comparing TCP and UDP in a regular small area



(a) Sent

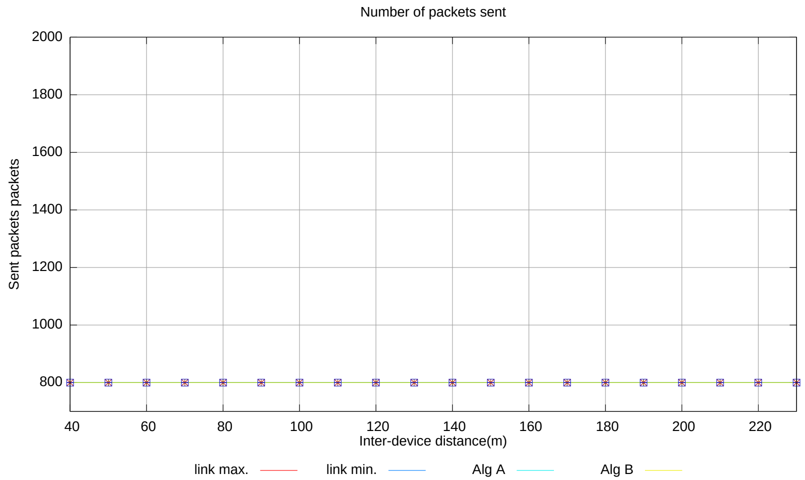


(b) Received

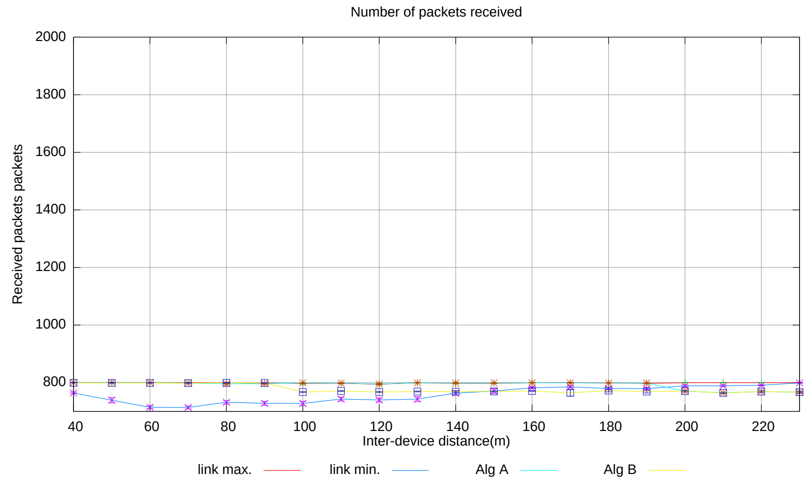


(c) Throughput

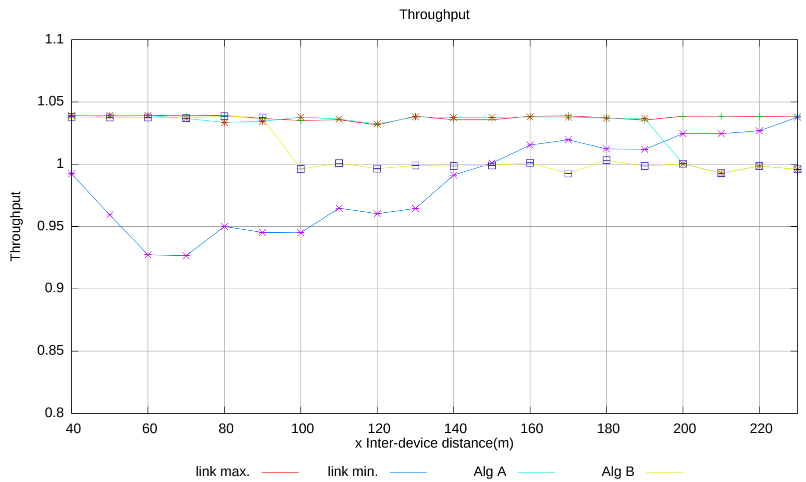
Figure F.1: *Application level packet events for TCP traffic in 800 packets in a regular small area scenario*



(a) Sent

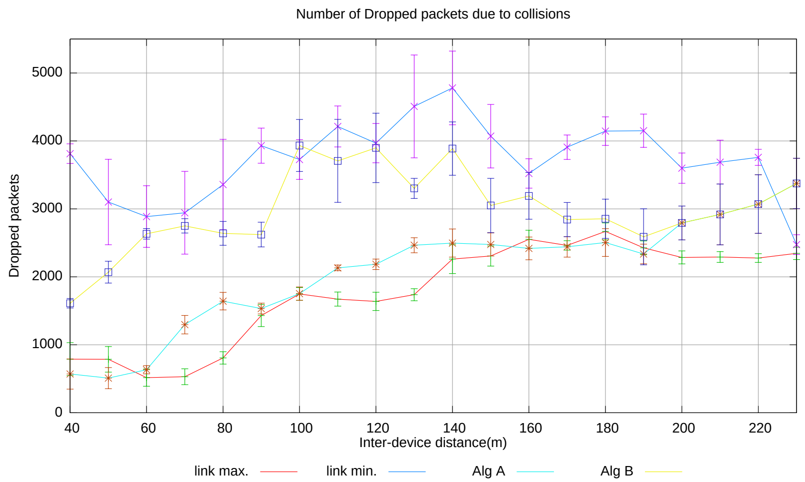


(b) Received

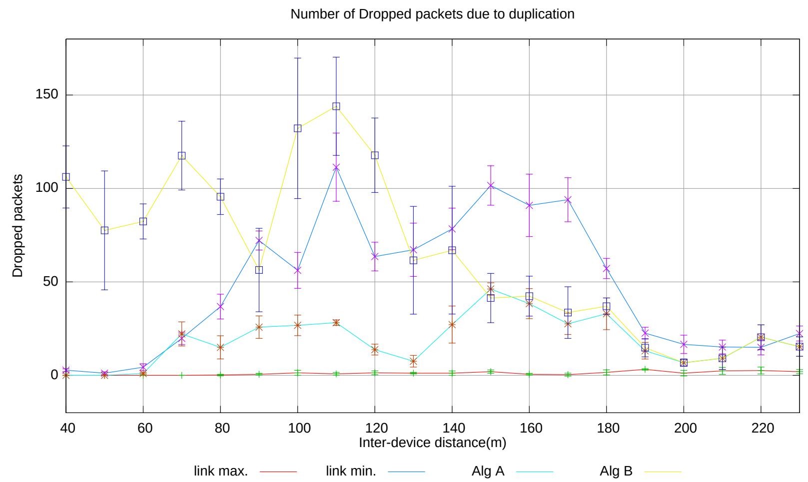


(c) Throughput

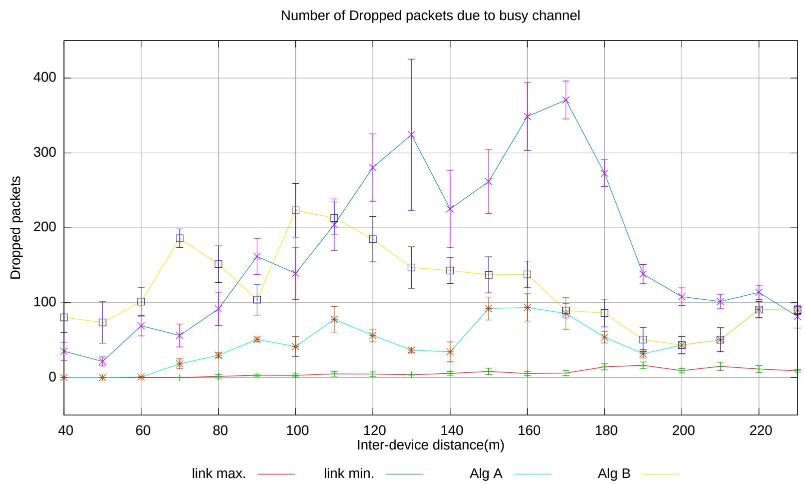
Figure F.2: *Application level packet events for UDP traffic in 800 packets in a regular small area scenario*



(a) Collided

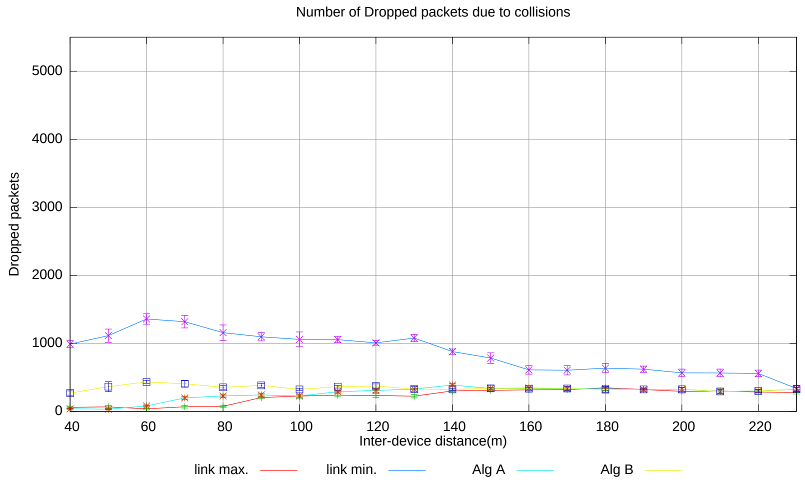


(b) Duplicated

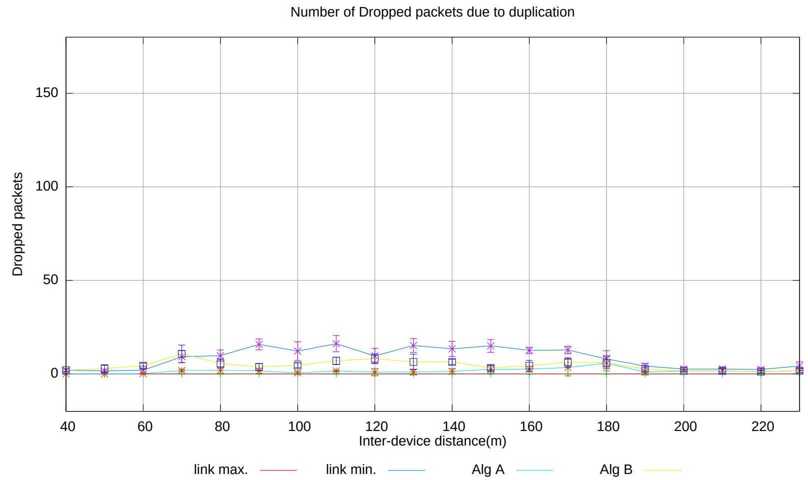


(c) Busy

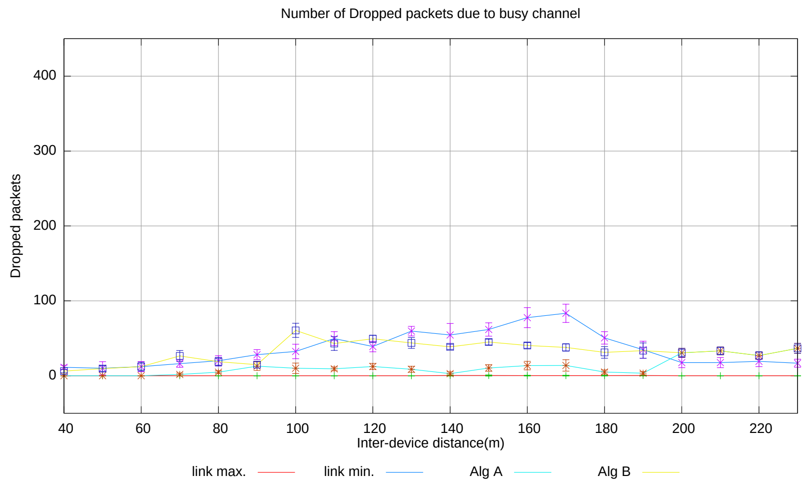
Figure F.3: *MAC level packet events for 800 TCP packets in a regular small area scenario*



(a) Collided



(b) Duplicated

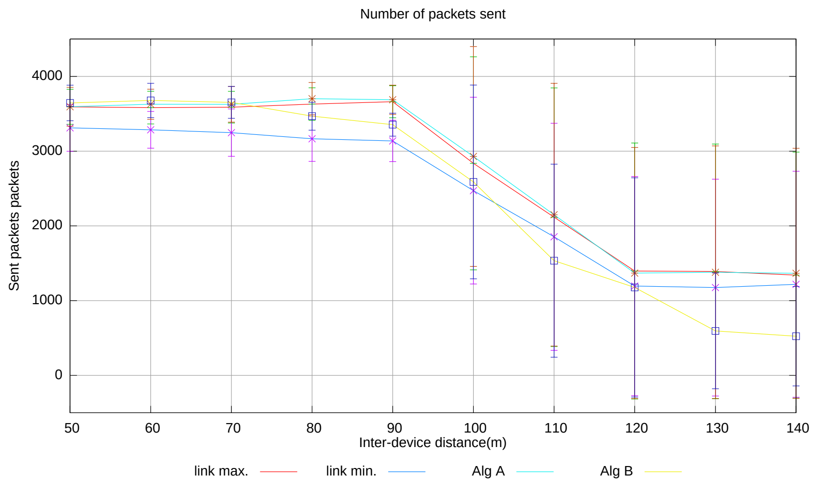


(c) Busy

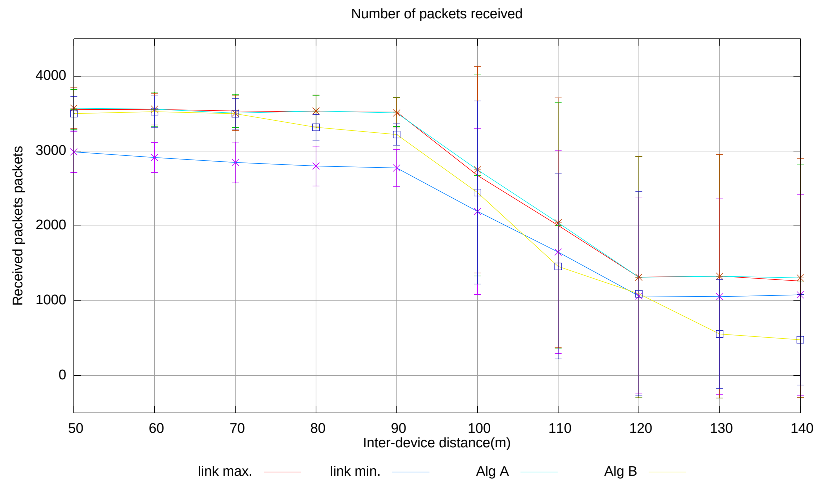
Figure F.4: *MAC level packet events for 800 UDP packets in a regular small area scenario*

Appendix G

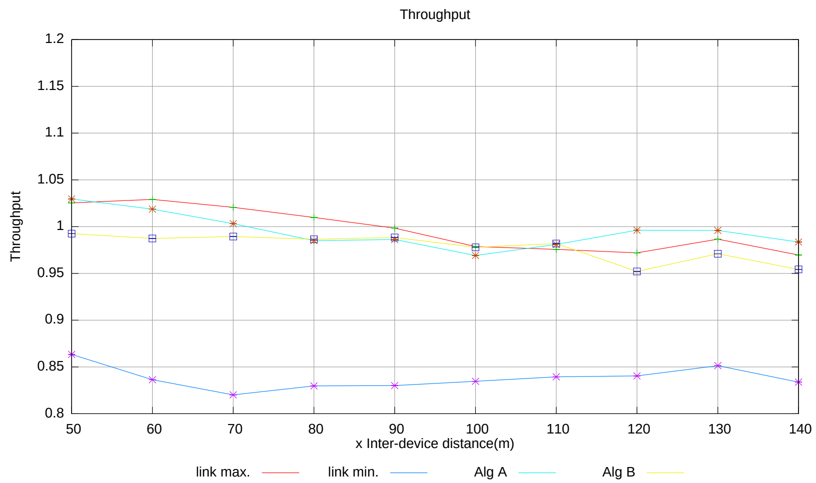
Graphs for comparing TCP and UDP in a random small area



(a) Sent

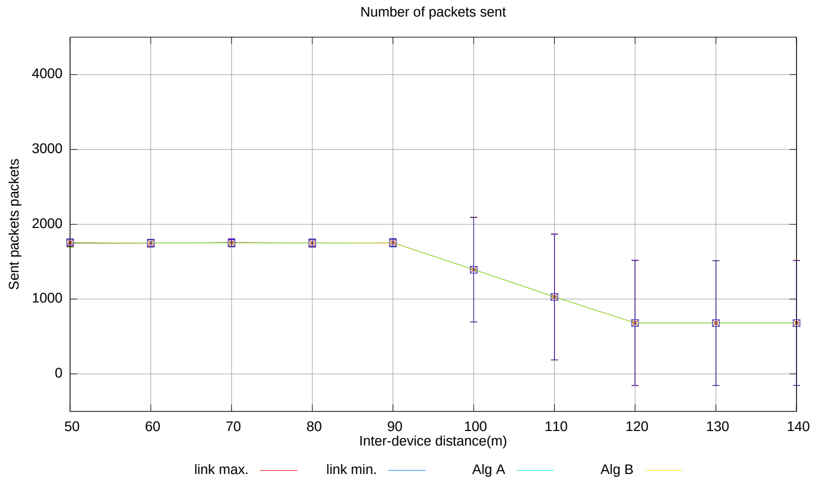


(b) Received

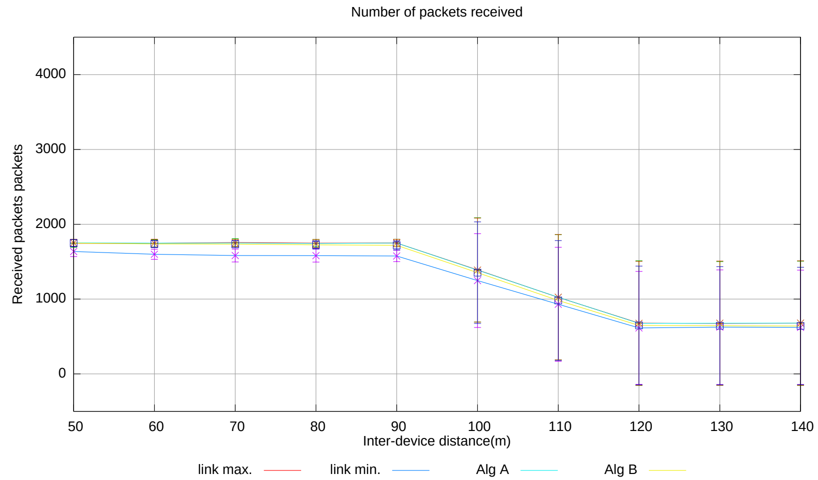


(c) Throughput

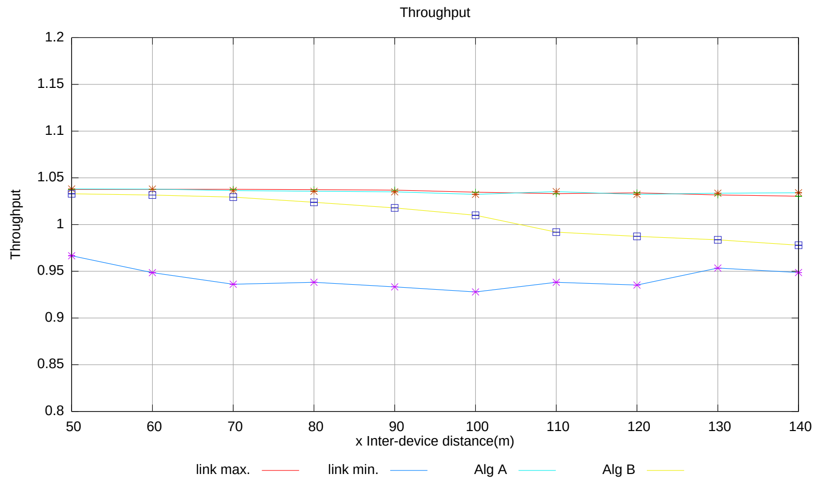
Figure G.1: *Application level events for 20 flows of TCP in a random small area scenario*



(a) Sent

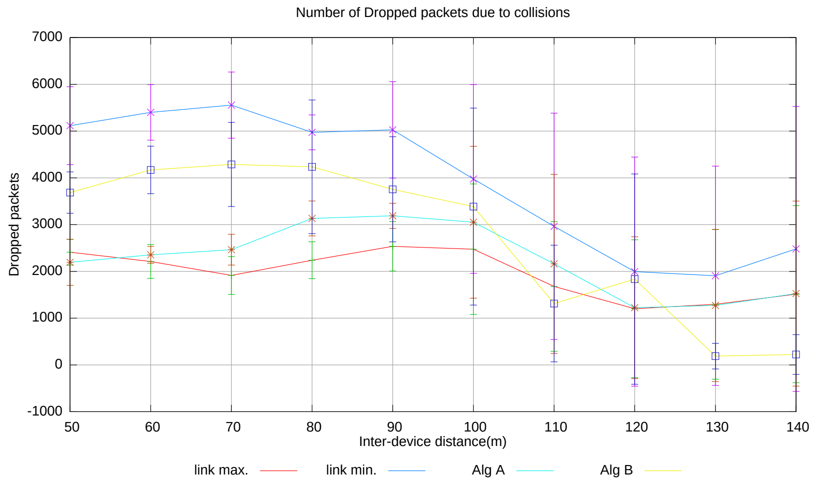


(b) Received

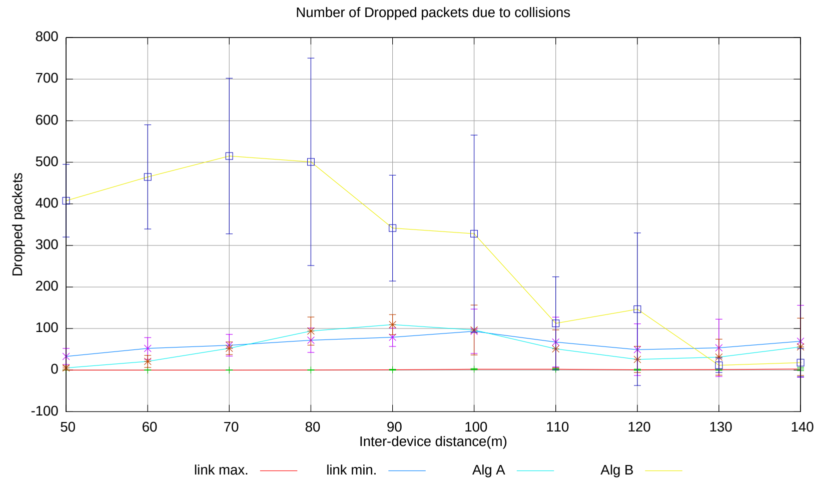


(c) Throughput

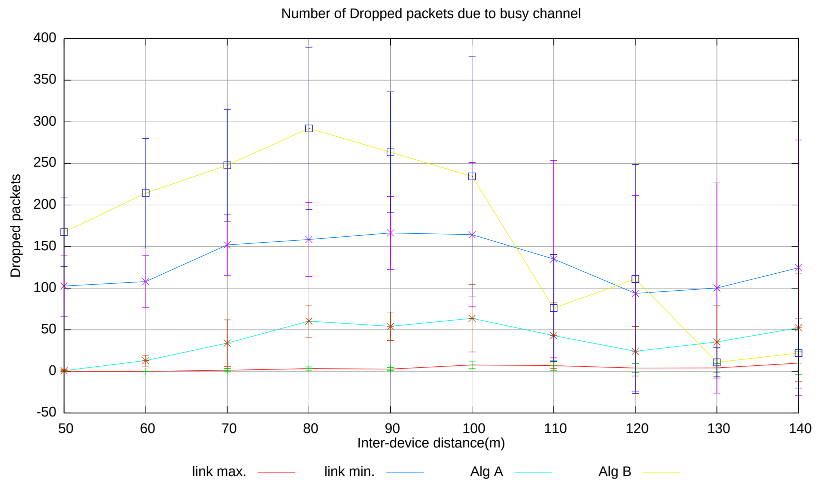
Figure G.2: *Application level events for 20 flows of UDP in a random small area scenario*



(a) Collided

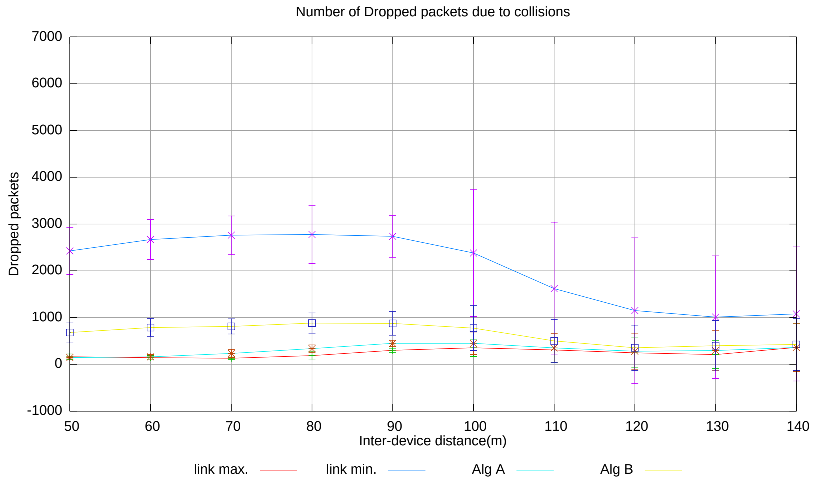


(b) Duplicated

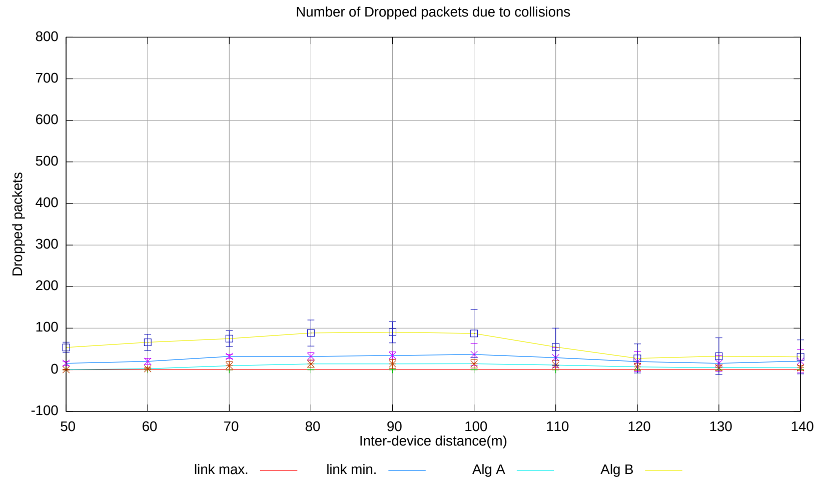


(c) Busy

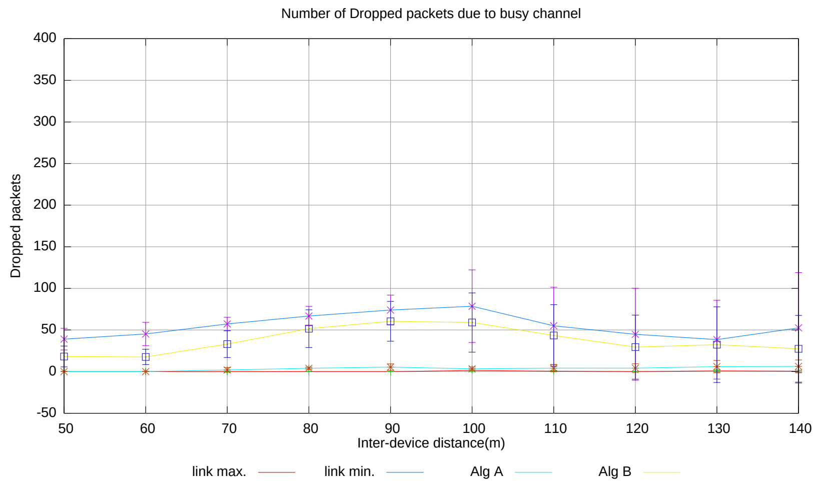
Figure G.3: *MAC level events for 20 TCP flows in a random small area scenario*



(a) Collided



(b) Duplicated

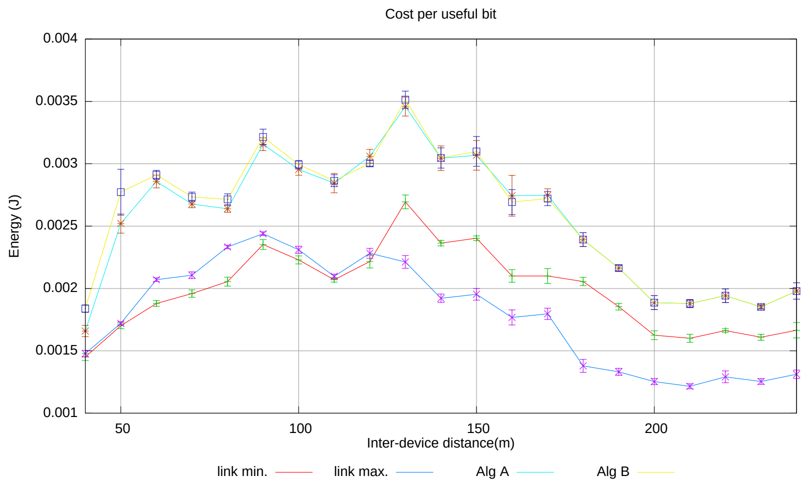


(c) Busy

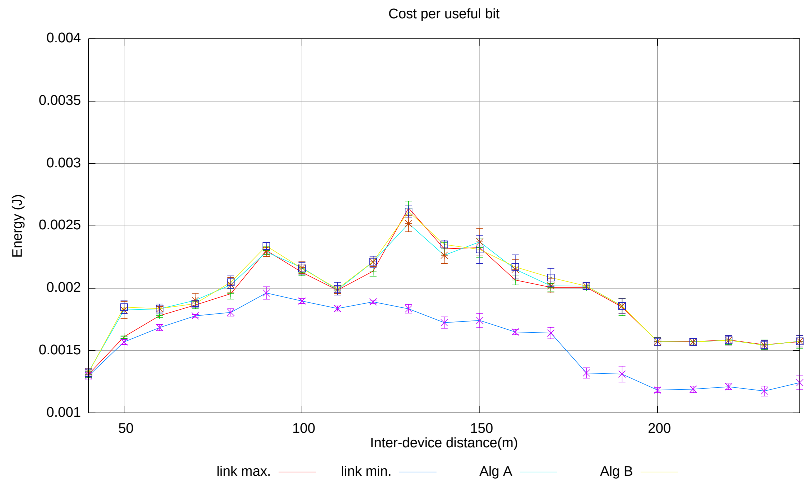
Figure G.4: *MAC level events for 20 TCP flows in a random small area scenario*

Appendix H

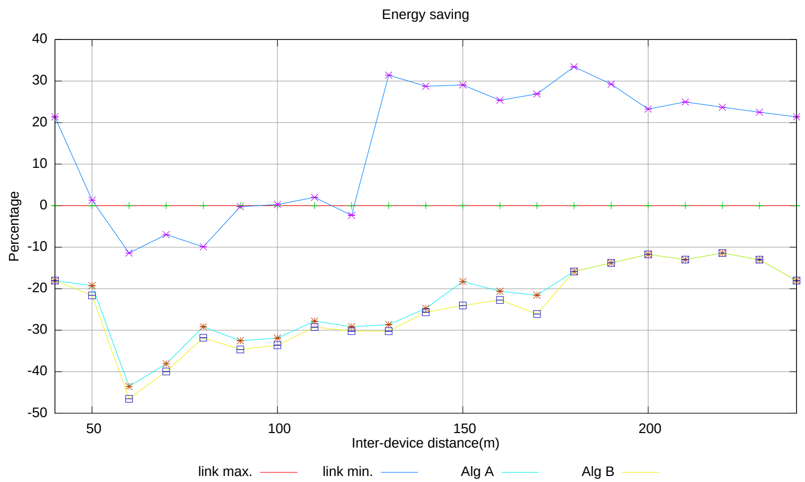
(S)DSDV Graphs for the random flows in a regular small area scenario



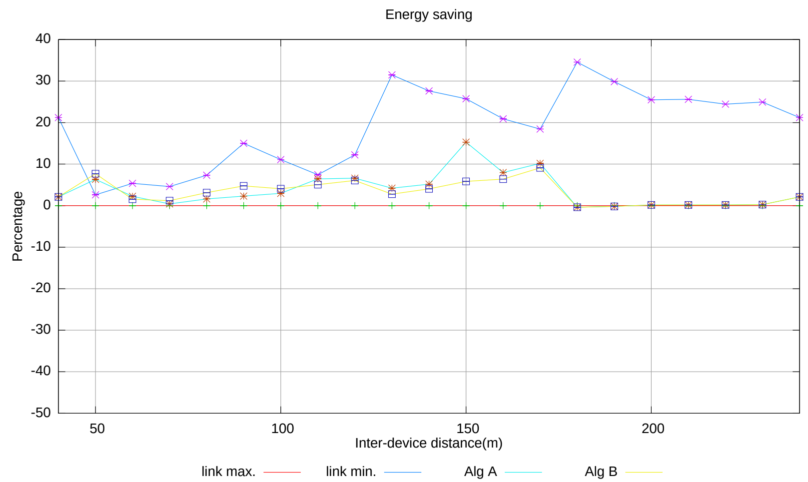
(a) C_{UB} - SDSDV



(b) C_{UB} - DSDV

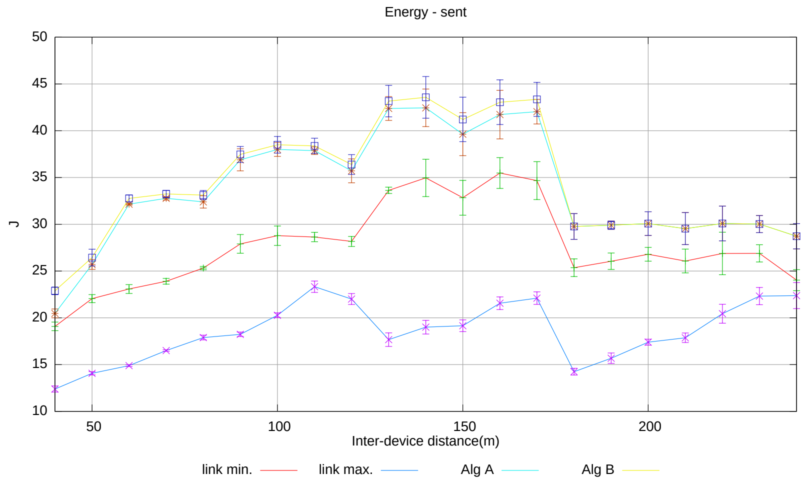


(c) Energy saving % - SDSDV

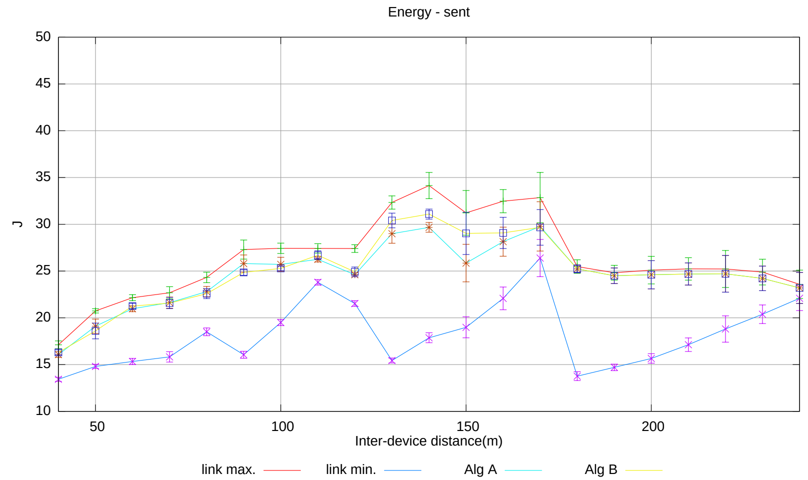


(d) Energy saving % - DSDV

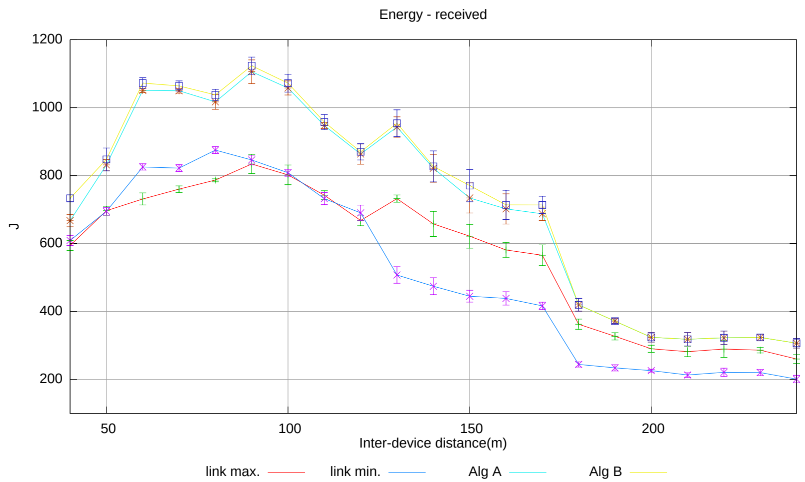
Figure H.1: *20 regular flows in a regular small area scenario (UDP) with routing using (S)DSDV (pt1)*



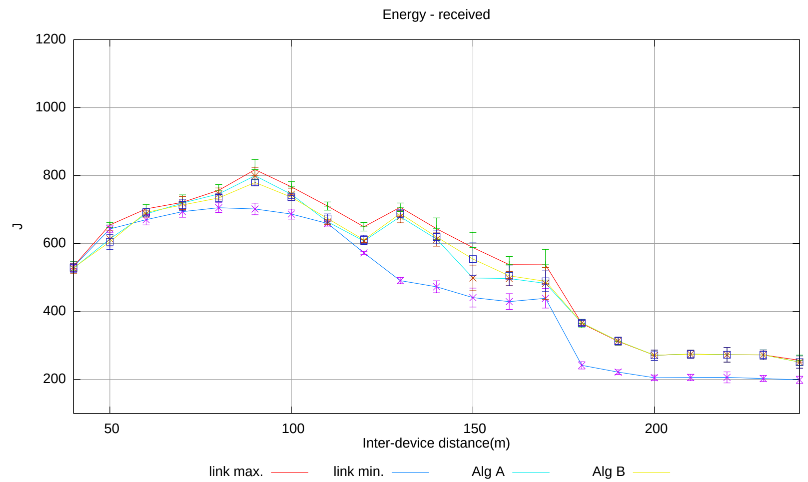
(a) E_{tx} - SDSDV



(b) E_{tx} - DSDV



(c) E_{rx} - SDSDV

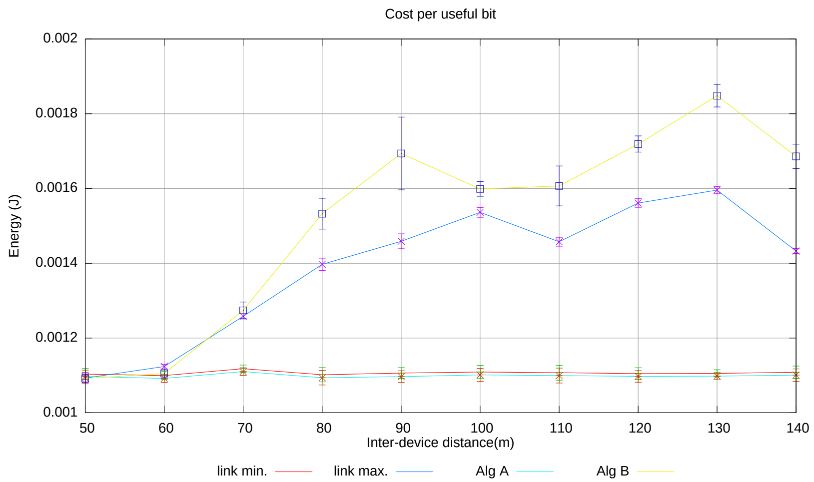


(d) E_{rx} - DSDV

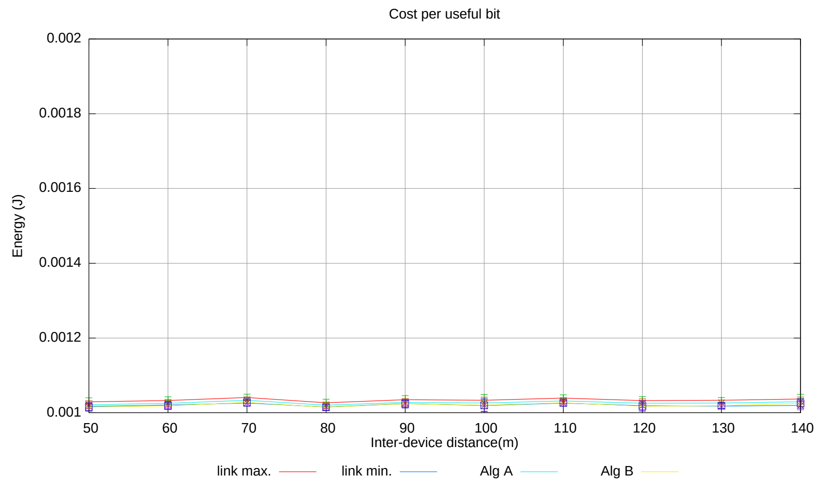
Figure H.2: *20 regular flows in a regular small area scenario (UDP) with routing using (S)DSDV (pt2)*

Appendix I

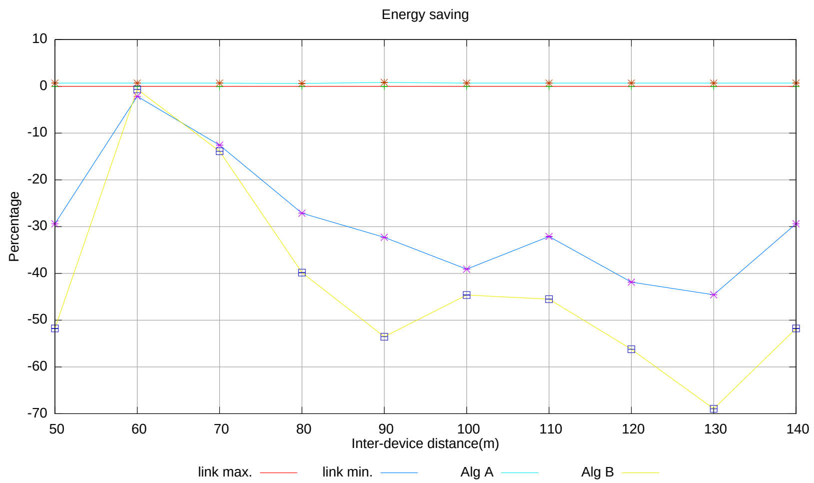
(S)DSDV Graphs for the random flows in a random small area scenario



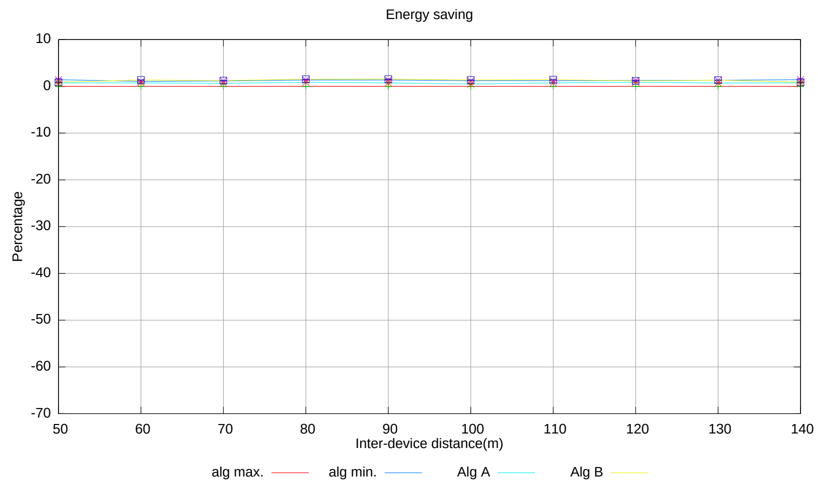
(a) C_{UB} - SDSDV



(b) C_{UB} - DSDV

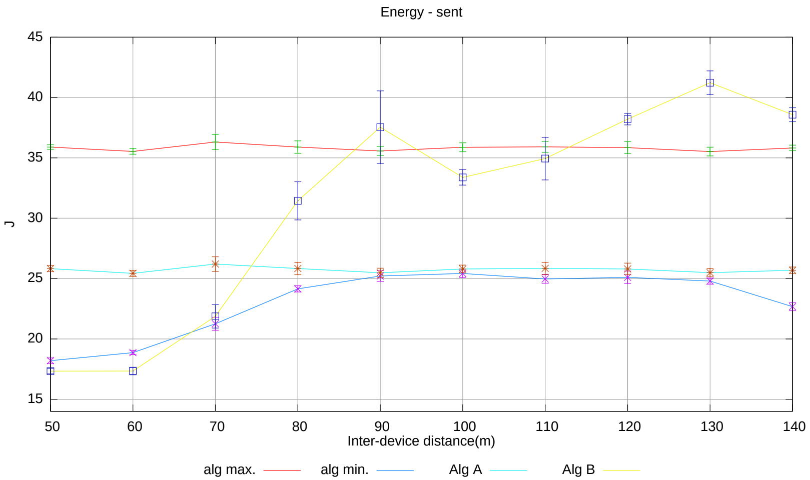


(c) Energy saving $\%$ - SDSDV

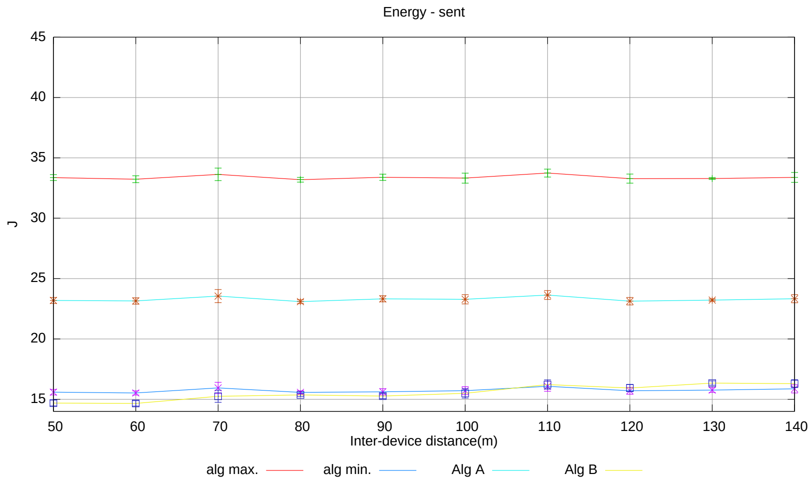


(d) Energy saving $\%$ - DSDV

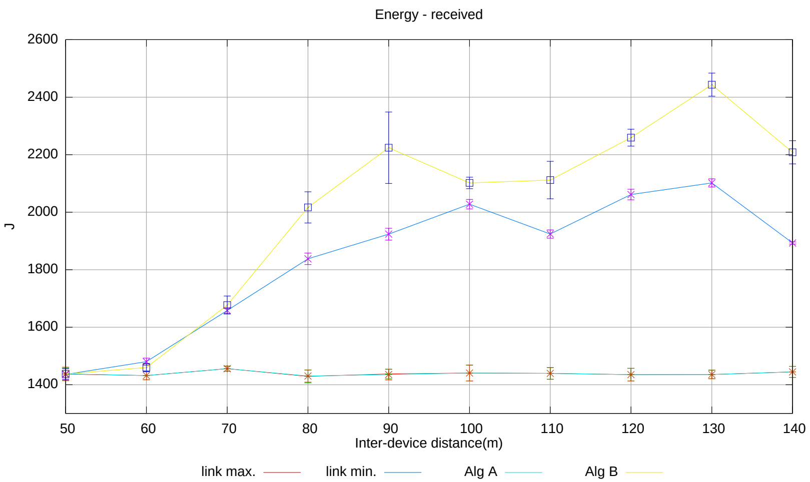
Figure I.1: *20 regular flows in a random small area scenario (UDP) with routing using (S)DSDV (pt1)*



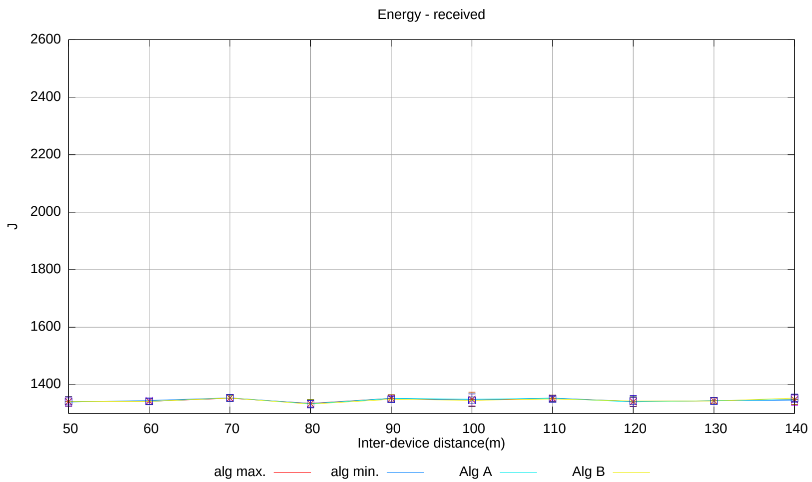
(a) E_{tx} - SDSDV



(b) E_{tx} - DSDV



(c) E_{rx} - SDSDV



(d) E_{rx} - DSDV

Figure I.2: *20 regular flows in a random small area scenario (UDP) with routing using (S)DSDV (pt2)*

Appendix J

Related publications

The following two papers have formed a substantial portion of chapter 3.

- Don't listen, talk: A comparative study of transmit and receive power for low power radios. Adam Greenhalgh, Stephen Hailes London Communications Symposium, University College London, 13th -14th September 2004 (LCS 2004).

- A summary of complex behaviour from simple power conserving protocols. Adam Greenhalgh, Stephen Hailes In Proc. of the 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'05), Berlin, Germany, 11-14 Sept. 2005.

I have collaborated with others and undertaken a number of related pieces of work which haven't formed portions of earlier chapters; these are :

- Landmark Guided Forwarding. MH.Lim, A.Greenhalgh, J.Chesterfield and J.Crowcroft. 13th IEEE International Conference on Network Protocols, 6th - 9th Novemeber 2005, Boston, MA

- Landmark Guided Forwarding: A hybrid approach for Ad Hoc routing MH.Lim, A.Greenhalgh, J.Chesterfield and J.Crowcroft. University of Cambridge Technical report.

- Hybrid routing: A pragmatic approach to mitigating position uncertainty in geo-routing. MH.Lim, A.Greenhalgh, J.Chesterfield and J.Crowcroft. University of Cambridge Technical report.

- Landmarks Guided Forwarding (poster) Lim Meng How,Jon Crowcroft, Adam Greenhalgh, Andrew Campbell ACM SIGCOMM Poster Session (Sigcomm 2004)

Bibliography

[1] Arena. World Wide Web, http://fnord.usc.edu/arena. 18

[2] FSR, internet-draft. World Wide Web, http://www.ietf.org/internet-drafts/draft-ietf-manet-fsr-00.txt. 32

[3] Manet working group. World Wide Web, http://www.ietf.org/html.charters/manet-charter.html. 20

[4] TORA, internet-draft, 2001. 21, 24, 26

[5] Agilent. Hp 54540c (discontinued) , 2 gsa/s - 1 gsa/s , 500 msa/s , 500 mhz. 35

[6] Zigbee Alliance. Zigbee alliance website. World Wide Web, http://www.zigbee.org/. 15, 16, 30

[7] A.D. Amis, R. Prakash, T.H.P. Vuong, and D.T. Huynh. Max-min d-cluster formation in wireless ad hoc networks. In *Proceedings of IEEE INFOCOM 2000*. IEEE, March 2000. 28

[8] Daniel S. Baum. Simulating the sui channel models. World Wide Web, http://www.ieee802.org/16/tg3/contrib/802163c-01_53.pdf, 2001. 51

[9] R.C. Bernhardt. Macroscopic diversity in frequency reuse systems. *IEEE Journal on Selected Areas in Communications*, Vol-SAC 5:862–878, June 1987. 51

[10] John Bresina, Gregory A. Dorais, Keith Golden, David E. Smith, and Richard Washington. Autonomous rover for human exploration of mars, 1998. 4

[11] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobicom 1998*, pages 85–97. ACM, 1998. 26, 48

[12] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 307–320, New York, NY, USA, 2006. ACM. 13

[13] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of 7th Annual International conference on Mobile Computing and Networking*, pages 85–96. ACM, July 2001. 21, 30

[14] C. Chiang, H. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks, 1997. 29

[15] Shane Eisenman Chieh-Yih Wan and Andrew Campbell. Coda: Congestion detection and avoidance in sensor networks. In *First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, November 2003. 53

[16] Renesas Technology Corp. Mitsubishi microcomputer, 3807 group, single-chip 8-bit cmos microcomputer. 35

[17] D.C Cox, R.R. Murray, and A.W. Noris. 800 mhz attenuation measured in and around suburban houses. *AT&T Bell Laboratory Technical Journal*, 673(6), July-August 1984. 51

[18] B. Das, E. Sivakumar, and V. Bhargavan. Routing in ad-hoc networks using a spine, 1997. 26, 28

[19] Samir R. Das, Charles E. Perkins, and Elizabeth M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. 21

[20] R. Dube et al. Signal stability based adaptive routing for ad hoc mobile networks, February 1997. 27

[21] Laura Marie Feeney and Martin Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Infocom*. IEEE, 2001. 33, 34, 43, 44, 68, 125

[22] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian. Poweraware routing in wireless packet networks. 30

[23] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian. Paro : A power-aware routing optimization scheme for mobile ad hoc networks, February 2001. 30

[24] J. Gomez, A.T. Campbell, M. Naghshineh, and C. Bisdikian. Paro: Supporting dynamic power controlled routing in wireless ad hoc networks. *ACM/Kluwer Journal on Wireless Networks (WINET)*, 2003. 45

[25] J. Gomez, A.T. Campbell, M. Nagshineh, and C. Bisdikian. Paro : A power-aware routing optimization scheme for mobile ad hoc networks, internet-draft, February 2001. 30

[26] Adam Greenhalgh and Stephen Hailes. Don't listen, talk: A comparative study of transmit and receive power for low power radios. In *London Communications Symposium, University College London, (LCS 2004)*, September 2004. 54

[27] Bluetooth Special Interest Group. The official bluetooth website. World Wide Web, `http://www.bluetooth.com/`. 15, 30

[28] UCLA PCL group. Glomosim web site. World Wide Web, `http://pcl.cs.ucla.edu/projects/glomosim/`. 16

[29] Monarch group at CMU. Monarch home page. World Wide Web, `http://www.monarch.cs.cmu.edu`. 16, 17, 24, 68

[30] NS group at ISI. Ns 2 home page. World Wide Web, `http://www.isi.edu/nsnam/ns/`. 16, 18, 51

[31] Haas. Wireless networks lab. World Wide Web, `http://www.ee.cornell.edu/~haas/wnlprojects.html`, September 2001. 28

[32] Haas. ZRP, internet-draft. World Wide Web, `http://www.ee.cornell.edu/~haas/Publications/draft-ietf-manet-zone-zrp-04.txt`, July 2002. 3, 21, 28

[33] Stephen Hailes. Z12, mobile systems lecture notes for UCL CS MSc in data communications networks and distrbuted systems, 1999. 19

[34] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences - 2000*. IEEE, 2000. 30

[35] Yih-Chun Hu and David B. Johnson. Caching stratefies in on-demand routing protocols for wireless ad hoc networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000)*. ACM, August 2000. 21, 24

[36] IEEE. IEEE 802.11 Wireless Local Area Networks. World Wide Web, `http://www.ieee802.org/11`. 4

[37] IEEE. IEEE 802.15 WPAN Task Group 1 (TG1). World Wide Web, `http://www.ieee802.org/15/pub/TG1.html`. 4, 15

[38] IEEE. IEEE 802.15 WPAN Task Group 4 (TG4). World Wide Web, `http://www.ieee802.org/15/pub/TG4.html`. 4

[39] IEEE. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements. World Wide Web, `http://standards.ieee.org/getieee802/download/802.11e-2005.pdf`. 58

[40] IEEE. Ieee standard for local and metropolitan area networks: Overview and architecture, December 1990. 14

[41] Opnet Technologies Inc. Opnet web site. World Wide Web, `http://www.opnet.com`. 16

[42] D. Johnson, Y. Hu, and D. Maltz. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4, request for comment (rfc). World Wide Web, `http://www.ietf.org/rfc/rfc4728.txt`, February 2007. 24

[43] David B. Johnson. Routing in ad hoc networks of mobile hosts. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, pages 158–163. IEEE, December 1994. 21

[44] David B. Johnson and David A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996. 3, 20, 21, 24, 25, 26

[45] David B. Johnson and David A. Maltz. Protocols for adaptive wireless and mobile networking. *IEEE Personal Communications*, 3(1), February 1996. 21

[46] L.R. Ford Jr. and D. R. Fulkerson. *Flows in Networks*. Princeton Univ. Press, 1962. 22

[47] Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *MOBICOM 2000*, pages 243–254. ACM, August 2000. 32

[48] Oliver Kasten. Energy consumption. October 2000. 21

[49] Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. 32

[50] R. Kravets, K. Schwan, and K. Calvert. Power-aware communication for mobile computers. World Wide Web, `http://citeseer.nj.nec.com/435118.html`, 1999. 19, 34

[51] Robin Kravets and P. Krishnan. Power management techniques for mobile communication. In *Proceedings of 4th Annual International conference on Mobile Computing and Networking*, pages 157–168. ACM, July 1998. 19, 34

[52] P. Krishna, N. Vaidya, M. Chatterjee, and D. Pradhan. A cluster-based approach for routing in dynamic networks, 1997. 28

[53] Q. Li, J. Aslam, and D. Rus. Online power-aware routing in wireless ad-hoc networks, 2001. 30

[54] Sha Liu, Kai-Wei Fan, and Prasun Sinha. Cmac: An energy-efficient mac layer protocol using convergent packet forwarding for wireless sensor networks. *ACM Trans. Sen. Netw.*, 5(4):1–34, 2009. 13

[55] Liran Ma, Qian Zhang, and Xiuzhen Cheng. A power controlled interference aware routing protocol for dense multi-hop wireless networks. *Wirel. Netw.*, 14(2):247–257, 2008. 14

[56] D. Maltz, J. Broch, and D. Johnson. Quantitative lessons from a full-scale multi-hop wireless ad hoc network testbed, 2000. 16, 68

[57] Maxim. Spi/microwirecompatible uart and $\pm$15kv esdprotected rs232 transceivers with internal capacitors. 35

[58] A.B. Mcdonald. *A mobility-based framework for adaptive dynamic cluster-based hybrid routing in wireless ad hoc networks*. PhD thesis, University of Pittsburgh, 2000. 28

[59] Jeffery Monks, Jean-Pierre Ebert, Wen-Mei W. Hwu, and Adam Wolisz. Energy saving and capacity improvement potential of power control in multi-hop wireless networks. *Computer Networks*, 41:313–330, 2003. 14

[60] Shree Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, 1996. 23

[61] NASA. Mars polar lander web side. World Wide Web, `http://mars.jpl.nasa.gov/msp98/index.html`. 4

[62] NASA. Mars pathfinder web site. World Wide Web, `http://science.ksc.nasa.gov/mars/default.html`, May 1998. 4

[63] OMNeT++. Omnet++ web site. World Wide Web, `http://www.omnetpp.org`. 17

[64] V.D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. 26

[65] Vincent D. Park, Joseph P. Macker, and M. Scott Corson. Applicability of the temporally-ordered routing algorithm for use in mobile tactical networks, 1998. 26

[66] G. Pei, M. Gerla, and X. Hong. Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility, 2000. 28

[67] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing, request for comment (rfc). World Wide Web, `http://www.ietf.org/rfc/rfc3561.txt`, July 2003. 22, 25

[68] Charles E. Perkins, editor. *Ad Hoc Networking*. Number ISBN 0-201-30976-9. Addison Wesley, 2001. 20, 22

[69] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. 1994. 20, 21, 22, 25

[70] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 80–100. IEEE, February 1999. 3, 20, 21, 24, 25, 26

[71] R. Ramanathan and M. Steenstrup. Hierarchically-organized, multihop mobile wireless networks for quality-of-service support. *Mobile networks and applications*, 3(1):101–119, 1998. 28

[72] Theodore S. Rappaport. *Wireless Communications Principles and Practice*. Number ISBN 0-13-375536-3. Prentice Hall, 1996. 7, 127

[73] Theodore S. Rappaport. *Wireless Communications Principles and Practice, Second Edition*. Number ISBN 0-13-042232-0. Prentice Hall, 2002. 1, 49, 50, 51

[74] RFM. Rfm tr1001, 868,35 mhz hybrid receiver. World Wide Web, `http://www.rfm.com/products/data/tr1001.pdf`. 35, 47

[75] Elizabeth M. Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, page 46 to 55, April 1999. 21, 27

[76] Inc Silicon Storage Technology. 29 series - 512 kbit (64k x8) page-write eeprom,sst29le512. 35

[77] Mark Stemm and Randy H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE (Institute of Electronics, Information and Communication Engineers) Transactions on Communications, Special Issue on Mobile Computing*, E 80-B(8):1125–1131, August 1997. 21, 34

[78] Ivan Stojmenovic and Xu Lin. Power-aware localized routing in wireless networks. 30

[79] Chai-Keong Toh. Associativity-based routing for ad-hoc mobile networks, 1996. 24, 27

[80] Tijs van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180, New York, NY, USA, 2003. ACM. 13

[81] Ya Xu, John Heidemann, and Deborah Estrin. Adaptive energy-conserving routing for multihop ad hoc networks. Research Report 527, USC/Information Sciences Institute, October 2000. 30, 32

[82] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of 7th Annual International conference on Mobile Computing and Networking*, pages 70 – 84. ACM, July 2001. 21, 32

[83] Wei Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. volume 3, pages 1567 – 1576 vol.3, 2002. 13

[84] Wei Ye, Richard T. Vaughan, Gaurav S. Sukhatme, John Heidemann, Deborah Estrin, and Maja J. Mataric. Evaluating control strategirs for wireless-networked robots using an integrated robot and network simulation, 2000. 18