On the Coordinated Use of a Sleep Mode in Wireless Sensor Networks: Ripple Rendezvous

Robert L. van Coppenhagen

November 2005

Submitted in fulfilment for the degree of Doctor of Philosophy

School of Electrical and Computer Systems Engineering
RMIT University
Melbourne, Australia
Abstract

It is widely accepted that low energy consumption is the most important requirement when designing components and systems for a wireless sensor network (WSN). The greatest energy consumer of each node within a WSN is the radio transceiver and as such, it is important that this component be used in an extremely energy efficient manner. One method of reducing the amount of energy consumed by the radio transceiver is to turn it off and allow nodes to enter a sleep mode. The algorithms that directly control the radio transceiver are traditionally grouped into the Medium Access Control (MAC) layer of a communication protocol stack.

This thesis introduces the emerging field of wireless sensor networks and outlines the requirements of a MAC protocol for such a network. Current MAC protocols are reviewed in detail with a focus on how they utilize this energy saving sleep mode as well as performance problems that they suffer from. A proposed new method of coordinating the use of this sleep mode between nodes in the network is specified and described. The proposed new protocol is analytically compared with existing protocols as well as with some fundamental performance limits. The thesis concludes with an analysis of the results as well as some recommendations for future work.
Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

The contents of this thesis are the results of original research, and have not been submitted for a higher degree at any other university or institution.

Robert L. van Coppenhagen
November 2005
Acknowledgements

I would like to thank all those people who have assisted and supported me as I completed my studies. Firstly I thank my supervisor Professor Andrew Jennings who has been singlehandedly patient with me as I have come to terms with the previously unfamiliar world of computer networks and the problem of exchanging information between different computing entities in this new field of distributed wireless communication. I am grateful to Professor Ian Bates and Ms. Lynda Gilbert who gave me the initial opportunity to pursue PhD candidature at RMIT University. Sincere thanks must go to Herr Doktor Heiko Rudolph for his insights, advice and being my mentor during my candidature.

Additionally, I thank my fellow research students in the RMIT building 55 research laboratory. Their stimulating discussions have sustained me through this labor. In particular I would like to thank Simon Mutzenich for his friendship and support while we have kept each other sane.

Thanks must also go to my Melbourne friends who have seen me continually drift off into a different world whenever I begin to think about my research. For their understanding and tolerance I am grateful. To my friends that are scattered all over this globe that we inhabit, I am extremely appreciative of our shared experiences as they have helped me become the person I am today.

Finally, I am indebted to my parents, sisters and brothers who have been constant sources of encouragement, motivation and inspiration through
the times when I thought that I had undertaken more than I was capable of. It has again reminded my that my family is indeed my strongest asset.
# Contents

Abstract .......................................................... i
Declaration .......................................................... ii
Acknowledgements .................................................. iii
List of Figures ....................................................... xii
List of Tables ....................................................... xv
List of Acronyms ..................................................... xvi

1 Introduction ......................................................... 1
   1.1 Wireless Sensor Networks .................................. 1
   1.2 Research Steps .............................................. 2
   1.3 Thesis Outline .............................................. 3
   1.4 Contribution of this Thesis ................................. 4

2 The Problem ......................................................... 6
   2.1 Proposed Applications for Wireless Sensor Networks .... 6
       2.1.1 The Target Tracking Task ............................. 7
       2.1.2 Forest Monitoring ..................................... 8
       2.1.3 Traffic Monitoring ................................... 10
   2.2 Design Requirements of Sensor Nodes ..................... 11
   2.3 Architecture of a Sensor Node ............................. 18
## CONTENTS

2.3.1 Power Unit ............................................. 19  
2.3.2 Sensing Components ................................. 19  
2.3.3 Computation Components .......................... 20  
2.3.4 Communication Components ....................... 20  
2.4 Energy Saving Mechanisms of Sensor Nodes .......... 25  
2.4.1 Multihop communication ............................ 25  
2.4.2 Sleep Mode ........................................... 27  
2.5 MAC protocols ........................................ 28  
2.5.1 Multiple Radio Channels ......................... 29  
2.5.2 Time Division Multiple Access (TDMA) ........... 29  
2.5.3 Unsynchronized Contention-Based ................. 32  
2.5.4 Synchronized Contention Based .................. 33  

3 Literature Survey ........................................... 35  
3.1 The Prototype Embedded Network (PEN) .............. 35  
3.2 The 802.11 Protocol .................................... 38  
3.2.1 The Application ..................................... 38  
3.2.2 MAC Frame Dialogue ............................... 39  
3.2.3 Power Save Feature .................................. 42  
3.3 The S-MAC 2001 Protocol .............................. 43  
3.3.1 S-MAC virtual cluster establishment ............... 51  
3.3.2 Implications of S-MAC 2001 ....................... 52  
3.4 The T-MAC Protocol ................................... 54  
3.4.1 Synchronization .................................... 59  
3.4.2 Implications of T-MAC ............................ 59  
3.5 The S-MAC 2003 Protocol ............................. 60  
3.6 Ripples .................................................. 60  
3.7 The DMAC Protocol .................................... 62  
3.7.1 Synchronization .................................... 66  
3.7.2 Implications of DMAC ............................ 66
CONTENTS

3.8 Conclusion ........................................... 67

4 Proposed Solution ................................. 69
  4.1 Ripples in a One Dimensional Network ........... 69
      4.1.1 Timeliness of Delivery ..................... 69
      4.1.2 Energy Efficient Adaptability ................. 72
      4.1.3 Independence of MAC layer .................. 77
  4.2 Ripples in a Two Dimensional Grid ............... 84
      4.2.1 Geo-Waves .................................. 86
  4.3 Routing Implications of Ripple Rendezvous ....... 95
      4.3.1 Unicast Traffic .............................. 95
      4.3.2 Broadcast Traffic ............................ 98
  4.4 Ripples Over a Random Topology ................. 98
      4.4.1 The Geographic Adaptive Fidelity (GAF) Algorithm 99
      4.4.2 Synchronisation ............................. 102
      4.4.3 GAF, R-MAC and Ripple Rendezvous ........... 103
  4.5 Initial Establishment of Ripple Rendezvous ...... 105
  4.6 Summary ........................................... 108

5 Comparison .......................................... 109
  5.1 Definition of performance metrics, parameters and scenario. 109
      5.1.1 Metrics of performance ....................... 109
      5.1.2 Parameters of comparison ..................... 111
      5.1.3 Comparison Scenario ......................... 115
  5.2 Performance Models .............................. 116
      5.2.1 Performance Limits .......................... 116
      5.2.2 Performance of S-MAC ....................... 119
      5.2.3 Performance of T-MAC ....................... 129
      5.2.4 Performance of Ripple Rendezvous ........... 133
  5.3 Comparison ....................................... 138
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Architecture of a Sensor Node</td>
<td>18</td>
</tr>
<tr>
<td>2.2</td>
<td>Layered model of a sensor node’s protocol stack</td>
<td>21</td>
</tr>
<tr>
<td>2.3</td>
<td>Total Power vs. Transmission distance for multiple hops [36]</td>
<td>26</td>
</tr>
<tr>
<td>3.1</td>
<td>Key to Radio State Diagram</td>
<td>37</td>
</tr>
<tr>
<td>3.2</td>
<td>A PEN Node Beaconsing to Establish Connectivity</td>
<td>37</td>
</tr>
<tr>
<td>3.3</td>
<td>802.11 MAC Frame Dialogue</td>
<td>41</td>
</tr>
<tr>
<td>3.4</td>
<td>Fragmentation and Overhearing Avoidance under S-MAC</td>
<td>46</td>
</tr>
<tr>
<td>3.5</td>
<td>Collision resolution under S-MAC</td>
<td>47</td>
</tr>
<tr>
<td>3.6</td>
<td>Packet transmission diagram</td>
<td>49</td>
</tr>
<tr>
<td>3.7</td>
<td>The reporting of an event along a long-haul multihop route under S-MAC</td>
<td>50</td>
</tr>
<tr>
<td>3.8</td>
<td>T-MAC and the Early Sleep Problem</td>
<td>56</td>
</tr>
<tr>
<td>3.9</td>
<td>T-MAC and delivering one packet through a 10 node network</td>
<td>58</td>
</tr>
<tr>
<td>3.10</td>
<td>Six nodes collaborating to form a ripple</td>
<td>61</td>
</tr>
<tr>
<td>3.11</td>
<td>DMAC’s ripple renewal mechanism</td>
<td>64</td>
</tr>
<tr>
<td>3.12</td>
<td>A Data Gathering Tree</td>
<td>65</td>
</tr>
<tr>
<td>4.1</td>
<td>An empty ripple</td>
<td>74</td>
</tr>
<tr>
<td>4.2</td>
<td>A ripple carrying traffic</td>
<td>76</td>
</tr>
<tr>
<td>4.3</td>
<td>State transition diagram of MAC modes</td>
<td>78</td>
</tr>
<tr>
<td>4.4</td>
<td>Behavior of a node in receive mode</td>
<td>78</td>
</tr>
<tr>
<td>4.5</td>
<td>Behavior of a node in send mode</td>
<td>79</td>
</tr>
</tbody>
</table>
4.6 Mode Diagram of a Ripple ................................. 80
4.7 Mode Diagram showing a left ripple giving way to a right ripple 81
4.8 State transition diagram of MAC modes for ripple that gives way ........................................... 82
4.9 Transmission range of a node in a grid network .......... 85
4.10 Phase offset for northern wave ............................... 86
4.11 Possible destinations from a node during the northern ripple . 87
4.12 Possible optimal routes between nodes in grid network .... 87
4.13 Two simultaneous intersecting routes ......................... 89
4.14 Blocked senders and receivers ................................ 90
4.15 Blocked senders and receivers for transmission to a diagonal neighbor ....................................... 91
4.16 Pseudo code for BoB transmission ............................ 92
4.17 An example of six routes travelling within a western ripple with a carrying capacity of 2 packets. ....................... 96
4.18 An example of five routes converging. ......................... 97
4.19 Virtual grid over random topology ............................. 100
4.20 Possible communication range from different positions within a virtual grid cell .................................. 104
4.21 A simple one dimensional network ............................. 106
5.1 Throughput of an Idealized protocol ......................... 118
5.2 Behavior of S-MAC 2001 ....................................... 120
5.3 Maximum throughput of S-MAC 2001 .......................... 122
5.4 Behavior of S-MAC 2003 ....................................... 125
5.5 Throughput of S-MAC 2003 ..................................... 128
5.6 Behavior of T-MAC .............................................. 130
5.7 Ripple Rendezvous delivering one packet through a 10 node network ............................................. 134
LIST OF FIGURES

5.8 Energy consumed vs. Route Length under initial protocol settings ........................................ 141
5.9 Time of Delivery vs. Route Length under initial protocol settings ........................................ 143
5.10 Energy x Delay vs. Route Length under initial protocol settings .......................................... 144
5.11 Energy vs. Route Length under modified ripple rendezvous rendezvous period ......................... 148
5.12 Delay vs. Route Length under modified ripple rendezvous rendezvous period ......................... 149
5.13 Energy-Delay vs. Route Length under modified ripple rendezvous rendezvous period ............... 150
5.14 Energy vs. Route Length for a packet of 30 bytes ................................................................. 154
5.15 Delay vs. Route Length for a packet of 30 bytes ................................................................. 155
5.16 Energy - Delay vs. Route Length for a packet of 30 bytes ..................................................... 157
5.17 Delay vs. Route Length for WaveLAN radio module ............................................................. 161
5.18 Energy vs. Route Length for WaveLAN radio module ........................................................... 162
5.19 Energy-Delay vs. Route Length for WaveLAN radio module .................................................. 164
5.20 Energy vs. Route Length for assumed low energy sleep mode of WaveLAN radio module .......... 167
5.21 Energy-Delay vs. Route Length for assumed low energy sleep mode of WaveLAN radio module ... 168
5.22 Energy vs. Route Length for reduced rendezvous period of T-MAC ......................................... 171
5.23 Delay vs. Route Length for reduced rendezvous period of T-MAC .......................................... 172
5.24 Energy-Delay vs. Route Length for reduced rendezvous period of T-MAC ................................. 173

B.1 Result of simulated annealing for a 10 by 10 grid network .................................................... 196
B.2 Cleaned rendezvous sequence for a 10 by 10 grid network .................................................... 198
LIST OF FIGURES

B.3  Diagonal rendezvous sequence for a 5 by 5 grid network . . . 200
B.4  An alternative route for a diagonal rendezvous sequence . . . 201
List of Tables

5.1 Parameters used in Performance models ............... 112
5.2 Parameters of RFM TR1001 Radio Module ............... 113
5.3 Parameters of Lucent WaveLAN PC card Radio Module . 113
5.4 Value of model variables .................................. 139
5.5 Fixed active times for S-MAC varieties with the TR1001 radio module ........................................... 140
5.6 Ripple Rendezvous maximum active times with the TR1001 radio module ........................................... 140
5.7 Energy consumed vs. Route Length under initial protocol settings ...................................................... 141
5.8 Time of Delivery vs. Route Length under initial protocol settings ...................................................... 142
5.9 Energy x Delay vs. Route Length under initial protocol settings ...................................................... 145
5.10 Energy consumption per node per rendezvous period under initial protocol settings ....................... 145
5.11 Throughput under initial protocol settings ............. 146
5.12 Energy vs. Route Length under modified ripple rendezvous rendezvous period ................................. 147
5.13 Delay vs. Route Length under modified ripple rendezvous rendezvous period ................................. 149
5.14 Energy-Delay vs. Route Length under modified ripple rendezvous rendezvous period ....................... 151
5.15 Average power consumption under modified ripple rendezvous rendezvous period ........................................ 151
5.16 Throughput under modified ripple rendezvous rendezvous period ..................................................... 152
5.17 Active times for S-MAC and ripple rendezvous varieties with a packet size of 30 bytes .......................... 153
5.18 Energy vs. Route Length for a packet of 30 bytes ............ 154
5.19 Delay vs. Route Length for a packet of 30 bytes ............. 156
5.20 Energy - Delay vs. Route Length for a packet of 30 bytes . 156
5.21 Energy consumption per node per rendezvous period for a packet of 30 bytes ........................................ 157
5.22 Throughput for a packet of 30 bytes .......................... 158
5.23 Value of model variables for WaveLAN radio module .......... 159
5.24 Active times for S-MAC and ripple rendezvous varieties with the WaveLAN radio module ..................... 160
5.25 Delay vs. Route Length for WaveLAN radio module .... 161
5.26 Energy vs. Route Length for WaveLAN radio module ...... 163
5.27 Energy-Delay vs. Route Length for WaveLAN radio module . 163
5.28 Energy consumption per node per rendezvous period for WaveLAN radio module ............................... 164
5.29 Throughput using the WaveLAN radio module .............. 165
5.30 Energy vs. Route Length for assumed low energy sleep mode of WaveLAN radio module ...................... 167
5.31 Energy-Delay vs. Route Length for assumed low energy sleep mode of WaveLAN radio module ............. 169
5.32 Energy consumption per node per rendezvous period for assumed low energy sleep mode of WaveLAN radio module ........ 169
5.33 Energy vs. Route Length for reduced rendezvous period of T-MAC .................................................... 170
LIST OF TABLES

5.34 Delay vs. Route Length for reduced rendezvous period of T-MAC ........................................ 172
5.35 Energy-Delay vs. Route Length for reduced rendezvous period of T-MAC ................................. 173
## List of Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>ACKnowledgement</td>
</tr>
<tr>
<td>BP</td>
<td>Backoff Period</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access (with) Collision Avoidance</td>
</tr>
<tr>
<td>CTS</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>CW</td>
<td>Contention Window</td>
</tr>
<tr>
<td>DIFS</td>
<td>Distributed Inter-Frame Space</td>
</tr>
<tr>
<td>DMAC</td>
<td>Data-gathering Medium Access Control</td>
</tr>
<tr>
<td>FRTS</td>
<td>Future Request To Send</td>
</tr>
<tr>
<td>GAF</td>
<td>Geographic Adaptive Fidelity</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>NIC</td>
<td>Network Interface Card</td>
</tr>
<tr>
<td>PEN</td>
<td>Prototype Embedded Network</td>
</tr>
<tr>
<td>RTS</td>
<td>Request To Send</td>
</tr>
<tr>
<td>SIFS</td>
<td>Short Inter-Frame Space</td>
</tr>
<tr>
<td>S-MAC</td>
<td>Sensor Medium Access Control</td>
</tr>
<tr>
<td>SP</td>
<td>Short Period</td>
</tr>
<tr>
<td>T-MAC</td>
<td>Timeout Medium Access Control</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Wireless Sensor Networks

At the time of writing, the search engine ‘google’ revealed about 477 million hits for the word ‘wireless’. The more academic ‘google scholar’ search engine gives about 587 thousand hits and the IEEE Xplore database returned just over 42 thousand articles relating to wireless research with 14 thousand of these featuring ‘wireless’ in the title. There is an enormous amount of interest in wireless communication.

A wireless sensor network is a network of sensing devices that allows information to be retrieved from the environment in which they are deployed. A user of the sensor network will in effect have their own senses augmented by the sensor network to provide them with digitally enhanced sensing abilities. There are many different challenges that need to be met prior to the realization of this technology. This thesis starts by examining a wireless sensor network from the perspective of the required functionality and gradually narrows its focus until it has one of the most significant problems squarely in its sights.

Then the fun begins ...
1.2 Research Steps

The steps taken up until this point reflect the highly dynamic nature of research into wireless sensor networks. This research project began with a review of many articles relating to wireless sensor networks until the problem of energy efficiency at the Medium Access Control layer was identified and deemed appropriate for PhD-level research. Soon after this, a proposed solution was developed after which a combinatorial optimization technique was used in an attempt to validate the proposed solution. This technique identified another possible solution, so an effort was made to simulate the proposed solution and compare this with an early version of the S-MAC protocol.

While the simulation was being developed, the T-MAC protocol and a later refinement to the S-MAC protocol were published causing the candidate to redefine the problem space. Much background knowledge was also gained at this time about computer network communication through tutoring duties the candidate performed.

Following this, the candidate was forced to refine the proposed solution in light of the new developments. The refined solution went beyond the capabilities of the developed simulation and all work on this aspect of the project was abandoned. It was thus decided to develop mathematical models of the protocols for their analysis.

The analytical models were used to generate results that were published in a paper titled ”Ripple Rendezvous for Wireless Sensor Networks” that formed the basis for a presentation titled ”Real time situational Awareness with Wireless sensor Networks” given at the 2005 Land Warfare Conference [52].
1.3 Thesis Outline

Chapter 1 has provided a brief introduction to the general theme of this thesis followed by a look at the research steps taken in completing this PhD project. This section gives an outline of the chapters and sections that make up this document followed by a section detailing the contributions that this thesis makes towards the eventual realization of wireless sensor networks.

Chapter 2 provides a definition and explanation of a wireless sensor network. Three example applications are given and design requirements are extracted from these. The reason for our concern with energy consumption is established as are some energy-saving mechanisms used within sensor networks. The concept of a protocol stack is introduced and our focus on the Medium Access Control (MAC) layer is established. The chapter concludes with a brief review of many different types of MAC protocols.

Chapter 3 provides an in depth examination of current uses of sleep mode and the selected subset of MAC protocols associated with them. The chapter begins with the protocol from which we derive the terminology used in this thesis and then moves onto a detailed look at the IEEE 802.11 MAC protocol. The S-MAC protocol is then examined and the important findings and mechanisms of this protocol are discussed followed by a look at the T-MAC protocol which builds heavily on S-MAC. While being very much more energy efficient than S-MAC, the T-MAC protocol does constrain the performance of the network considerably and this problem is discussed. A construct that is helpful for describing coordinated activity within a sensor network is introduced before the DMAC protocol is reviewed along with associated problems. The chapter concludes with some further design requirements of our proposed protocol.

Chapter 4 introduces the solution proposed by this thesis and begins by explaining its operation in a one dimensional network illustrating how the protocol meets the design requirements established at the end of the
previous chapter. The following section expands on the proposed solution by extending the solution to a two dimensional grid network. A coordination strategy is proposed along with the implication for higher layers in the protocol stack. A method of implementing this strategy in a network that has a random or ad hoc topology is discussed. This section makes use of the GAF topology control protocol which is explained in relevant detail.

Chapter 5 begins with a discussion of the comparison scenario along with performance metrics used as well as parameters used in the comparison. The next section is perhaps the most difficult to read as it establishes the various analytical performance models used to compare the selected protocols. The final section shows the comparison of current solutions with the proposed solution of this thesis. The comparison begins with an initial set of parameters which are modified to allow the reader to see how the protocols perform against each other as the parameters change. Finally the thesis is summarized and concluded in Chapter 6 where directions for future work are suggested.

1.4 Contribution of this Thesis

The original and novel contributions made in this thesis to the field of medium access control communication protocols for wireless sensor networks are:

• A strategy for organizing the activity of nodes within a network that seems to be appropriate for very large wireless sensor networks consisting of many thousands of nodes and covering areas of the order of square kilometers or greater

• The proposed protocol suite is evaluated in terms of both energy efficiency and the delay of information delivery and offers a better trade off between these two competing metrics.
• The proposed protocol operates independently of and makes minimal demands on the layers above and below it.
Chapter 2

The Problem

2.1 Proposed Applications for Wireless Sensor Networks

Generally speaking, a sensor network consists of a number of small devices, each capable of sensing, computation and wireless communication. These devices are deployed in the absence of existing infrastructure throughout an area about which information is to be sought. From this point on, each application for wireless sensor networks introduces its own set of requirements that influences design of the sensor nodes and thus the sensor network as a whole. Many small to medium scale applications have already been developed for prototype wireless sensor networks. These applications tend to be of the type that can be performed by wired sensor networks, but are more suitable to wireless sensor networks due to the cost and logistical advantage of not requiring the installation of cable infrastructure. Applications such as remote habitat monitoring of wildlife with a network of 32 sensor nodes [33] and the structural health monitoring of an office building with 25 sensor nodes [58] are examples of relatively small prototype wireless sensor networks. At the time of writing, the largest sensor network deployed to date is the Extreme Scale (ExScal) network by Ohio State University in which some 1000 sensor nodes were deployed over an area of just under 0.4 square kilometers [38].
Our focus is on large sensor networks, larger even than the ExScal network. In this context we regard remote surveillance and monitoring as the most challenging set of applications. Three example applications are provided to give the reader a high level perspective of the required functionality of a sensor network. Firstly, the target tracking task is discussed. This application is one of the most challenging applications for a sensor network and if this application can be supported, then almost any application can be. The forest monitoring application is another large scale application for wireless sensor networks, and finally the section concludes with the traffic monitoring application.

2.1.1 The Target Tracking Task

The target tracking task involves the rapid and dense deployment of many sensor nodes over the area to be monitored which is henceforth referred to as the sensor field. The deployment is likely to be carried out in an ad-hoc fashion from one or more air-borne vehicles eg. helicopters or Unmanned Airborne Vehicles (UAVs) to ensure full coverage of the sensor field with a relatively uniform dispersion density of sensor nodes. Once the nodes have been deployed in the sensor field, they organize themselves into a cooperative sensor network. After the network has self organized itself, it becomes ready to be used.

The users of the network may be located away from the sensor field and interact with the network through a fixed gateway node that connects the sensor network to the internet at large. Alternatively they may be co-located in the sensor field and may either be fixed (eg. a command center) or mobile (eg. a squad on patrol). A user within the sensor field will interact with the network by through their personal and portable wireless device.

A user of the network will enter the specifics of the required sensing task into the interface device which will then inject this task into the sensor
network. For the purposes of demonstration, a case in which users are interested in tracking animals is used. A sensing task can be of the form of a one-off report on the status of the environment within the network, eg. "Give me the number and location of four legged animals in region X, Y" where X and Y are two geographic points defining a rectangular subregion of the sensor field. Alternatively, the task may be of a periodic reporting nature, eg. "Give me an update every five seconds of the location and velocity of all horses within the sensor field for the next 10 minutes".

The region of interest is defined as the region within coordinates X and Y in the first example or the entire sensor field for the second example. It is that subset of nodes that is required to perform the actual sensing of the environment. This sensing task is propagated to the relevant sensor nodes in the region of interest who set about activating their appropriate sensors and begin 'listening' for the relevant signal. The sensors may be vibration sensors which are able to detect four legged animals by their identifiable seismic footprints [20]. When an event that the sensor node has been been tasked to be on the lookout for occurs within the region of interest, this information is returned to the user who originally tasked the network to perform this function.

The reader can now perhaps begin to have an appreciation for the required functionality of a sensor network and the nodes that constitute it. Before we look at the implications of the required application we present two more example of applications that are proposed for wireless sensor networks.

2.1.2 Forest Monitoring

A forest may be monitored by a wireless sensor network for several reasons. Firstly, to gather environmental data on the health of the forest eg. humidity, temperature, soil moisture, the presence of harmful chemicals etc. Secondly, to monitor the ecosystems within the forest such as native an-
The Problem

Animal population demographics. Thirdly for the purposes of protecting a plantation from theft, intrusion or damage. Finally, the most challenging application, the forest may be continuously monitored to detect an outbreak of fire.

Of these four example applications, we concentrate on the forest fire monitoring task. This application also involves the dense deployment of many sensor nodes throughout the sensor field, however rapid deployment may not be required. It may be possible to specifically place each sensor within the sensor field rather than deploying them from the air in an ad hoc fashion. Similarly there may by some assistance given to nodes in self organizing into a network and the sensor nodes will not be required to rapidly self organize since the network is unlikely to be deployed while there is a fire.

The users of the sensor network may be initially located far from the forest and be contacted by a gateway node that connects the sensor network to the outside world. When a fire is detected, an alert is sent from the nodes that detect the fire (and confirm this with other co-located nodes to avoid false alarms) to the gateway node, which then sends the alarm to the relevant system. When fire-fighting crews arrive at the forest, they interact with the sensor network directly to extract information regarding the current status of the blaze to assist extinguishing the fire. The fire fighting personnel will interact with the network directly in a fashion similar to the mobile ground-level users in the target tracking task by using a portable interface device mounted either on their person or carried within the fire fighting vehicle. The interface device will be used to query the network and receive the results of such network interrogation.

Apart from monitoring remote and possibly hostile environments, monitoring traffic within a city is another potential large-scale application of wireless sensor networks. Such an application is described next.
2.1.3 Traffic Monitoring

Monitoring traffic for the purposes of reporting and control within a city is another potential application of a wireless sensor network. This application involves the large-scale embedding of many sensor nodes into the road surface for the purposes of detecting the traffic that is present on the roads and then communicating this information to traffic control devices such as traffic lights which actuate their signals based on the detected traffic to achieve the desired traffic behavior.

The desired traffic behavior may be based on trying to optimize many different metrics of the performance of the traffic system. Firstly the actuation of the traffic system may be such as to maximize the throughput of traffic. The traffic system may behave so as to minimize the average delay of each vehicle or to reduce average mean square delay meaning that longer delays will be weighted more heavily. Alternatively the aim could be to reduce the traffic blocking rate which means that the traffic system will aim to prevent as few cars as possible from being stopped by the traffic lights.

Regardless of the aims of the traffic control system, the wireless sensor network is responsible for providing the information about the current state of traffic to the control algorithms at the actuators so that an intelligent decision can be made to optimize the chosen performance metric for the traffic control system.

For such an application the sensor nodes will be individually embedded into the road surface and thus a certain degree of network customization may be possible. However, the nodes may be placed to maximize their sensing abilities rather than their communication abilities. The information that the sensor nodes produce (about the traffic that they have sensed) will need to be regularly delivered to the actuation mechanism where the control decisions are made.

A wireless sensor network is a special category of wireless networks
that has several characteristics that differentiate it from other types of wireless networks. In the next section, the distinguishing features of sensor networks are revealed as the three applications are reviewed considering the implications of the desired functionality of wireless sensor networks.

2.2 Design Requirements of Sensor Nodes

The first constraint that becomes apparent when considering the applications in the previous section is the cost of the sensor nodes. In the target tracking task, the sensor nodes may be disposable. It may not be practical to collect the sensor nodes once the need for the sensor network is no longer present, especially as the number of sensor nodes deployed becomes large. For the forest monitoring application, some sensor nodes will be destroyed in the event of an outbreak of a fire within the forest. It is thus important that each sensor node be cheap enough to be regarded as being disposable.

For the target tracking task, the sensor nodes are required to be as small as possible. This is because firstly, to make the handling of many nodes simple, it is more convenient to manoeuvre the nodes en masse if they are each very small. The second reason is so that once they are deployed, they are as unobtrusive as possible so as not to disturb the phenomena being sensed. The size of the sensor nodes is less important for the other two applications. For monitoring traffic, since the nodes are embedded into the road, size does become a concern and the smaller they are, the easier they will be to deploy. Similarly for monitoring a forest smaller nodes are much easier to deploy than larger ones.

Once the nodes have been deployed in the sensor field they will need to organize themselves into a network that will facilitate the movement of information through it. Again, this requirement is the most stringent for the target tracking task in which nodes are deployed in an ad hoc manner and are left completely unattended until the network begins to be used. It
is also important that nodes organize themselves quickly since a particular target or sensing task may be the reason for deploying the network in the first place.

It is very important that the algorithms and operational routines that are developed for a wireless sensor network are still able to function as the network increases in size. It is entirely possible that for the three applications described, that a sensor field of the order of tens of kilometers will be employed, if not more\(^1\). It is important that the observed behavior of a small sensor network scale appropriately such that a sensor network with orders of magnitude more sensor nodes can function properly.

The nodes in the three applications described are not mobile. For the traffic monitoring application in which nodes are embedded into the roads surface, this is an obvious statement. For the target tracking task and to a lesser degree the forest monitoring application it does need to be said that while nodes are stationary for the vast majority of their operational life, that may be occasionally moved by events that take place within the sensor field, eg. a sensor node may be kicked down a hill by an animal moving around within the sensor field. These movements are not likely to occur very often and when they do the sensor node will once again be stationary in its new position.

Continuing along a similar theme, since the sensor networks will be operating in the real world, there is a need for them to adapt their behavior to accommodate events that occur or conditions that change. For example, considering the forest monitoring application, in the event of a fire some sensor nodes may be destroyed and the sensor network will have to modify its behavior to cope with the loss of those nodes that have been destroyed.

Apart from device failure, the radio signal may not be received by the

\(^1\)The ultimate size of a sensor network will be determined primarily by the cost of sensor nodes but also by just how useful they turn out to be. As such, it is difficult to specify a maximum size of a sensor network.
intended destination. A wireless channel is a notoriously unreliable medium for transmitting information through. A radio signal may be blocked by some physical mass. If this physical mass is stationary (e.g., a large rock or a stand of dense foliage) then the network will have to adapt to this permanent lack of communication between nodes that would otherwise be able to reach each other. If the channel is only temporarily blocked maybe due to some mobile physical mass (e.g., a large animal, group of animals, people or vehicles or even heavy rain) or perhaps due to a burst of noise in the channel, then the network must be able to cope with this change in the availability of wireless connectivity.

The sensor network should not constrain users of the network. In both the target tracking task and forest fire fighting application, any user of the network should be able to task any region of interest with a request for information from anywhere within the sensor field.

There is a need for the sensor network and thus each sensor node to have a long lifetime. For the applications discussed, nodes need to be able to function for long periods of time. The target tracking task may require the sensor network to last for days, months or years. For both the forest monitoring and the traffic monitoring application a network lifetime of several years is required.

Information must be able to move through the sensor network quickly. For the target tracking task, the request for information should be sent to the nodes within the region of interest rapidly so that the nodes can activate their sensors and begin to make note of the relevant events as they occur. The sensed information should also be returned to the user quickly, since the animals are mobile and the utility of the information decreases as it becomes out of date or 'stale'. Similarly, a sensor network that is monitoring a forest fire should allow this information to be reported to the relevant place in a

\[\text{the sensor network should both report the fire quickly to the gateway, and report the status of the blaze to the fire fighting crew quickly as a result of mobile network}\]
The Problem

timely manner since the lives of the fire-fighting crew may be endangered by consuming stale information. When monitoring traffic, a sensor network should allow the actuating decision makers to receive this information with only a small delay. This will avoid actuation decisions being made using information about the state of traffic that is stale which will degrade the performance of the traffic control system as a whole.

This brings us to the most important design requirement of sensor nodes. Since each node communicates via a radio transceiver, they do not require any existing cabling to be able to communicate with each other. This infrastructureless deployment of sensing and computing resources is one of the principle advantages of a wireless sensor network. The downside to this is that there are also no cables to bring electrical energy to each node and they must therefore rely on their own on-board energy supplies. To meet the small form factor requirement, it is not possible to attach large battery packs to each node. Once this energy supply is exhausted, a sensor node will no longer be able to function for some time at least since it is possible for sensor nodes to scavenge energy from their environment. Solar or vibration energy may be harvested, but the amount of energy derived from these sources is small and the performance of the network will be degraded while the node is not in an operational condition. It is thus imperative that sensor nodes use as little energy as is possible and that all systems that comprise a sensor node be designed to be as energy efficient as possible.

The constraints and design requirements of the wireless sensor network under consideration in this thesis are given in the following list. This is not meant to be an exhaustive list of all issues under consideration in the wireless sensor network research community, but a list of design requirements that are relevant to the research documented in this thesis.

Cost In order to be deployed on a large scale and to be disposable, each
sensor node will need to be as cheap as possible. A popular target is for each sensor node to cost less than a dollar [40]. One of the ways to achieve this per unit cost of one dollar is to allow for economies of scale and produce sensor nodes on a large scale as is required by the example applications given. Since this thesis is concerned with methods of communication, cost has little bearing.

**Small Size** In order to be deployed on a large scale, sensor nodes should be small. Some researchers aim for a size of 1 cm$^3$ [40], while [1] and [22] report that a size that is small enough to be suspended in the air may be required. This thesis is concerned with wireless communication protocols which have little bearing on the size of nodes.

**Self Organizing** Nodes will need to be able to organize themselves without aid. The time taken to self organize is important since this determines how soon after deployment a sensor network can be used. This property may be important depending on the target application. The forest monitoring task may be able to tolerate a time of hours or even days, whereas the target tracking task may require a network setup time of minutes or even seconds (in desperate situations). This requirement used to guide the design of our communication algorithms such that the network is able to support a wide range applications, some of which may require rapid network establishment.

**Scalability** It is vital that the operating procedures and communication protocols scale well. Techniques that rely on knowledge of the extent of the entire network or the location of each sensor node may be acceptable for small sensor network, but as the size of the network and the number of sensor nodes increase (from thousands to millions), these techniques make large demands of the nodes and become cumbersome and sometimes unworkable. This issue is very important with
The Problem

the solution presented in this thesis relying on issues relating to scaling rather than working against them.

Mobility While we will focus on situations in which nodes are not mobile, they may have their location modified by some entity or event that occurs within the sensor field. It is important that this does not degrade the sensor network. A solution is presented in which it is assumed that sensor nodes do not often move, however, there is a process that allows the network to cope with such a change.

Adaptability It is very important that a sensor network can cope with changes both within the sensor field as well as with changes in the demands that are made of it. A sensor network is likely to stand unutilized for long periods of time interspersed with periods of heavy use. The network must be able to modify its behavior to cope with these changes. Individual sensor nodes may cease to function or radio propagation conditions may change and these too, must be dealt with to maintain the integrity of the network. Hence adapting to changing conditions is a strict requirement.

Robustness The performance of the sensor network as a whole must be relatively unchanged in the face of both individual node and/or communication link failures. A node may fail due to the depletion of their energy supply or the actual device itself may be destroyed by some event that occurs in the sensor field. It is important that the performance of the application is not significantly affected in the event of such failures.

Rapid Information Delivery The applications that have been examined involve reporting on highly dynamic situations and it is essential that information be able to move rapidly through the network. The latency requirement is based on the speed of the phenomenon being reported
on and the faster a sensor network is able to report information, the more applications will become possible. It is believed that a latency in the order of seconds should be sufficient to support all but the most time-critical of applications. For situations in which a sensor network is being used for distributed data logging (such as habitat or environmental monitoring), this requirement for speedy information delivery becomes less important. However, even for these applications, speedy information delivery that results in a more current picture of the situation being monitored would be useful. Thus timeliness of delivery of information is a crucial design requirement of wireless sensor networks operating in a dynamic environment.

Extended Lifetime It is not feasible to attend to individual sensor nodes once they have been deployed and they must be able to function unattended for long periods of time. There are distinctive phases during this lifetime. First, the devices must be produced but not activated, for fear of diminishing their limited energy supply. Devices are activated just prior to deployment in the pre-deployment phase. This may be by some kind of induction loop that closes a circuit to activate each node\textsuperscript{3} or to even charge the nodes such that they begin to function. At this stage they can be programmed with their operational parameters. The third stage begins once the nodes have been deployed on the sensor field. This is the self-organization stage during which individual nodes discover each other and form a network. The fourth stage is when the network is used.

Low Energy Consumption Since the easiest way for a sensor node to reach the end of its operational lifetime is for it to exhaust its on board energy supplies and given the fact that the small form factor requirement means that each node cannot carry an abundant store of

\textsuperscript{3}activating millions of nodes by hand is not feasible
energy, it is absolutely imperative that each sensor node consume as little energy as possible. It is possible that the energy supply may be the largest component of a wireless sensor node (given that electronics can be miniaturized), thus the size of the energy supply may be the defining factor in satisfying the small form factor requirement.

In the next section the architecture of a sensor node is investigated and the sources of energy expenditure are examined. The largest source of energy consumption in a sensor node is also determined and then some strategies to reduce the energy consumption of a sensor node are introduced in the following section.

2.3 Architecture of a Sensor Node

In this section the architecture of various components that go towards making up a wireless sensor node are examined. We describe all parts of a node, leading to the section that is the focus of this thesis.

The diagram presented in figure 2.1, reproduced from [57], which presents the figure as the system architecture for a generic sensor node. We next examine each of the functional blocks in turn.
2.3.1 Power Unit

This is the device or system of devices that provides the energy by which the other three functional blocks are able to operate. The power unit typically includes a battery or capacitor in which energy is stored prior to deployment. Energy may be extracted from the environment while the node is deployed in the sensor field and this is known as energy scavenging.

Energy may be scavenged from a range of sources such as solar energy from the sun or vibration energy from the ambient environment. Please refer to [40] for a more comprehensive look at energy sources and energy scavenging techniques for wireless sensor nodes.

2.3.2 Sensing Components

These are the components that interact with the physical environment and produce a signal based on some environmental phenomenon. Each sensor node can have one or many sensors depending, again, on the application for which the sensor network is being used. With the advance in recent years of Micro Electro-Mechanical Systems (MEMS) technology has come an exciting new range of microscopic sized sensors that can determine a whole range of environmental demographics, including: temperature, humidity, vibration, light, sound, chemical and biological to name a few.

The analog signal from one sensor may be filtered or combined with the signal from another sensor^4 and is digitized by the analog to digital converter (ADC) contained within the processing unit. Ideally, the sensor devices have extremely low power requirements such as the vibration sensor used in [44] or better yet, the sensors may be able to use the energy from the phenomenon being sensed to produce the signal that is sent to the ADC. This thesis does not examine the technical details relating to the sensing unit of a node.

^4this is known as data fusion
2.3.3 Computation Components

These computation components include the micro processor that supports the operating system and the run-time algorithms that are used by the sensor node to govern its behavior. Both volatile and non volatile memory are respectively used for the temporary and longer-term storage of information.

Much work has been performed already in producing an energy efficient operating system for sensor nodes. TinyOS [63] [1] has been developed as an event driven operating system that is capable of dynamically altering its energy consumption in accordance with the level of work that it is required to perform. Please refer to [18] and [17] for a detailed discussion of the architecture of a sensor node including the TinyOS operating system.

To reduce the energy consumption of the computation components, a decision needs to be made that is again dependent on the application that is being performed. For applications that do not require many processor operations or instructions, using the metric of average power consumption per clock cycle may be applicable. However, for applications that do require a significant amount of processor operations, then the metric of energy consumption per operation becomes important. This, as [57] asserts, is the difference between low power and energy-efficient operations of the processing unit.

This functional block is connected to the sensing block via the analog to digital converter and is normally connected to the radio via a bus.

2.3.4 Communication Components

This block contains not only the physical radio hardware but also the routines within the processing unit that govern the behavior of the radio module. Conceptually, the communication components can be viewed as a protocol stack presented in figure 2.2 as reproduced from [61].

An explanation and discussion of a protocol stack may appear un-
A communication protocol is a standardized set of rules (or procedures) that govern the transmission of information between computing devices; ensuring, that if correctly followed, the aims of the protocol will be met. Routing, addressing, error detection, flow control and other communication tasks are described by protocols. Traditionally, protocols concerned with meeting similar goals are grouped together into layers and information is exchanged between computers (A and B) by having each layer on machine A communicate with the corresponding layer on machine B.

Each layer receives a protocol data unit (PDU) from the layer above, adds its own header (and sometimes footer), the format of which is always the same but the contents are dependent on the information received from the layer above and the layer’s own logic and passes it along with instructions to the layer below. Grouping similar protocols together like this allows a layer to be changed without requiring the entire protocol stack to be redeveloped. [61] holds that such a layered architecture is important for the design of a flexible protocol stack for wireless sensor networks. In essence each layer deals with a different abstraction of the problem of exchanging
information between two computers via a network.

The **physical layer** at the bottom of the protocol stack includes the physical interface between the computing device that wishes to send information to another machine and the medium or network to which it is connected. The specifics of the medium through which the device will transmit, the digital to analogue converter, the data rate, bit error rate, range of communication and other similar matters are determined by the operating parameters of this layer. According to [5] and [48], the physical radio module is the single largest source of energy consumption within a wireless sensor node. In the next section two techniques for reducing the energy consumption of the physical layer are discussed.

This thesis is primarily concerned with the **data link/medium access control (MAC) layer**. The role of the data link layer is to reliably deliver the PDU received from the network layer on the source device to the network layer on a specified destination machine [49] and [50]. It is access to the network that is a large consumer of energy in wireless sensor networks since accessing the medium requires the use of a node’s radio transceiver. This layer does not control how much energy a node consumes since it is obligated to send information received from the network layer. If the application requires a great deal of information to be moved through the network then nodes will consume much energy, however if nodes are not required to carry lots of information, less energy will be consumed. For example, referring to the target tracking task, a one-off report on the status within a particular region would consume less energy than reporting of a regular nature.

The MAC layer directly controls the physical layer and is responsible for ensuring that it is used in an energy efficient manner. The aim of this thesis is to investigate maximizing the energy efficiency of the use of the radio transceiver while impacting on the performance of the sensor network.
as little as possible. The protocol data unit for this layer is referred to as a *frame*. An introduction to the broad range of MAC protocols is given in section 2.5 with a detailed description of the functionality of a selected subset of these protocols given in the following chapter.

The layer that sits above the Data Link layer is the **Network/Routing Layer**, which is concerned with routing information through the network i.e. it is this layer that determines the subset of nodes that will be used to ‘carry’ this information through the network to reach the eventual destination. It is this layer that determines which node is the next node to receive the information as it is directed from source to destination. If a link is not available (due to noise or physical blocking as described in the previous section or because the device has failed or been destroyed) the logic and routines of this layer are responsible for selecting an alternate node to send the information to. In other words the responsibility for being adaptive to link failure belongs to the network/routing layer and thus there is a need for dynamic routing or routing on demand. This layer determines how energy consumption is distributed throughout the network.

To reduce the energy consumption of individual nodes at this layer, efficient route discovery algorithms should be used that do not require large amounts of information to be exchanged between nodes. To reduce the energy consumption of the network as a whole, routes should be used that require as few nodes within the network as is possible while maintaining a suitable link quality. To increase the lifetime of the network\(^5\) the network layer needs to use a variety of routes between a given source and destination node. Information at this layer is encapsulated into *packets*. A brief explanation of directed diffusion, an opportunistic and adaptive network layer routing protocol that has been developed specifically for wireless sensor networks, is given in appendix A.

---

\(^5\)network lifetime is a difficult metric to quantify, but can be generally defined as the time from deployment until the \(n^{th}\) sensor nodes fails due to exhausting its energy supply
The Problem

The **Transport layer** is responsible for end to end connectivity. In other words, the transport layer makes sure that the message that is sent by a source device is received by the specified destination device, regardless of the intermediate devices that help to route or carry the information.

The **Application layer** encapsulates the message from an application running on the source device and uses the services of the layers below it in the protocol stack to ensure that it is delivered to the specified application that is running on the destination device.

Both the transport and application layers can contribute to reducing the energy expenditure of each node by reducing the additional overhead that they contribute to the communication within the network. The information units that are used by these higher layers are commonly referred to as **datagrams**.

It should be said that this five-layered model of a protocol stack is well known to networking professionals as the standard five layer TCP/IP protocol stack. This model was developed in the context of wired network that do not have nearly the strict constraints that wireless sensor networks have. As Stallings [49] points out, the separation of protocols into layers does introduce inefficiencies as well as some duplication of services and data. This is the reason for the investigation of cross-layer optimization in the protocol stacks of wireless sensor nodes [55]. The drawback to optimizing the protocol stack is that it becomes less flexible in supporting different applications. This is why [61] maintains that grouping protocol functionality into layers as in figure 2.2 allows individual layers to be changed without requiring the redevelopment of the entire protocol stack. This represents one of the many trade offs in wireless sensor networks, the more optimal the protocol stack for one application, the less suitable it is for others.
2.4 Energy Saving Mechanisms of Sensor Nodes

As promised in the previous section, these are some methods of reducing the energy consumption of each node and the network as a whole.

2.4.1 Multihop communication

The first energy saving mechanism that has an influence on the work presented in this thesis, is the way that information is propagated through the network.

The energy ($e$) required to transmit a certain distance ($d$) can be roughly described by equation 2.1

$$e = d^n$$  \hspace{1cm} (2.1)

where $n$ has a value of between 2 and 5 depending on the radio propagation characteristics of the local terrain [62].

In free space, the exponent is 2 and as there are more impediments to the propagation of electromagnetic radiation that cause reflection, diffraction and scattering of the radio signal, the exponent tends upwards [42]. These value of the exponent $n$ is determined by factors that include; 1) the transmission frequency, 2) the antenna characteristics, 3) the presence and location of electromagnetic scatterers relative to the radio transceivers\(^6\) and 4) local terrain features [42].

Rather than transmitting the entire distance, if the sending node were to transmit with enough power to reach a closer intermediate node, and that intermediate node were to relay that message with enough transmit power to just reach the ultimate destination, then the total energy consumed would be less than the energy required to transmit the information directly from the sending node to the destination node. This is the basis for what is known as multihop communication.

\(^6\)both the radio that is transmitting and the radio that is receiving
Min and Chandrakasan point out in [36] however, that equation 2.1 is only an approximate energy model for radio transmission and does not take into account the fixed costs of producing a signal and thus, multihop communication does not always save energy. These observations are reproduced from [36] in figure 2.3.

What is argued in [36], is that each hop must be of a distance that is between an upper and a lower bound for multihop to be an energy efficient method of communication. These bounds are determined by the parameters of the physical radio module that is used. This condition is acknowledged and a technique known as topology control is discussed in the next subsection. This techniques is used to ensure that the transmission distance falls within the energy efficient range.
2.4.2 Sleep Mode

While it is actively operating, the radio transceiver of a sensor node can be in one of three states; it can be transmitting information; it can be receiving information; and it can be standing idle, during which it is neither transmitting or receiving information. A fourth state is possible, in which the radio transceiver is turned off and thus consumes much less power. This low power state is known as sleep mode and while it is an energy saving technique it must be used intelligently since the radio is not able to send or receive information while in this state and more importantly, a node is not able to be contacted to inform it that there is information that it needs to receive.

During this sleep mode, a sensor node powers down all non essential components as well as the radio transceiver to reduce its average power consumption. This thesis does not examine powering down components other than the radio transceiver. Since the radio transceiver can be the highest energy consumer of a sensor node, this is not believed to limit the findings of this thesis. It is the purpose of this thesis to develop a protocol that uses a sleep mode to save as much energy as possible while impacting on the performance of the network as little as possible. Within the protocol stack, each layer may be able to utilize sleep mode as an energy saving mechanism. The implications of this are now discussed.

If the application layer were to use a sleep mode, this would mean that all nodes on the sensor field that are running a particular application would become active, leaving nodes that are not running the application to save energy by entering sleep mode. Since all nodes on the sensor field are likely to be running the same application, this is not a terribly sensible idea. However, in the future this may be a possibility.

For the transport layer to utilize a sleep mode, this would mean that all nodes that are involved in transporting information from a particular
source node to some destination node\(^7\) would remain active while those that are not, are able to enter sleep mode to save energy. The author is not aware of any existing protocols that behave like this.

There are existing network layer protocols that use a sleep mode as a topology control protocol ([60] and [5]). Topology control is an effort to limit the number of neighbors that each node has within its local radio space. This is supported by the fact that some nodes that are closely co-located may be allowed to enter sleep mode without significantly affecting the coverage or the connectivity of the sensor network. Topology control protocols also provide some structure to the topology of a network that has randomly distributed nodes. In doing so, they are also able to ensure that for a given transmission range, that the node that actually receives the message lies within the energy efficient range as discussed in the previous subsection. A relevant discussion of the workings of the GAF topology control protocol, as presented in [60], is given in sub-section 4.4.1.

This brings us to the MAC layer which is the primary focus of this thesis. This is the layer that is most suited to using a sleep mode, since this layer is responsible for directly controlling the physical radio module when there is information to be sent or received. As such there are many MAC protocols that utilize a sleep mode to save energy and these are reviewed in the next section.

### 2.5 MAC protocols

Langendoen and Halkes present a detailed survey and comparison in [28] of many MAC protocols that have been proposed for wireless sensor networks. In a similar manner, various MAC protocols that utilize a sleep mode are discussed.

\(^7\)including all the intermediate relay nodes
2.5.1 Multiple Radio Channels

The Self-organizing Medium Access Control for Sensor networks (SMACS) [48] protocol effectively assigns a separate channel to each link or pair of nodes within the network using Frequency Division Multiple Access (FDMA). This requirement for many channels means that the radio consumes significantly more energy than a single channel radio [28]. The PicoRadio project [40] at the University of California (Berkeley) also utilizes a multiple channel radio in which orthogonal codes ensure collision-free channel access with Code Division Multiple Access (CDMA) as well as an always-on low power wakeup radio [28]. The Sparse Topology and Energy Management (STEM) [45][46] protocol uses two radio channels for sensor network communication. One channel is a control channel and the other is for transferring data. Nodes are not synchronized and thus have no idea as to when their neighbors will be awake. The solution to this is to have the control radio becoming active frequently to check if there is information for them to receive. Miller [35][34] also proposes using a wakeup radio as well as a main data radio. To reduce energy consumption, the wakeup radio is mainly in sleep mode, but periodically turns on to detect the presence of a signal indicating that it needs to turn on its data radio. This protocol does not function well in a multihop environment and results in a high latency.

All of the protocols mentioned in the preceding paragraph make use of multiple radio channels. Since we are interested in minimizing the energy consumption of a sensor node's radio transceiver, we are more interested in protocols that use a common radio channel. These are examined in the following paragraphs.

2.5.2 Time Division Multiple Access (TDMA)

Apart from using multiple channels to limit the effect of collisions, a MAC protocol can also be organized such that each node has a dedicated time
within which it is allowed to transmit information. Several MAC protocols utilize a TDMA format and these are now discussed.

The protocol proposed by Arisha et al. [2] has nodes forming into clusters in which each cluster has a fixed cluster-head that is connected to a wired back bone. The cluster head is responsible for assigning a TDMA slot to each node within the cluster. This protocol is unsuitable for sensor networks that are remotely deployed over a large area due the wired connectivity requirement of the cluster heads. Similarly, the Bit-Map-Assisted (BMA) MAC [30] also organizes nodes into clusters. Within each cluster, the cluster head assigns a TDMA slot to nodes that have information to report. Once the cluster head has gathered the information it transmits this directly to the base station. This protocol does not utilize multi-hop routing to report information from the cluster-heads to the base station. A Self-Stabilizing TDMA (SS-TDMA) is proposed in [26] which assumes that nodes are organized into a regular grid topology (triangular, square or hexagonal) where each node is aware of their own location. These assumptions are also used in this thesis (as shall be discovered later) however, the complexity of this protocol increases with the diameter of the network making it unsuitable for large sensor networks.

The Traffic Adaptive Medium Access (TRAMA) protocol employs a traffic adaptive distributed election scheme that selects receivers based on schedules announced by transmitters[41]. Time is divided into a signalling period during which nodes exchange topology and traffic information, and a transmission period which is divided into TDMA time slots during which data is exchanged in a collision free manner. Due to the definition of the length of these two time periods and the fact that nodes are required to listen to all information exchanged during the signalling period, a relatively high duty cycle is required. In addition, this protocol requires a significant amount of calculations at each time slot [6]. [41] concludes that TRAMA is
well suited to sensor network applications that are not sensitive to delays.

The **EMACs** (Eyes MAC protocol for Sensor networks) protocol [54] developed for the European EYES sensor network project is a TDMA based MAC protocol that also functions as a topology control protocol. EMACS has nodes self organizing to form an active connected backbone to the network surrounded by nodes that are not required to route data through the network which are allowed to fall into a passive state. Active nodes gather sensed information from passive nodes and forward this to other active/backbone nodes as the sensed information is routed through the network to its eventual destination. Each TDMA time slot is subdivided into three sections; a *communication request* section for passive nodes to request permission to report information; a *traffic control* section for the active node that owns the slot to coordinate communication; and a *data* section for transferring data.

The algorithms are relatively complex with each node having to decide on the role that it is required to play and although mention is made of the importance of latency and throughput, this protocol is in not compared with others using these metrics. In fact the only metric that is used in [54] is that of network lifetime\(^8\).

The Lightweight MAC (**LMAC**) protocol [56] by the same authors, builds upon EMACS and eliminates the *communication request* section from time slots. LMAC removes acknowledgement from the MAC dialogue leaving this to higher layers in the protocol stack. Time slot assignment is completely distributed and effectively ensures collision-free transmissions. However, the slot assignment algorithm requires nodes to listen to all traffic control sections meaning that a significant amount of energy is wasted [28].

Although TDMA protocols are inherently collision free, for wireless sensor networks they are complicated and require significant design trade

---

\(^{8}\) in this case, network lifetime is defined as time until 16 of 48 nodes in the network have exhausted their energy supplies
offs among many performance parameters such as flexibility, deployment scenario, node mobility, throughput and latency [28]. Thus we do not consider them further.

2.5.3 Unsynchronized Contention-Based

Rather than allocating a particular time for each node or link to communicate in a manner that does not allow collisions to occur, another approach is to have each node contend for the right to access the shared medium when they have information to send. This means that collisions will sometimes occur, however the advantage is that contention-based protocols generally tend to allow better latency. The first group of contention-based protocols that are examined are those in which nodes are not synchronized with each other.

Both preamble sampling [7] and low power listening [17] were developed independently despite utilizing very similar techniques implemented at the physical layer. The basic premise of both systems is to have a node with information to send precede the transmission with a relatively long preamble ($p$) compared to the length of the data transmission. After the preamble there is a start bit that signifies the beginning of the actual data. Other nodes perform carrier sensing every $p$ seconds and if they detect the presence of a preamble signal, they keep their receiver active until the start symbol and the data is received. By sensing the carrier every $p$ seconds the amount of time spent listening idly to the channel is reduced by a fixed factor of $p$ seconds divided by the time required to perform a carrier sense. Low power listening performs carrier sensing (which takes 30 $\mu$Sec) every 300 $\mu$Sec affording an improvement in idle listening of a factor of 10 [28]. This reduction factor of idle listening can be set arbitrarily by modifying the value of $p$ and a lower duty cycle means less idle listening.

---

9 carrier sensing is when a device uses its radio to determine the presence of a radio signal within the channel
but a greater latency. This is not the last time a trade off between energy consumption and latency is encountered in this thesis.

Both of these techniques are low level carrier sensing techniques that save energy at the physical layer. The techniques can be applied to any contention-based MAC protocol. Preamble sampling is coupled with ALOHA which is the first contention-based protocol that was developed [49] in which if a node has information to send, it sends it straight away. Low power listening is combined with CSMA (Carrier Sense Multiple Access) in which if a node has information to send it first senses the medium and if it is unoccupied, begins to transmit the data.

In a refinement of preamble sampling for the WiseNET sensor network program, El-Holydi also adopts a CSMA MAC protocol and piggybacks a nodes sampling phase offset onto acknowledgements to produce the WiseMAC protocol [8]. This allows nodes to gain knowledge of when their neighbors will become active to sample the medium and instead of transmitting a long preamble, allows nodes to wait until they know their neighbors will be awake saving energy in both the transmitter and the receiver.

The drawbacks to these techniques are that they involve nodes becoming active frequently to check if there is a transmission about to begin. As idle listening is only reduced by a fixed factor of 10 compared to the case when the radio transceiver is active all the time, these methods are not explored further as we would prefer to do very much better than this.

2.5.4 Synchronized Contention Based

This brings us to the ‘flavor’ of protocols that this thesis is concerned with, protocols in which nodes are synchronized with each other and become active relatively infrequently compared to the protocols in the previous section. These protocols are in effect a combination of TDMA and contention-based MAC protocols since they sleep for some period and then become active at
a common time to contend for the right to utilize the shared medium based on whether they have information to send or not. Nodes communicate via a single channel meaning that they use a simple radio module that consumes less energy than multi-channel radios. These protocols are examined in detail in the following chapter.
Chapter 3

Literature Survey

This chapter presents a review of single channel MAC protocols. It begins by examining some work from which we derive our terminology, followed by an examination of the general purpose 802.11 IEEE wireless standard. We then examine an early version of S-MAC, one of the first wireless sensor network channel access protocols, which identifies four sources of sensor network MAC protocol inefficiency and largely quenches three of these sources. We then examine T-MAC, which in solving the remaining source of inefficiency introduces problems of its own. After this we look at a later incarnation of the S-MAC protocol followed by a construct that helps explain coordinated activity in wireless networks. The last protocol to be examined appears to be quite similar to our solution but is not a truly independent MAC protocol due to its influence on the routing of packets. The chapter concludes by summarizing the relevant points from the reviewed literature and establishing the design requirements that our solution must meet.

3.1 The Prototype Embedded Network (PEN)

The PEN project [12] [13] [21] [51] sought to establish a network by embedding computers capable of wireless communication into a multitude of devices. These devices (known as PEN nodes) would then be capable of self configuring by utilizing information about their environment. The PEN was designed with three main parameters in mind: size, cost and power consumption. Size and cost were minimized by ensuring that the design of components was kept as simple as possible. It is the findings relating to energy expenditure that are of most interest to this thesis.

The process by which a node sleeps for most of its duty cycle and
then awakens periodically to meet with other nodes was termed *rendezvous* by researchers at the PEN project and we adopt this terminology. The *rendezvous period* ($T_{rp}$) is the time between successive ‘turn-on’ events or beacon intervals. The *rendezvous phase* ($\phi$) is the time between the start of the rendezvous period and when the node actually turns on. The *active time* ($T_a$) is the time for which a node is active before it enters sleep mode.

The PEN project has been developed for indoor use to provide direct connectivity between electronic devices. For example, to provide a protocol in which a smoke detector may discover and establish connectivity with an alarm bell [21]. There is no facility provided for a node to act as an intermediary node and play a role in assisting the delivery of data from a source to a destination that is out of range of the source. In other words, data is not delivered in a multi hop fashion.

The system proposed by the PEN uses rendezvous in an *on-demand* fashion, with nodes becoming active on an individual basis and independently of the rendezvous time of their neighbors. Nodes discover each other by waiting for a beacon signal when they become active and if they receive a beacon, they establish communication with their neighbors. They are not aware of when their neighbors will become active. Thus a node wishing to contact another must beacon continuously until a response is elicited.

Before we describe this dialogue depicted in the diagram, we must explain the diagrams themselves. A sensor node’s radio can be in one of four states\(^1\) and the representation of each state is commensurate with its relative energy cost as depicted in Figure 3.1. The line is thickest when transmitting and thinnest when in sleep mode. Time is represented on the vertical axis, so the state of each node changes with time as the reader moves down the diagram.

The process of beaconing to establish a link is depicted in Figure 3.2.

\(^1\)sleep, idle, receive or transmit
Since node A does not know when node B is going to become active, it must transmit a beacon signal at regular intervals to get node B’s attention. In this example, node A beacons unsuccessfully three times before establishing connectivity on the fourth attempt. Each time node A attempts to establish connectivity, it firstly listens to the medium to assess the availability of the medium, it then transmits its beacon signal and finally it waits for some period of time for a response and after receiving no response, it beacons again. [51] presents the traffic-dependent energy performance of whether it is better for source nodes to beacon when they have information to send or destination nodes beaconing to allow possible source know that they are awake (active).

---

\[2\] this is known as carrier sensing
This need for a source node to beacon to establish connectivity with a destination node represents the energy cost of establishing connectivity when nodes are unaware of their neighbor’s wake up schedule making the PEN rendezvous protocol more similar to preamble sampling and low power listening. Establishing connectivity in this manner also introduces a delay in the transmission of information - making timely delivery of long-haul multihop information improbable. For these reasons this thesis does not consider the PEN protocols further, however the terminology is retained.

3.2 The 802.11 Protocol

3.2.1 The Application

The 802.11 standard was prepared by the IEEE as a standard for the wireless LAN. The IEEE began work on the 802.11 project in 1990 to provide physical and medium access control specifications for wireless connectivity. 802.11 was recognized in 1997 with 1999 seeing the ratification of 802.11a and 802.11b standards [23]. Its main purpose is to provide wireless connectivity for mobile computers which may interface with a base station or allow a group of mobile computers to wirelessly connect with each other forming an ad-hoc network.

This is a summary of the relevant parts of the IEEE 802.11 protocols as given by [29] [19]. The 802.11 MAC protocol is a single channel, synchronized, contention-based protocol and all timing within the protocol is described in terms of time slots. When attempting to access the medium, nodes are only allowed to transmit at the start of a slot. The actual amount of time that a slot occupies being hardware dependent on the implementation of the physical layer. It is set to include the time required by a node to sense the medium and then modify the state of its radio electronics from receive mode to transmit mode [4]. This results in carrier sensing that avoids collisions by checking whether or not the medium is being used by another
device but it does not allow collisions to be detected during transmission\textsuperscript{3}. It is for this reason that the 802.11 MAC protocol is classified as a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) channel access protocol.

Nodes in an 802.11 network can operate in two modes. Firstly, they may be operating under the Point Coordination Function (PCF) where there is a controlling node\textsuperscript{4} that coordinates all of the traffic flow. The second mode is described by the Distributed Coordination Function (DCF) in which the same coordination logic is present and active in each node. It is the DCF mode that is most appropriate for the type of distributed network that is being considered in this document and we may thus neglect any further discussion of the PCF mode.

3.2.2 MAC Frame Dialogue.

This dialogue, is similar to that developed by the MACAW protocol [3], except that the 802.11 standard reintroduces the capability for wireless nodes to use carrier sensing hardware because the RTS-CTS control packets do not always prevent collisions from occurring and the exposed terminal problem is not a great concern in this situation. Also because of the reduced importance of the exposed terminal problem, the 802.11 protocol does not use the data sending (DS) control frame as the MACAW protocol does.

When a wireless node (the source) has information to transmit to another node (the destination) it must firstly wait for the medium to be unoccupied for DCF Inter Frame Space (DIFS) seconds. This DIFS time effectively acts as a channel reset, meaning that after being unoccupied for DIFS seconds, the channel is free for any node to contend for. After DIFS seconds, to minimize the risk of a collision, a source node chooses an integer

\textsuperscript{3}this is because the radio signal that is being transmitted will swamp any incoming radio signal

\textsuperscript{4}Typically a base station.
(n) at random where
\[ n \in [0, CW - 1] \]
where CW is the current value of its Contention Window.\(^5\) It then decreases n by one for every slot time the medium is unoccupied and the source node is said to be *backing off* while behaving like this. If the source detects that the medium becomes occupied while it is backing off (while it is decrementing n by one for every unoccupied slot time that passes) it freezes its back off timer (the value of n remains static) while the medium is occupied and restarts it when the medium becomes free again (after another DIFS seconds).

When n reaches zero the back off timer fires and the source transmits a Request To Send (RTS) control frame to the destination, which replies with a Clear To Send (CTS) control frame Short Inter Frame Space (SIFS) seconds later, if the medium is unoccupied within range of the destination. Upon receipt of the CTS, the source transmits the DATA frame SIFS seconds later. The destination receives the DATA frame, checks it for errors using a Cyclic Redundancy Check (CRC) and then transmits an acknowledgement (ACK) control frame (SIFS seconds later) if the DATA frame stands up to the CRC. A source node will detect a protocol breakdown by the non-reception of an ACK frame meaning that it will be required to attempt to re-send the data. This process is more clearly explained in Figure 3.3.

This mechanism attempts to ensure that when collisions occur, they occur between control frames only. This is good from the perspective of energy efficiency since retransmitting a short control frame is not as large a repetition of work as retransmitting and entire DATA frame, the RTS-CTS exchange explicitly alerts all nodes within range of the source and destination of an impending DATA transmission.

The purpose of having a contention period before accessing the shared medium (the radio space around the node) is to minimize the number

\(^5\)The Contention Window is explained shortly
of colliding packets. The chances of a packet colliding are dependent on two parameters of the situation,

- how many nodes are there are within range of each other and,
- how many nodes have information to deliver.

By having nodes choose a random number of slots to wait until they transmit, an attempt is made to ensure fair access to the medium, and since the node that eventually gains access to the medium will in effect have been chosen at random. The larger the contention window, then for the same above two parameter values, the probability of two nodes choosing the same back off time decreases and thus control packets are less likely to collide. It is however impossible to completely remove the chance of packets colliding.

The probability of a collision is managed by modifying the Contention Window (\(CW\)), which is doubled every time the medium access protocol fails to secure the medium for transmission of a DATA packet up to a maximum value \(CW_{\text{max}}\). When a successful attempt at transmission is made \(CW\) is reset to \(CW_{\text{min}}\) [37]. We will return to this discussion later after we have introduced the concept of ripple rendezvous, to discuss the value of \(CW_{\text{max}}\) as well as modifications to the method by which the contention window is managed. This value is very important in determining
the average energy consumed per node.

3.2.3 Power Save Feature

Nodes within an 802.11 network may utilize a power saving feature in which they periodically turn off their radio transceivers to save energy. At the start of the active time there is an Announcement Traffic Indication Message (ATIM) window during which control messages are broadcast (and then acknowledged) to nodes for which there is a stored and cached packet. After the ATIM window, nodes that have not received notification of impending traffic enter sleep mode until the beginning of the next window. Nodes that were notified of traffic remain active until the end of the next ATIM window. Under this system, traffic can be sent only one hop per rendezvous period. This is an amicable solution for the situation for which the 802.11 standard was devised, which was to wirelessly connect portable computers that are all within range of each other and would rarely need to utilize multihop communication. However, this serves to highlight the inefficiencies introduced when this wireless LAN protocol attempts to coordinate nodes in a wireless sensor network, something that it was not designed for.

The power saving mechanism used by the 802.11 standard has nodes within the same Independent Basic Services Set (IBSS) adopting the same rendezvous phase in that they enter sleep mode at the same time and then reawaken simultaneously [19] it does not specify how the rendezvous phase is coordinated between IBSS’s since this is not likely to occur in an Ad-Hoc network of devices such as laptop computers. However, in a sensor network with many neighbors lying outside the range of either the sender or the receiver, this can potentially lead to a situation in which each node becomes active once per rendezvous period at the same time as all of its neighbors to transmit information to them, as well as when each of its neighbors in different IBSS’s become active at their scheduled rendezvous times to receive
information from them. In the distributed networks that we are considering, for a node to become active many times per rendezvous period for a fixed amount of time represents a very inefficient method of operating. It is almost not worth going to sleep at all.

### 3.3 The S-MAC 2001 Protocol

This protocol was developed by researchers at USC/ISI working on the directed diffusion sensor network project and draws upon the 802.11 MAC dialogue and makes several modifications that tightens it to aggressively save energy, making it applicable to sensor networks. Nodes coordinate their sleep schedules using only local communication. A full treatment of the rendezvous phase assignment algorithm is given in Sub-Section 3.3.1.

In the frequently-cited paper that presents the S-MAC protocol [63], four sources of inefficiency in the operation of a sensor node’s radio transceiver are identified:

1. **Control Packet Overhead.** These are packets required to maintain the integrity of the network, and do not contribute to the ultimate purpose of the network, which is the delivery of application layer datagrams. There is no one particular solution to this problem, it is simply a fact that must be considered by protocol designers.

2. **Collisions.** When transmissions collide they must be rebroadcast, this repetition of work is a source of energy waste. Although collisions are not eliminated by the S-MAC protocol, the effect of them is minimized by a technique known as message-passing.

3. **Overhearing.** When a node receives a transmission for which it (the node) is not the intended destination, since the frame will be discarded when the physical layer hands it to the MAC layer, the node wastes

\[6\] since the frame is not addressed to this node
energy receiving it. The problem of overhearing has largely been solved by the S-MAC protocol.

4. **Idle listening.** This is any time that a node spends listening for traffic but none arrives - it is the largest source of wasteful energy consumption in sensor networks.

As mentioned previously, since the activity within the network can be described as being bursty there may be extended periods of time during which the network is idle. Guo et al. [14] suggest that idle listening can be responsible for up to 90% of energy consumption in wireless sensor networks, however the exact figure is largely dependent on the individual application and usage scenarios being considered. We shall show in this section that although the S-MAC 2001 protocol presents a solution to the problem of idle listening, there are remaining inefficiencies that present the opportunity for further refinement.

S-MAC 2001 is not adaptive to the level of traffic within the network with nodes adopting a fixed duty cycle regardless of the level of traffic present in the network. In the study conducted in [63], sensor nodes would sleep for one second and then become active for 300 ms. This allowed messages to be delivered within one rendezvous period with the only delay in delivering information being the sleep delay incurred while waiting for the network to awaken. This sleep delay is inherent to a network that utilizes a sleep mode. Once active, the network may stay active long enough to ensure timely delivery of information depending on the setting of the operational parameter and the length of the route.

Major improvements of the MAC dialogue were also developed. The first of these is the technique of 'message passing' or MAC-level fragmentation which as we shall demonstrate, largely solves the problem of collisions and/or transmission errors. Once a link is established, the data packet is fragmented and transmitted in a burst with each fragment receiving an ac-
knowledgement. In the experimental validation each 'message' consisted of 10 fragments of approximately 40 bytes each with each control frame being about 10 bytes.

The second development is that of overhearing avoidance, a technique that has nodes going to sleep if a node overhears the establishment of a neighboring link (a neighboring node is involved in a link with a third node). A node is prevented from forming a link anyway since to do so would be to interfere with the first link. Under S-MAC 2001, all MAC frames (RTS, CTS, DATA and ACK frames) contain a duration field.

Radio state diagrams are again used to represent the dialogue between nodes implementing both the message passing and overhearing avoidance mechanisms. If necessary, please refer to Figure 3.1 for an explanation of the state representation. Figure 3.4 depicts a four node network in which nodes are only within range of their immediate neighbors. In this example node B has one packet of information to be delivered to node C which is fragmented into ten segments and the two techniques of message passing and overhearing avoidance are depicted.

At A1, node A goes to sleep after overhearing B's RTS and reawakens at A2 since the initial RTS contains the expected duration of the transmission. D's case is similar except that it goes to sleep after overhearing C's CTS that also contains a duration field allowing node D to reawaken at D2 when the transmission is finished and the medium is again free to contend for. Both node A and D have saved energy by sleeping during B's transmission to C while they are not able to form a link anyway since to do so would be to interfere with B's transmission to C.

During B1 node B backs off to contend for the medium after which B sends to (and then receives from) node C the RTS-CTS handshake that establishes a link. During B2, node B transmits the data fragments in a burst with each fragment being individually acknowledged.
Figure 3.4: Fragmentation and Overhearing Avoidance under S-MAC
Fragmentation reduces the cost (in terms of energy efficiency) of a collision by trading off the somewhat higher energy cost and the slightly longer transmission time\textsuperscript{7} of fragmentation against the cost of retransmitting an entire relatively large DATA packet. The link is terminated during B3 with the reception by B of the ACK for the last fragment.

Figure 3.5 demonstrates the performance of S-MAC for the case of a collision which can occur in an S-MAC network although this case is uncommon. In this figure, node B has a packet to send to node C as in the previous example, but this time node D also has a packet to send to C.

Both B and D start backing off at the beginning of D1. B transmits its RTS to C (which D cannot hear) and unfortunately D sends its RTS

\textsuperscript{7}Due to the extra ACKs and DATA frame headers that must be transceeed
to C at the same time as C sends the CTS to B which is received without corruption since B is out of range of D. Both C and D are unaware of the other’s transmission (since carrier sense is not possible during transmission in wireless communication [49]). During D2, D waits for a CTS from C which doesn’t arrive, after which D backs off and transmits a RTS again which this time collides with the first data fragment corrupting it. This is represented in black on nodes C’s time line.

Since the first fragment was corrupted, C does not acknowledge it, so at B1, B transmits the first fragment again. During D3, D backs off for a long time this time having failed to secure the medium twice, in fact it is still backing off when it overhears hears C’s ACK to B (for the successful reception of the first fragment) which contains the expected duration of the link and goes to sleep to further avoid overhearing. Node D reawakens at D4, when the medium is again free to compete for. Node A awakes before the link is finished since the link has been extended due to the collision. It overhears the beginning of the last fragment (the header) which tells it of the new expected duration, and sleeps until the new link-completion time.

Message passing helps avoid collisions by having both the source and destination nodes fill the radio space around them with their corresponding data fragments and acknowledgements. These relatively short frames alert all potentially interfering nodes of the ongoing transmission, as well as the time the link is expected to be terminated. Message passing also significantly reduces the cost of collision since when they do occur, only the much shorter data fragment need be retransmitted as opposed to the entire data frame.

Comparing Figures 3.4 and 3.5, we can see that the behavior of the nodes is relatively similar for both of these cases. Both A and D sleep (to avoid overhearing) for almost the same length of time and the packet is also transmitted from B to C in approximately the same time in both cases. These two cases illustrate how the effects of collisions are reduced
Figure 3.6: Packet transmission diagram

by message passing and how overhearing avoidance allows nodes to save energy by sleeping. It is noted here that experiments conducted in the study of the T-MAC protocol (see next section) noticed problems caused by overhearing avoidance. These problems result from when nodes would sleep on overhearing a RTS or CTS but subsequently the link failed to be established. Since the nodes were sleeping, they were not aware that the neighboring link had failed. A decrease in the throughput of the network was reported with the associated increase in latency and decrease in efficiency. We maintain however, that the energy saved by nodes using the overhearing avoidance mechanism may outweigh the slight decrease in performance.

Looking at the operations of the node at this level becomes rather confusing, let’s have a look from a higher level of abstraction. Rather than looking at the situation from the MAC level and observing the exchange of MAC-level Frames, if we consider the situation from the Data Link Layer (DLL) perspective, this results in a much less cluttered diagram and also allows us to represent the transmission of multiple network-layer packets.

Figure 3.6 presents the same situation as presented in Figures 3.4 and 3.5, that is, node B transmits one packet to node C. During the transmission of the packet from B to C, the extent of the radio transmission range is denoted so as to help depict the overhearing avoidance behavior of neighboring nodes.

Having established the meaning of these link-level node diagrams, it
is now possible to depict more complex network scenarios. The situation in Figure 3.7 is for a 10 node network (nodes A to J) in which node A has a packet of information that is sent to node J in a multihop fashion.

Shortly after the network becomes active, node A transmits the information to node B which in turn sends it to node C. This process continues as the packet is carried through the network bound for node J. There is a short time interval between the transmission of each packet to reflect the average time spent backing off before each node accesses the channel.

In this case, even though the network has been tuned to remain active for only just long enough for all the information transactions to take place, for most of the time that nodes are active, they are in idle mode. It is also interesting to note that S-MAC has a network performing better with respect to the average energy per node metric for higher traffic levels since each node will spend more time in sleep mode while engaged in overhearing avoidance. We shall return to this diagram in Sub-Section 5.2.2 when discussing the performance model of the S-MAC 2001 protocol.
Because there is no controlling or coordinating infrastructure to allocate a particular time at which a node needs to become active, a node chooses its active and sleep times based on the virtual cluster to which it belongs. The process by which a network self organizes itself into these virtual clusters is the topic of the next section.

### 3.3.1 S-MAC virtual cluster establishment

A major improvement on the power save protocol used by the 802.11 protocol suite is how individual nodes self organize to coordinate their rendezvous phases using only local rules and communication with nodes that are in their local radio space. When a node (node A) first becomes active (it comes to life for the first time having been deployed in the sensor field, this is at the beginning of phase 3 of its operational life), it firstly listens for random amount of time $t_1$ where

$$t_1 \in [0, T_1]$$

after which, there are three possibilities:

1. If it hears nothing, then it chooses a random rendezvous phase and immediately broadcasts it "I'm going to sleep in $t_2$ seconds", where

$$t_2 \in [0, T_2]$$

in the form of a SYNC frame. A node behaving like this is said to be a *synchronizer*.

2. If it hears another node’s schedule before choosing one for itself, then it simply adopts that rendezvous phase, broadcasts a SYNC (after backing off for a collision-avoiding random-delay of $t_{bo}$ seconds) where

$$t_{bo} = n . slotTime$$

and

$$n \in [0, CW - 1]$$
such a node is termed a follower.

3. If it hears another node’s (node B’s) schedule after it has already chosen its own schedule it must reply with its own schedule (to let node B know that it too must become an *inter-cluster node*) and follows both, but it only broadcasts its own schedule.

The larger $T_1$ becomes, the larger the size of the virtual clusters, because nodes will be willing to wait for a longer time to hear another node’s schedule before choosing their own. What this means, is that as $T_1$ becomes larger, nodes must wait in idle mode (awaiting the arrival of a synchronizing schedule) for longer and waste their limited energy reserves on idle listening. A longer $T_1$ also means a longer time for the network to self organize. Thus, there is a trade off between virtual cluster size and initial energy expenditure and network self organization time.

Information that is being delivered through the network in a multihop fashion, may have to spend a proportion of the time traversing the cluster boundaries while waiting for the next cluster to become active. Thus the advantage of larger clusters, is that information is able to be delivered faster. Larger clusters are achieved by setting a larger $T_1$ however this causes the network to take longer to self organize and consume more energy during the self organization phase 3 of the network.

Synchronization issues due to clock drift are not a significant concern for S-MAC since a node is active for significantly longer than the clock drift. All times that are exchanged between nodes are relative (eg. in 3.5 seconds) rather than absolute (eg. at 2:05:03.5 pm)[63].

### 3.3.2 Implications of S-MAC 2001

For the rest of this document we consider a situation in which a network is organized into just one virtual cluster. This simplification is a best-case scenario and places an upper bound on the performance of an S-MAC net-
work since DATA packets do not have to traverse inter-cluster boundaries and as such they are do not incur another sleep delay while multi-hopping from source to sink.

The primary implication of the S-MAC protocol, as it relates to energy expenditure, is that the average power consumption of a node is relatively independent of the amount of traffic that it carries. This is because the three active modes of a radio transceiver (transmitting, receiving and idle) all consume very similar amounts of energy when compared with sleep mode. The downside to the fixed duty cycle is that the lifetime of a sensor node is extended by a predefined factor of the ratio of the rendezvous period to the active time and also means that the amount of time an S-MAC 2001 node spends engaged in idle listening is inversely proportional to the traffic demands on the network i.e. the less traffic there is, the more time a node spends engaged in idle listening. This reinforces the remaining problem of idle listening.

When a network has nodes scheduled to become active for a fixed period of time, a trade off is established between average energy consumption and time taken to deliver information. Consider one extreme case in which nodes become active only for enough time to transmit one DATA packet before returning to sleep mode. Nodes in this case will be operating at a very low average power consumption, however since one DATA packet can only be transmitted one hop every rendezvous period before the nodes return to sleep mode, the message will take an integer multiple (one for each hop) of rendezvous period seconds to reach its eventual destination.

The other extreme is for the sleep time of nodes to be very small in which case the nodes hardly sleep at all. In this case a DATA packet can be multi-hopped very quickly from source to destination, but each node will be operating at close to its highest average power consumption (when compared with sleep mode) and thus not very efficiently at all. These two
examples represent the trade off between energy efficiency and timeliness of delivery.

We conclude by saying that the S-MAC 2001 protocol identifies and then addresses four sources of energy inefficiency in MAC protocols for wireless sensor networks. The problem of overhearing is eliminated without the need for any further control packets and the effects of collisions are significantly reduced by the technique of *message passing*. There is an additional amount of protocol overhead introduced by message passing which represents a trade off between the probability of collision and the energy cost of retransmission. The problem of idle listening increases as the level of traffic decreases and is thus not solved. At low levels of traffic, nodes spend time that would be used to communicate wasting energy idly.

### 3.4 The T-MAC Protocol

This protocol [53] was developed at the Technical University of Delft in an effort to address the largest source of inefficiency remaining in the S-MAC protocol, that of idle listening. This protocol uses the virtual clustering scheme developed for S-MAC, with the main difference being that the active period of a node is adaptive to the amount of traffic that is present within the network.

The T-MAC protocol draws heavily on the S-MAC protocol with the main refinement being that it is adaptive to the level of traffic within the network. T-MAC has nodes communicating their cached DATA packets with each other in a burst when they first become active. From when a node initially becomes active, it will sleep if it has not experienced an activation event for some period $T_i$. There are five activation events;

1. Their wake up timer fires.

2. Data (any data) is received via radio.
3. Communication is sensed on the radio.

4. Reception of an acknowledgement for a packet just sent, or finishes transmitting an acknowledgement of a packet just received.

5. Neighboring communication finishes (determined through prior overheard RTS or CTS)

A T-MAC node resets its idle listening timer \(T_i\) every time an activation event occurs and if this timer fires before another activation event occurs, it enters sleep mode until activation event 1 occurs and it reawakens in the next rendezvous period. This is referred to as *idle listening shutdown*. The value of \(T_i\) is set to the contention period \(T_i\) (which under T-MAC is fixed) plus the time taken for a RTS-CTS control packet exchange all multiplied by a factor of 1.5 just to be sure.

[53] describes two main types of communication that sensor nodes are required to deal with. The first type is that of local unicast and broadcast communication which is termed *local gossip*. Sensor nodes engage in this type of communication pattern while they are exchanging information with each other while information is being fused or aggregated. The second type of communication pattern is after the local data fusion has occurred and this aggregated packet of information is returned to a sink node along a long-haul multi-hop route.

T-MAC performs well for nodes engaged in local data fusion when they do not enter sleep mode because there is nearby network activity that they can hear. T-MAC uses the same virtual clustering mechanism that S-MAC uses, which again we assume is considered to involve all nodes in the network adopting the same rendezvous phase.

van Dam [53], acknowledges the *early sleep* problem of T-MAC that occurs after the data has been fused locally and is being returned to sink node along a long haul (longer than two hops) multi hop route. This early
All nodes become active
Node A
Node B
Node C
Node D
Node E

Node A transmits packet 1 to B
Node A hears no more network activity and sleeps

Node B
Node C
Node D
Node E

Since nodes D and E have not heard any network activity within their idle listening timeout period they have prematurely entered sleep mode before they are needed to carry the information

Node C attempts to initiate a link with node D but is unable to do so since node D is asleep

Figure 3.8: T-MAC and the Early Sleep Problem

The early sleep problem is most easily described in Figure 3.8 which does not depict Node A's RTS frame or the CTS control frame from B that C overhears which causes it (C) to stay active, but does show the failed RTS frame sent by node C. This problem basically has nodes entering sleep mode prematurely because they have not sensed the ongoing network activity. The developers of the T-MAC protocol recognize this problem and have developed a mechanism by which the range of communication is extended by the use of Future Request To Send (FRTS) control frames. When a node overhears a CTS frame, it responds by sending its own FRTS frame to let its neighboring nodes know in advance of the network activity that they cannot directly hear. The FRTS frames require a modification of the basic MAC dialogue to include a Data Sending (DS) control frame similar to the MACAW [3] protocol. This DS frame provides enough time for neighboring nodes to send a FRTS frame without having them collide with the first part of the DATA frame.

The FRTS mechanism introduces problems of its own including a

\^after overhearing a CTS frame
higher control frame overhead. There is also an increase in the value of $T_i$ which results in an increased energy expenditure of every node in the network since every node waits for longer before engaging in idle listening shutdown. For a full explanation of these problems, please refer directly to the source [53]. However the FRTS mechanism unreliably\(^9\) extends the range from two to three hops and the protocol still suffers from the early sleep problem. To ensure timely delivery of information, T-MAC needs to be tuned to the maximum expected route length by adjusting the rendezvous period ($T_{rp}$). This tuning need to be done prior to deployment during phase 2 of its operational life since it is not feasible to contact each node with instructions to modify their rendezvous period once they have been deployed. For an expected maximum route length of $N_{max}$, the rendezvous period will have to be approximately $\frac{T_{od}}{N_{max}}$ to be able to deliver the information in $T_{od}$ seconds. This means that nodes will be turning on more often which in turn increases the associated energy cost per node. Once more a trade off is established between energy efficiency and timeliness of information delivery.

Returning now to the network scenario presented in Figure 3.7 for ten nodes along a DATA route where node A has one packet to be delivered to node J, we see the T-MAC version of the same situation presented in Figure 3.9.

From Figure 3.9, all nodes become active at the same time and node A backs of before transmitting the packet to node B. In forming the link, node B sends a CTS control packet which node C overhears and transmits a FRTS control packet (before entering sleep mode to avoid overhearing) which node D receives (and then enters sleep mode since it will not receive anything until at least A has sent the packet to B), thus nodes C and D are prevented from the early sleep problem.

After node B finishes receiving the packet, it backs off, before trans-\(^9\)due to colliding FRTS frames
mitting the packet to node C. During the formation of the link, node C sends a CTS control frame to B which D overhears and transmits a FRTS control frame that node E does not receive since it has already had its $T_i$ timer fire and entered sleep mode. After node C has finished receiving the packet it backs off and transmits the packet to node D. During the formation of the link node D returns a CTS control frame to node C and this time no FRTS control frame is sent because node E is asleep. The failed attempts by node D to establish a link with node E (which is already asleep) are not depicted. This process continues in the two subsequent rendezvous periods as the packet makes its way to its ultimate destination (node J).

The T-MAC protocol is very good at reducing the energy consumption of sensor network nodes. Since it has nodes spending very little energy engaged in idle listening when compared to the energy spent sending and receiving information. In the study conducted in [53], T-MAC was reported to
use 96% less energy than the S-MAC protocol, the amount being dependent on the specific traffic scenario as well as the parameters chosen; including the rendezvous period, idle listening timeout, active time, time required to transmit a packet, the relative energy cost of the four radio modes and the size of the contention window.

3.4.1 Synchronization

Synchronization between nodes becomes an important issues for T-MAC since nodes are no longer active for very much longer than clock drift. In fact some difficulty in maintaining synchronization was reported in [53] with nodes losing connectivity with each other after about 10 minutes of operation. The solution to this problem was relatively easy in that when nodes would exchange SYNCH packets with each other, they would modify their internal clocks only by 50% of the difference between their schedule and the schedule received from the other node. This technique was found to have nodes maintaining perfect connectivity after 10 hours.

3.4.2 Implications of T-MAC

T-MAC trades off the timely delivery of information for the ability to expend energy almost exclusively on either sending information to or receiving information from other nodes in the network. In other words T-MAC is adaptive to level of traffic within the sensor network with nodes spending the majority of energy doing useful work (transporting information through the network). However, the early-sleep problem combined with the fact that T-MAC uses the same rendezvous phase assignment algorithm as S-MAC, information is not able to be delivered in a timely manner along long-haul multihop routes.
3.5 The S-MAC 2003 Protocol

S-MAC 2003 is essentially the same as S-MAC 2001 except that the active time is set to be equal to the time required to transmit one packet of information. This means that each node becomes active for $DIFS + CW_{\text{max}} + T_{\text{pkt}}$ seconds whether or not there is any traffic to be carried. S-MAC 2003 also uses a technique called adaptive listening in which nodes become active again (after sleeping to avoid overhearing) for a short period after a neighboring link has finished. This means that if the node were to be the next hop along the data path, then it would be able to receive it in the same active period rather than having to wait for the next one [62]. This short period is not clarified in [62] so we must assume that it is similar to the idle listening timeout developed by the creators of the T-MAC protocol.

According to [32], this means that information can only be sent two hops per rendezvous period making it functionally similar to T-MAC operating without the use of FRTS control packets. When organized like this, S-MAC 2003 trades much lower energy consumption for a decrease in the ability to deliver information in a timely manner. S-MAC 2003 also places a strict upper bound on the number of network-layer packets that can be relayed per rendezvous period. In fact, it essentially limits network nodes to handle one packet per rendezvous period within their two hop neighborhood.

3.6 Ripples

Before moving to the next related protocol, a wireless multihop communication construct, developed specifically for this thesis is introduced - the ripple. A ripple is a technique by which successive source-destination pairs establish pre-arranged connectivity in a cascading manner that facilitates timely delivery of a packet of information. A node’s active time is divided
Figure 3.10: Six nodes collaborating to form a ripple equally into a receive-half and a send-half that reflects the half-duplex nature of a single-channel wireless link\textsuperscript{10}. A node, when it awakes, firstly listens for inbound transmissions from its upstream neighbor. Once this time expires, it then waits for the next node on the route to turn on, transmits the information that it has for the newly awake downstream neighbor and then enters sleep mode.

Figure 3.10 depicts a simple example in which six nodes have coordinated their rendezvous phases to be staggered along the data route from node A to node F. For demonstration, DATA fragmentation is not depicted however a ripple can carry multiple data fragments as long as the active time \((T_a)\) is set to twice the time required to successfully transmit one packet of information. The ripple has a \textit{carrying capacity} of one packet.

The receive half of node A’s active time is not depicted since it does not have an upstream neighbour and it is the original source of the information. Node F is the ultimate destination so the send half of its active time is not depicted. Node A commences backing off at A\textsubscript{1} which is when node A knows that node B has just become active. A transmits a

\textsuperscript{10}a single channel radio cannot send and receive information at the same time
RTS, receives a CTS from B, transmits the DATA frame and then receives the ACK from B. Node A goes to sleep at A2 since it has done all that it can in this rendezvous period. From node B’s perspective it awakens at B1 and almost immediately forms a link with node A (as it responds to node A’s RTS) during the first half of its active time. At B2 (which is also when node C awakens) node B changes from receive mode to send mode and begins the process of forming a link with node C and transmitting the information.

Comparing the behaviour of nodes A to F of Figure 3.10 with the behaviour of nodes D to I of Figure 3.7, nodes spend very much less time listening idly when ’rippling’ than when behaving as S-MAC nodes.

3.7 The DMAC Protocol

The approach that DMAC [32] takes to allow a sensor network to deliver information in a timely manner as well as operate at a very high level of energy efficiency is to allow DMAC nodes to self organize their rendezvous phases to form data gathering trees. Each data route in the tree (from a leaf node to the root) has nodes behaving so as to ripple information from the leaf (source) node to the root (sink) node. We firstly examine the behavior of nodes along one particular route from source to sink. We then talk about the two dimensional implementation of DMAC and see that problems occur when routes merge and multiple child nodes both have information to transmit to parent nodes.

DMAC designers have modified the MAC dialogue to be just a DATA-ACK handshake since they argue that the small amount of information that sensor nodes need to send does not justify the RTS-CTS handshake to establish the link. As such, each node becomes active for two slot times \( (2 \times \mu) \) per ripple, where one slot is

\[
\mu = BP + CW + DATA + SP + ACK
\]
Where

- BP = back off period (similar to DIFS in the 802.11 protocol)
- CW = the size of the fixed contention window
- DATA = the time required to transmit a packet of information
- SP = short period (similar to SIFS in the 802.11 protocol)
- ACK = time required to transmit an ACK control frame

One slot is just enough time for one packet of information to be transmitted successfully from a child node to a parent node in the data gathering tree.

For the case in which there is no information to be delivered from leaf to root, leaf nodes do not need to become active and all branch and root nodes become active for $\mu$ seconds before they sleep until the next rendezvous period because they know that there is no information to be sent. When there is one packet to be delivered from leaf to root it is sent during a node’s regularly scheduled active time.

So far we have only discussed a ripple that can carry one packet of information per rendezvous period, however DMAC is adaptive to the level of traffic within the network. If a node has more than one packet to send, the node schedules another ripple shortly after the first ripple separated by a certain amount of time\(^{11}\) to allow the packet within the first ripple to get clear of the radio space around the node with a second packet. DMAC achieves this by piggybacking More To Send (MTS) information onto the MAC frames that are sent (DATA, ACK).

This process is illustrated in Figure 3.11 through the use of a network scenario similar to the one depicted in Figure 3.10 except in this case node A has two packets of information to be sent to node F. One can also see that the MAC dialogue has been altered to just a large DATA frame followed by an relatively short ACK frame.

\(^{11}\text{typically } 3 \times \mu\)
Node A becomes active at A1, backs off and then transmits its first DATA frame. This frame has a more data flag set within it that node B recognizes and confirms in its ACK frame that it returns to A. Node A sleeps for $3\mu$ seconds to give the first packet a chance to get clear before reawakening and transmitting its second DATA frame. In this second frame the more data flag is not set since A does not have any more information to send. As packet 1 is transported through the network the more data flag will be set at each hop and thus all nodes will reawaken $3\mu$ seconds later to carry the second packet. One can now see how the ripple renewal mechanism in DMAC works and allows nodes to consume energy in a highly efficient manner similar to that of T-MAC while at the same time allowing information to be delivered in a timely manner by solving the data forwarding interruption problem.

Now that it has been shown how data is transported from a leaf node to the root node we now look at how a DMAC network behaves when data paths are combined to form a data gathering tree such as the one depicted in Figure 3.12.
We first consider the case where nodes A3 and B3 both have a packet of information to send to node A2. Since A3 and B3 are siblings in the tree they have the same rendezvous phase. When they enter the send part of their active time only one of them can win the channel to send their information packet. Either node will not have the more data flag set when they send their DATA frame since they do not have more information to send (it is their sibling node that has more data to send) and because of this the node that loses out in contending for the medium will have to queue their packet until the next active period.

To solve this problem DMAC uses a data prediction scheme in which nodes become active again $3\mu$ seconds after receiving a packet of information. This means that after receiving every packet of information a node will go to sleep and reawaken after $3\mu$ second to see if there is any more information that needs to be carried through the network. The means that every rendezvous period a node spends at least one receive slot ($\mu$ seconds) in idle mode\textsuperscript{12}. However, as the Authors of DMAC state - the reduction in latency that DMAC affords is worth the extra energy expenditure.

Another problem that occurs with DMAC is when nodes on different branches are still within interference range of each other as is the case for

\textsuperscript{12}this is a similar behavior to the $T_i$ seconds that T-MAC nodes stand idle for every rendezvous period
nodes B3 and C3 in Figure 3.12. If both B3 and C3 have information to send to their respective parent nodes in the same rendezvous period, then only one of them will be able to transmit since they are within range of each other. The data prediction scheme does not schedule an extra ripple for the node that loses out in contention for the medium and so that node will have its packet of information delayed until the next rendezvous period.

To resolve this problem, DMAC uses a control frame called a More To Send (MTS) frame that is sent after a packet is transmitted. This MTS control frame explicitly informs a node’s parent to become active 3\(\mu\) seconds later. To accommodate the MTS frame, the value of \(\mu\) must be increased to include the time required to transmit an MTS frame. A node will send an MTS frame if either:

1. It has not sent a packet because the channel is busy or,

2. If it receives an MTS frame from a child node. This is aimed to alert all nodes along the data route to schedule an extra ripple.

Although the MTS frame is very short, there is an associated energy and latency cost paid, however this is needed to allow the uninterrupted flow of packets along different branches in the data gathering tree.

3.7.1 Synchronization

The paper that presents the DMAC protocol [32] deals with the question of synchronization very briefly by stating that local synchronization is adequate for maintaining reliable connectivity since each node need only be aware of its neighbor’s schedule. The synchronization achieved has a precision three orders of magnitude smaller (in time) than the size of \(\mu\).

3.7.2 Implications of DMAC

Having explained the operation of the DMAC protocol, the reader can now see that DMAC nodes spend the majority of their energy transmitting or
receiving information (doing useful work) as well as allowing information to be transported through the network with minimal delay. Unfortunately DMAC is not independent of other layers in the protocol stack. According to the developers of S-MAC in their work in building a protocol stack for sensor networks [61] this independence of layers is an important criterion in the design of sensor network protocols.

What DMAC does, is combine aspects of the MAC/Data Link Layer with aspects of the network layer above it that is responsible for routing when it establishes its data gathering trees. This is most clearly stated in [32] . . .

" we should note that this comparison is only applicable under the specific data gathering tree scenario for unidirectional communication flow from multiple sources to a single sink. S-MAC is in fact a general-purpose MAC that can handle simultaneous data transmissions and flows between arbitrary source and destination. For applications that require data exchange between arbitrary sensor nodes, DMAC cannot be used "

3.8 Conclusion

In summary, S-MAC presents solutions to all four sources of energy inefficiency identified, but the solution to idle listening leaves plenty of room for improvement. The T-MAC protocol is far more energy efficient than S-MAC, but suffers from the early sleep problem during long-haul source to sink reporting. Finally there is a routing/MAC protocol (DMAC) that is also extremely energy efficient and allows timely delivery but constrains the possible routes that information can take through the network to just that of a static data-gathering tree.

So, what is needed is a system that
1. incorporates the energy efficient mechanisms of S-MAC (message passing and overhearing avoidance),

2. is adaptive to the level of traffic within the network as the T-MAC protocol is (the energy expenditure of each node is proportional to the level of traffic within the network),

3. allows packets to be delivered quickly along long-haul source to sink reporting as the DMAC protocol does, and

4. is independent of the routing layer that traditionally sits above the MAC layer and allows routes to be dynamically created on demand or as they are available.
Chapter 4
Proposed Solution

Having introduced the reader to the concept of a ripple, it is shown how ripples can be implemented to pursue the four objectives outlined at the end of Chapter 3. We begin by describing the operation of a one dimensional network and explain how our ripples differ from those used by the DMAC protocol. We then move on to examining how ripples can be implemented in a two dimensional grid network and the routing implications are discussed. The chapter continues by relaxing the requirement for a grid network by incorporating some related research that allows ripple rendezvous to operate in an unordered sensor network topology. The chapter is concluded with a discussion of the initial establishment of ripples within the network.

4.1 Ripples in a One Dimensional Network

Referring back to the discussion of T-MAC in section 3.4 we see that this protocol is well suited for the traffic pattern in which nodes exchange messages locally in what is known as local gossip. It is only when information must be sent along a long-haul multihop route that the protocol suffers from a significant delay in delivering information. The rest of this thesis is concerned with the delivery of information along an extended multihop route. A discussion on how ripple rendezvous supports the exchange of locally broadcast information is given in section 6.2.

4.1.1 Timeliness of Delivery

When a MAC protocol uses a sleep mode duty cycle as an energy saving technique, there is a penalty paid in terms of a delay in the transport of
network layer packets through the network in their delivery to their ultimate destination. This delay has been termed sleep delay by the researchers who brought us the DMAC protocol [32] and it is incurred by the packet while waiting for the MAC layer to become active before the packet can be transmitted to the next node along the route. Given that nodes become periodically active every $T_{rp}$ seconds and that a network layer packet could arrive at the MAC layer at any time, then the initial sleep delay will be on average $\frac{T_{rp}}{2}$ seconds long.

In the model of S-MAC 2001 used in this document, a short route may only incur an initial sleep delay since the network may be tuned to allow delivery in one active period. However, as the route length increases, it becomes increasingly difficult to achieve this. Depending on how S-MAC 2001 is tuned, the delivery of a packet may span multiple active periods separated by a delay while the network sleeps. For a network using S-MAC 2003, a packet will incur a similar delay after two hops and the T-MAC protocol incurs this delay after three hops\footnote{This is when the FRTS mechanism is used, otherwise T-MAC incurs a sleep delay after just two hops}. This delay after a certain number of hops has been termed the data forwarding interruption problem by developers of DMAC. Given that a packet must wait until the next time the MAC layer becomes active, this data forwarding delay is approximately $T_{rp}$ seconds every 2 hops for S-MAC 2003 and 3 hops for T-MAC.

The reader will recall from section 3.6 that a ripple is a pre-arranged sequence of source-destination pairs successively establishing connectivity along a data path. This rippling behavior means that information can be delivered in a timely manner. Let us examine this closer.

For a network in which nodes have coordinated their active times to form a ripple, there is an initial sleep delay incurred. Since the active times are staggered along the network (which is also the data route in a one dimensional network), there is no further delay of the packet until it arrives...
Proposed Solution

at its eventual destination. In other words there is no data forwarding delay.

To ensure rapid delivery of packets by the MAC layer (except for an initial sleep delay), each node must have its active time set to be twice the time required by the MAC protocol to transmit one network-layer packet as for the DMAC protocol. This is the carrying capacity of the ripple that was introduced in Section 3.6 of Chapter 3. The active time must be twice the time required by the MAC protocol to transmit a packet since each node must first receive the packet from its upstream neighbor in receive mode and then transmit it to its downstream neighbor in send mode.

Unlike the ripples employed by DMAC, the ripples used by the Ripple Rendezvous scheme do not schedule additional ripples to adaptively handle variations in the level of traffic. An explanation as to why this is the case is left until the discussion of ripples in a two dimensional network. For a single ripple to carry more than one packet of information the active time of each node must be increased to provide the capacity for more packets to be transmitted within the ripple. For example, to increase the carrying capacity of a ripple to be able to carry three packets of information then the active time of each node must be set to be six times the time required to transmit one packet. This capacity for more packets to be delivered has an effect on the delay in delivering packets to their ultimate destination. By increasing the carrying capacity of the ripple, there is the effect of decreasing the speed at which the ripple progresses through the network and as such there is an associated decrease in the speed of delivery.

Thus we reach a fundamental trade off when discussing ripples. For more information to be delivered per rendezvous period, the carrying capacity of the ripple must be increased and by increasing the carrying capacity of the ripple one increases the delay in delivering information. Again, we state that this topic will be discussed further in the section on the two dimensional implementation of ripples.
A ripple is almost entirely characterized by the active time \( (T_a) \) of each node that coordinates its rendezvous phase to form a ripple. It is this value that determines both the amount of information that can be delivered per rendezvous period and the speed at which the information is delivered. The *carrying capacity* of a ripple is the amount of information that can be transmitted or received in half of the time a node is active for \( (T_a/2) \) and thus there is an upper bound on the amount of information that can be carried by a ripple in one rendezvous period. It is the ripple that the nodes have organized themselves to be part of (and more specifically, the value of \( T_a \)) that determines the speed that information is delivered at. The ripple propagates through the network at a rate of \( T_a/2 \) per hop meaning that a packet of information advances one hop along its route every \( T_a/2 \) seconds.

### 4.1.2 Energy Efficient Adaptability

The MAC protocol that operates with Ripple Rendezvous is called Ripple-MAC or R-MAC for short and is very similar to the T-MAC protocol. To ensure the energy expenditure is proportional to the level of traffic within the network, an idle listening shutdown mechanism is also used. To recap, this is where nodes shift all of their communication to the start of their active time and transmit their backlog of cached packets. This has nodes either sending or receiving information unless they are sleeping to avoid overhearing a neighboring transmission. Nodes will then enter sleep mode until their next scheduled active period if they have not sensed any network activity for some time \( T_i \).

When a node’s \( T_i \) timer fires, it enters sleep mode until its next scheduled active period and can do so without causing the early sleep problem because it becomes active at the same time as the upstream neighbor finishes receiving the last of the information that it will receive in its current active period. Rephrased, the upstream neighbor has received all of
the information that will be forwarded to the node and if there is a time of radio silence for $T_i$ seconds, it means that the upstream neighbor has no more information for the node which can enter sleep mode without causing the data forwarding interruption problem.

We now examine two cases for a six node, linear network. The first case is for a network that does not have any packets to be delivered and is depicted in Figure 4.1. In this case only the second half of node A’s active period is shown since it does not have any upstream neighbors and only the receive part of node F is shown since it does not have any downstream neighbors.

Referring to Figure 4.1, we pick up the example at B1 when node B becomes active to receive information from node A (and node B does not know that this time there is no packet to be sent to it in this active period). After $T_i$ seconds, node B has not sensed any network activity and so its idle listening shutdown timer fires and node B returns to sleep mode. The next node to become active is node C which does so at C1. Similarly node C becomes active and waits for a time of $T_i$ seconds before shutting down to prevent unnecessary idle listening.

This process continues at nodes D, E and F with each becoming active (effectively to check with their upstream neighbors if there is information that must be carried) and then shutting down to avoid idle listening (in other words, when it becomes apparent that there is no information to be carried, nodes return to sleep mode). In this case nodes do not become active during the send half of their active time since they do not have any information to send to their downstream neighbors. From this we see that idle listening is restricted to $T_i$ seconds per rendezvous period.

The second case depicted in Figure 4.2 is when node A has one large multi-fragment packet to deliver to the node F and Node C has a small, single fragment packet bound for node E. The large packet is fragmented.
Figure 4.1: An empty ripple
and sent in a burst like the S-MAC protocol since message passing is an energy efficient method of sending large packets.

In Figure 4.2, node A becomes active at A1, backs off, establishes a link, transmits the multiple DATA fragments in a burst (as described in the S-MAC section) until A receives a final ACK from B after which A goes to sleep at A2, as it has no more information to send.

B awakes at B1 and immediately starts its $T_i$ timer. However before the timer fires, it receives the RTS from A and responds with a CTS after which it starts to receive the DATA fragments (B sends an ACK for each fragment correctly received) ending with the final ACK. B then hears no further network activity so at B2 its $T_i$ timer fires, and node B shuts down to avoid idle listening. B reawakens at B3 since it knows that node C is active and waiting and firstly backs off, establishes a link, transmits the DATA payload, receives an ACK and then goes to sleep at B4 since its MAC buffer is empty.

Node C awakes and receives the large multi fragment packet from B and then shuts down after its $T_i$ timer fires. It then reawakens in transmit mode and backs off before transmitting the large multi-fragment packet. C then backs off again before transmitting the small single fragment packet.

Node D finishes receiving the large multi-fragment packet at D1 and before its $T_i$ timer fires, it receives the single fragment packet. Node D then becomes active again as node E becomes active and transmits the multi-fragment packet followed by the single fragment packet. Node E finishes receiving the large multi-fragment packet at E1 and then receives the small packet. We include this small packet to illustrate the point that ripples can carry packets with varying sizes as long as each node limits the amount of information to the amount that can be transmitted in $\frac{T_a}{2}$ seconds.

The reader may be curious to know why each node backs off before attempting to establish a link when the each node is not competing with any
Figure 4.2: A ripple carrying traffic
others for the medium since they are the only ones in transmit mode at the time. This is correct for the one dimensional network that we are examining, however, as we shall see when considering a two dimensional network, it is necessary to have this contention period.

4.1.3 Independence of MAC layer

The radio-state diagrams (Figure 4.1 and Figure 4.2) are large, bulky and represent information that is too detailed to depict the situation clearly as we begin to discuss the collective behavior of sensor nodes. Having shown how nodes in a Ripple Rendezvous network adaptively scale their energy consumption as the level of traffic in the network increases, we can present a general case that is independent of the level of traffic in the network by describing the state of the MAC protocol as either:

1. *dormant mode* - when no information is transceived since the node’s radio is off,

2. *receive mode* - when the node receives information from its upstream neighbor. During this time, a node may actually be receiving (a RTS or DATA frames from its upstream neighbor), transmitting (a CTS or ACK frames to its upstream neighbor) or in sleep mode (either because it is avoiding overhearing or its $T_i$ timer has fired, meaning that the upstream neighbor has no more packets to send)

3. *send mode* - when the node sends information to its downstream neighbor. During this time a node may actually be transmitting (a RTS or DATA frames to its downstream neighbor), receiving (a CTS or ACK frames from its downstream neighbor) or in sleep mode (it is avoiding overhearing or since it has finished sending all the packets that it has to send).
Figure 4.3: State transition diagram of MAC modes

Figure 4.4: Behavior of a node in receive mode

Figure 4.3 depicts the state transition diagram for a node that forms part of a ripple. A transition timer indicates when the MAC should change between one of the three modes.

For reference, a simplified version of the behavior of the R-MAC protocol while in receive mode is depicted in Figure 4.4. Similarly, the send mode diagram is depicted in Figure 4.5.

Figure 4.6 represents both cases depicted in Figures 4.1 and 4.2 with each node being represented by a single thin line when in sleep mode (similar to previous diagrams). When there is connectivity between two nodes (both their radios are on), this is represented by a rectangle enclosing both node time lines that is labelled with an arrow showing the direction of packet
movement and each node’s MAC state is labelled as either receive mode (with an R) or send mode (with an S). These diagrams are independent of the level of traffic within the network as long as the level of traffic does not exceed the amount of information that can be sent by a node in $\frac{T_a}{2}$ seconds. While this may appear to be a severe constraint on the usefulness of the Ripple Rendezvous scheme, it shall be demonstrated in the section on two dimensional ripples that while this does limit the amount of information that can be carried by the network, the problem may not be as restrictive as first thought.

Any node can deliver information to any other node that is to the right of it (as long as there is remaining capacity within the ripple) by transmitting to their right-hand neighbor during the ripple that moves to the right (the right ripple). In a one dimensional network the ripple effectively ‘sweeps’ along all possible routes in the network, allowing information to be transmitted within it as it passes through the network.

To allow information to flow quickly in the other direction, another ripple moving in the opposite direction is required. This left ripple has

Figure 4.5: Behavior of a node in send mode
nodes behaving in a similar manner as during the right ripple and allows information to be delivered from any node to any other node that is to the left of it. These ripples may avoid each other by having the left ripple wait until the right ripple has moved through the entire network before the left ripple commences its progression through the network, however as the network increases in size this becomes increasingly difficult to achieve, thus this solution does not scale well. To implement both ripples in a large scale network, one avoids the other by giving way as depicted in Figure 4.7.

There must be sufficient time at node E for the right ripple to pass between the receive and transmit halves of the left ripple. There is not enough time for the send half of of the left ripple immediately after receive mode for fear of interfering with packets that may be transferred during the right ripple. The send half is scheduled for a time that gives the right ripple a chance to get clear of the local radio space. This means that information moving to the left will be delayed slightly. This delay is denoted as $T_{gw}$ in Figure 4.7 as well as minimum and maximum values for this delay.

The minimum value of $T_{gw}$, is $2T_a$ seconds. This consists of half an active period ($\frac{T_a}{2}$) prior to the node becoming active in receive mode for the
Figure 4.7: Mode Diagram showing a left ripple giving way to a right ripple
Proposed Solution

Figure 4.8: State transition diagram of MAC modes for ripple that gives way

right ripple\(^2\) plus an entire active period (\(T_a\)) during the right ripple as well as another half an active period (\(\frac{T_a}{2}\)) afterwards to allow the right ripple to move out of the local radio space. The maximum delay can similarly be described as \(3T_a\) seconds. The initial delay caused prior to the arrival of the right ripple can be up to but not including one and a half active periods. If there were to be one and a half active periods or more between the end of node E’s receive period for the left ripple and the commencement of node E’s receive period for the right ripple, Node D would be able to receive all transmissions from node E without fear of collisions with node C’s transmissions. Node D would then delay the progress of the left ripple until the right ripple had a chance to get clear of the local radio space. The other delays of an active period during the occurrence of the right ripple and another half an active period to allow the right ripple to get clear remain unchanged. The precise value of \(T_{gw}\) relates to how many times the active period fits into the rendezvous period.

The state transition diagram that represents the changes in MAC mode of the node that forms the ripple that gives way is depicted in Figure 4.8.

\(^2\)this is the ripple that is NOT giving way
Proposed Solution

If there is a greater energy cost borne by the node that is responsible for the ripples avoiding each other (in this case, node E) then if the rendezvous period \( (T_{rp}) \) for the left ripple were slightly different to \( T_{rp} \) for the right ripple, this would result in the actual point of intersection being distributed across many nodes in the network over multiple rendezvous periods during the lifetime of the sensor network.

To be clear, a node will schedule one ripple to give way to the other when it calculates that the time when it becomes active for a ripple moving in one direction is within \( \frac{3}{2}T_a \) seconds of the time that it becomes active for the ripple moving in the opposite direction. Intersecting ripples will only be a problem of both ripples are carrying traffic and the transmission of information within one ripple may interfere with transmission of information within the neighboring ripple.

There are a variety of options available when determining which ripple gives way to which. Ripples may take turns giving way to each other so that information moving in a particular direction is not unfairly disadvantaged. An alternative is to specify that the left ripple always gives way to the right or visa versa. Precisely which option is the better and how great a problem this may be is the subject of future research to be conducted.

It is now apparent how the ripples that we have developed differ from those used by the DMAC algorithm. Under the DMAC protocol, each ripple is used to carry just one packet of information. If there are more packets to be carried to the sink node, more ripples are dynamically scheduled, one after the other, until all packets are delivered to the sink. This is fundamentally different to the ripples that we have developed in that our ripples have nodes becoming active, receiving all the information that can be sent to them (remembering that this is the amount of information that can be received in half of a node’s active time) and then transmitting this information (minus the packets that are ultimately destined for this node.
Proposed Solution

and plus packets that the node has for other downstream nodes) to the next node along the ripple pathway. We acknowledge that this means that packets of information will not be delivered as quickly as DMAC and also that there is an upper bound on the amount of information that can be delivered per rendezvous period, but as we shall see in the next section that discusses two dimensional networks that this is necessary to allow information to be delivered between arbitrary source and sink nodes.

Having shown how information can be delivered in a timely and energy efficient manner from an arbitrary node to any other node in a one dimensional network, we now turn our attention achieving this result in a more practical and real world scenario - the two dimensional network.

4.2 Ripples in a Two Dimensional Grid

Having shown how a one dimensional network can be organized to form two ripples that allow information to be delivered in a timely and energy efficient manner between arbitrary nodes, the challenge is now to organize a two dimensional network to do the same.

Before continuing, we need to state our assumptions about the underlying topology of the two dimensional network that being considered. Having discussed a one dimensional network in which nodes are only within range of their nearest neighbors, this restriction is kept and the logical extension to that of a grid topology that has nodes only being within range of their nearest neighbors is made. Such a network topology is depicted in Figure 4.9.

This grid topology may very well appear to be a ludicrous assumption for the topology of a sensor network which requires thousands of nodes to be distributed in an ad-hoc fashion over a wide area, as does the use of a very simplistic disc model to reflect the characteristics of radio propagation. For the time being, these assumptions will suffice until section 4.4, in
which a topology control technique is discussed to bring some order to the random dispersion of nodes as well as a modification to the MAC protocol is discussed that allows for non-uniform radio propagation.

In a two dimensional network, ripples are said to ‘overlap’. What is meant, is that nodes in the network that are correlated in their geographic location have the same rendezvous phase. This means that many nodes become active simultaneously and form what is termed a wave of connectivity that sweeps through the network.

There are many ways that ripples can be organized to overlap and form waves in a two dimensional grid network, however in allowing the formation of waves, information can only be sent along a subset of all possible routes. Three different methods of scheduling the activity of nodes in a two dimensional network were investigated. The third and final method is presented next. The first two patterns that are the result of early exploration of the design space are included in Appendix B.
Figure 4.10: Phase offset for northern wave

4.2.1 Geo-Waves

This rendezvous scheme\textsuperscript{3} has four waves and is achieved by scheduling successive rows and columns of nodes to become active in a cascading manner. Each of these waves is named after the general direction in which they allow packets to be delivered (north, south, east and west).

Figure 4.10 shows how the phase is offset for nodes during the northern wave. The color that represents each node is indicative of its phase offset with the darker colors having a greater offset. The northern and southern waves have rows of nodes with the same phase offset and the eastern and western waves have columns of nodes with the same phase offset. If the network is large enough, there may be more than one geo-wave in each direction moving through the network at the same time.

During the northern waves, a node can send information to any other downstream node within the range as indicated by Figure 4.11 with the source node being portrayed in black, possible destinations shown in grey and unreachable destinations shown in white. Remember that the MAC protocol is responsible for delivering a packet at each hop and it is the routing protocol that decides which route a packet takes to reach a specified destination. Two example routes are also shown to reinforce this explanation.

\textsuperscript{3}the term \textit{Rendezvous Scheme} is used to refer to the sequence in which nodes in a network become active before returning to sleep mode
Thus a packet can be delivered from an arbitrary source node to an arbitrary destination by having the MAC protocol wait for the appropriate wave (north, south, east or west) as specified by the routing layer and then the MAC protocol sending the packet within it. It is possible to see now how almost every route is possible under the geo-wave rendezvous scheme giving the routing protocol flexibility in deciding how a packet should be routed from a sensing node to a sink node and making few demands of it.

In a grid network such as the one we have been discussing, the number of minimum-hop (or optimal) routes available between a source node and a sink node decreases as the sink node becomes oriented closer to the diagonal relative to the source node. This point is illustrated by Figure 4.12, which depicts the possible routes between a sensing node (indicated as a black circle) and different sink nodes (drawn in grey) for a three hop route.

As depicted in this figure, there are seven possible three-hop routes.
that a packet can take to get from a node to the node that is directly to the north of it. There are six routes to get to the node that is one hop to the east (or west, by symmetry) of it, three possible routes for a node that is two hops to east of it and, although not depicted, it should be clear that only one three-hop route is possible from a node to another node that is three hops to the north as well as three hops to the east. It is assumed that the route that a packet will travel along is as close to the line that connects the sensing and sink nodes as the grid network will allow, ie. for the purposes of discussion, straight line routing is assumed.

Another feature of the geo-wave front rendezvous scheme is that it is not possible for a packet to travel along a sub-optimal route while travelling within one wave front (either north, south, east or west). For a packet to be transmitted from sensing node to sink node along a sub-optimal route, a combination of waves must be used. For example if the sub-optimal route depicted in time slices 1 to 8 in Figure B.3 were to be specified for nodes using the geo-wave rendezvous scheme, the packet would complete the northerly part of its route during the northern wave, the packet would incur a sleep delay at the top-left node until the eastern wave allows it to complete the remainder of its route. To be clear, it is possible for information to be delivered along any route under a geo-wave rendezvous scheme and the route chosen does not have a significant influence on the energy expended by each node along the data route\textsuperscript{4}.

The time of delivery, however, will be affected. Since the wave that the packet changes to can arrive at any time within the next rendezvous period depending on where in the network the change takes place, the packet will on average experience a delay of $\frac{T_{rp}}{2}$ seconds each time the packet changes the wave that it is being carried within.

\textsuperscript{4}it will of course have an effect on the total energy consumption metric if the route requires more nodes to carry the packet through the network
Figure 4.13: Two simultaneous intersecting routes

to be delivered between arbitrarily chosen nodes, along arbitrarily chosen optimal routes, that is energy efficient and is also delivered in a timely fashion. However we have only examined the situation for very low levels of traffic. Our discussion holds for traffic levels starting from zero up to one packet of information that does not encounter another packet on its journey from sensing node to sink node per rendezvous period. The next situation that requires scrutiny is when simultaneous routes intersect. What is meant by ‘two simultaneous routes’ is two packets that have been received at the MAC layer within the previous rendezvous period, who’s routes intersect while they travel within the same wave. Two simultaneous intersecting routes are depicted in Figure 4.13 with each of the two routes indicated in a different shade of grey as well as the extent of the transmission range of nodes involved in carrying the packets through the network. This is to give the reader a reference point for the concepts and processes we now discuss.

These routes result from packets that have arrived at the MAC layer
Proposed Solution

Figure 4.14: Blocked senders and receivers

(Along with instructions for which node to transmit the packet to and which wave to send it within) within the last rendezvous period for them to be travelling within the same wave and so this is not likely to occur often. However as the size of the network, the length of the routes, the number of users, the number of reportable events and the level of interest increases so does the chance of simultaneous packet routes intersecting. R-MAC operates within the ripple rendezvous scheme such that it manages to work one packet around another as they are both transmitted towards their respective destinations.

The problem that can occur is a variation of the early sleep problem that T-MAC suffers from. Referring to Figure 4.14, a node that is blocked (nodes BS1, BR1, BS2 and BR2; BS stands for blocked sender and BR stands for blocked receiver) due to the transmission of packet 1 from node A to node B is not be able to either send or receive a second packet to/from another node (either nodes R1, S1 or R2, S2; R signifies a node in receive mode and S is a node in send mode) that are out of range of the transmission of packet 1 and are thus unaware of the situation. Node R1 will sleep early while waiting for a transmission from node BS1 and node S1 will waste energy trying impotently to establish a link with node BR1 before giving up and sleeping early. Both situations will result in the second packet being delayed until the next active period.

The solution to this problem is similar to that of T-MAC in that the MAC dialogue is modified to include a DS control frame and a control
frame is transmitted by a blocked receiver node that is not the intended destination of the packet at the same time as the DS control frame. Only blocked receivers need transmit, since the transmission will inform both nodes R and S of the situation. This control frame is termed a Broadcast on Block (BoB) frame. The BoB transmitted by node BR1 will collide with the BoB transmitted by node BR2 at both nodes A and B as in the T-MAC protocol, but nodes S1, R1, S2 and R2 receive their corresponding BoB control frames uncorrupted by collision.

A more complex situation occurs when a node transmits to its diagonal neighbor as depicted in Figure 4.15. In this situation there are three blocked receivers and three block sending nodes. The difficulty in this situation is that unlike the situation depicted in Figure 4.14, not all blocked nodes will receive both the RTS and CTS control frames that establish the link, in fact the only nodes that do receive both of these control frames are node BR2 and BS2. Thus node BR1, upon overhearing a CTS control frame addressed to a node that is out of range (as node A is) must transmit a BoB control frame that alerts nodes R1 and S1 of the impending transmission as well as the expected completion time (since this information is contained within both the RTS and CTS frames). Node BR3 upon overhearing a RTS control frame to a node that is out of range (as node B is) must transmit a BoB control frame that informs nodes R2 and S2 of the impending transmission as well as when to reawaken to either send or potentially receive a
Proposed Solution

if (this node is in receive mode)
{
    wait();    // wait until a frame is received
    if (node overhears RTS for node B from node A
        && this node is NOT in line with node A)
        && node B is out of range of this node
        //determined from address of node B 
        //contained within the RTS frame
    {
        wait();    //wait until after the CTS frame 
        //has been transmitted by node B
        transmit_BoB_frame();
        //while node A transmits a DS frame
    }
    //if (node overhears CTS for node A from node B)
    transmit_BoB_frame();
    //while node A transmits a DS frame
}

Figure 4.16: Pseudo code for BoB transmission

second packet (if there is one).

From these two situations, the algorithm that determines a neighboring node’s transmission of a BoB control frame is given in figure 4.16 in pseudo code. A node (X) is said to be 'in line' with another node (Y) if node X is directly opposite node Y in the direction that the wave is moving in. Thus node B is in line with node A in the example shown in figure 4.14 and node BR2 is in line with node A in figure 4.15.

MAC addresses can be allocated according to the grid coloring problem that ensures that there no address is repeated within a certain range. Nodes are then able to determine the location of another node in the grid from its address. This also means that MAC addresses need not be as long as network addresses which should be unique to every node in the network. This will also have the effect of reducing the additional overhead introduced by the MAC protocol.
Proposed Solution

The reason for excluding nodes from sending a BoB if they are in line is to prevent node BR2 from sending a BoB which would interfere with the BoB sent by node BR3.

These BoB frames serve to alert two-hop neighbors on both sides of the transmission of the ongoing and impending network activity so that the neighbors do not waste energy and/or erroneously shut down before they are required to relay information. The subsequent behavior of nodes R and S depends on the route of the second packet that they have to send or receive as well as other traffic commitments within the network.

A node in receive mode that receives a BoB will wait for $T_i$ seconds before sleeping until the expected completion time of the link when it will reawaken when the medium is free to possibly receive further information from the blocked node. The reason it waits for $T_i$ seconds before sleeping is to give a chance for nodes that are beyond two hops from the transmission to transmit to the two hop neighbor. For example, node R2 waits for $T_i$ seconds before sleeping in case either node S2 or its blank neighbor next to it has a packet destined for node R2.

A node will only be active in transmit mode if it has traffic to send, otherwise it sleeps. A node in transmit mode that receives a BoB will sleep until the expected completion time of the link if it only has a packet for the blocked node. If it has other traffic commitments destined for other neighbors beyond two hops, then it can set about fulfilling them while packet 1 is being transmitted.

If the link were to be extended beyond its advertised completion time (due either to a collision or because of channel errors or noise) then two hop nodes that are sleeping to avoid overhearing would reawaken at the previously advertised completion time and would face the same problem that saw the introduction of the BoB control frame in the first place. It is thus necessary for nodes, that realize that the link they have established
will go beyond the time that was advertised, to reinitialize the link at the previously advertised completion time. Nodes A and B that have established a link that has been extended (for whatever reason) will need to re-form the link with a RTS-CTS-DS:Pause and subsequent BoB control frame exchange which allows the information that the link has been extended to reach the two hop neighbors. This higher control frame overhead is required to ensure unhindered relaying of packets from source to sink.

The carrying capacity of the waves needs to be great enough such that nodes can work packets around each other. For two packets to work around each other, the waves will need a carrying capacity of two packets or greater, for three packet routes to intersect, the carrying capacity will need to be three packets or greater. If there is more traffic than the waves are able to carry, then some packets will be left behind and will have to wait for the next wave in the next rendezvous period. It must be said that although one node has a maximum traffic load per rendezvous period, there is nothing to stop many routes through the network carrying traffic. It is only when the routes are within interference range of each other that there is a problem.

Implementing the north and south waves is now fully explained with the two waves avoiding each other as shown in Figure 4.7. Mention should be made of the situation when perpendicular waves intersect (north/south intersecting with east/west). It is not possible to have these waves avoiding each other, however for this to be a problem, then the point at which two perpendicular wave intersect must also coincide with the point at which two simultaneous routes intersect. If we are using slightly different rendezvous periods for different waves, then this point of intersection will occur at different points in the network each rendezvous period, and so this case will be a rare event indeed. A worst case scenario is that one of the packets may be left behind and will have to be transmitted in the next rendezvous period.
The packet will incur an extra delay of $T_{rp}$ in its delivery (this is hardly a train smash) and on average, packets will still be delivered in a timely manner. This is an issue to be explored in further research.

In summary, the geo-wave rendezvous scheme meets all of the design requirements outlined in Section 3.8, namely that of timely delivery (due to the staggered rendezvous phase or a ripple rendezvous network), energy efficiency (because R-MAC uses the idle listening shutdown mechanism) while allowing the energy saving mechanisms derived by S-MAC to operate (message passing and overhearing avoidance) as well as allowing packets to be delivered between arbitrary sensing and sink nodes along a wide selection of optimal routes.

4.3 Routing Implications of Ripple Rendezvous

4.3.1 Unicast Traffic

Unicast traffic is traffic that is destined for one particular node in the network rather than a subset of nodes. The primary restriction that Ripple Rendezvous places on the routing of information is that the number of simultaneous intersecting routes must not exceed the carrying capacity of the wave they are being transmitted within, otherwise information will be left behind that will make its way to the sink node within the next wave in the next rendezvous period. This is not desirable since we would like information to be delivered in a timely manner. So just how great a restriction is this?

It is possible for multiple packets to travel through the network within the same wave, there is only a problem when the routes of these packets intersect. Figure 4.17 depicts an example of six different packets all travelling within the same wave in a 20 by 20 grid network. The example wave has a carrying capacity of two packets. The diagram is similar to Figure 4.13 with each node in the grid network shown as a circle with the
nodes lying along the route of each packet depicted as a filled grey circle. An arrow depicts the next hop in the route and the extent of a node’s radio range is depicted as a larger circle around the node.

The reader can see that the restriction of two simultaneous intersecting routes (for a wave with a carrying capacity of two packets) is not as severe a restriction as may have initially been thought when discussing a one dimensional network.

When routes converge to a particular sink node\(^5\), the carrying capacity restriction of a wave becomes a greater limiting factor, in fact the carrying capacity dictates the number of packets that can be delivered to a particular sink node per rendezvous period. For a user to be able to receive more packets than the carrying capacity of the wave allows, (s)he will need to have an interface device that is capable of interacting with many sensor nodes if they are to be the recipient of multiple packets along differ-

\(^5\)this type of traffic has become known as convergecast
Another example scenario is given in Figure 4.18 that depicts a situation in which five routes converge to a particular sink node. Unlike the previous diagram, the extent of the communication range is not depicted and nodes that lie along the packet route are not shaded. A sink node is labelled as such and this is the node where all the routes converge to.

The reader's attention is drawn to the large grey circle that represents the range of the interface device that a user of the sensor must have to be able to receive all five packets in the same rendezvous period for a wave with a carrying capacity of two packets. In effect, the user needs to interact with the network as a collection of nodes rather than just interfacing with one particular node.

In summary, yes there are restrictions placed on the routing and delivery of packets within a sensor network that has implemented the Ripple Rendezvous scheme. These restrictions, however, only begin to apply when the level of traffic within the network begins to exceed the carrying capacity of the waves. It is therefore up to the application designers to tune the
network prior to deployment just as S-MAC and T-MAC need to have their parameters tuned for the specific situation that they will be used for.

A network using Ripple Rendezvous is tuned by adjusting the parameters $T_{rp}$ and $T_a$ to give the required performance of the network as dictated by the particular application and desired level of use.

4.3.2 Broadcast Traffic

Information may also be sent to a subset of nodes within the network rather than just to one. This may be under the local gossip traffic pattern in which nodes share information with their immediate neighbors or during network-wide flooding of information as is common with some routing protocols.

For a node to distribute information to all of its eight immediate neighbors, the same information must be transmitted four times (once during each wave-front). The reason for this is that because ripple rendezvous schedules activity to be staggered throughout the network and there is never a time during which a node is active at the same time as all of its immediate eight neighbors.

Similarly, if information needs to be flooded throughout the network then the source node may be required to transmit the information during each of the four geo-waves. Nodes that will receive this flooded information are the reachable downstream neighbors similar to those indicated in figure 4.11 as possible destinations. For the information to be carried through the network within the geo-wave of connectivity, the restriction remains that the number of coincidental transmissions remain within the carrying capacity of the constituent ripples.

4.4 Ripples Over a Random Topology

Having shown in the previous section, that ripple rendezvous and R-MAC operate in a manner that is suitable for a sensor network with a regular grid
Proposed Solution

4.4.1 The Geographic Adaptive Fidelity (GAF) Algorithm

The GAF algorithm [60] was also developed by the sensor network researchers at USC/ISI working on the directed diffusion sensor project and used as a topology control mechanism to support a routing protocol in sensor networks. Although we are not directly concerned with the problem of routing, the topology control mechanism of the GAF algorithm is of interest to us.

GAF has nodes determining their topological equivalence through the use of geographical information. Geographical information may be determined by GPS or even better, by other location determination methods that do not require each node to expend energy to access the GPS infrastructure. These methods allow a small subset of nodes to use GPS information to determine their location directly and then this information is propagated through the network using beacon signals and distributed multilateration to allow all other nodes in the network to determine their own location [43] with centimeter precision. A virtual grid is superimposed upon the sensor-net and nodes that lie within the same grid cell are determined to be topologically equivalent. The grid is defined such that a node within one grid cell will be able to communicate with all nodes in the eight surrounding cells.

In Figure 4.19, for node B to be able to communicate with all nodes in the surrounding 8 cells it will need a radio range \( r \) great enough to be able to reach node A. Using simple trigonometry, this results in

\[
 r \geq 2\sqrt{2} d \tag{4.1}
\]

Since the radio range is fixed and predetermined for its energy effi-
ciency, it is the size of the grid cells that must be

\[ d \leq \frac{r}{2\sqrt{2}} \]  \hspace{1cm} (4.2)

Nodes that are equivalent can enter sleep mode as long as one remains active to represent the grid cell. The GAF algorithm has three states: *sleeping, discovery and active*. When a node is first deployed, it is in the discovery state which has the node discovering its equivalent neighbors in the same grid cell. The nodes exchange messages with each other and in doing so develop a ranking of all nodes in discovery mode. Based on this ranking system, they organize for one of them to transition to the active state for an agreed upon time \( T_{active} \) while the rest enter sleep state. The node that remains active is selected according to several ranking criteria including a node’s remaining energy supply meaning that nodes with greater remaining energy supplies are preferred to represent the grid cell. Nodes that are in sleep state will enter discovery state after some time \( T_{sleep} \) at which time they will coordinate with other nodes that are in discovery mode to again select a node to remain active to represent the grid cell. Care has been taken
in the design of the GAF algorithm to ensure that only one node is active at a time within each grid cell. For more information on the details of the GAF protocol please refer to [60].

The developers of GAF report that network lifetime increases proportionally to the density of nodes deployed within a given area. A four-fold increase in node node density increases the lifetime of the network between three and six times.

GAF represents a very important development in the field of wireless sensor networks as it takes advantage of one of the strengths of sensor networks, that of redundancy. By allowing nodes to take turns at representing their grid cell, the sensor network becomes less vulnerable to node failures i.e. if a node fails then the GAF algorithm ensures that another node takes over. It is the GAF algorithm that is responsible for maintaining a representative node for each grid cell and thus the robustness of the network.

There are a number of limitations and consideration to the effectiveness of the GAF protocol. The most significant of these is the fact that GAF assumes an idealized radio propagation model and effectively guesses at connectivity between two nodes rather than directly measuring it. This means that links may be non-existent which requires that the routing protocol be able to adapt to this link unavailability. This also means that GAF must make conservative estimates of the radio range $r$ and this results in less than optimally energy efficient operation. To address this shortcoming, the developers of the GAF protocol have developed the CEC (Cluster-based Energy Conservation) Algorithm [59].

The developers of GAF report in [60] that the protocol is not affected by moderate location error, however they do expect that the protocol will fail completely when the location error approaches the size of the grid cells. The protocol is robust under geographic location error that is correlated
between different nodes. This is due to the fact that GAF relies on relative node position when nodes are allocated to grid cells.

There is an associated energy cost with accessing geographical information through GPS infrastructure, however since nodes are static this only need be done once. As mentioned in Chapter 2 nodes in a sensor network are not mobile although this is not to say that they cannot be moved. Each time a sensor node moves to a new location it will in effect leave the network and will have to go through though the protocols by which nodes join the network. Each time a node moves it will have to again determine its new geographic location, but again the node need only do this once meaning that the energy cost need not be that significant.

The GAF algorithm approximately produces a network with a regular grid structured topology as is discussed. We acknowledge that a nodes radio will spill over into grid cells beyond the eight immediately neighboring cells. However this problem is remedied by making a relatively small modification to the R-MAC protocol.

4.4.2 Synchronisation

For nodes to reliably rendezvous with other nodes it is necessary that they be synchronized. For nodes in a network that is organised under the GAF algorithm, this is not likely to be a problem since information is derived from GPS, nodes can also use this system to retrieve synchronization information. It is desirable for the stationary sensor nodes to use global synchronization so that mobile nodes (that are also synchronised with the sensor net) know when to rendezvous with the sensor net no matter where they are within the network.

Accessing GPS infrastructure does have an associated energy cost. The frequency with which clocks need to be re-synchronised depends on the rate at which their internal chronometers drift over time. With traditional
quartz oscillators, clock drift is of the order of $1 \times 10^5$ [63]. However, recent discoveries in the field of MEMS technology [25] provide synchronization with a clock drift that is approximately five orders of magnitude better than this for a small energy cost.

4.4.3 GAF, R-MAC and Ripple Rendezvous

It can be seen that R-MAC and Ripple Rendezvous will function as previously described for a grid network when applied to a GAF network topology as long as packets travel through the network without moving within range of each other. Thus we must examine the situation when nodes that do lie within range of each other each try to transmit a packet within a wave.

As mentioned earlier, Ripple Rendezvous and R-MAC must be able to deal with the fact that a node’s radio transmission will spill over into cells beyond their eight immediately neighboring cells. What this means is that nodes that lie beyond the neighboring cells may also be blocked from either sending or receiving a second packet. To depict the extent of this problem the reader is referred to Figure 4.20 which depicts the transmission range of a node in eight possible positions within a virtual grid cell. The grid cell in question is depicted in darkest grey with the eight neighboring cells shown in the next darkest grey. The eight different positions at the extremities of the grid cell are shown as small black circles with the corresponding range from each position shown as a larger circle. Grid cells that that may possibly be within range are shown in light grey. The factors that determine whether or not a node in the dark grey cell can communicate with a node in a light grey cell are both of their exact positions within their corresponding grid cells.

The problem then, is for blocked nodes to inform their neighbors (that do not overhear both or either of the RTS, CTS control frames) that they are blocked. Since there are several nodes (each in a different grid cell)
Proposed Solution

Figure 4.20: Possible communication range from different positions within a virtual grid cell

that may be blocked, it is desirable to avoid the situation in which multiple nodes on the same side of the transmission, transmit a BoB control frame causing these BoB control frames to collide with each other at the intended recipients. To be clear, the intended recipients of the BoB frames are nodes that lie outside the range of the RTS and/or CTS frames but are within range of nodes that are blocked by the ongoing transmission.

The proposed solution to this problem is to modify the R-MAC protocol to increase the length time allowed for the DS control frame as in the T-MAC protocol except rather than the sending node transmitting a DS frame, it simply remains silent for the equivalent length of time. This allows nodes to contend for the right to transmit a BoB based on their geographic position. A node can be up to three grid cells away from the sending or receiving node and need to transmit a BoB frame. A node that is three cells away (a three cell neighbor) and is in a position that results in it being blocked (it is within range of the sending, receiving or both nodes) will transmit a BoB immediately. A node that is two grid cells away and
that is in a position that results in it being blocked, will transmit a BoB after first listening for a transmission from a three cell neighbor, and if it doesn’t begin to hear a BoB from a three cell neighbor, then it will transmit one. Similarly, a one cell neighbor will first listen for a transmission from a three cell neighbor, then a two cell neighbor and if it still hasn’t begun to receive a BoB, will transmit one itself. This means that the length of the DS control frame needs to be extended by two slot times. The reader is asked to recall from the discussion of the 802.11 protocol in section 3.2 that a slot time is a length of time used by nodes when initially contending for the right to transmit a frame.

This BoB contention mechanism is a system that has nodes, furthest from the ongoing transmission but still blocked by it, transmitting a frame to alert their neighbors that they will be blocked for a certain period. In the process, all the neighbors of closer blocked nodes will also be alerted of the ongoing transmission. This mechanism also allows us to remove the reliance in the disc model of radio propagation although we do retain our assumption of symmetrical radio propagation (if node A can transmit to node B, then we assume that node B will be able to transmit to node A). If an obstacle (eg, a rock or dense foliage) were to restrict the range of a node’s transmission, the BoB contention mechanism still results in the furthest blocked nodes informing all of their neighbors of the ongoing network activity as well as the expected completion time.

4.5 Initial Establishment of Ripple Rendezvous

Nodes use a combination of geographical information and a reference rendezvous phase for a particular grid cell and particular time. That is, they are supplied with a reference to a particular point in space-time during the predeployment phase two of its operational life. A node, after being deployed, during the self organizing phase three, establishes its geographic location
and self organizes under the GAF topology control algorithm, calculates its offset from the reference grid cell in both time and space and adjusts its own rendezvous phases accordingly. Nodes that are deployed can thus calculate the rendezvous phase that corresponds to the grid cell that they are located within as soon as they establish their geographic location relative to the reference location, time and rendezvous phase they have been supplied with.

This point is illustrated with a simple example using a one dimensional network. Consider a network of 10 nodes with one of the nodes lying in the reference cell (node 1) as shown in Figure 4.21. Note that it is not even necessary for the reference cell to lie within the actual area of the network, it is only required for all nodes to have something in common for them to all refer to.

During the pre-deployment phase, the nodes are programmed with:

1. A particular location that corresponds to a particular grid cell,

2. A set of particular rendezvous periods that all nodes in the network will use. The north-south waves may have slightly different rendezvous periods to the east-west waves so that the places where they intersect are distributed over the network i.e. the intersection points are different each rendezvous period. In this example only the formation of the east ripple is demonstrated and a rendezvous period of 1 second is chosen arbitrarily.
3. A set of times that a node in the reference grid cell becomes active for each of the four waves eg. 12:00:00.03 for the east wave.

4. How long each node will be active for at each wave. A $T_a$ of 0.02 seconds is used.

Having been deployed, determined its geographic location and self organized under the GAF algorithm, a node then calculates how many grid cells away from the reference location it is and adjusts its rendezvous phases accordingly.

So for node 2 to calculate its rendezvous time for the east ripple it knows that it will have to turn on $\frac{T_a}{2}$ seconds (10 msec) after node 1 and it knows it should turn on at 12:00:00.04. Similarly node 2 (being 2 cells away from the reference cell) turns on $2 \times \frac{T_a}{2}$ seconds later (at 12:00:00.05). This process is executed at each node, so that node 10 calculates that it should become active at 12:00.13 to form part of the east ripple. In other words, once a node establishes its location it is able to determine its place within the larger network structure and starts to behave accordingly.

Rather than using geographic location to initially assign a unique set of rendezvous phases for each node in the network, there is also the option of seeding the network from some point in the network. The seed node would have a set of four rendezvous phases which it would broadcast to all of its neighbors. The neighboring nodes would then modify their own set of rendezvous phases based on their relative location to the seed node and then broadcast their set of rendezvous phases to their neighbors which would in turn modify their own set of rendezvous phases based on their relative position and then broadcast them to their neighbors. This way the network would self organize from a central seed point and require the use of relative positioning rather than global positioning information. This method would however mean that the network takes a lot longer to self-organize before it can be utilized to perform its role (that of reporting sensed information to
nodes that request it). It also provides a central point of failure in the self organization of the network. For these reasons this method of self organizing has not been significantly explored.

4.6 Summary

To summarize the capability of nodes coordinating their active times under the Ripple Rendezvous scheme and operating with the R-MAC protocol, we find that all the design requirements outlined at the end of the last Chapter in section 3.8 are met. Once again these are for a system that:

1. incorporates the energy efficient mechanisms of S-MAC (message passing and overhearing avoidance). These are able to be fully implemented by R-MAC,

2. is adaptive to the level of traffic within the network as the T-MAC protocol is (the energy expenditure of each node is proportional to the level of traffic within the network). This holds up to a maximum level of traffic within the network. This issue is further clarified in the next chapter.

3. allows packets to be delivered in a timely manner and although they are not delivered as quickly as possible they are still delivered relatively quickly. Just how quickly is revealed in the next chapter.

4. is independent of the routing layer that traditionally sits above the MAC layer. All routes are possible under the wave front Ripple rendezvous scheme although some suboptimal length routes may experience a slightly increased latency as the route of packet travels within different waves.
Chapter 5

Comparison

Having described the operation of both varieties of the S-MAC protocol (the basic version of 2001 and the enhancements of 2003) as well as the T-MAC protocol, in this chapter they are compared with ripple rendezvous as well as an idealized protocol that defines the limits of performance. The chapter begins with a definition of the how the protocols will be compared and the basis for the parameters chosen. Analytical models are then constructed by which the protocols can be compared. Finally, all protocols are compared using a range of parameters, from section one and the mathematical models derived in section two.

5.1 Definition of performance metrics, parameters and scenario

In this section the metrics that will be used to compare the S-MAC and the T-MAC protocols with the Ripple Rendezvous rendezvous scheme are improved. The parameters of physical, MAC/Link and network layers are then defined. Finally we define the operational scenarios to be used for comparison.

5.1.1 Metrics of performance

The first metric that is considered is the average energy consumption per node in an empty network. This is the amount of energy consumed by each node in a network that is standing completely idle. This is an important metric since a sensor network may stand unutilized for large periods of time...
between sensed activity.

When the sensor network is used, the average energy consumption per node per packet gives an indication of how energy efficient the protocol is. Only nodes on the route are considered rather than every node in the network, since most of the nodes in the network will not form part of the data route or even be neighbors of the nodes that do.

It is the nodes along the data route that are most significantly affected since they must bear the full energy cost of firstly receiving the packet from their upstream neighbor and then transmitting it to the downstream neighbor as dictated by the network layer routing protocol. The one hop (immediate) neighbors are only slightly affected since it is assumed that all protocols use the overhearing avoidance mechanism of S-MAC\(^1\) and as soon as they receive either the short RTS or CTS control frames, they enter sleep mode until the expected completion time of the link as previously described. Nodes that are outside the range of the nodes involved in delivering the packet are unaffected since to them the network appears to be empty.

With the overhearing avoidance mechanism, neighboring nodes of a packet transfer do not experience a significant increase in their energy consumption since they sleep until the packet has been transferred. If energy consumption were to be averaged across all nodes in the network, the increased energy consumption in nodes that do form the DATA route would be lost among the many nodes that do not.

The problem of timely delivery of information through a sensor network has been discussed many times in this thesis and it is appropriate that a metric that takes this into consideration be used. The time of delivery metric is introduced. This metric starts from the time at which the MAC layer at the source node receives the packet from the network layer above it and ends when the sink node finishes receiving the packet.

\(^1\)this is a valid and worthwhile energy saving technique and also this assumption helps to normalize the comparison of the protocols.
What is also needed is a metric for comparing different systems that takes the conflicting requirements of energy consumption and packet latency into account. Lindsey et. al. [31] propose such a metric for data collection in wireless sensor networks. The metric proposed is the multiple of the average delay in delivering a message and the average energy consumption of each node.

The final metric that we shall consider is that of throughput of information within the sensor network. This reflects the maximum amount of information that can be transported or ’carried’ through the sensor network.

5.1.2 Parameters of comparison

In this section we continue to form the model that is used to compare the link layer/MAC protocols by determining the ranges of the parameters used in the model. These parameters and an explanation of them are given in table 5.1.

We firstly examine the possible values for the power consumption of the radio for the four possible modes of operation (sleep, idle, receive and transmit).

For the purposes of comparison, the parameters for two different radio modules are used. The first is the RFM TR1001 radio module that is used for the European wireless sensor network EYES project [16] [54]. This radio has a bandwidth of 115.2 kbps and a sleep mode that consumes 15 microWatts of power [45]. The relevant parameters are presented in table 5.2.

The second radio module that is used as the physical layer in the analytical comparison of MAC protocols is that of the WaveLAN radio module as measured in [10]. Similar to the parameters of the TR1001 module presented in table 5.2, the parameters for the WaveLAN module are presented in table 5.3.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_s$</td>
<td>This is the power consumed by the radio while it is in sleep mode.</td>
</tr>
<tr>
<td>$P_i$</td>
<td>This is the power consumed by the radio while it is idle. It is capable of receiving information but it is not doing so.</td>
</tr>
<tr>
<td>$P_r$</td>
<td>This is the power consumed by the radio while it is receiving information.</td>
</tr>
<tr>
<td>$P_t$</td>
<td>This is the power consumed by the radio while it is transmitting information.</td>
</tr>
<tr>
<td>$T_{bo}$</td>
<td>The is the average time that a node spends backing off before commencing transmission of the frame sequence involved in sending a network layer packet to some destination. This is set to the average time taken from when a node begins to contend for the medium (by backing off) and finishes when the node starts to transmit the RTS control frame.</td>
</tr>
<tr>
<td>$T_{pkt}$</td>
<td>This is the time taken by a node to transmit the entire frame sequence involved in sending a complete network layer packet. It is defined as the time from when a node begins sending the RTS control packet and ends when the node finishes receiving the ACK control frame from the receiving node.</td>
</tr>
<tr>
<td>$T_i$</td>
<td>This is the length of time that the idle listening timer is set to. If a node has not sensed any network activity (including its own) for this period of time it will shut down to avoid any further idle listening.</td>
</tr>
<tr>
<td>$T_{rp}$</td>
<td>This is the time between successive turn on events. This figure determines how frequently a node in the network becomes active in a given period of time.</td>
</tr>
<tr>
<td>$T_a$</td>
<td>This represents the fixed amount of time that nodes in an S-MAC network are active. This also represents the maximum amount of time that nodes are active for under a ripple rendezvous scheme.</td>
</tr>
<tr>
<td>$T_{sp}$</td>
<td>This represents the short period that nodes utilizing the 'adaptive listening' feature of S-MAC 2003 become active for after sensing some network activity during their reduced fixed duty cycle.</td>
</tr>
<tr>
<td>$N$</td>
<td>This is the number of hops in the route from source node to sink node.</td>
</tr>
</tbody>
</table>

Table 5.1: Parameters used in Performance models
**Comparison**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth</td>
<td>115.2 kbps</td>
</tr>
<tr>
<td>Sleep mode</td>
<td>15 µW</td>
</tr>
<tr>
<td>Idle mode</td>
<td>14.4 mW</td>
</tr>
<tr>
<td>Receive mode</td>
<td>14.4 mW</td>
</tr>
<tr>
<td>Transmit mode</td>
<td>36 mW</td>
</tr>
</tbody>
</table>

Table 5.2: Parameters of RFM TR1001 Radio Module

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth</td>
<td>11 Mbps</td>
</tr>
<tr>
<td>Sleep mode</td>
<td>47 mW</td>
</tr>
<tr>
<td>Idle mode</td>
<td>739 mW</td>
</tr>
<tr>
<td>Receive mode</td>
<td>900 mW</td>
</tr>
<tr>
<td>Transmit mode</td>
<td>1.346 W</td>
</tr>
</tbody>
</table>

Table 5.3: Parameters of Lucent WaveLAN PC card Radio Module

From the figures presented in tables 5.2 and 5.3, it is possible to determine energy required per bit transmitted and received. This is the sum of the power of receive and transmit mode divided by the bandwidth of that particular radio module. Using this metric, the TR1001 radio consumes 0.4375 µJ/bit whereas the WaveLAN radio costs less than half at 0.2042 µJ/bit. The point being made is that while sending and receiving information, the WaveLAN radio is better suited for wireless sensor network since it consumes less energy per bit. The challenge then is to reduce the amount of time spent in idle mode.

The number of hops that we look at in our comparison is partly determined by the distance that a packet is needed to be sent. It is conceivable that a packet may be required to be transmitted over distances up to the order of kilometers. Another factor determining the required number of hops is the range of the radios used by the sensor nodes. For the low bandwidth EYES radio module, the range is between 15 and 30 metres [15]. This is limited when compared to the range of higher bandwidth 802.11 NICs which can be 50, 100 or even 250 meters. Analytical results for routes that have up to 50 hops in them are presented.
The size of the packet that is delivered from the network layer to the MAC layer largely determines the time it takes to be transmitted. Packet size is determined by the application as well as the overhead added by the application, transport and routing layers. The amount of information in a packet is assumed to be 300 bytes as in the S-MAC 2001 protocol evaluation [63].

Another factor that influences the time taken to transmit a packet is the frame sequence and control overhead that is added by the MAC layer. In this comparison, again the parameters used by S-MAC 2001 in [63] are used. Control frames (RTS, CTS and ACK) each have a 6 byte header and 2 byte cyclic redundancy check (CRC) making them 8 bytes each. The packet is fragmented and each fragment encapsulated into a DATA frame. Each data frame has a 6 byte header, a 30 byte payload and a 2 byte CRC making each data frame 38 bytes long. In the transmission of a 300 byte packet, it is fragmented into ten sections requiring ten data frames plus ten acknowledgements plus the initial RTS-CTS control frame exchange. Thus a 300 byte packet requires a frame sequence of 476 bytes. The times taken by the distributed inter frame space (DIFS) and short inter frame space (SIFS) are neglected.

The average time spent backing off is determined by the size of the contention window which for both T-MAC and R-MAC is fixed and we assume the same for S-MAC. This is given in [28] as 9.15 milliseconds. The size of the contention window is determined by the expected level of traffic within the network. For a network in which there is expected to be a large amount of traffic, the contention window need to be set to a larger value to avoid two nodes choosing the same value. For a network in which two nodes rarely contend for the medium the contention window can be made smaller. This holds for the simple backoff algorithm used by the T-MAC protocol, i.e. a fixed contention window with a random selection with uniform probability
from within this range. The length of time taken by the contention window is assumed to scale with the bandwidth of the radio that is used by the physical layer.

Research has developed a superior backoff algorithm that reduces the amount of time that nodes spend backing off and thereby increasing the throughput of the network [27]. We shall see that it is the size of the contention window that is largely responsible for determining the energy consumption of both the T-MAC and ripple rendezvous protocols.

5.1.3 Comparison Scenario

To standardize the comparison between the link layer protocols we need to standardize the topology of the network, so rather than using a network that has been organized by the GAF\(^2\) algorithm, we use a network that has a regular grid layout. The reason for this is that much of the performance of the protocols with a GAF-like topology is determined by exactly where the node lies within the individual grid-cell. Thus, to simplify the comparison between the protocols the variability in the topology is simplified to a regular grid network.

Having revealed that the remaining issue is with the delivery of a packet to some distant destination along a long-haul multihop route, we examine how the different protocols perform under this traffic pattern. We shall look at the average delivery of a packet from one node to some other distant node that is \(N\) hops away and examine the performance of the protocols as \(N\) is varied. The relative performance of the protocols thus hold only under this traffic scenario.

\(^2\)see sub-section 4.4.1
5.2 Performance Models

We now present models of the performance metrics defined in subsection 5.1.1.

When inspecting the delivery of a packet from the node that gathers the aggregated packet to the sink node that receives and processes the packet, only the relay nodes involved are considered. In other words, the energy consumption of the sensing and sink nodes do not contribute to our metric of average energy per node. This is a reasonable assumption since the average energy consumed per node including the sensing and sink nodes approaches that of the average energy consumed per node excluding the sensing and sink nodes as the length of the data route (the number of hops $N$) increases. Again the focus on large scale networks is re-stated in which long multi hop routes are used.

5.2.1 Performance Limits

When examining the performance of a sensor network there are some limits that just cannot be traversed. There is always going to be some energy cost of transmitting a packet through a network and this packet is always going to take some time, so what are they? Here we formulate the performance metrics of an ideal MAC protocol that uses the RTS-CTS-DATA-ACK frame dialogue that all three of the protocols that are being compared do.

The minimum energy consumed per rendezvous period is described by the power consumed while in sleep mode multiplied by the period in question. Thus the equation that describes the minimum ideal energy expenditure in an empty network per rendezvous period is described by equation 5.1.

\[\text{minimum energy} = \text{power in sleep mode} \times \text{period}\]

\[\text{even though ideally, nodes do not become periodically active every rendezvous period, but rather rely on their idealized 'psychic' abilities to turn on as soon as they are needed to receive a packet, that is, the ideal case assumes that in some way, nodes turn on precisely when needed. The energy consumption per rendezvous period is determined for comparison with the other protocols.}\]
Comparison

\[ E_c = P_s T_{rp} \] (5.1)

Multihop communication reduces the speed at which information can travel through a single channel communication network\(^4\), even one that is active all of the time and can respond to a packet as soon as a node’s MAC layer receives one. This is because each packet must be received from the previous device\(^5\) and then transmitted to the next node\(^6\) and the minimum time that a packet will take to be sent from a sensing node to a sink node is dependent on the time required for a node to receive and then send the packet multiplied by the number of hops along the data route. If we denote the time required to transmit one packet as \(T_{pkt}\), then for a data route of \(N\) hops, the time of delivery \((T_{oD})\) will be

\[ T_{oD} = N.T_{pkt} \] (5.2)

The minimum energy expended by each node along the data route during the delivery of the packet is the energy expended to receive and then transmit the information and can be described using the power consumption while in receive mode \(P_r\) and transmit mode \(P_t\) as well as the energy consumed in the remaining time while it is in sleep mode \(P_s\).

\[ E_d = T_{pkt}\cdot(P_r + P_t) + (N.T_{pkt} - 2.T_{pkt})\cdot P_s \] (5.3)

There is also a maximum to the amount of information that can travel through a multihop wireless network given that packets cannot travel through the network too close in both space and time for fear of colliding with each other. We restrict our discussion of throughput to the amount of information that can travel along a single route through a network with

\(^4\)being a single channel network the nodes are connected by a half-duplex link and a device cannot send information as it receives the information

\(^5\)unless it is the source node

\(^6\)unless it is the sink node
the requirement that no other route comes within transmission range of nodes involved in the route. The reason for this is that if throughput along multiple routes were to be considered, then the throughput would depend somewhat on the routes themselves within the network being considered.

The situation presented in figure 5.1 assumes that node A has many buffered packets to be sent to node H. Node A sends packet 1 to node B and must then wait until packet 1 has moved away from the local radio space. Node A must wait while node B sends the packet to node C and must also wait again as node C sends the packet to node D since node B’s transmissions in response to a second packet from node A would be received by node C causing a collision. Node A can only begin to send packet 2 to node B once node C has finished sending packet 1 to node D. This process continues through the network and through time as more packets are sent. By giving a packet the chance to get clear of the local radio space, and evenly spacing...
the packets along the route, the maximum density of concurrent traffic in the network is attained. Since the maximum number of nodes along the route are utilized in relaying information through the network, this situation gives the maximum possible throughput along a particular route through the network. After the initial latency, as described by equation 5.2, nodes begin to receive a packet every $3T_{pkt}$ seconds. Thus the maximum number of packets that can arrive per rendezvous period ($ppT_{rp}$) can be described by equation 5.4.

$$ppT_{rp} = \frac{T_{rp}}{3T_{pkt}}$$

Equations 5.2, 5.3 and 5.4 are fundamental properties of multihop networks and provide benchmarks against which we can compare the different protocols.

5.2.2 Performance of S-MAC

The S-MAC 2001 protocol has nodes adopting a fixed duty cycle, that is, the duty cycle is specified prior to the nodes being deployed in the field. This means that every rendezvous period, nodes become active for a predetermined and fixed period of time before returning to sleep mode. The energy consumed per node per rendezvous period in an empty network is thus easily expressed as equation 5.5;

$$E_e = T_a.P_i + (T_{rp} - T_a).P_s$$

Figure 5.2 represents the behavior of a series of nodes along a data route under the S-MAC 2001 protocol. The network has been tuned to allow a packet to be transmitted three hops per rendezvous period. Equations are presented that are valid for any tuning of the network, in other words, for any value of the parameters $T_a$ and $T_{rp}$.

The parameters of the model determine how many complete rendezvous periods are required as well as the length of the remaining partial
Comparison

Figure 5.2: Behavior of S-MAC 2001

rendezvous period to deliver a packet from source to sink along an $N$ hop route. The average number of hops that are possible during the active time ($T_a$) of an S-MAC 2001 network is described by equation 5.6

$$N_{T_a} = \frac{T_a}{(T_{bo} + T_{pkt})} \tag{5.6}$$

The number of complete rendezvous periods ($i$) is calculated using the value $N_{T_a}$ where the symbol $\uparrow$ is used to signify rounding up to the nearest integer and similarly the symbol $\downarrow$ indicates rounding down to the nearest integer.

$$i = \frac{N}{N_{T_a}} \downarrow = \frac{N(T_{bo} + T_{pkt})}{T_a} \downarrow \tag{5.7}$$

The integer value $i$ represents the average number of full rendezvous periods taken for the packet to complete most of its journey. The number of remaining hops is given by $N - iN_{T_a}$. Thus the total time for a packet to travel from source to sink node is the initial sleep delay which we assume
to be on average approximately half the rendezvous period \( \left( \frac{T_{rp}}{2} \right) \) plus the delay of \( i \) rendezvous periods \( (iT_{rp}) \) plus the delay of the remaining few hops \( ((N - iN_{Ta})(T_{bo} + T_{pkt})) \). Putting all of these together results in equation 5.8.

\[
T_{uD} = iT_{rp} + (N - iN_{Ta})(T_{bo} + T_{pkt}) + \frac{T_{rp}}{2} \quad (5.8)
\]

The energy consumed by each relay node along the data route from when the MAC layer at the source node receives the packet until the MAC layer at the sink node passes the packet to the network layer is the sum of the energy required to transmit the packet \( (T_{pkt}.(P_t + P_r)) \) plus the energy consumed by each node while in sleep mode during overhearing avoidance \( (2.T_{pkt}.P_s) \). For the remainder of their active time, nodes are idle. The time spent in idle mode is calculated by working out the total amount of time a node is active during the time described by equation 5.8 and then subtracting times we know that it cannot be idle. This means that a node is active for \( iT_A + (N - iN_{Ta})(T_{bo} + T_{pkt}) \) seconds. We must subtract \( 2T_{pkt} \) seconds to exclude the two times it is in sleep mode for overhearing avoidance and another \( 2T_{pkt} \) seconds when it is in receive and then transmit mode while each node actually handles the packet. The amount of time spent sleeping by each node is on average \( \frac{T_{rp}}{2} \) seconds during the initial sleep delay, plus all of the complete rendezvous periods is \( i(T_{rp} - T_A) \) seconds as well as twice while engaged in overhearing avoidance \( (2T_{rp}) \) seconds.

Putting all of these together results in equation 5.9 describing the average energy consumed per node along the delivery route.

\textsuperscript{7}The first and last relay nodes do not engage in overhearing avoidance twice, but for the simplicity of the equations it is assumed that they do. This improves the performance of S-MAC slightly because the first and last relay nodes actually spend \( T_{pkt} \) seconds in idle mode rather than sleep mode as the other relay nodes do.
Figure 5.3: Maximum throughput of S-MAC 2001

\[ E_d = T_{pkt} \cdot [P_r + P_t] + [iT_a + (N - i.N_Ta) \cdot (T_{bo} + T_{pkt}) - 4T_{pkt}] \cdot P_t + \ldots \]
\[ \ldots \left( \frac{T_{rp}}{2} + 2T_{pkt} + i.(T_{rp} - T_a) \right) \cdot P_s \]

Equation (5.9)

Because this equation models the packet approaching and then moving away from the nodes, it holds for all \( N \geq 4 \).

The maximum throughput of the S-MAC 2001 protocol is determined by the fixed active time that the network is tuned to. For illustration purposes, again a network that has been tuned to allow three consecutive hops per active period is shown in figure 5.3.
In calculating the maximum throughput of S-MAC 2001 it is assumed that packets are transmitted through the network at maximum density, similar to that of the ideal protocol except that now the time spent contending for the medium must also be taken into account. This assumption is made to determine an upper bound of the throughput i.e. the maximum throughput that S-MAC 2001 is capable of producing given that on average nodes will spend $T_{bo}$ seconds contending for the medium each time a packet is transmitted.

With this assumption, the maximum throughput is similar to that of the throughput for the ideal protocol, but also taking into account the average time spend contending for the medium ($T_{bo}$) and the reduced level of activity of nodes. If each packet is to be transmitted on average after $T_{bo} + T_{pkt}$ seconds each hop, and the network is active for $T_a$ seconds every rendezvous period, then assuming maximum traffic density in the network while it is active, the throughput per rendezvous period can be expressed as equation 5.10.

$$ppT_{rp} = \frac{T_a}{3(T_{bo} + T_{pkt})}$$  \hspace{1cm} (5.10)

In reality, the throughput will be less than this due to colliding frames and randomly selected nodes\textsuperscript{8} gaining access to the medium resulting in a less than maximum density of traffic moving through the network. However this does allow us to analytically determine an upper bound.

As already stated, the modifications to the S-MAC protocol in 2003 mean that the active time of each node is reduced to the time required to transmit one packet of information. This means that the active time will be at least the maximum contention window plus the time required to transmit a packet of information. The adaptive listening feature means that nodes reawaken from overhearing avoidance (following the completion of a neigh-

\textsuperscript{8}randomly selected based on the random number selected from within the contention window
bouring transmission) for a short period of time. This short period of time is assumed to be the maximum contention window plus the time required for an RTS control frame to be transmitted. The S-MAC 2003 protocol is difficult to model as for some of the time it is behaving as the S-MAC 2001 protocol, but at other times (when it is engaged in adaptive listening) it behaves more like the T-MAC protocol. A diagrammatic representation of the behaviour of an S-MAC 2003 network is given in figure 5.4.

The equation that describes the empty network performance of S-MAC 2003 is the equation 5.5 with an active time ($T_a$) that has been reduced to the time required to transmit one packet. It is evident how reducing the active time to that required to transmit one packet reduces the amount of idle listening compared to S-MAC 2001 which can allow multiple packets to be transmitted in one active period depending on how it is tuned prior to deployment.

It is assumed that the packet commences its journey from source to sink during the regularly scheduled active time. With this assumption, the behaviour of nodes along the data path can be divided into two groups - nodes that receive a packet during the regularly scheduled active time (odd numbered relay nodes along the route) and nodes that receive the packet during their adaptive listening behaviour period (even numbered relay nodes).

Referring to figure 5.4 the behaviour of the odd numbered relay nodes while they handle a data packet is as follows. Each node becomes active as regularly scheduled and are idle until contacted with a request to receive a packet. After receiving the packet, each node is idle while backing off prior to requesting the next neighbouring node on the route to receive the packet - after transmitting the packet, the node sleeps. In the next rendezvous period, the node reawakens and is idle until it enters sleep mode to avoid overhearing the next neighbours transmission. After this, the adaptive listening algorithm requires the node to become active again for a
Figure 5.4: Behavior of S-MAC 2003
short period (denoted as $T_{sp}$ in figure 5.4) after which the node sleeps until the next rendezvous period. While dealing with with the packet, the energy expended by the nodes are: the energy required to receive and then send the packet ($T_{pkt}(P_r + P_t)$); each node is idle for an average of $3T_{bo} + T_{sp}$ seconds; each node sleeps for approximately $T_{rp} - T_a$ seconds per rendezvous period. For the rest of their active time during the the transmission from source to sink, the nodes are idle. To calculate this we take $\frac{N}{2}$ rounded up to the nearest integer, subtract two for the two active periods that each node is dealing with the packet giving a figure of $(\frac{N}{2} - 2)T_a$ seconds. This makes these equation valid for route lengths in which $N \geq 4$.

The behaviour of the even numbered relay nodes along the data path while dealing with a packet is as follows. Each node becomes active during its regularly scheduled active time and is idle until it sleeps to avoid overhearing the inbound packet. After this, the idle listening algorithm means that they reawaken and become active for a short period (a maximum of $T_{sp}$ seconds) however after an average of $T_{bo}$ seconds they are contacted by their upstream neighbours with a request to receive the packet.\(^9\) When they have finished receiving the packet, they enter sleep mode until the next rendezvous period when they reawaken and immediately start backing off and are thus idle for an average of $T_{bo}$ seconds until they initiate a link with their downstream neighbour during which the packet is transmitted. After transmitting the packet, the nodes are idle for an average of $T_{bo}$ seconds until they sleep to avoid overhearing. While they are sleeping to avoid overhearing, their active time expires and they do not reawake until the next rendezvous period. For the rest of the time during the the transmission from source to sink, the nodes are idle while they are active.

The energy expended by each even numbered node is almost identical to that of the odd numbered nodes except that each node is idle for $4T_{bo}$

\(^9\)this illustrates the adaptive listening mechanism in action
Comparison

seconds rather than $3T_{bo} + T_{sp}$ seconds. $T_{sp}$ is close to $2T_{bo}$ since the short period is the maximum contention window (which we assume to be fixed as in the T-MAC protocol) plus the time taken for a RTS transmission. If we neglect the time taken to transmit the small control frame and consider that $T_{bo}$ is the average of random selections from within the range $[0, CW - 1]$, we can see how $T_{sp} \approx 2T_{bo}$.

With this slight approximation we can derive an expression for the average energy consumption of each relay node along the data route during the transmission of a packet from source node to sink node as the energy required while transmit the packet for $T_{pkt}$ seconds plus the energy consumed while standing idle for $4.5T_{bo} + \left( \frac{N}{2} \uparrow - 2 \right) T_A$ seconds. Finally, each node sleeps for an average of $\frac{T_{rp}}{2}$ seconds during the initial sleep delay, plus $(\frac{N-1}{2}) \downarrow (T_{rp} - T_a) + T_{pkt}$ seconds during the packet journey from source to sink.

Putting all of these times together with the power consumption in each mode give us equation 5.11.

$$E_d = T_{pkt} \cdot (P_r + P_t) + \left[ 4.5T_{bo} + \left( \frac{N}{2} \uparrow - 2 \right) T_A \right] \cdot P_i + \ldots$$

$$\ldots \left[ \frac{N-1}{2} \downarrow (T_{rp} - T_a) + T_{pkt} + \frac{T_{rp}}{2} \right] P_s$$

(5.11)

This equation holds for all $N \geq 4$

The equation that described the time of delivery of a packet along an $N$-hop route is determined by the initial sleep delay of $\frac{T_{rp}}{2}$ seconds plus the number of rendezvous periods taken which is defined as $\frac{N-1}{2} \downarrow . T_{RP}$ seconds plus the time taken in the last one or two hops depending on whether $N$ is odd or even. Thus the time of delivery is given by equation 5.12

$$T_{oD} = \frac{T_{rp}}{2} + \frac{N-1}{2} \downarrow . T_{rp} + T_{bo} + T_{pkt} \quad \text{for all } N \mod 2 = 1$$

$$= \frac{T_{rp}}{2} + \frac{N-1}{2} \downarrow . T_{rp} + 2(T_{bo} + T_{pkt}) \quad \text{for all } N \mod 2 = 0$$

(5.12)

To determine an upper bound for the throughput along a route the
Figure 5.5: Throughput of S-MAC 2003
same assumption of maximum traffic density is made as for S-MAC 2001. The situation presented in figure 5.5 depicts the fact that since only two hops are possible per rendezvous period under S-MAC 2003 and since a packet needs to have travelled three hops before the next one can be transmitted (to achieve the maximum traffic density), we find that two packets are delivered every three rendezvous periods.

Another assumption is made in favor of S-MAC 2003, in that figure 5.5 assumes that in the second rendezvous period, node A will be able to successfully transmit packet 2 to node B rather than entering sleep mode even though it has not heard the transmission of packet 1 from node C to node D. From this the maximum throughput of S-MAC 2003 is given by equation 5.13.

\[
ppT_{rp} = \frac{2}{3} \tag{5.13}
\]

### 5.2.3 Performance of T-MAC

We now produce some equations that can be used to model the performance of the T-MAC protocol. Again these equations are used to model the behavior of all nodes in an empty network and the relay nodes along a particular data route that are carrying one network-layer packet.

Rather than referring back to Figure 3.9, Figure 5.6 presents an expanded and annotated figure to use for this chapter. T-MAC has nodes waking every \(T_{rp}\) seconds and if they do not detect any network activity for a time \(T_i\), they return to sleep mode until their next scheduled active time. Thus it is evident that in an empty network, each node consumes energy as described by equation 5.14.

\[
E_c = T_i\cdot P_i + (T_{rp} - T_i)\cdot P_s \tag{5.14}
\]

Nodes adaptively modify their active times in the presence of network-layer traffic that must be delivered. This effectively ensures that a node’s
Figure 5.6: Behavior of T-MAC
energy expenditure is proportional to the level of traffic within the network. Each of the nodes along the data route expend energy receiving and then transmitting the packet. Thus, the energy consumed sending and receiving information per node is

$$T_{\text{pkt}}. (P_r + P_t)$$  \hspace{1cm} (5.15)

The number of times each node along the data route becomes active during the delivery of a network layer packet can be described in terms of the integer values $N^{10}$ and $3^{11}$ as well as the integer result of their division $\left(\frac{N}{3}\right)$. The number of times that each node becomes active during the delivery of a packet from source node to sink node is described by:

$$\left(\frac{N}{3}\right)^\uparrow$$  \hspace{1cm} (5.16)

Each node is idle for $T_i$ seconds every time they become active and is in idle mode for different multiples of $T_{bo}$ seconds. The first, fourth, seventh etc. relay nodes are idle for $3T_{bo}$, the second, fifth, eighth etc. are idle for $4T_{bo}$ seconds and the third, sixth, ninth etc are idle for $5T_{bo}$ seconds. On average each node is idle for $4T_{bo}$ seconds during the delivery of the packet. Combining these observations with equation 5.16 gives the time spent in idle mode during the delivery of one packet from source node to sink node:

$$\left(\frac{N}{3}\right)^\uparrow . T_i + 4T_{bo}$$  \hspace{1cm} (5.17)

Similarly, nodes engage in overhearing avoidance different numbers of times. The first, fourth etc nodes sleep to avoid overhearing just once. The second, fifth etc. nodes sleep twice and the third, sixth etc. sleep three times$^{12}$. Each node on average sleeps twice to avoid overhearing while the packet is in their local radio space. Each node is in sleep mode for an average of $\frac{T_{\text{S}}}{2}$ seconds during the initial sleep delay, plus $\frac{N}{3}$ $\uparrow$ times at $(T_{rp} - T_i)$

$^{10}$the number of hops along the data route
$^{11}$the maximum number of hops per rendezvous period under T-MAC
$^{12}$due to the FRTS control frame
seconds each time minus the times that we know it is not sleeping, that is, $2T_{pkt}$ seconds while it firstly receives and then sends the packet and for an average of $4T_{bo}$ seconds as determined for equation 5.17 during which it is idle. This results in the average amount of time spent in sleep mode being described by equation 5.18.

$$\frac{T_{rp}}{2} + \frac{N}{3} \uparrow (T_{RP} - T_i) - 2T_{pkt} - 4T_{bo}$$

Combining the equations, 5.17 and 5.18 with the power consumption of idle and sleep mode ($P_i$ and $P_s$) with equation 5.15, the average energy consumed per node is given by:

$$E_d = T_{pkt}.(P_r + P_i) + \left(\frac{N}{3} \uparrow T_i + 4T_{bo}\right).P_i + \ldots \left(\frac{T_{rp}}{2} + \frac{N}{3} \uparrow (T_{rp} - T_i) - 2T_{pkt} - 4T_{bo}\right).P_s$$

Combining equations 5.20 and 5.21 with the times taken per hop ($T_{pkt} + T_{bo}$) and per rendezvous period ($T_{rp}$) we find the time of delivery

The number of rendezvous periods taken for a packet to be delivered along an $N$ hop data route is described by the value of $i$ as

$$i = \left(\frac{N - 1}{3}\right) \downarrow$$

The number of remaining hops in the last rendezvous period (just before the packet is delivered to the sink node) is expressed as

$$N - 3i$$

Combining equations 5.20 and 5.21 with the times taken per hop ($T_{pkt} + T_{bo}$) and per rendezvous period ($T_{rp}$) we find the time of delivery
Comparison

described as:

\[ T_{oD} = \frac{T_{rp}}{2} + i.T_{rp} + (N - 3.i).(T_{bo} + T_{pkt}) \] (5.22)

We see that equation 5.22 has the time of delivery of a packet being largely dominated by \( T_{rp} \) which is much larger than both \( T_{pkt} \) and \( T_{bo} \). This conforms with the described performance of T-MAC.

The T-MAC protocol has nodes remaining active as long as they can hear ongoing local communication. Considering a situation in which node A (from figure 5.6) has an infinite supply of packets to be sent through the network along a particular route; assuming a maximum traffic density within the network it is possible for T-MAC’s adaptive duty cycle to occupy the entire rendezvous period.

T-MAC operating with the FRTS\(^{13}\) control frames operating at maximum traffic density, has every fourth node pair exchanging a packet of information while the others sleep to avoid overhearing. We can thus define the upper bound for the number of packets put through the network per rendezvous period as described by equation 5.23

\[ ppT_{rp} = \frac{T_{rp}}{4(T_{bo} + T_{pkt})} \] (5.23)

In reality, what happens is, as reported in [28], that because T-MAC uses a fixed contention window\(^{14}\), as the number of nodes contending for the medium increase, collisions between competing RTS control frames limit the throughput that can occur.

5.2.4 Performance of Ripple Rendezvous

Before we produce the mathematical model, it is useful to review the performance of Ripple Rendezvous using a situation with which we are familiar.

\(^{13}\)here we admit that collisions between FRTS control frames could very well reduce the effectiveness of the protocol to a level that makes it not worth using them.

\(^{14}\)and thus does not have a contention resolution scheme that is able to respond to fluctuations in the level of traffic
Recalling the scenario depicted in Figures 3.7 and 3.9 in which one packet is delivered along a ten node route, Figure 5.7 depicts how nodes in a Ripple Rendezvous network behave under the same traffic scenario. Purely for demonstration purposes, a ripple that has a carrying capacity of two network-layer packets is depicted.

Figure 5.7 depicts the parameters $T_{bo}$, $T_i$ and $T_{pkt}$ as well as depicting the length of time each node spends in receive and then send mode respectively ($\frac{T}{2}$). This diagram depicts what has been explained so far about the benefit of using the Ripple Rendezvous scheme that is, energy efficient operation and timely delivery of information.
A model of the performance of a sensor network that has implemented the wave front Ripple Rendezvous scheme and R-MAC is presented along the same lines as for the minimal performance limits (the ideal protocol), S-MAC 2001, S-MAC 2003 and T-MAC.

Ripple rendezvous has nodes becoming active four times per rendezvous period (once for each wave) and thus the expression for the energy consumption per node per rendezvous period for an empty network under ripple rendezvous is

\[ E_e = 4T_i.P_i + (T_{RP} - 4T_i).P_s \]  

(5.24)

On average, there will be an initial sleep delay of \( \frac{T_{rp}}{2} \) seconds and each hop then takes \( \frac{T_p}{2} \) seconds with the last hop taking \( T_{hl} + T_{pkt} \) seconds to be completed. There is also an additional delay introduced when waves give way to waves moving in the other direction and just how this delay is calculated is explained next.

Referring back to the discussion of how one ripple can give way to the other (see Figure 4.7) we here make the assumption that the left wave always gives way to the right.\(^{15}\) From the point where the waves intersect, the wave that gives way (the left wave), recommences with \( T_{rp} - 2T_a \) seconds until it must give way again. The wave takes \( \frac{T_p}{2} \) seconds to advance each hop and the next node has the right wave scheduled to occur a further \( \frac{T_p}{2} \) seconds earlier. The left wave can make \( \frac{(T_{rp} - 2T_a)}{T_a} \) hops in a row before having to give way to the incoming right wave.

There are three factors that cause the delay in giving way. The first is the delay as the right wave approaches. The second is the delay of the left wave as the right wave is active at the node where the intersection takes place and the third is the delay caused by waiting for the right wave to get clear of the local radio space so as not to potentially interfere with

\(^{15}\)the terms 'left' and 'right' waves are chosen to be consistent with the ripples in figure 4.7 and these names are kept for ease of referral
information being transmitted within the wave.

The initial delay is a minimum of $\frac{T_a}{T}$ plus the remaining fraction of an active period i.e. $\text{rem}\left[\frac{(T_{rp} - 2T_a)}{T_a}\right]T_a$ seconds. The delay during the transmission is simply $T_a$ seconds, and afterwards there needs to be a gap of $\frac{T_a}{2}$ seconds to allow the right wave to get clear.

In an $N$ hop route within a wave that does give way, there will be $\frac{N}{\text{rem}\left[\frac{(T_{rp} - 2T_a)}{T_a}\right]}$ wave intersections causing an additional delay of $2T_a + \text{rem}\left[\frac{(T_{rp} - 2T_a)}{T_a}\right]T_a$ seconds at each intersection.

Since the traffic scenario being considered is of the long haul delivery of a packet of information between arbitrary source and sink nodes, on average\(^\text{16}\) half the traffic will travel within waves that give way. We may thus halve this additional delay when considering the average time of delivery of a packet from arbitrary source node to arbitrary sink node.

The general description of the time of delivery for an $N$ hop route is given by equation 5.25.

$$T_{oD} = \frac{T_{rp}}{T} + (T_{bo} + T_{pkt}) + (N - 1)\frac{T_a}{T} \ldots$$

$$\ldots + \frac{N}{\text{rem}\left[\frac{(T_{rp} - 2T_a)}{T_a}\right]} \left(2T_a + \text{rem}\left[\frac{(T_{rp} - 2T_a)}{T_a}\right]T_a\right)$$

for all $N > 1$

(5.25)

Equation 5.25 is used to establish the number of rendezvous periods that elapse during the delivery of the packet. This number is denoted as $N_{rp}$ and is described as

$$N_{rp} = \frac{T_{oD}}{T_{rp}}$$

(5.26)

On average each node will be idle for $T_i$ seconds, four times\(^\text{17}\) per rendezvous period with the number of rendezvous periods taken to deliver the packet given by $N_{rp}$.

\(^{16}\)since the source and destination nodes are unrelated to which wave gives way to the other

\(^{17}\)once for each wave front
During the delivery of a packet from source to sink, each relay node turns on during the wave front that the packet is transmitted within\(^{18}\), and waits while the upstream node backs off for an average of \(T_{bo}\) seconds, it then receives the packet which takes \(T_{pkt}\) seconds after which there is no more traffic so the node will shut down after standing idle for \(T_i\) seconds. When it awakes in send mode, it is idle while it backs off for an average of \(T_{bo}\) seconds before establishing a link and transmitting the packet which takes \(T_{pkt}\) seconds. A node will turn off as soon as it has finished sending the packet since it knows that there are no more packets to be sent. Each node will also spend \(T_i\) seconds in idle mode during each of the other 3 waves in which there is no traffic in this comparison scenario. In the subsequent rendezvous periods\(^{19}\) nodes will turn on stand idle four times per rendezvous period for \(T_i\) seconds each time. Each node is in idle mode during the delivery of the packet for a total of \(2T_{bo}\) seconds plus four times the number of rendezvous periods as specified by equation 5.26 plus for an average of another \(2T_i\) seconds during the initial sleep delay in which the node becomes active for two other waves that do not carry information in the direction of the sink in this comparison scenario.

The time spent in sleep mode is the number of rendezvous periods multiplied by the time taken by each rendezvous period \((N_{rp}.T_{RP})\) plus the initial sleep delay \((\frac{T_{rp}}{2})\) minus the average of twice that the node becomes active for other waves that do not carry information towards the sink during the initial sleep delay, minus the time spent receiving and transmitting the packet\((2T_{pkt})\) and minus the time spent in idle mode \((2T_{bo} + 4N_{rp}.T_i)\). The energy consumed by each node in relaying a packet from source to sink is given by equation 5.27.

\(^{18}\)that is, the wave that is moving in the general direction of the sink

\(^{19}\)when the downstream neighbors are still be carrying the packet to its destination
\[ E_d = T_{pkt}(P_r + P_t) + (2T_i + 2T_{bo} + 4T_i . N_{rp})P_i \ldots \]
\[ \ldots + \left( \frac{T_{rp}}{2} - 2T_i - 2(T_{bo} + T_{pkt}) + (T_{rp} - 4T_i)N_{rp} \right) . P_s \quad (5.27) \]

To describe the throughput of ripple rendezvous along a particular route, no assumptions need to be made, as has already been described, the average maximum number of packets that can travel along a particular route is the number of packets that can be transmitted in half the active time, this is described by equation 5.28.

\[ ppT_{rp} = \frac{T_a}{2(T_{bo} + T_{pkt})} \quad (5.28) \]

Having described the operation of a Ripple Rendezvous and R-MAC network in laborious detail and having derived mathematical equations that model the behavior of nodes that implement these protocols, we now use the derived mathematical equations to compare the four protocols with each other as well as with the minimum performance limits derived at the start of this section.

5.3 Comparison

5.3.1 Initial Protocol Settings

We first discuss the settings of the parameters used to compare the different protocols.

Starting with S-MAC 2001, there are three different 'flavors', with the first of these representing a fixed duty cycle that has been tuned to allow a packet to be transmitted an average of five hops in a single active period. This is represented on the graphs as S-MAC 2001-5. Similarly, S-MAC 2001-10 and S-MAC 2001-20 represent different settings of the S-MAC 2001 protocol that are tuned to allow an average of 10 and 20 hops respectively, each time they become active.
For S-MAC 2003 there is just one variety shown since its active time is set to the maximum contention interval plus the time required to transmit one packet. This protocol uses the adaptive listening feature as mentioned. Similarly, only one variety of T-MAC is used in the comparison.

There are three different settings for the ripple rendezvous protocol that represent different 'carrying capacities' of the ripples. The first one (RR-2) represents a ripple rendezvous scheme in which two packets can be carried along the same route within each wave. RR-5 and RR-10 represent waves that allow 5 and 10 packets respectively.

We initially use parameters for the TR1001 radio module as specified in table 5.2.

Initially a network layer packet is assumed to be 300 bytes and remembering from before, that fragmentation and control packet overhead means that a 300 byte network layer packet requires 476 bytes of information to be exchanged at the MAC layer.

The size of the contention window is taken from [28] which uses a 46 kbps radio to compare MAC protocols and a contention time of 9.15 ms. This contention time is then scaled to the bandwidth of the radio that is being used.

The final parameter that needs clarification is that of the rendezvous period. Initially, a rendezvous period of one second is used. Having defined these parameters, the values for the variables used that depend on these parameters in the performance models are shown in table 5.4.

The fixed active times for the three different tunings of S-MAC 2001 and S-MAC 2003 are presented in table 5.5. These are the length of time

<table>
<thead>
<tr>
<th>Model Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{pkt}$</td>
<td>33.056ms</td>
</tr>
<tr>
<td>$T_{bo}$</td>
<td>1.827ms</td>
</tr>
<tr>
<td>$T_i$</td>
<td>3.793ms</td>
</tr>
</tbody>
</table>

Table 5.4: Value of model variables
that each S-MAC variety becomes active for every rendezvous period and they need to be specified prior to the deployment of the network.

The T-MAC protocol does not have any predefined active time, although each node becomes active for at least $T_i$ seconds every time the network becomes active. As already indicated in table 5.4 this is for a time of 3.793 ms each rendezvous period.

The active times of the three varieties of the ripple rendezvous protocol are presented in table 5.6. These are the maximum times that nodes are able to be active for. The actual time that each node is active for is dependent on the level of traffic within the network. It is these active times that determine the speed with which information will be able to move through the network as well as the volume of information that can be carried.

It is evident that from table 5.6 that only RR-2 is able to operate with a rendezvous period of 1 second and even then, waves give way every four hops. This immediately illustrates a restriction of the ripple rendezvous protocol i.e., the protocol settings need to be such that there is enough time for the network to form four sets of wave fronts. Nevertheless, the results of the setting are presented in figure 5.8 for all protocols except RR-5 and

---

### Table 5.5: Fixed active times for S-MAC varieties with the TR1001 radio module

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Fixed active time</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-MAC 2001-5</td>
<td>174.4 ms</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>348.8 ms</td>
</tr>
<tr>
<td>S-MAC 2001-20</td>
<td>697.6 ms</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>36.7 ms</td>
</tr>
</tbody>
</table>

### Table 5.6: Ripple Rendezvous maximum active times with the TR1001 radio module

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Maximum Active Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>RR-2</td>
<td>146.8 ms</td>
</tr>
<tr>
<td>RR-5</td>
<td>367.1 ms</td>
</tr>
<tr>
<td>RR-10</td>
<td>734.2 ms</td>
</tr>
</tbody>
</table>

---
RR-10 which do not work with the TR1001 radio module and a rendezvous period of one second.

Results are presented in figure 5.8 with a subset of the numerical results for the various protocols and settings given in table 5.7 for route lengths of 10, 20, 30, 40 and 50 hops respectively. The average energy consumed per node involved in carrying the information through the network

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>1.670 mJ</td>
<td>1.675 mJ</td>
<td>1.680 mJ</td>
<td>1.685 mJ</td>
<td>1.690 mJ</td>
</tr>
<tr>
<td>2001-5</td>
<td>4.806 mJ</td>
<td>9.854 mJ</td>
<td>14.90 mJ</td>
<td>19.95 mJ</td>
<td>25.00 mJ</td>
</tr>
<tr>
<td>2001-10</td>
<td>4.794 mJ</td>
<td>9.826 mJ</td>
<td>14.86 mJ</td>
<td>19.89 mJ</td>
<td>24.92 mJ</td>
</tr>
<tr>
<td>2001-20</td>
<td>4.794 mJ</td>
<td>9.817 mJ</td>
<td>14.84 mJ</td>
<td>19.87 mJ</td>
<td>24.89 mJ</td>
</tr>
<tr>
<td>2003</td>
<td>3.436 mJ</td>
<td>6.151 mJ</td>
<td>8.867 mJ</td>
<td>11.58 mJ</td>
<td>14.30 mJ</td>
</tr>
<tr>
<td>T-MAC</td>
<td>2.056 mJ</td>
<td>2.265 mJ</td>
<td>2.473 mJ</td>
<td>2.751 mJ</td>
<td>2.960 mJ</td>
</tr>
<tr>
<td>RR-2</td>
<td>2.254 mJ</td>
<td>2.637 mJ</td>
<td>2.949 mJ</td>
<td>3.332 mJ</td>
<td>3.644 mJ</td>
</tr>
</tbody>
</table>

Table 5.7: Energy consumed vs. Route Length under initial protocol settings
Table 5.8: Time of Delivery vs. Route Length under initial protocol settings

for an ideal protocol does not significantly increase as the length of the route increases. This is because nodes are only active when they are required to firstly receive and then transmit the packet and for the rest of the time they are in sleep mode. The T-MAC protocol is the most energy efficient real protocol as has already been described and the ripple rendezvous protocol closely follows the T-MAC protocol in terms of energy efficiency. It is evident how the improvements in the S-MAC 2003 protocol make it more energy efficient than all varieties of the S-MAC 2001 protocol, however the improvements do not result in a protocol that is nearly as energy efficient as both the T-MAC and ripple rendezvous protocols. The three varieties of the S-MAC 2001 protocol all consume very similar amounts of energy since they are all active for the same amount of time in the delivery of a packet from source to sink with only the amounts of time spent in sleep mode varying. Since the energy consumed while in sleep mode with the TR1001 radio module is very small when compared to the energy consumed while active, they appear to consume almost identical amounts of energy. Basically, for each hop that the packet makes, the other nodes along the route are idle20.

The results of the time of delivery are also equally unsurprising and these are presented in figure 5.9 with table 5.8 again giving a subset of these results in numerical form. The ideal time of delivery is the time required to firstly receive and then transmit the packet along the route. All of the real

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>0.3306 s</td>
<td>0.6611 s</td>
<td>0.9917 s</td>
<td>1.322 s</td>
<td>1.653 s</td>
</tr>
<tr>
<td>2001-5</td>
<td>1.674 s</td>
<td>3.674 s</td>
<td>5.674 s</td>
<td>7.674 s</td>
<td>9.674 s</td>
</tr>
<tr>
<td>2001-10</td>
<td>0.8488 s</td>
<td>1.849 s</td>
<td>2.849 s</td>
<td>3.849 s</td>
<td>4.849 s</td>
</tr>
<tr>
<td>2001-20</td>
<td>0.8488 s</td>
<td>1.198 s</td>
<td>1.849 s</td>
<td>2.198 s</td>
<td>2.849 s</td>
</tr>
<tr>
<td>2003</td>
<td>4.570 s</td>
<td>9.570 s</td>
<td>14.570 s</td>
<td>19.570 s</td>
<td>24.570 s</td>
</tr>
<tr>
<td>T-MAC</td>
<td>3.535 s</td>
<td>6.570 s</td>
<td>9.605 s</td>
<td>13.53 s</td>
<td>16.57 s</td>
</tr>
<tr>
<td>RR-2</td>
<td>1.800 s</td>
<td>3.442 s</td>
<td>4.781 s</td>
<td>6.422 s</td>
<td>7.761 s</td>
</tr>
</tbody>
</table>

20except for the nodes that are sleeping to avoid overhearing
protocols suffer from an initial sleep delay which is half a rendezvous period. The protocol that delivers information the quickest is S-MAC 2001 since it can be tuned to remain active for a relatively long time. This is evident by the S-MAC 2001-20 protocol delivering a packet along a 50 hop route in just under 3 seconds caused largely by the fact that it is mostly active during its operation. In this case it is active for 698 milliseconds every second and sleeps for the remaining 302 milliseconds. S-MAC 2001-10 is the next fastest for similar reasons and the readers attention is drawn to the time of delivery for S-MAC 2001-10 which is the same as S-MAC 2001-20 up until 10 hops this is because for route length greater than 10 hops must span multiple active periods and the packet is delayed by nodes entering sleep mode. The ripple rendezvous-2 protocol is slightly faster than S-MAC 2001-5 with the T-MAC being slower still because every three hops the packet is delayed until the next rendezvous period and similarly the slowest is S-MAC 2003 in which a packet is delayed until the next rendezvous period every two hops.
It has been shown how the requirements of timeliness of delivery and energy efficiency compete with each other in that for a protocol to deliver information quickly it must sacrifice some energy efficiency and visa versa, this issue is also discussed in [39]. This has been demonstrated in the difference between the 2001 and the 2003 versions of the S-MAC protocol with the 2003 having better performance in terms of energy efficiency, yet the 2003 incarnation of S-MAC performs significantly worse in terms of the time of delivery. It is therefore useful to use a metric that takes both of these competing requirements into consideration. The \( \text{Energy} \times \text{Delay} \) metric is used for just this purpose. The results of the multiplication of the numbers presented in figures 5.8 and 5.9 are presented in figure 5.10. The performance of an ideal protocol is not included in figure 5.10 because as table 5.9 shows, the ideal protocol is four orders of magnitude better than all of the real protocols under this metric.

From the results presented in figure 5.10 as the length of the route
Comparison

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>0.4413 µJs</td>
<td>1.107 µJs</td>
<td>1.666 µJs</td>
<td>2.228 µJs</td>
<td>2.793 µJs</td>
</tr>
<tr>
<td>2001-5</td>
<td>8.047 mJs</td>
<td>36.21 mJs</td>
<td>84.56 mJs</td>
<td>153.1 mJs</td>
<td>241.8 mJs</td>
</tr>
<tr>
<td>2001-10</td>
<td>4.069 mJs</td>
<td>18.17 mJs</td>
<td>42.33 mJs</td>
<td>76.56 mJs</td>
<td>120.9 mJs</td>
</tr>
<tr>
<td>2001-20</td>
<td>4.069 mJs</td>
<td>11.76 mJs</td>
<td>27.44 mJs</td>
<td>43.66 mJs</td>
<td>70.92 mJs</td>
</tr>
<tr>
<td>2003</td>
<td>15.70 mJs</td>
<td>58.87 mJs</td>
<td>129.2 mJs</td>
<td>226.7 mJs</td>
<td>351.3 mJs</td>
</tr>
<tr>
<td>T-MAC</td>
<td>7.267 mJs</td>
<td>14.88 mJs</td>
<td>23.75 mJs</td>
<td>37.24 mJs</td>
<td>49.05 mJs</td>
</tr>
<tr>
<td>RR-2</td>
<td>4.058 mJs</td>
<td>9.076 mJs</td>
<td>14.10 mJs</td>
<td>21.40 mJs</td>
<td>28.28 mJs</td>
</tr>
</tbody>
</table>

Table 5.9: Energy x Delay vs. Route Length under initial protocol settings

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Joules</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-MAC 2001-5</td>
<td>$2.52 \times 10^{-3}$</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>$5.03 \times 10^{-3}$</td>
</tr>
<tr>
<td>S-MAC 2001-20</td>
<td>$1.01 \times 10^{-2}$</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>$5.44 \times 10^{-4}$</td>
</tr>
<tr>
<td>T-MAC</td>
<td>$6.96 \times 10^{-5}$</td>
</tr>
<tr>
<td>Ripple Rendezvous</td>
<td>$2.33 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 5.10: Energy consumption per node per rendezvous period under initial protocol settings

increases, it is the RR-2 protocol that manages to balance the competing requirements of energy efficiency and timeliness of delivery best out of the non-ideal protocols as the route length becomes large. T-MAC is performs slightly better than S-MAC 2001-20 which is in turn, better than S-MAC 2001-10 and then S-MAC 2001-5, with S-MAC 2003 having the worst performance under this metric.

The figures presented so far indicate the performance of the network while information is being delivered. This is not the typical situation for the majority of nodes in the sensor network that are not used to deliver information, they are idle since they are not involved in the transport of information from source node to the sink. Thus it is useful to examine the relative energy consumption per node of an empty network. The figures in table 5.10 represent the energy consumed per node per rendezvous period in a network that is standing unutilized.

It is here that the the penalty that S-MAC 2001 pays for being able
Comparison

<table>
<thead>
<tr>
<th>Protocol</th>
<th>$pp_{T_p}$</th>
<th>bytes/$T_p$</th>
<th>bytes/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>10.08</td>
<td>3025</td>
<td>3025</td>
</tr>
<tr>
<td>S-MAC 2001-5</td>
<td>1.667</td>
<td>500.0</td>
<td>500.0</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>3.333</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>0.667</td>
<td>200.0</td>
<td>200.0</td>
</tr>
<tr>
<td>T-MAC</td>
<td>7.167</td>
<td>2150</td>
<td>2150</td>
</tr>
<tr>
<td>RR-2</td>
<td>2</td>
<td>600.0</td>
<td>600</td>
</tr>
</tbody>
</table>

Table 5.11: Throughput under initial protocol settings

to deliver information quickly is evident. Nodes must have a relatively high energy expenditure every rendezvous period to adopt such a highly active duty cycle which allows a packet to be delivered quickly (as S-MAC 2001-20 and to lesser degrees S-MAC 2001-10 and -5 have done). T-MAC consumes the least energy per rendezvous period as is to be expected with ripple rendezvous consuming four times this amount since nodes are active four times per rendezvous period.

The throughput of the protocols are presented in table 5.11. These figures are determined from the relevant throughput equations as derived in section 5.2. The figures presented in column two are the throughput in packets per rendezvous period. The third column is the throughput in terms of bytes per rendezvous period calculated from the number of packets per rendezvous period multiplied by the size of the packets. Finally bytes per second are calculated by dividing the figures in the third column by the rendezvous period. Since a rendezvous period of one second is used initially, the last two columns of figures are the same.

From table 5.11, the reduced throughput of a distributed multihop wireless network is evident. This is due to due to restrictions on the density of concurrent traffic as well as the MAC protocol overhead. The maximum ideal throughput of 3025 bytes per second is significantly less than the bandwidth of the radio module which is 14400 bytes per second. The real protocol with the highest throughput is T-MAC with this protocol being less than
Comparison

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>1.670 mJ</td>
<td>1.675 mJ</td>
<td>1.680 mJ</td>
<td>1.685 mJ</td>
<td>1.690 mJ</td>
</tr>
<tr>
<td>2001-5</td>
<td>4.806 mJ</td>
<td>9.854 mJ</td>
<td>14.90 mJ</td>
<td>19.95 mJ</td>
<td>25.00 mJ</td>
</tr>
<tr>
<td>2001-10</td>
<td>4.794 mJ</td>
<td>9.826 mJ</td>
<td>14.86 mJ</td>
<td>19.89 mJ</td>
<td>24.92 mJ</td>
</tr>
<tr>
<td>2001-20</td>
<td>4.794 mJ</td>
<td>9.817 mJ</td>
<td>14.84 mJ</td>
<td>19.87 mJ</td>
<td>24.89 mJ</td>
</tr>
<tr>
<td>2003</td>
<td>3.436 mJ</td>
<td>6.151 mJ</td>
<td>8.867 mJ</td>
<td>11.58 mJ</td>
<td>14.30 mJ</td>
</tr>
<tr>
<td>T-MAC</td>
<td>2.056 mJ</td>
<td>2.265 mJ</td>
<td>2.473 mJ</td>
<td>2.751 mJ</td>
<td>2.960 mJ</td>
</tr>
<tr>
<td>RR-2</td>
<td>2.052 mJ</td>
<td>2.095 mJ</td>
<td>2.138 mJ</td>
<td>2.198 mJ</td>
<td>2.241 mJ</td>
</tr>
<tr>
<td>RR-5</td>
<td>2.110 mJ</td>
<td>2.263 mJ</td>
<td>2.416 mJ</td>
<td>2.569 mJ</td>
<td>2.722 mJ</td>
</tr>
<tr>
<td>RR-10</td>
<td>2.404 mJ</td>
<td>2.916 mJ</td>
<td>3.330 mJ</td>
<td>3.842 mJ</td>
<td>4.255 mJ</td>
</tr>
</tbody>
</table>

Table 5.12: Energy vs. Route Length under modified ripple rendezvous rendezvous period

the ideal throughput due to the added time spent backing off before initializing a link as well as one in four nodes able to be active due to the FRTS control frames. S-MAC 2001-20 comes close to T-MAC due to is large fixed duty cycle and the fact that traffic can be packed more densely since S-MAC 2001 does not use FRTS control frames as T-MAC does. S-MAC 2001-10 is the next best protocol. RR-2 is next, closely followed by S-MAC 2001-5 with S-MAC 2003 having by far the lowest throughput.

5.3.2 Modifying the rendezvous period of ripple rendezvous

All of the parameters are kept the same except the the rendezvous period for the ripple rendezvous protocols is changed to 5 seconds to allow the RR-5 and RR-10 to have enough time to form four sets of wave fronts. the rendezvous period for the other protocols is kept at 1 second. with a rendezvous period of five seconds, RR-2 waves give way every 32 hops, RR-5 every 11 hops and RR-10 every four hops. The effect on energy consumption is the first metric to be examined with this parameter change with the results of the change presented in figure 5.11. Again a subset of results are given numerically in table 5.12.

The first thing that needs to be explained is the different performance of the three varieties of the ripple rendezvous protocol under this
Figure 5.11: Energy vs. Route Length under modified ripple rendezvous rendezvous period

metric. This metric is the energy consumed by nodes along the route of the packet during its delivery. Since the RR-5 an RR-10 are tuned to have higher carrying capacities, the ripple advance through the network at a slower rate. This means that nodes expend energy for a greater amount of time which in turn affects their performance under this metric. It is also evident now that RR-2 performs better than T-MAC after a route length of 12 hops and RR-5 is better after 28 hops since the rendezvous period for ripple rendezvous is greater, meaning that nodes do not become active as often and thus do not expend as much energy. RR-10 performs significantly worse due to the fact that the waves advance at a much slower rate as well as the fact that every four hops the waves must give way.

Looking at the time of delivery results presented in figure 5.12 and table 5.13 we see that the performance of RR-2 actually improves for routes longer than about 30 hops, despite suffering from a greater initial sleep delay due to the longer rendezvous period. The better performance in time of
Figure 5.12: Delay vs. Route Length under modified ripple rendezvous rendezvous period

Table 5.13: Delay vs. Route Length under modified ripple rendezvous rendezvous period
delivery is caused by the decreased amount of give-way delay. We see that although RR-5 performs poorly at route lengths shorter than about 10 as the route length gets above about 35 hops it performs better than T-MAC. RR-10 is definitely the worst performing protocol, since it has been tuned to be able to carry a large amount of information and also it is only able to make four successive hops before the waves must give way and when it does give way, because the $T_a$ parameter is relatively large, each give way delay requires a significant amount of time meaning that RR-10 is not able to deliver information in a timely manner for the present parameters.

Figure 5.13 shows the Energy-Delay metric vs. route length performance of all real protocols. Again the results of the ideal protocol are not shown for this metric because as table 5.14 shows, the ideal protocol performs between four and five orders of magnitude better than the real protocols. From these results, it is evident that RR-2 begins to perform better than all non-ideal protocols once the length of the route surpasses
Comparison

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>0.4413 μJs</td>
<td>1.107 μJs</td>
<td>1.666 μJs</td>
<td>2.228 μJs</td>
<td>2.793 μJs</td>
</tr>
<tr>
<td>2001-5</td>
<td>8.047 mJs</td>
<td>36.21 mJs</td>
<td>84.56 mJs</td>
<td>153.1 mJs</td>
<td>241.8 mJs</td>
</tr>
<tr>
<td>2001-10</td>
<td>4.069 mJs</td>
<td>18.17 mJs</td>
<td>42.33 mJs</td>
<td>76.56 mJs</td>
<td>120.9 mJs</td>
</tr>
<tr>
<td>2001-20</td>
<td>4.069 mJs</td>
<td>11.76 mJs</td>
<td>27.44 mJs</td>
<td>43.66 mJs</td>
<td>70.92 mJs</td>
</tr>
<tr>
<td>2003</td>
<td>15.70 mJs</td>
<td>58.87 mJs</td>
<td>129.2 mJs</td>
<td>226.7 mJs</td>
<td>351.3 mJs</td>
</tr>
<tr>
<td>T-MAC</td>
<td>7.267 mJs</td>
<td>14.88 mJs</td>
<td>23.75 mJs</td>
<td>37.24 mJs</td>
<td>49.05 mJs</td>
</tr>
<tr>
<td>RR-2</td>
<td>6.556 mJs</td>
<td>8.232 mJs</td>
<td>9.970 mJs</td>
<td>12.51 mJs</td>
<td>14.40 mJs</td>
</tr>
<tr>
<td>RR-5</td>
<td>8.833 mJs</td>
<td>15.38 mJs</td>
<td>22.73 mJs</td>
<td>30.88 mJs</td>
<td>39.83 mJs</td>
</tr>
<tr>
<td>RR-10</td>
<td>22.15 mJs</td>
<td>52.33 mJs</td>
<td>83.20 mJs</td>
<td>129.5 mJs</td>
<td>173.4 mJs</td>
</tr>
</tbody>
</table>

Table 5.14: Energy-Delay vs. Route Length under modified ripple rendezvous rendezvous period

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Watts</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-MAC 2001-5</td>
<td>$2.52 \times 10^{-3}$</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>$5.03 \times 10^{-3}$</td>
</tr>
<tr>
<td>S-MAC 2001-20</td>
<td>$1.01 \times 10^{-2}$</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>$5.44 \times 10^{-4}$</td>
</tr>
<tr>
<td>T-MAC</td>
<td>$6.96 \times 10^{-5}$</td>
</tr>
<tr>
<td>Ripple Rendezvous</td>
<td>$5.86 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

Table 5.15: Average power consumption under modified ripple rendezvous rendezvous period

17 hops. RR-5 exceeds all other protocols for route lengths greater than 35 hops. RR-10 begins as the worst performing protocol for short routes but manages to outstrip the performance of S-MAC 2003 after 17 hops and S-MAC 2001-5 after 37 hops.

Because the ripple rendezvous protocol is using a different rendezvous period to the other protocols now, the metric of energy consumed per node per rendezvous period in an empty network is no longer a valid comparison. The solution to this problem is to look at the average power consumption. This is calculated by simply dividing the average energy consumed per node per rendezvous period, by the time taken by each rendezvous period. The results of this are presented in table 5.15.

We see from the results presented in table 5.15 that ripple rendezvous has the lowest average power consumption. This value is the same for all
Comparison

<table>
<thead>
<tr>
<th>Protocol</th>
<th>$ppT_{rp}$</th>
<th>$\text{bytes}/T_{rp}$</th>
<th>$\text{bytes/sec}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>10.08</td>
<td>3025</td>
<td>3025</td>
</tr>
<tr>
<td>S-MAC 2001-5</td>
<td>1.667</td>
<td>500.0</td>
<td>500.0</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>3.333</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>0.667</td>
<td>200.0</td>
<td>200.0</td>
</tr>
<tr>
<td>T-MAC</td>
<td>7.167</td>
<td>2150</td>
<td>2150</td>
</tr>
<tr>
<td>RR-2</td>
<td>2</td>
<td>600.0</td>
<td>120</td>
</tr>
<tr>
<td>RR-5</td>
<td>5</td>
<td>1500</td>
<td>300</td>
</tr>
<tr>
<td>RR-10</td>
<td>10</td>
<td>3000</td>
<td>600</td>
</tr>
</tbody>
</table>

Table 5.16: Throughput under modified ripple rendezvous rendezvous period setting of ripple rendezvous because all setting or ripple rendezvous become active for the same amount of time in an empty network, the difference between them is the rendezvous phase difference between nodes. The power consumption of ripple rendezvous is lower even than T-MAC since T-MAC has nodes become active once per second, and ripple rendezvous has nodes becoming active four times every five seconds with RR-2 facilitating the delivery of information significantly faster than T-MAC as the required route becomes large.

The new throughput figures for the ripple rendezvous protocols are presented in table 5.16 and here the disadvantage of increasing the rendezvous period for ripple rendezvous is noted in that the throughput in terms of the maximum number of bytes per second is significantly reduced. This is due to the fact that there are fewer waves scheduled to occur and thus less information can be carried within the waves per second.

### 5.3.3 Reducing the size of packets

Another way of ensuring that four ripple rendezvous waves can coexist within the network at the same time, is to reduce the size of the packet that the network layer exchanges between source and sink nodes. For this we assume a packet size of 30 bytes, and we retain the MAC protocols as they are, i.e. the RTS-CTS mechanism is still used to initiate a link and an acknowledge-
Comparison

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Active time</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-MAC 2001-5</td>
<td>30.66ms</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>61.32ms</td>
</tr>
<tr>
<td>S-MAC 2001-20</td>
<td>122.6ms</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>7.959ms</td>
</tr>
<tr>
<td>RR-2</td>
<td>31.84ms</td>
</tr>
<tr>
<td>RR-5</td>
<td>79.59ms</td>
</tr>
<tr>
<td>RR-10</td>
<td>159.2ms</td>
</tr>
</tbody>
</table>

Table 5.17: Active times for S-MAC and ripple rendezvous varieties with a packet size of 30 bytes

ment is still required for each DATA fragment, however only one fragment is required to transmit the packet. As such, a 30 byte packet requires 62 bytes to be exchanged in the MAC level frame dialogue. With the change in the size of this parameter, the only variable that is altered in the performance models is the value of $T_{pkr}$ which changes from a figure of 33.06ms for a packet of 300 bytes to 4.306ms for a packet of 30 bytes of information.

The change in the time required to transmit a complete network layer packet requires the re-tuning of the S-MAC and ripple rendezvous protocols. The new values of their $T_a$ variables are given in table 5.17

For the ripple rendezvous active times listed in table 5.17, RR-2 has ripples giving way to each other every 29 hops, RR-5 every 10 hops and RR-10 every 4. The rendezvous period for ripple rendezvous is reset to one second in this subsection.

The results of the average energy consumed by each node during the delivery from source node to sink node are presented in figure 5.14 and table 5.18.

The three varieties of the ripple rendezvous protocol exhibit significantly different performance levels under this metric. This is largely due to the difference in the time of delivery due to the slower moving ripples and the different number of times that waves must give way to each other. Both S-MAC and T-MAC are more energy efficient than all varieties of the ripple
Comparison

Figure 5.14: Energy vs. Route Length for a packet of 30 bytes

Table 5.18: Energy vs. Route Length for a packet of 30 bytes
Figure 5.15: Delay vs. Route Length for a packet of 30 bytes

...rendezvous protocol for route lengths less than seven hops. RR-2 becomes more energy efficient than all protocols for route lengths greater than nine hops and RR-5 exhibits superior performance than all (except RR-2) for routes greater than 15 hops. RR-10 becomes better than all of the S-MAC protocols for route lengths greater than 15 hops, however it never becomes more efficient than T-MAC, even as route lengths approach 50 hops.

Looking at the time of delivery as presented in figure 5.15 and table 5.19 we see that again it is the three varieties of the S-MAC 2001 protocol that perform best under this metric until the route length becomes such that the packet must wait until the next rendezvous period to continue its journey through the network. RR-2 becomes the best protocol in terms of time of delivery for route lengths over 20 hops at which stage S-MAC 2001-20 cannot deliver a packet in one active period and so the time of delivery must span two or more active periods separated by a time in which the
network is asleep. RR-5’s performance is located between S-MAC 2001-20 and S-MAC 2001-10 for almost all route lengths and RR-10 has a similar time of delivery to that of S-MAC 2001-5. T-MAC has the second worst time of delivery with S-MAC 2003 having the highest latency.

Combining the energy and time of delivery into the energy-delay metric, we see from figure 5.16 and table 5.20 that RR-2 performs the best of all for route lengths greater than seven hops. RR-5 becomes the second best performing for hops greater than 20 hops, with even the relatively slow RR-10 performing comparably with S-MAC 2001-10. S-MAC 2001-20 is the best performing of the comparison protocols, being located between RR-5 and RR-10 at longer route lengths. The worst performing protocol for this situation is S-MAC 2003 followed by S-MAC 2001-5 and then T-MAC.

### Table 5.19: Delay vs. Route Length for a packet of 30 bytes

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>7.488 mJs</td>
<td>18.79 mJs</td>
<td>28.26 mJs</td>
<td>37.80 mJs</td>
<td>47.38 mJs</td>
</tr>
<tr>
<td>2001-5</td>
<td>1.338 mJs</td>
<td>6.307 mJs</td>
<td>14.92 mJs</td>
<td>27.19 mJs</td>
<td>43.11 mJs</td>
</tr>
<tr>
<td>2001-10</td>
<td>0.4826 mJs</td>
<td>2.743 mJs</td>
<td>6.798 mJs</td>
<td>12.65 mJs</td>
<td>20.29 mJs</td>
</tr>
<tr>
<td>2001-20</td>
<td>0.4826 mJs</td>
<td>1.085 mJs</td>
<td>4.120 mJs</td>
<td>5.715 mJs</td>
<td>11.32 mJs</td>
</tr>
<tr>
<td>2003</td>
<td>3.368 mJs</td>
<td>13.26 mJs</td>
<td>29.62 mJs</td>
<td>52.46 mJs</td>
<td>81.78 mJs</td>
</tr>
<tr>
<td>T-MAC</td>
<td>2.131 mJs</td>
<td>5.316 mJs</td>
<td>9.757 mJs</td>
<td>17.60 mJs</td>
<td>24.97 mJs</td>
</tr>
<tr>
<td>RR-2</td>
<td>0.3490 mJs</td>
<td>0.4646 mJs</td>
<td>0.6465 mJs</td>
<td>0.7904 mJs</td>
<td>0.9462 mJs</td>
</tr>
<tr>
<td>RR-5</td>
<td>0.6410 mJs</td>
<td>1.197 mJs</td>
<td>1.899 mJs</td>
<td>2.746 mJs</td>
<td>3.739 mJs</td>
</tr>
<tr>
<td>RR-10</td>
<td>1.533 mJs</td>
<td>4.470 mJs</td>
<td>7.946 mJs</td>
<td>13.51 mJs</td>
<td>19.14 mJs</td>
</tr>
</tbody>
</table>

Table 5.20: Energy - Delay vs. Route Length for a packet of 30 bytes
Figure 5.16: Energy - Delay vs. Route Length for a packet of 30 bytes

Table 5.21: Energy consumption per node per rendezvous period for a packet of 30 bytes

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Joules</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-MAC 2001-5</td>
<td>$4.561 \times 10^{-4}$</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>$8.972 \times 10^{-4}$</td>
</tr>
<tr>
<td>S-MAC 2001-20</td>
<td>$1.779 \times 10^{-3}$</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>$1.296 \times 10^{-4}$</td>
</tr>
<tr>
<td>T-MAC</td>
<td>$6.956 \times 10^{-5}$</td>
</tr>
<tr>
<td>Ripple Rendezvous</td>
<td>$2.332 \times 10^{-4}$</td>
</tr>
</tbody>
</table>
Comparison

<table>
<thead>
<tr>
<th>Protocol</th>
<th>$pp T_{rp}$</th>
<th>bytes/$T_{rp}$</th>
<th>bytes/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>77.42</td>
<td>2323</td>
<td>2323</td>
</tr>
<tr>
<td>S-MAC 2001-5</td>
<td>1.667</td>
<td>50.00</td>
<td>50.00</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>3.333</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>S-MAC 2001-20</td>
<td>6.667</td>
<td>200.0</td>
<td>200.0</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>0.667</td>
<td>20.00</td>
<td>20.00</td>
</tr>
<tr>
<td>T-MAC</td>
<td>40.77</td>
<td>1223</td>
<td>1223</td>
</tr>
<tr>
<td>RR-2</td>
<td>2</td>
<td>60.00</td>
<td>60.00</td>
</tr>
<tr>
<td>RR-5</td>
<td>5</td>
<td>150.0</td>
<td>150.0</td>
</tr>
<tr>
<td>RR-10</td>
<td>10</td>
<td>300.0</td>
<td>300.0</td>
</tr>
</tbody>
</table>

Table 5.22: Throughput for a packet of 30 bytes

The average energy consumed per node per rendezvous period in an empty network is presented in table 5.21 and it is noted that both ripple rendezvous and T-MAC perform the same as in the original configuration since the empty network energy consumption of these protocols is determined by the size of the contention window and thus the time spent in idle mode while waiting to determine if there is going to be a packet transmission or not. The size of the packet being carried through the network has no influence on this metric. The S-MAC protocols on the other hand, experience a decrease in the empty network energy consumption since their fixed active times are reduced due to the decrease in the time required to transmit a packet and as such they do not spend as much time in idle mode as they do when the packets are larger. The S-MAC 2003 protocol now consumes less than the ripple rendezvous protocols since the time required to transmit a packet ($T_{pkt} = 4.306ms$) is now similar to the idle listening shutdown timer ($T_i = 3.793ms$) and ripple rendezvous is idle for $T_i$ seconds four times per rendezvous period whereas S-MAC 2003 is idle for $cw + T_{pkt}$ once per rendezvous period.

It is noted first in the throughput figures given in table 5.22 that there is a reduced maximum throughput in the ideal protocol. This is caused by the relative increase in control packet overhead at the MAC level for
Comparison

<table>
<thead>
<tr>
<th>Model Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{pk}$</td>
<td>346.2 µs</td>
</tr>
<tr>
<td>$T_{bo}$</td>
<td>19.13 µs</td>
</tr>
<tr>
<td>$T_{i}$</td>
<td>39.7 µs</td>
</tr>
</tbody>
</table>

Table 5.23: Value of model variables for WaveLAN radio module

information exchange. The additional amount of protocol overhead similarly reduces the throughput of T-MAC and even though the size of the packets has decreased by a factor of 10, there is only a six fold increase in the number of packets that can be sent per second. Since the S-MAC varieties and ripple rendezvous are tuned relative to the size of the packets, their throughput decreases exactly with the decrease in the size of the packet.

5.3.4 Replacement of the radio module

In addition to changing the parameters of the MAC layer (the rendezvous period) or parameters of the network layer (the size of the packet) it is also possible to make more ‘room’ within a one second rendezvous period by increasing the bandwidth of the radio transceiver that is used. To achieve this, the parameters of the TR1001 radio module are replaced with the parameters of the Lucent WaveLAN PC card that are presented in table 5.3.

The change in the parameters of the radio module has the effect of changing several of the variables in the performance models and the new values for these are presented in table 5.23 as the results of the parameters of the TR1001 module are presented in table 5.4. A packet size of 300 bytes is used.

The most significant factor in the change to the WaveLAN radio module is that now the sleep mode of the radio now consumes a significant amount of energy when compared the the three active modes. The sleep:idle:receive:transmit ratio of the four modes of operation for the TR1001 radio is 1:960:960:2400 whereas this ratio is approximately 1:15.7:19.1:28.6
Comparison

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Active time</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-MAC 2001-5</td>
<td>1.827ms</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>3.654ms</td>
</tr>
<tr>
<td>S-MAC 2001-20</td>
<td>7.307ms</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>384.4µs</td>
</tr>
<tr>
<td>RR-2</td>
<td>1.538ms</td>
</tr>
<tr>
<td>RR-5</td>
<td>3.844ms</td>
</tr>
<tr>
<td>RR-10</td>
<td>7.689ms</td>
</tr>
</tbody>
</table>

Table 5.24: Active times for S-MAC and ripple rendezvous varieties with the WaveLAN radio module for the WaveLAN card.

By changing the bandwidth of the radio module, the active times for the four varieties of the S-MAC protocol which are tuned to allow a certain number of packet transmissions in each active period are also modified. Similarly the active times of the three varieties of ripple rendezvous are also altered. The new fixed active times for the S-MAC protocol and the maximum active times of the ripple rendezvous protocol are presented in table 5.24.

With the much higher bandwidth of the WaveLAN radio module, RR-2 must give way approximately every 3000 hops, RR-5 approximately every 1200 hops and RR-10 every 597 hops when a rendezvous period of one second is used. It is also clear that the active times of S-MAC and ripple rendezvous are very much smaller than a rendezvous period of one second. However, in order to demonstrate the effects of changing the parameters of the performance models, we maintain a rendezvous period of one second for now.

The average time of delivery is the first metric that is examined since this has a large effect on the energy consumed during the delivery of a packet. We see from the results presented in figure 5.17 and table 5.25 that all varieties of the ripple rendezvous protocol have superior times of delivery than all other protocols once the route lengths become greater than what
Comparison

Figure 5.17: Delay vs. Route Length for WaveLAN radio module

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>3.462 ms</td>
<td>6.924 ms</td>
<td>10.39 ms</td>
<td>13.85 ms</td>
<td>17.31 ms</td>
</tr>
<tr>
<td>2001-5</td>
<td>1.502 s</td>
<td>3.502 s</td>
<td>5.502 s</td>
<td>7.502 s</td>
<td>9.502 s</td>
</tr>
<tr>
<td>2001-10</td>
<td>0.5037 s</td>
<td>1.504 s</td>
<td>2.504 s</td>
<td>3.504 s</td>
<td>4.504 s</td>
</tr>
<tr>
<td>2001-20</td>
<td>0.5037 s</td>
<td>0.5073 s</td>
<td>1.504 s</td>
<td>1.507 s</td>
<td>2.504 s</td>
</tr>
<tr>
<td>T-MAC</td>
<td>3.500 s</td>
<td>6.501 s</td>
<td>9.501 s</td>
<td>13.50 s</td>
<td>16.50 s</td>
</tr>
<tr>
<td>RR-2</td>
<td>0.5073 s</td>
<td>0.5150 s</td>
<td>0.5227 s</td>
<td>0.5304 s</td>
<td>0.5380 s</td>
</tr>
<tr>
<td>RR-5</td>
<td>0.5177 s</td>
<td>0.5369 s</td>
<td>0.5561 s</td>
<td>0.5753 s</td>
<td>0.5946 s</td>
</tr>
<tr>
<td>RR-10</td>
<td>0.5350 s</td>
<td>0.5734 s</td>
<td>0.6119 s</td>
<td>0.6503 s</td>
<td>0.6887 s</td>
</tr>
</tbody>
</table>

Table 5.25: Delay vs. Route Length for WaveLAN radio module
can be delivered in one active period. This causes the packet to be delayed until the next time nodes become active to continue its journey through the network. The performance of the ideal protocol is not included in figure 5.17 since as can be seen from the numerical results presented in table 5.25, the ideal protocol results in a delay in delivering information that is orders of magnitude better than all of the real protocols. It is now that the difference in the performance of the various protocols becomes most apparent. It is clear that all the protocols except the ripple rendezvous protocols are restricted in how quickly they can deliver a packet of information through the network by how many hops the packet can make in one active period before the network returns to sleep mode. S-MAC 2003 performs worst in time of delivery followed by T-MAC and then the three varieties of S-MAC 2001 in increasing order of fixed active times. It is also evident that the time of delivery of the three varieties of ripple rendezvous increases slightly as the carrying capacity of the waves increases.
Comparison

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>0.9077 mJ</td>
<td>1.070 mJ</td>
<td>1.233 mJ</td>
<td>1.396 mJ</td>
<td>1.559 mJ</td>
</tr>
<tr>
<td>2001-5</td>
<td>72.90 mJ</td>
<td>169.4 mJ</td>
<td>266.0 mJ</td>
<td>362.5 mJ</td>
<td>459.0 mJ</td>
</tr>
<tr>
<td>2001-10</td>
<td>25.99 mJ</td>
<td>75.52 mJ</td>
<td>12.50 mJ</td>
<td>174.5 mJ</td>
<td>224.1 mJ</td>
</tr>
<tr>
<td>2001-20</td>
<td>25.98 mJ</td>
<td>28.68 mJ</td>
<td>78.04 mJ</td>
<td>80.74 mJ</td>
<td>130.1 mJ</td>
</tr>
<tr>
<td>2003</td>
<td>213.1 mJ</td>
<td>449.4 mJ</td>
<td>685.8 mJ</td>
<td>922.1 mJ</td>
<td>1158 mJ</td>
</tr>
<tr>
<td>T-MAC</td>
<td>212.4 mJ</td>
<td>353.4 mJ</td>
<td>494.5 mJ</td>
<td>682.6 mJ</td>
<td>823.7 mJ</td>
</tr>
<tr>
<td>RR-2</td>
<td>48.22 mJ</td>
<td>48.58 mJ</td>
<td>48.94 mJ</td>
<td>49.31 mJ</td>
<td>49.67 mJ</td>
</tr>
<tr>
<td>RR-5</td>
<td>48.71 mJ</td>
<td>49.61 mJ</td>
<td>50.52 mJ</td>
<td>51.43 mJ</td>
<td>52.33 mJ</td>
</tr>
<tr>
<td>RR-10</td>
<td>49.52 mJ</td>
<td>51.33 mJ</td>
<td>53.15 mJ</td>
<td>54.96 mJ</td>
<td>56.77 mJ</td>
</tr>
</tbody>
</table>

Table 5.26: Energy vs. Route Length for WaveLAN radio module

Since the energy metric being used looks at the energy consumed by nodes involved in carrying a packet during the delivery of the packet, and since ripple rendezvous delivers a packet so quickly compared to the other protocols, we expect it to perform much better using this metric and indeed it does as can be seen from figure 5.18 and table 5.26. In fact the time of delivery is the main factor in the performance of the protocols under this metric. This is because time spent in sleep mode now significantly contributes to the energy consumed by nodes in the network. Another interesting result is that for the ideal protocol, the average energy consumed per node during the delivery of a packet begins to increase as the route length increases, this is also caused by the significant amount of energy consumed by the radio in sleep mode.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>2.423 nJs</td>
<td>7.411 nJs</td>
<td>12.80 nJs</td>
<td>19.32 nJs</td>
<td>26.97 nJs</td>
</tr>
<tr>
<td>2001-5</td>
<td>0.1094 Js</td>
<td>0.5933 Js</td>
<td>1.463 Js</td>
<td>2.719 Js</td>
<td>4.361 Js</td>
</tr>
<tr>
<td>2001-10</td>
<td>0.013088 Js</td>
<td>0.1135 Js</td>
<td>0.3130 Js</td>
<td>0.6116 Js</td>
<td>1.009 Js</td>
</tr>
<tr>
<td>2001-20</td>
<td>0.013088 Js</td>
<td>0.01455 Js</td>
<td>0.1173 Js</td>
<td>0.1217 Js</td>
<td>0.3257 Js</td>
</tr>
<tr>
<td>2003</td>
<td>0.9592 Js</td>
<td>4.270 Js</td>
<td>9.944 Js</td>
<td>17.98 Js</td>
<td>28.38 Js</td>
</tr>
<tr>
<td>T-MAC</td>
<td>0.7435 Js</td>
<td>2.297 Js</td>
<td>4.698 Js</td>
<td>9.216 Js</td>
<td>13.59 Js</td>
</tr>
<tr>
<td>RR-2</td>
<td>0.02446 Js</td>
<td>0.02502 Js</td>
<td>0.02558 Js</td>
<td>0.02615 Js</td>
<td>0.02672 Js</td>
</tr>
<tr>
<td>RR-5</td>
<td>0.02521 Js</td>
<td>0.02663 Js</td>
<td>0.02809 Js</td>
<td>0.02958 Js</td>
<td>0.03111 Js</td>
</tr>
<tr>
<td>RR-10</td>
<td>0.02649 Js</td>
<td>0.02943 Js</td>
<td>0.03252 Js</td>
<td>0.03574 Js</td>
<td>0.03910 Js</td>
</tr>
</tbody>
</table>

Table 5.27: Energy-Delay vs. Route Length for WaveLAN radio module
Figure 5.19: Energy-Delay vs. Route Length for WaveLAN radio module

<table>
<thead>
<tr>
<th>Protocol</th>
<th>milliJoules</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-MAC 2001-5</td>
<td>48.26</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>49.53</td>
</tr>
<tr>
<td>S-MAC 2001-20</td>
<td>52.06</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>47.28</td>
</tr>
<tr>
<td>T-MAC</td>
<td>47.03</td>
</tr>
<tr>
<td>Ripple Rendezvous</td>
<td>47.11</td>
</tr>
</tbody>
</table>

Table 5.28: Energy consumption per node per rendezvous period for WaveLAN radio module

It is observed by looking at the results of the energy-delay metric as presented in figure 5.18 and table 5.26, that the ripple rendezvous protocols do not increase significantly in the energy delay metric as the route length increases. As has been explained for the time of delivery metric, all varieties of the ripple rendezvous protocol perform better than the other protocols (except, of course, the ideal minimum) once the route length becomes such that the other protocols can no longer deliver the packet within one active period.
Comparison

<table>
<thead>
<tr>
<th>Protocol</th>
<th>$ppT_{rp}$</th>
<th>bytes/$T_{rp}$</th>
<th>bytes/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>962.3</td>
<td>288900</td>
<td>288900</td>
</tr>
<tr>
<td>S-MAC 2001-5</td>
<td>1.667</td>
<td>500.0</td>
<td>500.0</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>3.333</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>0.667</td>
<td>200.0</td>
<td>200.0</td>
</tr>
<tr>
<td>T-MAC</td>
<td>684.3</td>
<td>205300</td>
<td>205300</td>
</tr>
<tr>
<td>RR-2</td>
<td>2</td>
<td>600.0</td>
<td>600.0</td>
</tr>
<tr>
<td>RR-5</td>
<td>5</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>RR-10</td>
<td>10</td>
<td>3000</td>
<td>3000</td>
</tr>
</tbody>
</table>

Table 5.29: Throughput using the WaveLAN radio module

The other effect of a relatively high energy consumption sleep mode is to produce very similar empty network energy consumption in all of the protocols. It is evident from the figures presented in table 5.28 that the empty network energy consumption is dominated by the energy consumed while in sleep mode. From table 5.3 the power consumption of sleep mode for the waveLAN radio is 47 $mw$ and due to the increased bandwidth, nodes are not required to be active for very long (see table 5.24) when compared to the rendezvous period of one second. This explains why the energy consumption per node per rendezvous period in an empty network is dominated by the energy consumed in sleep mode.

With the vastly increased bandwidth of the WaveLAN radio it is evident from table 5.29 that the ideal maximum throughput is several orders of magnitude greater than what is allowed by the pre-tuned settings of both the S-MAC and ripple rendezvous protocols. It is however unlikely that such a large throughput will be required in a wireless sensor network.

5.3.5 Assumption of a low energy sleep mode

The next situation that is examined is one in which an assumption is made. The power consumption of the WaveLAN 802.11 radio module in sleep mode consumes significant amounts of energy. This is because for laptops and other mobile devices that use such a Network Interface Card (NIC), the
energy consumption, while being a problem, is not nearly as critically significant as it it in the field of wireless sensor networks. This is because, if a laptop computer were to exhaust its battery-supplied power then the user may very well be inconvenienced for a period of time, however this would only be until a power point was located and the battery recharged. The user would then be able to resume using their computer while its energy supply was being replenished. Alternatively, the user simply replaces the battery in their laptop. In other words, the sleep mode energy consumption of the wireless NIC is negligible when compared to the other devices on the system. The point being made is that while the power consumption of the radio in sleep mode is a problem it is not not large enough to motivate the reduction of the energy consumption of sleep mode to a level much below where it is at the moment (47 mW) since the other components of a laptop (DVD/CD drive, CPU, display screen) also have large power requirements totaling approximately 18 Watts [11].

All that is required of a radio module in sleep mode is a clock which allows the module to know when to become active. As such, we now make the assumption that a WaveLAN-like radio module that has been fully optimized specifically for a wireless sensor network could have a sleep mode in which the power consumption is like that of the TR1001 radio module. The first results of this assumption are presented in figure 5.20 and table 5.30.

It is observed that under the situation in which a high bandwidth radio module is used that has a low energy sleep mode available, that the energy consumed by all varieties of ripple rendezvous during the delivery of a packet is very similar to that of the minimum energy expenditure of the ideal protocol even as the route lengths increase to 50 hops. Once again, it is the increasing amount of time spend in idle mode that decides the performance of the protocols under this metric. T-MAC consumes more energy than ripple rendezvous (especially as the route length increases),
Figure 5.20: Energy vs. Route Length for assumed low energy sleep mode of WaveLAN radio module

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>0.7775 mJ</td>
<td>0.7777 mJ</td>
<td>0.7777 mJ</td>
<td>0.7777 mJ</td>
<td>0.7778 mJ</td>
</tr>
<tr>
<td>2001-5</td>
<td>2.477 mJ</td>
<td>5.208 mJ</td>
<td>7.939 mJ</td>
<td>10.67 mJ</td>
<td>13.40 mJ</td>
</tr>
<tr>
<td>2001-10</td>
<td>2.461 mJ</td>
<td>5.177 mJ</td>
<td>7.892 mJ</td>
<td>10.61 mJ</td>
<td>13.32 mJ</td>
</tr>
<tr>
<td>2001-20</td>
<td>2.461 mJ</td>
<td>5.161 mJ</td>
<td>7.876 mJ</td>
<td>10.57 mJ</td>
<td>13.29 mJ</td>
</tr>
<tr>
<td>2003</td>
<td>1.761 mJ</td>
<td>3.256 mJ</td>
<td>4.752 mJ</td>
<td>6.247 mJ</td>
<td>7.743 mJ</td>
</tr>
<tr>
<td>T-MAC</td>
<td>1.019 mJ</td>
<td>1.152 mJ</td>
<td>1.285 mJ</td>
<td>1.462 mJ</td>
<td>1.596 mJ</td>
</tr>
<tr>
<td>RR-2</td>
<td>0.9392 mJ</td>
<td>0.9402 mJ</td>
<td>0.9412 mJ</td>
<td>0.9422 mJ</td>
<td>0.9432 mJ</td>
</tr>
<tr>
<td>RR-5</td>
<td>0.9405 mJ</td>
<td>0.9431 mJ</td>
<td>0.9456 mJ</td>
<td>0.9482 mJ</td>
<td>0.9507 mJ</td>
</tr>
<tr>
<td>RR-10</td>
<td>0.9428 mJ</td>
<td>0.9479 mJ</td>
<td>0.9530 mJ</td>
<td>0.9581 mJ</td>
<td>0.9632 mJ</td>
</tr>
</tbody>
</table>

Table 5.30: Energy vs. Route Length for assumed low energy sleep mode of WaveLAN radio module
but significantly less than all 'flavors' of S-MAC. Of the S-MAC protocols, S-MAC 2003 consumes the least energy and again all different tunings of S-MAC 2001 (5,10 and 20) consume almost identical amounts of energy during the delivery of the packet. This is because all of the different tunings spend the same amount of time in idle mode while the packet is being delivered.

Results for the time of delivery metric for are not presented since it is the same as in the previous subsection.

The energy-delay metric presented in figure 5.21 and table 5.31, shows that ripple rendezvous becomes better than all other protocols once the route length increases above five hops and performs approximately two orders of magnitude better than the other protocols as the route length reaches 50 hops. However the ideal minimum is still between four and five orders of magnitude less than ripple rendezvous illustrating that there is still room for improvement.

Lastly, the empty network energy consumption is presented in table
Comparison

Table 5.31: Energy-Delay vs. Route Length for assumed low energy sleep mode of WaveLAN radio module

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>2.153 (nJ)s</td>
<td>5.383 (nJ)s</td>
<td>8.076 (nJ)s</td>
<td>1.076 (nJ)s</td>
<td>13.46 (nJ)s</td>
</tr>
<tr>
<td>2001-5</td>
<td>3.720 (mJ)s</td>
<td>18.23 (mJ)s</td>
<td>43.68 (mJ)s</td>
<td>80.04 (mJ)s</td>
<td>127.3 (mJ)s</td>
</tr>
<tr>
<td>2001-10</td>
<td>1.239 (mJ)s</td>
<td>7.784 (mJ)s</td>
<td>19.75 (mJ)s</td>
<td>37.16 (mJ)s</td>
<td>60.00 (mJ)s</td>
</tr>
<tr>
<td>2001-20</td>
<td>1.239 (mJ)s</td>
<td>2.618 (mJ)s</td>
<td>11.84 (mJ)s</td>
<td>15.94 (mJ)s</td>
<td>33.27 (mJ)s</td>
</tr>
<tr>
<td>2003</td>
<td>7.925 (mJ)s</td>
<td>30.93 (mJ)s</td>
<td>68.90 (mJ)s</td>
<td>121.8 (mJ)s</td>
<td>189.7 (mJ)s</td>
</tr>
<tr>
<td>T-MAC</td>
<td>3.566 (mJ)s</td>
<td>7.489 (mJ)s</td>
<td>12.20 (mJ)s</td>
<td>19.74 (mJ)s</td>
<td>26.32 (mJ)s</td>
</tr>
<tr>
<td>RR-2</td>
<td>0.4764 (mJ)s</td>
<td>0.4841 (mJ)s</td>
<td>0.4919 (mJ)s</td>
<td>0.4997 (mJ)s</td>
<td>0.5074 (mJ)s</td>
</tr>
<tr>
<td>RR-5</td>
<td>0.4868 (mJ)s</td>
<td>0.5063 (mJ)s</td>
<td>0.5258 (mJ)s</td>
<td>0.5455 (mJ)s</td>
<td>0.5652 (mJ)s</td>
</tr>
<tr>
<td>RR-10</td>
<td>0.5043 (mJ)s</td>
<td>0.5435 (mJ)s</td>
<td>0.5831 (mJ)s</td>
<td>0.6230 (mJ)s</td>
<td>0.6633 (mJ)s</td>
</tr>
</tbody>
</table>

Table 5.32: Energy consumption per node per rendezvous period for assumed low energy sleep mode of WaveLAN radio module

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Joules</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-MAC 2001-5</td>
<td>(1.366 \times 10^{-3})</td>
</tr>
<tr>
<td>S-MAC 2001-10</td>
<td>(2.715 \times 10^{-3})</td>
</tr>
<tr>
<td>S-MAC 2001-20</td>
<td>(5.415 \times 10^{-3})</td>
</tr>
<tr>
<td>S-MAC 2003</td>
<td>(2.991 \times 10^{-4})</td>
</tr>
<tr>
<td>T-MAC</td>
<td>(4.435 \times 10^{-5})</td>
</tr>
<tr>
<td>Ripple Rendezvous</td>
<td>(1.324 \times 10^{-4})</td>
</tr>
</tbody>
</table>
Comparison

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>0.7775 mJ</td>
<td>0.7776 mJ</td>
<td>0.7777 mJ</td>
<td>0.7777 mJ</td>
<td>0.7778 mJ</td>
</tr>
<tr>
<td>2001-5</td>
<td>2.477 mJ</td>
<td>5.208 mJ</td>
<td>7.939 mJ</td>
<td>10.67 mJ</td>
<td>13.40 mJ</td>
</tr>
<tr>
<td>2001-10</td>
<td>2.461 mJ</td>
<td>5.177 mJ</td>
<td>7.892 mJ</td>
<td>10.61 mJ</td>
<td>13.32 mJ</td>
</tr>
<tr>
<td>2001-20</td>
<td>2.461 mJ</td>
<td>5.161 mJ</td>
<td>7.876 mJ</td>
<td>10.57 mJ</td>
<td>13.29 mJ</td>
</tr>
<tr>
<td>2003</td>
<td>1.761 mJ</td>
<td>3.256 mJ</td>
<td>4.752 mJ</td>
<td>6.247 mJ</td>
<td>7.743 mJ</td>
</tr>
<tr>
<td>T-MAC</td>
<td>0.9683 mJ</td>
<td>1.067 mJ</td>
<td>1.166 mJ</td>
<td>1.299 mJ</td>
<td>1.398 mJ</td>
</tr>
<tr>
<td>RR-2</td>
<td>0.9392 mJ</td>
<td>0.9402 mJ</td>
<td>0.9412 mJ</td>
<td>0.9422 mJ</td>
<td>0.9432 mJ</td>
</tr>
<tr>
<td>RR-5</td>
<td>0.9405 mJ</td>
<td>0.9431 mJ</td>
<td>0.9456 mJ</td>
<td>0.9482 mJ</td>
<td>0.9507 mJ</td>
</tr>
<tr>
<td>RR-10</td>
<td>0.9428 mJ</td>
<td>0.9479 mJ</td>
<td>0.9530 mJ</td>
<td>0.9581 mJ</td>
<td>0.9632 mJ</td>
</tr>
</tbody>
</table>

Table 5.33: Energy vs. Route Length for reduced rendezvous period of T-MAC

5.32. Here we see that the S-MAC 2001 protocols consume significantly more energy every rendezvous period than the other protocols with the energy expenditure increasing as their fixed active time increases. S-MAC 2003 consumes the next greatest amount of energy with ripple rendezvous consuming with the lowest empty network energy consumption being attributed to T-MAC which consumes only 2.957 times the energy that would be consumed if the network were to be in sleep mode for the entire rendezvous period.

The throughput is unaffected by a low energy sleep mode and the throughput characteristics for this situation are as for those presented in table 5.29.

5.3.6 Modify rendezvous period of T-MAC

Finally, to allow a direct head-to-head comparison of T-MAC with ripple rendezvous, the rendezvous period of T-MAC is set to a quarter of a second while the rendezvous period for the other protocols is kept at one second. This means that both T-MAC and ripple rendezvous have nodes becoming active four times per second. All other parameters are kept as in the previous section.

Referring to figure 5.22 and table 5.33 it is clear that the energy
Figure 5.22: Energy vs. Route Length for reduced rendezvous period of T-MAC

Consumed during the delivery of a packet improves for T-MAC. This is due to the fact that the packet is being delivered faster now due to T-MAC's increased level of activity, although nodes become active the same number of times during the delivery of a packet as for when T-MAC used a rendezvous period of one second, nodes now spend less time in sleep mode and thus the nodes expend less energy.

Looking at how the time of delivery presented in figure 5.23 and table 5.34 improves for T-MAC, we see that T-MAC has a similar latency to the S-MAC 2001 protocol that uses a rendezvous period of one second that has been tuned to allow 10 hops per active period.

Looking at the performance of T-MAC under the Energy-Delay metric presented in figure 5.24 and table 5.35, it is evident that T-MAC now becomes the second best performing real protocol behind the ripple rendezvous protocols. However as the route tends towards 50 hops the Energy-Delay of T-MAC is still an order of magnitude greater than all varieties of ripple
Figure 5.23: Delay vs. Route Length for reduced rendezvous period of T-MAC

<table>
<thead>
<tr>
<th>Protocol</th>
<th>10 hops</th>
<th>20 hops</th>
<th>30 hops</th>
<th>40 hops</th>
<th>50 hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>3.462 ms</td>
<td>6.924 ms</td>
<td>10.39 ms</td>
<td>13.85 ms</td>
<td>17.31 ms</td>
</tr>
<tr>
<td>2001-5</td>
<td>1.502 s</td>
<td>3.502 s</td>
<td>5.502 s</td>
<td>7.502 s</td>
<td>9.502 s</td>
</tr>
<tr>
<td>2001-10</td>
<td>0.5037 s</td>
<td>1.504 s</td>
<td>2.504 s</td>
<td>3.504 s</td>
<td>4.504 s</td>
</tr>
<tr>
<td>2001-20</td>
<td>0.5037 s</td>
<td>0.5073 s</td>
<td>1.504 s</td>
<td>1.507 s</td>
<td>2.504 s</td>
</tr>
<tr>
<td>T-MAC</td>
<td>0.8753 s</td>
<td>1.625 s</td>
<td>2.376 s</td>
<td>3.375 s</td>
<td>4.125 s</td>
</tr>
<tr>
<td>RR-2</td>
<td>0.5073 s</td>
<td>0.5150 s</td>
<td>0.5227 s</td>
<td>0.5304 s</td>
<td>0.5380 s</td>
</tr>
<tr>
<td>RR-5</td>
<td>0.5177 s</td>
<td>0.5369 s</td>
<td>0.5561 s</td>
<td>0.5753 s</td>
<td>0.5946 s</td>
</tr>
<tr>
<td>RR-10</td>
<td>0.5350 s</td>
<td>0.5734 s</td>
<td>0.6119 s</td>
<td>0.6503 s</td>
<td>0.6887 s</td>
</tr>
</tbody>
</table>

Table 5.34: Delay vs. Route Length for reduced rendezvous period of T-MAC
Figure 5.24: Energy-Delay vs. Route Length for reduced rendezvous period of T-MAC

Table 5.35: Energy-Delay vs. Route Length for reduced rendezvous period of T-MAC
comparison

rendezvous.

The empty network power consumption for the protocols are as for table 5.32 except that since T-MAC has nodes becoming the same number of times as ripple rendezvous the power consumption for T-MAC is also 0.1324 mW.

Since the active time of T-MAC is adaptive to the level of traffic within the network and can consume the entire rendezvous period if traffic needs to be carried, the throughput for T-MAC is as given in table 5.29.

Having presented many different results for a variety of parameters and model variables, in the next chapter we collate these implications of the results and make a number of conclusions. We again re-state that the protocols are compared only for the traffic pattern defined at the start of the chapter in which a single pattern is unicast along an extended multihop route to some distant destination. It is expected that ripple rendezvous will not perform as well under the traffic pattern known as local gossip. Hang in there, only one chapter to go!!
Chapter 6

Summary, Conclusion and Recommendations

6.1 Summary

To recap the work presented in this thesis, the findings are summarized in this section.

6.1.1 Chapter One: Introduction

A brief introduction to the field of wireless sensor networks is given in this chapter followed by a review of the research steps undertaken in the completion of this thesis. An outline of the thesis is presented followed by a summary of the significant, original and novel contributions made by this thesis.

6.1.2 Chapter Two: Defining the Problem

Chapter two presents the problem that is the focus of this thesis, beginning with a broad overview of wireless sensor networks and three examples of the 'flavor' of applications that are considered. From these applications some design requirements of the sensor nodes are developed. The relevant requirements such as low energy consumption, scalability, adaptability and low latency are noted and requisite features not affected by this research such as size and cost are excluded. The architecture of a sensor node is
examined with a focus on the communication components of a sensor node. This is achieved with the introduction of the concept of a protocol stack and after a brief explanation of each layer in the protocol stack, the MAC layer, that is the focus of this work, is explained. Two energy saving techniques are then introduced that have a significant impact on the solution presented; multihop routing and sleep mode. The chapter concludes with a look at the broad range of MAC protocols that have been proposed for wireless sensor networks that take different approaches from using multiple channels for communication to implementing a TDMA channel access scheme to allowing nodes to contend for the medium in an unsynchronized manner, culminating in our interest in synchronized, single channel, contention-based MAC protocols.

6.1.3 Chapter Three: Current solutions to the Problem

Chapter three examines the current solutions to the problem of developing an energy efficient MAC protocol for wireless sensor networks. The chapter begins with the Prototype Embedded Network (PEN) which gives this work much of the terminology that is used. The IEEE 802.11 MAC protocol is then examined in detail such that the reader may become familiar with issues relating to a wireless MAC protocol. The S-MAC 2001 protocol is then discussed and it is revealed that the energy cost of the three active modes of a radio transceiver (transmit, receive and idle) can be considered to consume similar amounts of energy when compared to the energy consumption of sleep mode. The S-MAC 2001 protocol effectively predetermines the energy consumption of a sensor node’s radio by stipulating that a node becomes active for a fixed length of time $T_a$ every rendezvous period $T_{rp}$. While this protocol makes much progress towards minimizing the energy consumption of a sensor node’s radio transceiver with message passing and overhearing avoidance being important mechanisms for sensor nodes to
adopt, it was also shown that much room for improvement remains, with the energy consumed by a node’s radio transceiver while it is in idle mode accounting for a significant portion of total energy expenditure.

The T-MAC (Timeout-MAC) protocol ensures that the energy consumed by a sensor node’s radio transceiver is proportional to the demands made of the MAC layer. In other words, its energy expenditure is related to the amount of work that it is required to do. It achieves this with a mechanism that puts a node into sleep mode if it has not sensed or performed any communication for a certain length of time $T_i$. This timeout mechanism effectively eliminates energy wasted by a radio standing idle. However, the result of the timeout mechanism is that a packet may only be sent two hops every rendezvous period. Thus, the MAC frame dialogue is modified to include a short (FRTS) control frame that extends the range that a packet may be sent to three hops.

A further refinement to the S-MAC protocol in 2003 is discussed. This incarnation of S-MAC reduces the fixed active $T_a$ of a node to that length of time required to transmit one packet. A mechanism known as adaptive listening has a node staying active for a short period after reawakening from sleeping due to overhearing avoidance in case they are the next node to receive the packet. This means that a packet can only be transmitted two hops every rendezvous period. In reducing the energy consumption of the radio transceiver, a higher latency in delivering a packet is produced.

The concept of a ripple is then introduced prior to the discussion of the next protocol which uses a very similar technique. A ripple is simply a sequence of pre-arranged cascading connectivity by a set of nodes. The next protocol, DMAC, has nodes arranging themselves into data gathering trees that ‘ripple’ from the leaf nodes to the root node, meaning that nodes consume energy in proportion to the level of traffic that they must carry as well as delivering the information in a timely manner. The downside
to the DMAC protocol is that it does not allow the independent operation of the network/routing layer above it since information must travel along a predefined route. This lack of routing flexibility means that DMAC is unsuitable and is not considered further.

Chapter three concludes by stating that what is needed is a MAC protocol that 1) incorporates the energy saving mechanisms of S-MAC, 2) is energy efficient and 3) allows timely delivery of packets 4) between arbitrary source and sink nodes.

6.1.4 Chapter Four: Proposed Solution

Chapter four begins by further discussing the concept of a ripple in one dimension. The active time $T_a$ of each node is divided equally into two halves. The first half is for receiving information from its upstream\footnote{upstream' denotes nodes that become active $\frac{T_a}{2}$ seconds before this node becomes active} neighbor(s)\footnote{when multiple ripples overlap a node can have multiple neighbors as well as multiple peers} and the second half is for transmitting information to its downstream\footnote{nodes that become active $\frac{T_a}{2}$ seconds after this one.} neighbors. The ripples presented in this document are then differentiated from the ripples used by DMAC in that information is not delivered at a 'per packet' speed, but because a ripple has a strict limit on the maximum amount of information received and transmitted per rendezvous period, a packet is delivered at a speed dictated by the 'carrying capacity' of the ripple. The cascading connectivity that constitutes a ripple is described such that it is evident that information may be delivered in a timely manner. To ensure that energy consumption is proportional to the level of traffic and the network layer is not significantly constrained, the R-MAC protocol which operates within ripple rendezvous is introduced.

The next section seeks to implement these one dimensional ripples in a two dimensional grid network topology. A method of scheduling the
activity of nodes in a network or *rendezvous scheme* is presented. This method has ripples overlapping to form wave fronts of cascading connectivity within the network. The rendezvous scheme has four different types of wave fronts moving through the network. Each wave front consists of either a column or row of nodes becoming active at the same time (they have the same rendezvous phase ($\phi$) to form a geo-wave. This scheme has nodes becoming active four times per $T_{rp}$. Only all minimum hop-length routes are possible within one geo-wave and if so desired, sub-optimal routes are possible when passage from source to sink is made up of a combination of geo-waves. The details of what happens when two simultaneous routes intersect are examined as well as the BoB control frame mechanism of the R-MAC protocol that allows nodes to work their respective packets around each other as they journey from source to sink.

The next section discusses the routing implications of the geo-wave rendezvous scheme and demonstrates there is a degree of tuning that needs to take place to ensure that the expected level of traffic is met. There is also a limitation in how much simultaneous information can converge to a single node requiring an interface device with a greater range than the sensor nodes posses.

The next section discusses a method of implementing a ripple rendezvous scheme in a network that, rather than having a grid topology, has nodes distributed in an ad hoc topology. The construction of such a network relies on the GAF topology control protocol. The section begins with how, with the use of a virtual grid and their own location, nodes are able to determine their topological equivalence and elect what is effectively a cluster head that represents each grid cell. The requirement for high precision synchronization is met by stating that the process by which nodes determine their geographic location is also able to provide them with synchronization information. This produces an approximately grid structured network. A
node’s signal, under this arrangement, will spill over into virtual grid cells beyond those immediately neighboring and the R-MAC protocol needs to be modified to one in which nodes in receive mode who overhear a RTS or CTS control frame will transmit a BoB control frame based on their distance from the overheard frame. This ensures that as many neighbors as possible who may be affected by the transmission of a packet are alerted to the impending transfer.

The self organizing routines are then discussed in which during the pre-deployment phase, the nodes are supplied with a set of reference coordinates and rendezvous times. During the self organizing phase of the network, nodes are able to use this information to determine their place in the larger network structure and as they begin to behave accordingly, the network becomes ready to use. The chapter concludes by stating that R-MAC and the ripple rendezvous protocol meet all four of the design requirements listed at the end of chapter three.

6.1.5 Chapter Five: Comparison of Solutions

The chapter begins by defining the performance metrics, the parameters and the comparison scenario to be used. The performance metrics used in the comparison are: 1) The energy consumed per node by a network that contains no traffic (the network is unutilized); 2) The time of delivery, which is from when the MAC layer at the source node receives a packet until the MAC layer at the sink node finishes transmitting the final acknowledgement for the last DATA fragment (at which stage the MAC layer reassembles the packet and delivers it to the network layer); 3) The energy consumed per node on the route during the time of delivery; 4) The product of the time of delivery and the energy consumed during the time of delivery. This metric accounts for the competing demands of timely delivery and energy efficient operation; and 5) The maximum throughput of a single route.
The parameters used in the performance models are discussed. Starting with the bandwidth and energy consumption of the RFM TR1001 and Lucent WaveLAN radio modules. The reasons for considering routes consisting of up to 50 hops are then given. Parameters influencing the time required to transmit a packet are discussed as are issues relating to the average time required to contend for the medium (average back off time). The comparison scenario is presented in which the average of many routes over varying route lengths is considered. A grid network is used to remove any variation introduced by using a random, ad hoc topology in which the performance of the protocols is strongly influenced by the exact location of the nodes.

The next section presents mathematical models that are used to analytically determine the performance metrics. Models of the limits are determined to give an idea of how a perfectly ideal MAC protocol would perform. Equations that model the five performance metrics are presented along with their derivation for both incarnations of the S-MAC protocol (2001 and 2003), T-MAC and geo-wave ripple rendezvous with R-MAC.

The performance models are examined for a set of initial protocol settings including the use of the TR 1001 radio module, a packet size of 300 bytes and a rendezvous period of one second. S-MAC 2001 has three different settings represented in which a packet can be sent on average 5, 10 and 20 hops per active period respectively. Similarly ripple rendezvous has three different implementations reflecting the fact that each node has the capacity to transmit 2, 5 and 10 packets per rendezvous period per geo-wave.

These initial parameters do not favor ripple rendezvous with only the RR-2 setting being able to operate. There is only just enough room for RR-2, which has waves giving way to each other every four hops. In fact the only metric in which RR-2 performs best is under the energy-delay metric for route lengths greater than 17 hops.
The rendezvous period used by ripple rendezvous is then modified to five seconds in an attempt to make room for the waves. This results in ripple rendezvous having the lowest empty network power consumption. For the energy consumed during the delivery of a packet, RR-2 is best for routes greater than 15 hops and RR-5 is next best for routes greater than 35 hops. Both of these varieties hold the same status under the energy-delay metric for similar route lengths. The disadvantage of an increased rendezvous period for ripple rendezvous is a significant decrease in the maximum throughput of all varieties.

An alternative approach to make more room for the formation of wave fronts, is to decrease the size of the network layer packets to 30 bytes while keeping all other parameters the same as the initial settings. Under the energy consumed during the delivery of a packet, RR-2 is better than all other protocols for routes greater than 10 hops RR-5 is better for routes greater than 15 hops and RR-10 becomes better than all S-MAC flavors (but not better than T-MAC) for routes greater than 17 hops. RR-2 becomes the best performing protocol in terms of delay in delivering a packet for routes greater than 20 hops. RR-2 and RR-5 perform best under the energy-delay metric for routes greater than 10 and 20 hops respectively. Because the size of the contention window has not changed, the empty network energy consumption of ripple rendezvous and T-MAC does not change, however the performance of all flavors of S-MAC improves because their fixed active time is decreased.

The parameters for the TR1001 radio are replaced with the parameters for the Lucent WaveLAN radio module and all other parameters are kept at the initial settings. The effects of the increased bandwidth of the new radio are immediately evident in the fact that all varieties of ripple rendezvous now have superior performance in terms of time of delivery. Since information is delivered so quickly, Ripple rendezvous nodes do not expend
as much energy as all other protocols for route lengths greater than 20 hops. Similarly ripple rendezvous performs best under the energy-delay metric for routes greater than 20 hops. Due to the fact that the sleep-mode energy consumption is relatively high, the empty network energy consumption is very similar for all protocols.

Assuming a sleep-mode energy consumption for the WaveLAN like that of the TR1001 sees ripple rendezvous performing better than all other protocols under all other metrics except for time of delivery for short routes\(^4\), throughput (as has been explained) and empty network energy consumption (T-MAC is the only protocol that is better). For a head-to-head comparison of T-MAC vs. ripple rendezvous, the rendezvous period for T-MAC is set to a quarter of a second and thus T-MAC and ripple rendezvous have nodes becoming active the same number of times per second. While this modification does improve the performance of T-MAC in terms of the time of delivery, it does not improve T-MAC beyond ripple rendezvous.

This concludes the summary of this thesis. The next section discusses the results and makes some conclusions and recommendations.

6.2 Discussion and Conclusion

The emerging field of wireless sensor networks has been explored in a way that focusses on large-scale applications. The importance of the MAC/data link layer in determining the energy consumption of a wireless sensor node has been revealed and current systems that organize the use of the radio module have been discussed. It has been revealed in this discussion that there is a trade off between low-energy operation and speedy delivery of information. Thus a new way of coordinating the activity of nodes in a wireless sensor network is proposed. The proposed method, known as Ripple Rendezvous, presents a new and better trade off between the amount of

\(^4\)S-MAC always beats ripple rendezvous in time of delivery for routes that are able to be delivered in one active period.
information that can be delivered along a long-haul multihop route through the network per unit time and the speed with which it is delivered.

From the results of the comparison of the protocols, it is evident that Ripple Rendezvous has superior performance over the other protocols examined, when the active time is small compared to the rendezvous period as in the case when a smaller packet size is used with the TR1001 radio and for all cases where the Lucent WaveLAN radio is used. The energy efficient operation is made most apparent when there is a low energy sleep mode available for the radio module to use.

It should also be noted that it is the MAC protocol with an idle listening shutdown mechanism like that used by T-MAC and R-MAC that ensures that a node uses its radio transceiver in an energy efficient manner. It is the ripple rendezvous protocol which determines when nodes become active that allows information to be delivered in a timely manner. The virtual cluster establishment algorithms developed by S-MAC and adopted by T-MAC allow nodes to coordinate their active times using local interactions only and this is important to ensure that these protocols function for large networks. It is argued that if nodes use information about their location they can intelligently coordinate their active times to form ripples within the network which will allow information to move through the network quickly.

Ripple rendezvous and R-MAC either allow nodes to consume the same amount of energy as the other protocols examined in this document, but in doing so allow information to be delivered significantly faster, or to ensure the same latency as the other protocols ripple rendezvous will consume less energy. Under the right protocol parameters Ripple rendezvous and R-MAC may even be able to consume less energy and allow information to be delivered faster than all of the other protocols examined.

The relevant design requirements that were established in chapter 2 are now reviewed with a discussion of how this thesis has managed to meet
them.

**Rapid information delivery** Ripple rendezvous presents network designers with a choice between the speedy delivery of information and the amount of information that can be delivered along a particular route per rendezvous period. If more information is required to be transmitted through the network then the ripples of connectivity can be set to have a greater carrying capacity but as explained they will move through the network slower. There is of course the option of reducing the rendezvous period to increase the throughput of ripple rendezvous per unit time, however this will have an associated extra energy cost with nodes becoming active more frequently and thus expending more energy. This is because each time a node becomes active, it will waste some energy engaged in idle listening (while waiting for its idle listening timer to fire) regardless of the traffic that it carries.

One of the most important advantages of the ripple rendezvous scheme is that the speed with which information can be transmitted through the network is fairly independent of the rendezvous period. The initial sleep delay between when a packet arrives at the MAC layer on the source node until it starts being transmitted to the first relay node does depend on the length of the rendezvous period with the sleep delay being on average half of the rendezvous period. From this point on, the information is transmitted through the network at the speed with which the ripple of connectivity traverses the network.

For example, using a WaveLAN-like radio module, for RR-10 the ripple advances through the network about every 4 milliseconds per hop meaning that it moves at about 250 hops per second. It is thus possible for an application that requires a latency of, say, 10 seconds to use a ripple rendezvous scheme with a rendezvous period of 10 seconds. This will give a maximum time of delivery for routes up to 250 hops of about
11 seconds and an average time of delivery of 6 seconds. S-MAC and T-MAC are not able to modify the rendezvous period with as much flexibility because their time of delivery is more closely correlated with the rendezvous period. In this thesis it has been assumed that there is little or no communication between the MAC layer and higher layers. If information from the MAC layer were to be made available to higher layers, then information could be sensed and aggregated just in time for a packet to be sent within a ripple from source to destination. This would eliminate the initial sleep delay from the time of delivery and thus in the example just given, a latency of less than one second can be achieved.

The rendezvous period then determines how frequently a packet can be delivered from source to sink rather than the latency of the packet. It is the ability to move information through the network in a very energy efficient and timely manner that makes ripple rendezvous and R-MAC a valuable contribution to the developing field of wireless sensor networks. This opens up a whole new area of application involving real-time monitoring and reporting of information to anywhere in the network and thus significantly improves the utility of wireless sensor networks.

Low energy consumption It is not possible for the MAC/link layer protocol to ensure a low energy consumption because if this layer is required by the higher layers in the protocol stack to transmit a lot of information, then much energy will be consumed. The best that the protocols at this layer can do is ensure that all communication is highly efficient, with energy only spent on sending or receiving information. This is largely the case with the R-MAC protocol which has nodes wasting energy through idle listening only while waiting for their idle listening shutdown timer to fire and while nodes are backing off as
Adaptability The R-MAC protocol that operates within ripple rendezvous means that the energy expended by a node’s radio transceiver is proportional to the level of traffic that the node is required to ’carry’. This means that the energy expended by a node is adaptive to the demands made of the MAC/link layer. There is a limit to the extent to which a node can adapt and this is defined by the carrying capacity of the ripples used by the network.

Robustness It is not the role of the MAC layer to deal with node or link failures and thus the robustness of an application that is utilizing ripple rendezvous and R-MAC is largely dependent on the network layer routing protocol that is selected. For the application to be robust in the event of node and/or link failures, it is necessary that a dynamic routing algorithm be used such as Directed Diffusion which is briefly outlined in Appendix A.

Extended lifetime Again, this requirement has little to do with the MAC/link layer. All that this layer can do is ensure energy efficient use of the radio transceiver. It is the network/routing layer that is responsible for selecting the route that is used and thus where the energy will be consumed in the network. See appendix A for a brief introduction to directed diffusion, an application/network layer protocol suite developed specifically for wireless sensor networks.

Self-organizing The algorithm by which a network organizes itself to form a ripple rendezvous scheme is given in section 4.5. Each node in the network is supplied with the same reference information and once nodes have gone through their neighbor discovery and GAF topology control algorithms, they are able to use their current location in space and time relative to the reference information to calculate their
Summary, Conclusion and Recommendations

own required behavior. The total time to self organize is not expected to be significantly greater with the addition of the ripple rendezvous algorithms since ripple rendezvous self-organization makes use of information already established by the GAF topology control routines and the only expected additional delay being while computing the adjustment of the reference information to a node’s actual location.

Scalability The geo-wave ripple rendezvous scheme scales well to large networks. In fact, it is not until the network becomes large (and thus long multi-hop routes become utilized) that the scheme begins to exhibit superior performance to the other protocols. This is both a testament to the scalability of Ripple rendezvous and the problems with the scalability of the other protocols. For example S-MAC 2001 has increased problems with idle listening as the route length increases. T-MAC has greater problems with the time of delivery as the route length increases and S-MAC 2003 has more problems with both of these metrics as the route length becomes larger.

Mobility As we have stated, nodes are not mobile in a wireless sensor network but they can be relocated within the sensor network by some entity or event that occurs within the sensor field. The mechanism that deals with the repositioning of a node is the GAF protocol which even has the capacity to maintain control over the topology in an environment with highly mobile nodes.

Ripple rendezvous performs better than T-MAC for long-haul multi-hop routes. For local communication and broadcast such as when an event has been sensed and local nodes aggregate their sensed information or information is flooded throughout the network, a common rendezvous phase is desirable and T-MAC is likely to perform better under this local gossip traffic pattern (see Section 3.4). The reason that ripple rendezvous does not
perform as well under this communication primitive is because not all of a node’s immediate neighbors are awake at the same time due to the staggered nature of their activity. Ripple rendezvous has been designed to support the timely and energy efficient delivery of information along long-haul multihop routes and to support the full operation of sensor network traffic it is necessary to combine ripple rendezvous with a protocol that supports the energy efficient operation under the local gossip traffic primitive. T-MAC may be integrated into ripple rendezvous by using the same radio channel, however a preferable solution would be to use a separate radio channel so that the two different types of communication do not interfere with each other. This will allow mobile devices to retain synchronization with the network regardless of their location within it. This need not contribute to a significant extra energy cost since the T-MAC protocol ensures that the extra energy cost is proportional to information that must be exchanged anyway.

It is concluded that a network using a MAC protocol with an idle listening shutdown mechanism that coordinates the active times of nodes to be staggered along a route is capable of facilitating the timely delivery of information along long routes and highly energy efficient operation. When the staggered active times are allowed to overlap to form geo-waves, a MAC protocol such as R-MAC is required to ensure that routes that intersect will be able to keep the information moving within the geo-wave and thus ensure that as long as the carrying capacity is not exceeded, information can be delivered in an energy efficient and timely manner between arbitrary source and destination nodes.

This means, finally, that ripple rendezvous will be able to support traffic patterns such as data gathering trees used by DMAC (see section 3.7) as well as the convergecast traffic pattern (see section 4.3) as the protocol has been described as long as the total information that is transmitted within the tree or convergecast scenario does not exceed the carrying capacity of
the geo-wave.

If the carrying capacity is exceeded, information will be delayed until the next rendezvous period at which stage it may be able to resume its journey to the destination as long as there is available capacity within the geo-wave. If the volume of information is in excess of the carrying capacity for only a short time, then ripple rendezvous should be able to relieve the backlog of information relatively quickly and the information need not be significantly delayed. If however, the carrying capacity is exceeded every rendezvous period, then information will build up and form a backlog that ripple rendezvous is unable to recover from. Thus the selection of the setting of the carrying capacity of the ripples is an important parameter for network designers to consider.

6.3 Recommendations and Future Research

Having concluded that the ripple rendezvous protocol suite is well suited to large sensor networks that require information to be delivered in an energy efficient and timely manner between arbitrary source and destination nodes, some recommendations are now made and areas identified that the author sees as the next stages in seeing ripple rendezvous and R-MAC implemented in an operational prototype sensor network.

Because the focus of this work is on large sensor networks, simulation of typical application scenarios should be explored. This will allow examination of the effects of different routing protocols to be established as well as determining the appropriate operational parameters to be used for ripple rendezvous. In other words, the most suitable setting for the carrying capacity ($T_a$) and the rendezvous period ($T_{rp}$).

A simulation would also allow the further investigation of alternative implementations of ripples within an ad hoc network that do not rely on global location and synchronization as well as alternative methods of self
organizing.

The simulation may also be used to evaluate the effectiveness of an alternative MAC protocol that has been envisaged. Rather than implementing the RTS-CTS-DS(BoB)-DATA-ACK handshake, the proposed MAC protocol uses a RTC-OTC-BoB-DATA-ACK handshake. RTC is a Request To Carry frame which is preceded by a contention period as prior to the RTS control frame. OTC is an Offer To Carry control frame which is preceded by a second contention period from which nodes select a delay in responding based on their suitability to route the packet to its ultimate destination based on information contained in the RTC frame. A node’s suitability could be based on how close it lies to the optimal route, its remaining energy supplies and the presences of a clear radio channel. Such a protocol represents a combined MAC/Routing layer protocol. Such a protocol may allow the development of a three layer protocol stack with just a physical layer, the above MAC/routing layer and an application layer. This may result in an extremely energy efficient protocol stack for wireless sensor networks and is considered to be worthy of further investigation.
Appendix A

Directed Diffusion

This appendix presents a summary of the Directed Diffusion routing paradigm as presented in [9] and [20] for a more detailed explanation of the specifics of directed diffusion, please refer directly to the sources.

Directed Diffusion was developed with three requirements in mind; that of robustness, scaling and energy efficiency. Robustness is achieved by allowing sensed information to be sent along multiple paths. By exploiting the inherent redundancy within a sensor network, the impact of link failure on the performance of the application can be reduced. Scalability is ensured by making sure that only local communication rules are used to establish routes or draw down information from nodes with information to send. Energy efficiency is achieved by allowing nodes to aggregate their responses to network interrogation.

Essentially what happens in a network that is using directed diffusion as an application/network protocol is that an interest for information about some phenomenon is injected into the network at some arbitrarily chosen sink node. This interest is then propagated to the region of interest using directional flooding to establish some initial interest gradients. A gradient can be thought of as the strength of the interest in the named phenomena of the sensing task, for example a stronger interest or a higher gradient may be reflected by an interest that has a more frequent update criterion.
Information is then returned from the region of interest to the sink node. At this stage the sink node can make a selection from all of the routes that the information has been returned along. The selection of which routes to use can be based on many different criteria such as the hop count, the latency or the total energy cost of the route. More complex metrics such as the route on which nodes have the maximum energy reserves or even the route with the maximum minimum energy reserve of a node on it. There are also local rules built into the directed diffusion routines for route repair in the event of link failure.

Regardless of the reason for selecting a particular route, the routes that are empirically better than others are reinforced by re-sending the original sensing task along the selected routes but with a stronger level of interest. This way, multiple paths that meet the selection criteria are opportunistically established from the source nodes to the sink using only local rules and communication. Directed diffusion incorporates the top three layers of sensor node’s protocol stack (see figure 2.2), namely the application, transport and network layers.
Appendix B

Central and Diagonal Waves

B.0.1 Central Waves

This was the first rendezvous scheme developed and was done so by the use of the combinatorial optimization technique known as Simulated Annealing [24]. Under this technique, a set of parameter values that represent a complex system are subject to a cost or objective function that determines their ‘fitness’. A population of sets of parameter values is subject to rounds of random changes and if the random change results in a better cost as described by the objective function, then this random change is accepted and the set of parameter values is updated to include the random change.

If the random change results in a worse cost for a set of parameter values, it is accepted subject to chance. This chance is determined by the current ‘temperature’ variable where there is a higher chance of a less fit set of parameter values being accepted at a higher temperature. After each round of random changes the value of the ‘temperature’ is reduced which reflects a decrease in the probability of accepting a random change that results in a less ‘fit’ set of parameter values.

This allows a set of parameter values that has settled into a local maximum (or minimum, depending on the objective function) to ‘climb’ out of it and (hopefully) continue on its way towards the global minimum (or maximum). There is no guarantee that the simulated annealing technique
will produce the globally optimal solution, however it has been shown to be an effective strategy for searching the solution space of combinatorial optimization problems.

In modelling a grid network the set of parameter values (or configuration) was built as a series of boolean matrices. Each matrix represented a time slice of an \( n \times n \) grid network that was long enough for two nodes to completely transceive a packet and each node in the network could either be active or sleeping during each time slice. After every random change, the configuration was tested for validity and if the configuration was not valid then the change was recomputed until a valid configuration was produced.

A configuration was defined as being valid if it allowed full network connectivity. Full network connectivity was defined as being the possibility for each node in the network to be able to send a packet in a multihop fashion to any other node in the network within one rendezvous period.

The objective function for each configuration was determined by counting the number of times nodes in the network became active in all time slices and it was this figure that the simulated annealing algorithm was required to minimize. Figure B.1 presents one of the best configurations that our simulated annealing search strategy of the solution space produced. An active node is depicted as a filled circle while a node in sleep mode is drawn as a clear circle. In the diagram there are nine time slices presented as well as an example of the multi-hop route that a packet must take to get from the top-left node to the bottom left node.

We must be careful to clearly explain these diagrams. Figure B.1 is quite similar to Figures 4.6 and 4.7 since each node is depicted as being either active or in sleep mode. A node that is in sleep mode has its radio transceiver in one state - turned off. A node that is active can have its radio in any of the four possible states. A node’s radio can be turned off (the node is in sleep mode) either because it is engaging in overhearing avoidance or
Figure B.1: Result of simulated annealing for a 10 by 10 grid network
has engaged in idle listening shutdown. The node can be either sending or receiving information (in which case the radio will either be in transmit or receive mode). Finally, a node could be in idle mode either while waiting for its $T_i$ timer to fire or because it is backing off before attempting to transmit a packet. The exact state of a node’s radio is dependent on the traffic that is present in the network at the time.

Nodes that lie along the data route will do the transmitting and receiving of the information. Neighbors of the nodes along the data route will only be in idle, receive (while overhearing a neighboring control packet) or sleep mode (while avoiding overhearing). Nodes that lie outside the range of nodes involved in routing the packet will operate as though there is no traffic in the network and only be in idle mode before their $T_i$ timers fire at which stage they will enter sleep mode.

The reader can see that the optimization process results in a disorderly sequence of nodes becoming active and sleeping and in many cases, there is only one possible route between two nodes. Nodes are active for an average of approximately 2.5 time slices per rendezvous period. This was achieved by have certain ‘key’ nodes becoming active more than twice per rendezvous period and having them included in many routes.

An attempt to ‘clean’ this solution into an orderly arrangement is presented in Figure B.2. Here the reader can the origin of the name we have given for this type of rendezvous scheme; that of central waves. This is because the rendezvous phase offset of a node is determined by its distance from the center of the network. In the cleaned up version presented, each node is active (on average) for approximately 4 time slices per rendezvous period\(^1\). This means that nodes are on in receive mode twice and in send mode twice per rendezvous period.

Such a solution, while interesting, is not appropriate for several rea-\(^1\)nodes at the center of the network are active for three time slices per $T_{rp}$ and nodes on the outside of the network turn on twice
Figure B.2: Cleaned rendezvous sequence for a 10 by 10 grid network

1. This does not allow true MAC layer independence, since the routing protocol must have detailed knowledge of the rendezvous scheme to be able to route information between two nodes.

2. Such a rendezvous scheme has the effect of causing packets to be routed through the center of the network. This causes the nodes in the center of the network to bear a greater proportion of the energy cost in the network delivering packets which will in turn have the effect of shortening the lifetime of the network.

3. For such a rendezvous scheme to be established in a network, it will require each node in the network to have knowledge of the extent of the network and specifically where the center of the network is so that each node can calculate when it is supposed to become active and
when it is supposed to return to sleep mode as it 'ripples' information towards the center of the network and away from it each rendezvous period.

For the reasons stated above the search continued for a two dimensional implementation of ripples that meets the design requirements established in section 3.8.

B.0.2 Diagonal Waves

In an attempt to address that largest deficiency of central waves, that of causing information to be directed through the center of the network, another rendezvous scheme was developed. This scheme has nodes in lines that run diagonally to the grid axis of the network becoming active in succession. These are termed Diagonal Waves and are presented in Figure B.3 along with examples of the routes that information can take to get from the bottom-left node to the top-right node (time slices 1 to 8), as well as from the top-right node to the center node (time slices 9 to 12).

Figure B.3 shows the north-east wave moving through a five by five grid network, followed by the south-west wave. It is not a requirement of diagonal ripples that the south-west wave must wait until the north-east wave has moved through the network as they can avoid each other as described in section 4.1 that discusses one dimensional ripples. Only a 5 by 5 network is depicted since the diagonal waves require more time slices to ensure that full network connectivity is maintained. The movement of the north-east and south-west waves through the network are independent of each other (each can occur at any time during the rendezvous period as long as they 'give way' to each other). It is also possible for diagonal waves to be arranged such that they move through the network from from the north-west to the south-east and visa versa, without affecting their operation.

The effect of this rendezvous scheme is to restrict routing through
the network to a Manhattan style routing. In the example routing presented in time slices 1 to 8 of Figure B.3, one can see that the packet travels firstly to the north (slices 1 to 4) and then to the east (5 to 8). Routing is not restricted to a strict-Manhattan type of routing and an alternate route is presented in Figure B.4 that only depicts the north-east wave.

Rather than using a strict-Manhattan route, a stepped-Manhattan route is used to move a packet from the bottom-left node to the top-right node. In fact there are many of these stepped-Manhattan routes available to be used by the routing protocol. The actual route that is used does not affect the time of delivery of the packet since the packet is being transmitted within the wave which moves at a predetermined rate.

The advantages of a diagonal wave rendezvous scheme are that nodes become active twice per rendezvous period and also that traffic is not directed through the center of the network as it is when central ripples are
Figure B.4: An alternative route for a diagonal rendezvous sequence

used. A significant disadvantage is that for sink nodes that do not lie exactly either the north, south, east or west of the source node only suboptimal routes are available to reach them. This is demonstrated in both Figures B.3 and B.4 in which in both cases eight hops are used to move a packet from the bottom-left node to the top-right node because nodes cannot transmit to their diagonal neighbours\(^2\). The minimum required route length is a route of four hops that takes a packet diagonally through the center of the network. The inability of nodes to send information to their diagonal neighbors results in the subset of allowable routes under this rendezvous scheme excluding many of the optimal routes that are possible.

A strict-Manhattan style of routing may very well have a beneficial effect on the lifetime of a sensor network, since routes involve nodes that are distributed towards the outside of a network rather than straight-line routing which takes the packet through the nodes closer to the center. According to the research presented in [47], it is the nodes towards the center of the network that bear more of the energy load when optimal routes are used since they are more likely to be required to relay a packet towards its ultimate destination. [47] presents a routing scheme that occasionally selects less than optimal routes so that the energy cost of carrying a packet through the

\(^2\)A diagonal neighbor is a neighboring node that lies to the north-east, south-east, south-west or north-west
network is distributed over a larger proportion of the nodes.

This routing strategy relies on the ability to alternate between using optimal routes and using sub optimal alternate paths such that energy consumption is distributed over a larger subset of nodes within the network. Diagonal waves simply do not allow many of the optimal routes to be used and this restriction on the routing of information may be less than satisfactory. In subsection 4.2.1 of the main body of the thesis, a rendezvous scheme is presented that is believed to satisfy all the design requirements outlined in section 3.8, however the findings presented there also hold for the diagonal and central waves presented in this Appendix. The decision as to which rendezvous scheme is the better is the subject of future research to be conducted.
Bibliography


ExScal Research Group - http://www.cast.cse.ohio-state.edu/exscal/ 

energy and delay between MAC protocols for wireless sensor networks. 
In Advanced Communication Technology, 2004. The 6th International 

[40] Rabaey, J., Ammer, M., da Silva, J. J., Patel, D., and Roundy, 
S. Picoradio supports ad hoc ultra-low power wireless networking. 

[41] Rajendran, V., Obraczka, K., and Garcia-Luna-Aceves, J. J. 
Energy-efficient collision-free medium access control for wireless sensor 

[42] Rappaport, T. S. Wireless Communications, Principles and Practice, 

[43] Savvides, A., Han, C. C., and Strivastava, M. B. Dynamic 
fine-grained localization in Ad-Hoc networks of sensors. In 7th Annual 
International Conference on Mobile Computing and Networking (Rome, 
Italy, 2001), ACM Press New York, NY, USA.

[44] Sazonov, E., Jonoyan, K., and Jha, R. Wireless Intelligent Sensor 
Network for Autonomous Structural Health Monitoring. In SPIEs 11th 
Annual International Symposium on Smart Structures and Materials 

[45] Schurgers, C., Tsiatsis, V., Ganeriwal, S., and Srivastava, 
M. Optimizing sensor networks in the energy-latency-density design 
TY - JOUR.


