UNIVERSITY of CALIFORNIA
Santa Barbara

Medium Access Control and Energy Maps for Delay-tolerant Wireless Sensor Networks

A dissertation submitted in partial satisfaction of the requirements for the degree
Doctor of Philosophy
in
Electrical and Computer Engineering

by

Min Kyoung Park

Committee in charge:

Professor Volkan Rodoplu, Chair
Professor Jerry D. Gibson
Professor John J. Shynk
Professor João P. Hespanha

October 2006
The dissertation of Min Kyoung Park is approved.

______________________________
Professor João P. Hespanha

______________________________
Professor John J. Shynk

______________________________
Professor Jerry D. Gibson

______________________________
Professor Volkan Rodoplu, Committee Chair

October 2006
Medium Access Control and Energy Maps for Delay-tolerant Wireless Sensor Networks

Copyright © 2006

by

Min Kyoung Park
To my parents,

for their support, encouragement, and sacrifices.
Acknowledgements

First, I would like to thank my principal advisor, Prof. Volkan Rodoplu, for his guidance, mentorship, encouragement, and support. He has always been insightful in his comments and suggestions throughout my graduate studies.

I would like to thank Professors Jerry Gibson, John Shynk, and João Hespanha for their suggestions throughout my dissertation work. I would also like to thank Prof. Ronald Iltis for his guidance in the early stages of my graduate study.

My research and social life are greatly enriched by my colleagues and friends, including Sang-Uk Ryu, Jiyun Byun, Sukseung Hwang, Vishwanath Venkataraman, Taehyuk Kang, Seung-Jun Kim, Andrew Brown, Sunwoo Kim, Carl Thorner, Ann Nitikman, Kyung-Lee Park, and anyone I might be forgetting. I would also like to thank Val de Veyra and other ECE staff members at UCSB for their administrative help.

Finally, I would like to express sincere gratitude to my family for their endless love and support.
Curriculum Vitæ
Min Kyoung Park

Education

October 2006  Doctor of Philosophy  
Department of Electrical and Computer Engineering  
University of California, Santa Barbara

June 2003  Master of Science  
Department of Electrical and Computer Engineering  
University of California, Santa Barbara

March 2000  Master of Science  
Department of Information and Communications  
Gwangju Institute of Science and Technology, Korea

March 1998  Bachelor of Science  
Department of Electronics Engineering  
Chunnam National University, Korea

Professional Employment

2005 - 2006  Teaching Assistant, Electrical and Computer Engineering, University of California, Santa Barbara

2002 - 2005  Graduate Student Researcher, Electrical and Computer Engineering, University of California, Santa Barbara

Fields of Study

MAC protocol design, energy-efficient network design, delay-tolerant networks, wireless sensor and ad hoc networks.

Honors and Awards

April 2006  ECE Department Dissertation Fellowship at UCSB
2001 - 2002  International ROTARY Club Scholarship

Publications

Papers


Patents

Abstract

Medium Access Control and Energy Maps for Delay-tolerant Wireless Sensor Networks

by

Min Kyoung Park

This thesis develops new techniques for the analysis and design of energy-efficient, delay-tolerant wireless sensor networks. We describe how to efficiently manage the energy resource for two types of delay-tolerant networks: underwater ecological sensing networks, and terrestrial, energy-limited, mobile sensor networks. For underwater networks, we develop a distributed, scalable, energy-efficient medium access control (MAC) protocol that works despite long, unknown propagation delays of the underwater acoustic medium. In our protocol, the use of relative time stamps and the sleep mode of sensor units allow the nodes to operate in a synchronized environment. Further, we develop novel methods that are robust in that they adapt to the changes of a network such as channel variations, new node deployment, loss of synchronization, and node failures.

For delay-tolerant, energy-limited terrestrial networks, we develop methodologies for handling high mobility in order to manage the limited node energy supplies. We propose a novel framework to share, retain and refine end-to-end energy metrics in the joint memory of the nodes, over time scales over which this information can be spread to the network and utilized for energy planning decisions. We construct “energy maps,” which are maps of the end-to-end en-
ergy metrics in space, in order to enable energy optimization in high-mobility networks. We show how to (1) compute the spatial derivatives of energy potentials in high-mobility networks, (2) construct energy maps on-demand via path integration methods, and (3) distribute, share, fuse, and refine energy maps over time by information exchange during encounters.

We develop an algorithm for energy optimization, based on the energy maps, that finds the optimal bit allocation strategy to minimize the energy consumption, subject to a delay constraint. We show that significant energy savings are obtained by leveraging network mobility and the energy maps, when compared with a competing algorithm that allocates the traffic at a constant rate without utilizing the energy map. These techniques enable energy optimization and planning for mobile energy-limited networks.
Contents

List of Tables xii
List of Figures xiii

1 Introduction 1

  2.1 Introduction .................................................. 10
  2.2 Proposed MAC Protocol for UWANs .......................... 12
    2.2.1 Basic Idea ............................................... 12
    2.2.2 Initialization of the MAC Protocol ...................... 14
    2.2.3 Data Transmission After Initialization ................ 17
    2.2.4 Handling Newcomers .................................... 19
    2.2.5 Handling Node and Synchronization Failures .......... 21
  2.3 Addressing Channel Variations ............................. 23
  2.4 Simulation Results ........................................ 26
    2.4.1 No Propagation Delays .................................. 27
    2.4.2 Long Propagation Delays ............................... 29
    2.4.3 A Realistic Network Scenario ......................... 31
    2.4.4 Self-configuration of the Protocol .................... 34
  2.5 Summary .................................................. 36

3 Energy Maps for Large-scale, Mobile Networks 38
  3.1 Introduction ................................................ 40
  3.2 Relationship to Current State of Knowledge ............... 44
  3.3 Construction of the Energy Map for
      Time-invariant Energy Potentials .......................... 48
    3.3.1 System Description .................................. 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.2</td>
<td>End-to-end Measurement of the Energy Potential</td>
<td>49</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Localized Link Energy Cost Measurements</td>
<td>51</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Construction of Energy Potential from Local “Field” Measurements</td>
<td>55</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Collecting and Distributing the Spatial Energy Measurements</td>
<td>60</td>
</tr>
<tr>
<td>3.3.6</td>
<td>Adaptive Cell-sizing Approach</td>
<td>72</td>
</tr>
<tr>
<td>3.4</td>
<td>Construction and Dissemination of the Energy Map for Time-varying Energy Potentials</td>
<td>79</td>
</tr>
<tr>
<td>3.5</td>
<td>Summary</td>
<td>90</td>
</tr>
<tr>
<td>4</td>
<td>Energy-efficient Transmission Utilizing an Energy Map</td>
<td>91</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>92</td>
</tr>
<tr>
<td>4.2</td>
<td>Relationship to Current State of Knowledge</td>
<td>95</td>
</tr>
<tr>
<td>4.3</td>
<td>Basic Assumptions</td>
<td>97</td>
</tr>
<tr>
<td>4.4</td>
<td>Traffic Allocation Algorithm using Dynamic Programming</td>
<td>101</td>
</tr>
<tr>
<td>4.5</td>
<td>Simulation Results</td>
<td>108</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
<td>119</td>
</tr>
<tr>
<td>5</td>
<td>Conclusions</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>131</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Computation of $\nu$ values from $t$ to $(t + 2)$ for Figure 4.7 . . . . . . 105

5.1 Analogy between point-to-point and network QoS concepts developed in this thesis. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
List of Figures

2.1 Basic idea of UWAN MAC Protocol. ........................................ 13
2.2 An example network configuration. .......................................... 15
2.3 Initialization phase of nodes for Figure 2.2 carried out at initial network deployment. ............................................................... 16
2.4 The transmission packet structure and listen duration for newly joining nodes in the data transmission phase. ......................... 17
2.5 Modification of a cycle period during data transmission phase. ... 18
2.6 A new node $F$ joins the network. ............................................ 20
2.7 (a) Handling loss of synchronization, (b) Handling a node failure. 22
2.8 Listen durations placed before and after the scheduled wake-up time. 24
2.9 A scenario of a receive-receive collision with propagation delays and a solution of the collision using the guard time policy. .... 25
2.10 The two cases in which a packet collides with another packet in a cycle. ................................................................. 27
2.11 Fraction of energy wasted due to collisions when ignoring propagation delay. ............................................................. 29
2.12 Fraction of energy wasted due to collisions with propagation delays. 30
2.13 Fraction of energy wasted due to collisions under a realistic network scenario. ............................................................. 33
2.14 Fraction of energy wasted due to collisions when 5 new nodes are added. ................................................................. 35
3.1 Energy potential from end-to-end measurements under the Random Waypoint Model .................................................. 43
3.2 Dominated region based on Joules per bit-meter ....................... 53
3.3 Energy field constructed from averaged local measurements ....... 57
3.4 Virtual, projection paths obtained via the path integration method 58
3.5 Error percentage of the path integration method compared with global end-to-end energy measurements ........................................ 59
3.6 Algorithm for energy metric dissemination and filtering ............ 66
3.7 Percentage error of data dissemination with respect to end-to-end global measurements of the energy potential. ....................... 68
3.8 Percentage error of data dissemination with respect to path integration with global knowledge of measurements ....................... 69
3.9 Number of measurements broadcast to neighbors, parameterized by \( W \) ................................................................. 71
3.10 The ratio of the standard deviation of energy measurements to the average energy with uniform cells .......................... 74
3.11 Cell tessellation with adaptive cell-sizing for the RWP model..... 76
3.12 The ratio of the standard deviation of energy measurements to the average energy with adaptive cells .......................... 77
3.13 Performance comparison between with uniform and adaptive cells: Percentage error of data dissemination with respect to end-to-end global measurements of the energy potential. .......... 77
3.14 Mobility models for time-varying case ................................. 81
3.15 Time-varying node densities at different time sequences ............ 83
3.16 Time-varying energy metrics at different time sequences .......... 84
3.17 An interpolation of the transmission vector and the link energy cost for a cell for which no measurement is available. ............... 87
3.18 Instantaneous percentage errors for the time-invariant case and the time-varying case with varying \( S \) ................................. 88
4.1 Energy planning and optimization utilizing an energy map .......... 93
4.2 Session distributions of two nodes ...................................... 97
4.3 Energy cost distribution for a given sample cell ......................... 99
4.4 Parameters selection from the given energy cost distribution ......... 100
4.5 An energy trellis diagram with 3 stages in time \( t \) .................... 103
4.6 Finding an optimal solution via the graphical method at stage 1 .... 104
4.7 An energy trellis diagram with 3 stages in time \( t \), using \( \nu \) function ................................................................. 105
4.8 An example of \( \nu \) computation with sample energy costs .......... 107
4.9 Energy performance for the time-invariant case ....................... 111
4.10 Energy performance for the time-varying case ....................... 112
4.11 Performance with varying the parameters \( L \) and the number of states for a given session: time-invariant case ..................... 115
4.12 Performance with varying the parameters \( L \) and the number of states for a given session: time-varying case ..................... 116
4.13 Performance comparison of energy planning: optimization based on disseminated energy maps versus that based on actual energy maps with global information on the energy metrics. 118

5.1 Measurement of a link delay. 125

5.2 A reference grid for a mobile destination, where $R$ denotes a reference point for collecting the raw measurements, $S$ denotes the source node, and $D$ denotes the destination node. 129
Chapter 1

Introduction

The wireless networks of the next 10 years are expected to be made up of a plethora of tiny microprocessor-sensor units embedded in clothes, shoes, cars, and buses. We envision that these sensor networks will be extensions of the wired and wireless networks today (which include conventional devices such as laptops and PDAs); however, in terms of the number of devices deployed, the microprocessor-sensor units are expected to be very large-scale and dwarf the existing networks in the number of nodes. Such networks present a number of research challenges, including the design of scalable solutions that range from the design of MAC layer protocols to the design of novel routing protocols that can operate in the presence of thousands to millions of these microprocessor-sensor devices.

A second feature that is expected of the networks of the next 10 years is that in geographic scope, they will be extended into unconventional domains: the interplanetary Internet, underwater acoustic communications, and communication in remote or difficult terrains are three such venues to extend the global networks to
space, to the oceans, and to remote locations that currently have no connectivity. Such extensions into unconventional geographic domains are already exemplified by several efforts within the IETF as well as supported by the NSF. The recent delay-tolerant networking (DTN) paradigm [1][2][3] seeks to develop networking solutions that can deliver asynchronous, delay-tolerant data to remote villages in Africa, by a combination of message-ferrying [4] and other DTN solutions. In addition, recently, new focus areas are emerging both within the ACM community, on underwater networks, as well as within the oceanography, limnology and marine science communities in developing new ecological underwater sensing solutions to collect data from the oceans, and connect these ocean observatories via a global network (operating over satellite and optical links) under the recent efforts of the ORION project [5], and the CREON coral reefs project [6]. In terms of the research challenges addressed, there are significant overlaps between the DTN community that has formed within the ACM, and the community forming from multiple disciplines to address underwater acoustic networks. We expect that the future networks will have a wider variety of data traffic, including delay-tolerant and error-tolerant data (not currently addressed in today’s TCP/IP networks), and that these networks will extend into remote geographical areas not currently included in the scope of the Internet.

Delay-tolerant networks were first proposed in 2003 by IETF drafts and in [1]. These networks are defined in the literature by the following 4 properties:

1. Very large delays (on the order of minutes to days) that are tolerable by the applications.
2. Relatively low data rates.

3. Limited memory and energy of end devices.

4. Frequent network partitions due to node mobility.

We shall call the microprocessor-sensor units “sensors” from now on; however, it must be kept in mind that these units include microprocessors with significant processing capability but which must operate in the presence of severely limited battery supplies. In fact, the energy limitations are so significant that the design of these networks call for approaches that are very different from the networking solutions developed so far for both the wired Internet as well as wireless LANs.

The paradigm of delay-tolerant networking was introduced in 2003 by IETF drafts and [1][2][3].) The literature on DTN’s so far has addressed the following problems: the asynchronous message delivery format (“bundles”) were introduced in the original papers [1][2]. Routing protocols [7][8][9][10] have been proposed, including work [11] that uses encounter frequencies. Definitions including contact, encounter, and their durations, were introduced in [12], and proposals [13][14] were made to solve the problems of end-to-end delivery, using replication and erasure methods. The main challenge in DTN routing algorithms is to address and leverage the multiple disconnections [15] that occur over a very long period. Message ferrying methods [4], as well as node localization methods [16] in the presence of mobile robots have also been proposed. The fertility of the topic for network layer design, and the need for new models have become very clear.

In order to more concretely define the design constraints and environments in which these sensor networks will be addressed, we focus in this thesis on the
following two concrete applications:

- Underwater ecological sensing ("eco-sensing") networks
- Terrestrial, energy-limited, mobile sensor networks.

The common thread that link these two application areas are the properties (1)-(4) of the DTN paradigm; namely, large tolerable delays, low data rates, and limited energy of the nodes. In Chapter 2, we focus on properties (1)-(3) in the design of underwater eco-sensing networks; that is, the nodes are assumed stationary. For the design of terrestrial, energy-limited, mobile sensor networks, we focus on all properties (1)-(4) with special emphasis placed on mobility.

It is important to note that since the data traffic is delay-tolerant; that is, it is not the maximization of the data rate that is important but rather the reliable delivery of small amounts of sensor data under severe energy but rather loose delay constraints. This paradigm is very different from the majority of the literature that focuses on maximizing the data rate. Hence, some very novel features that are very relevant to sensor networks will emerge in the analysis.

In this thesis, we address three of the major challenges that arise in delay-tolerant, energy-limited sensor networks. The first challenge is the design of medium access control (MAC) protocols in underwater networks where the propagation delays are large and unknown. This key link-layer property of large delay presents significant challenges in network synchronization, and the traditional methods of tight synchronization as in RF networks are infeasible. In Chapter 2, we present the full design of a MAC protocol that solves this challenge of unknown, large propagation delays.
The second challenge that this thesis addresses is the challenge of mobility in energy-limited, delay-tolerant networks. Traditional methods that address routing and energy optimization in mobile networks declare such networks to be intractable above a certain amount of mobility that is determined by individual node velocity. That is, it is commonly believed that the larger the individual node velocities, the less tractable these networks become. This thesis shows that this is not the case. In particular, in Chapter 3 of this thesis, we show how to separate the components of node mobility from the QoS metrics, such as end-to-end energy consumption, between physical locations in space. We propose “energy maps”, which are maps of the end-to-end energy metrics between physical locations in space. This approach is a very novel approach for mobile networks. For example, traditional mobile routing protocols presume that the path from node $A$ to node $B$ is gone if any of these nodes in the path changes: Proactive routing protocols would require table updates as often as such changes in the actual paths occur, and reactive routing protocols would initiate node discovery (or use node repair) under such a change of even a single node in the path. The notion that we use, which has also appeared in trajectory-based routing in the recent literature [17], is to conceive of network paths not as sequences of nodes but rather as “paths through space”. We show that the end-to-end energy consumption between physical locations in space is roughly invariant, for example, for the Random Waypoint Model, even though the nodes in between might be changing locations. This information is very valuable in designing routing and energy optimization and planning protocols in high-mobility networks, if the information on energy consumption between physical locations in these networks
can be disseminated fast enough to all the nodes that are likely to use these metrics in their decisions. We propose novel dissemination algorithms and show how the information on energy metrics can be disseminated very fast in these networks and thus made available for use.

The energy metrics between physical location through real networks is time-varying. In the second half of Chapter 3, we address the time-varying energy metrics and show how they can be handled. We show that even when the energy metrics between physical locations are time-varying, they can be disseminated to the nodes and put to use. The usefulness of this becomes more apparent in the context of energy planning and optimization in high-mobility networks.

The third challenge is the problem of energy planning and optimization through high-mobility networks. In the past, this problem has also been deemed infeasible due to the high amount of node mobility that was believed to limit the performance of protocols based on QoS. However, with the tools of energy maps developed in Chapter 3, we show that the energy optimization and planning problem becomes feasible in high-mobility networks. This is due to two reasons: first, the energy maps allow the energy metrics to be distributed fast to the other nodes, and these maps can be quite accurate, as we show in Chapter 3, when the energy metrics are invariant. When the energy map is time-varying, it turns out that even though the amount of time variation of the metric (which is separate from node velocity) places limits on what can be tracked on a location-by-location basis, from the perspective of energy optimization over a high-mobility network spread over a terrain, the ordering of the physical locations in terms of their energy metric characteristics remains roughly invariant; that is, when the energy
map is shifting and forming cusps to the time variations (related to contraction, expansion or circulation of node densities), the distortions in the energy map (e.g. when viewed in as a three-dimensional graph) are small when the map is viewed as a geometric surface. For this reason, the energy optimization over the terrain using this geometric surface is still feasible, as will be demonstrated in Chapter 4. This result is very surprising since it would appear that when the energy potential is time-varying, serious limitations should appear on energy optimization problems. However, we show that this is not the case, and the energy values that are collected from the energy maps (in Chapter 3) give accurate enough estimates to lead to a small performance loss compared with the situation where global information on the energy values is available.

The rest of this thesis is organized as follows: In Chapter 2, we describe our medium access control protocol for the battery-limited devices in the delay-tolerant applications such as in underwater environments. The proposed protocol not only operates despite the very long, unknown propagation delays of the underlying medium but is also decentralized and scalable. We also discuss the energy efficiency of our protocol in detail in the simulation results of the second chapter.

In Chapter 3, we develop a novel, localized, scalable framework for constructing and sharing the energy maps over the deployment region. This enables energy planning and optimization for large scale, high-mobility wireless networks. In the first half part of Chapter 3, we study the spatial, time-invariant, end-to-end energy characteristics of highly mobile networks and show how to construct, share, and refine the energy maps. In the second part, we examine a more realistic network scenario that addresses the time-varying energy potentials, and we apply our
algorithms to construct the energy maps under time-varying energy potentials.

In Chapter 4, given that a node can obtain the energy maps of the deployment region via the algorithms developed in Chapter 3, we show how the node can transmit its traffic more efficiently in the energy-limited regime subject to the time deadline imposed by applications. We propose an optimal traffic allocation algorithm based on dynamic programming and show its energy savings over a competing allocation scheme.

Finally, in Chapter 5, we summarize this thesis and discuss some of the broad impacts of this thesis: First, we describe how our MAC protocol can serve as a primer for the development of energy-efficient MAC protocols for future underwater sensor networks. Second, we explain how our framework can be extended to the construction of end-to-end delay maps which enable delay-aware planning, and further develop an analogy between the quality of a wireless point-to-point link and end-to-end network QoS metrics.
Chapter 2


In this chapter, we propose a distributed, scalable, energy-efficient MAC protocol that works despite long, unknown propagation delays of the underwater acoustic medium. This protocol can be used for delay-tolerant applications such as underwater ecological sensor networks between energy-limited nodes. Our protocol differs significantly from ALOHA, MACA, and MACAW protocols in that energy is the main performance metric in our case rather than bandwidth utilization. We show that under a realistic underwater sensor network scenario, our MAC protocol wastes only 3 percent of the transmit energy due to collisions, when the average number of neighbors is 4, and the duty cycle is 0.004. This distributed, scalable MAC protocol has the potential to serve as a primer for the development
of energy-efficient MAC protocols for future underwater sensor networks.

2.1 Introduction

The development of MAC protocols for Underwater Wireless Acoustic Networks (UWANs) is challenging due to energy limitations, long propagation delays, low data rates, and difficulty of synchronization in underwater environments [18][19][20][21][22]. Our aim in this chapter is to develop an energy-efficient MAC protocol that will operate under large propagation delays in underwater ecological sensing and monitoring applications. We concentrate on a dense network of hundreds of sensor nodes with a small node spacing (e.g. up to 100 meters). Such short distances extend battery life by using low-power transmissions [23][24]. In addition, we exploit the idea of “sleep mode” [23][25] to save the nodes’ energy via relatively low duty cycles. In this chapter, we focus on delay-tolerant data such as conductivity, temperature and depth (CD) measurements, which are the fundamental measurements for oceanography, limnology, and marine science applications [26].

The recent design of energy-efficient MAC protocols concentrated on terrestrial sensor networks [25][27][28][29][30][31] and the techniques that have been developed are not suitable for the challenging underwater acoustic communication medium [32] that experiences very large propagation delays (of 1 second over 1.5 km). In past acoustic network deployments, FDMA was used (e.g. in the 1998-1999 SeaWeb [33] and in [34]), but was found to be restrictive and inefficient in terms of bandwidth utilization. SeaWeb 2000 [21] favored a CSMA/CA
solution with RTS/CTS exchange; however, the problem with the handshaking
based protocols [35][36] [37][38][39] is that they exacerbate the end-to-end delays
that are incurred, especially in underwater sensor networks with a large number
of nodes.

A recent underwater networking solution [40] proposed by Xie and Gibson,
seeks to achieve lower and more predictable end-to-end delays in the network by
use of a base station that computes routes for all underwater sensors. This obvi-
ates the RTS/CTS exchange and significantly reduces both the propagation delay
and jitter along each route; however, there are issues regarding the scalability of
this centralized solution to a large number of nodes. The topology establishment
phase uses a clever design for the distribution of CDMA pseudo-noise sequences
to a tree of nodes in the network; however, the near-far problem [41][42][43] that
would arise from the use of CDMA codes in such a network has not been ad-
dressed.

In the recent literature for terrestrial sensor networks, PEDAMACS [28] is an
energy-efficient MAC protocol in which the nodes are synchronized by a common
base station. PEDAMACS uses TDMA for access, and due to its centralized
structure, achieves high utilization of the bandwidth as well as energy efficiency.
Self-configuring protocols that use a distributed TDMA scheme were proposed
in [29]. The S-MAC protocol [27] improves upon the existing 802.11 DCF. Re-
cent papers [25][30] improve upon S-MAC by proposing fast path schemes that
leverage upper-level topology information to allow fast-routing, and bypass the
delay associated with the length of the sleep-listen cycle. The problem with the
application of all of these terrestrial protocols to underwater acoustic networks
is their assumption that the propagation delay between the nodes is very small. Further, the recent improvements [25][30] to S-MAC use a scheme that converges to a global schedule for the entire network, which would be untenable for an underwater acoustic network. In contrast, our protocol does not require global schedules.

The rest of this chapter is organized as follows: In Section 2.2, we describe our energy-efficient MAC protocol. In Section 2.3, we address the channel variations that may occur due to mobility or platform motion and propose enhancements to the basic protocol. In Section 2.4, we quantify the fraction of energy wasted due to collisions at the MAC layer, and exhibit network self-reconfiguration upon the deployment of new nodes into an existing network. In Section 2.5, we conclude with a summary of this chapter.

2.2 Proposed MAC Protocol for UWANs

2.2.1 Basic Idea

The basic idea of the protocol is illustrated in Figure 2.1, which explains how to achieve a locally synchronized schedule even in the presence of long, unknown propagation delays.

* Determination of “listen” cycles : In Figure 2.1, node A broadcasts a SYNC signal (the shaded rectangle in the figure) at the beginning of its cycle period, and then goes to sleep by turning off its transceiver circuits to save energy. This SYNC signal announces node A’s transmission cycle period “$T_A$”. Assume that node B is located near node A. When node B joins the network, it first listens
to the channel for this SYNC signal to achieve frame synchronization with node A. (The white rectangles in the figure indicate node B’s receptions of node A’s SYNC signals.) After achieving frame synchronization with node A, node B decodes the length of the transmission cycle period ($T_A$) from the message. This *explicit stamping of the transmission cycle period* (rather than any absolute wake-up time) allows node B to wake up at exactly the correct time in the next cycle to listen to node A without any knowledge of the value of the propagation delay, as long as the propagation delay remains fixed from one cycle to the next (and the clock drift is not significant in one cycle). This localized protocol holds for any pair of nodes ($A$, $B$). Further, this scheduling algorithm does not require any adjustments to the nodes’ clocks since absolute timing information is obviated.

In this protocol, a collision may occur in two ways: first, a “transmit-receive collision” may occur in which a node is transmitting while other nodes’ packets
arrive at its receiver and collide with the node’s own transmission. Since the transmit power is usually much larger than the power required for successful packet reception, the received packet would not be decoded properly at the node in this case. Second, a “receive-receive collision” may occur if more than two data packets arrive at a node and overlap in duration, in which case the node cannot decode either of the data packets.

**Determination of transmit times**: The topology control layer keeps track of the neighbors of a node for whose transmissions a node needs to wake up at the proper time. Even though the listen times are predetermined using the above scheme, the initial transmission time is selected at random and independently by each node. However, once a node chooses a certain transmission start time, it sticks to its schedule by transmitting its data at that time again in the next cycle. Since the nodes have clocks that are at random offsets from each other, if the cycle period is much longer than the transmit duration, then the probability of collisions will be small. (This will be quantitatively demonstrated in Section 2.4.)

### 2.2.2 Initialization of the MAC Protocol

We use the network in Figure 2.2 as an example network configuration throughout this chapter. In this figure, each node sets its transmission range by using localized topology control [23], and each circle indicates the transmission range of the node at its center\(^1\). For this given network topology, the broadcasting of control packets in the initialization period is illustrated in Figure 2.3. In this figure,\(^\text{1}\)

---

\(^1\)The transmission ranges are circular in this diagram for illustration purposes to indicate neighbor relationships. In reality, these will not be circular, and the proposed MAC protocol does not depend on the circularity of these ranges.
Figure 2.2: An example network configuration.

every node’s time axis is shown separately. A node’s own transmission is shown as shaded, and the receptions of that transmission at the other nodes are shown as white rectangles with a label on top of the white rectangle that indicates from which node that packet was received. In the initialization period, for the distribution of the transmission schedules among the nodes, each node \( i \) broadcasts its SYNC packet, and remains awake until the beginning of the next cycle in order to receive its neighbor nodes’ SYNC packets. The SYNC packet contains the node’s transmission cycle period \( T_i \) which tells its neighbors that it will transmit data again after this time period. The cycle period \( T_i \) is initially fixed to the same value \( T_o \) for all nodes in order for them to initialize their transmission/listen schedules.

In Figure 2.3, every node selects its transmission start time at random in the interval \((0, T_o)\) and broadcasts its SYNC packet to its neighbors. Because node \( A \)’s start time happens to be the first, we see in this figure that node \( A \) broadcasts its SYNC packet first. Then, node \( A \)’s neighbors (nodes \( B, C, \) and \( D \)) receive
node A’s SYNC packet during initialization which allows them to schedule their wake-up times (which will take place after the initialization period). Node A will also schedule particular wake-up times as shown in this figure for its neighbor nodes B, C and D after it hears their schedules during this initialization period. It should be noted that the ratio of transmission duration $\tau$ to the period $T_o$ (namely, the “duty cycle”) is in fact very low, (e.g., $\tau/T_o = 0.004$), and the packets in this figure have not been drawn to scale.

Each node maintains its neighbor table in the topology control layer. In our protocol, each node runs its initialization periodically (e.g. every 10 cycles). In addition to this periodic initialization, each node can go into the initialization phase on demand. For instance, assume that the topology control layer requires...
that at least 3 neighbors be maintained per node for network connectivity. If the number of its currently synchronized neighbors is less than 3, then the node goes into the initialization period with an increased transmit power, and gathers its neighbors’ transmission schedules again.

\subsection{2.2.3 Data Transmission After Initialization}

After the initialization phase is complete, each node knows when it needs to wake up again to receive data from its neighbors. After this initialization, nodes follow their established schedules and begin sending data. Figure 2.4 shows the structure of the data transmission packet and the listen period to listen to any potential newcomers during the data transmission phase. The transmit duration has been shaded and is followed by a listen duration. The transmission duration has three distinct parts: “Missing”, “SYNC” and “Data Tx”. The “Data Tx” corresponds to the part where actual data is being sent. We now explain the first two control functions: In our protocol, each node $i$ compares the neighbor list with the list of nodes from which node $i$ has successfully received signals. After this comparison, node $i$ generates the “Missing node list” and sends the list of “Missing” neighbor nodes in the header of the data packet in its next cycle.
During regular operation, every node keeps sending in its SYNC header a cycle period stamp, which may be different from the one ($T_o$ seconds) in the initialization period. By using the SYNC message in the header, a node has the option to change its current cycle period, and its neighbors can decode the modified SYNC message and change their wake-up times for that node. Figure 2.5 shows this modification of a cycle period during the data transmission phase. In this figure, node $A$ changes its next cycle period from $T_o$ to $T_A$, and node $B$ decodes the modified SYNC message and wakes up to listen to node $A$ after the new cycle period $T_A$. If a node $B$ loses contact with node $A$ during this modification, it will recover node $A$ as a neighbor via the Missing neighbor list. This recovery procedure will be detailed in Section 2.2.5.

We now explain the Listen duration shown in Figure 2.4. After the transmission of data, a node does not go to sleep right away but rather goes into an idle listening mode. In the listening mode, the node is still awake but operates at low-power. If it hears something, it will go into the receive mode. This additional
listen duration is used to hear newcomers and improves robustness. A scenario in which a new node joins will be discussed in the next section in detail. The length of the listen duration is chosen as a practical parameter which involves a trade-off: a very long duration decreases the energy efficiency of the protocol (due to idle period energy consumption), but a very short duration might not be enough to catch some of the newcomers’ messages.

2.2.4 Handling Newcomers

We use the topology shown in Figure 2.2 to show how the protocol handles newcomers. In this figure, node $F$ joins the network, and nodes $B$ and $E$ become its neighbors. Figure 2.6 shows the packet transmissions that occur in this scenario. After node $F$ joins the network, whenever it hears from another node, it sends a HELLO packet back to that neighbor to inform the neighbor of its transmission schedule. In Figure 2.6, since node $F$ happens to receive data from node $B$ first, it sends a HELLO packet back to node $B$ right after its data reception is complete. (Since each node puts an additional listen duration after each transmission as explained in Section 2.2.3, the existing node can hear from the newcomer’s HELLO message during this duration.) This HELLO packet from $F$ contains a time stamp $\Delta_B$, which is defined as the number of seconds from the beginning of the HELLO packet to the beginning of the scheduled SYNC packet of node $F$. By the same argument as in the basic idea of the protocol (in Section 2.2.1), this mechanism allows node $B$ to wake up in order to listen to node $F$ exactly $\Delta_B$ seconds after it receives node $F$’s HELLO packet. After this, in the figure, at its selected transmission start time, node $F$ broadcasts its
SYNC packet to achieve initialization with its neighbors $B$ and $E$. When node $B$ decodes this SYNC packet of node $F$, it schedules its wake-up time to receive future data transmissions from node $F$. In order to acknowledge the successful reception of this HELLO packet, node $B$ sends back an ACK to node $F$ at $B$’s next data transmission time, and node $F$ ensures its synchronization with node $B$ by keeping sending its HELLO until it hears the ACK from node $B$. A similar procedure is repeated for neighbor node $E$ in the figure. At the end, node $F$’s data transmission begins at its scheduled time, namely $T_o$ seconds after its SYNC packet.
2.2.5 Handling Node and Synchronization Failures

A node may not receive data at a scheduled wake-up time due to a bad channel condition from the sender, or due to a sender node failure. Whenever the receiver node does not hear from the sender at a scheduled time, the receiver puts the sender in its Missing node list explained in Section 2.2.3. For instance, in Figure 2.7(a), assume that a data delivery from node $i$ to node $j$ has failed in cycle $t_m$ of node $i$. Then, node $j$ will announce via its Missing header that it has not heard from node $i$. In case the delivery failed due to a bad link state from node $i$ to node $j$, node $i$ will decode the Missing header from node $j$, and discover its own delivery failure or possible loss of synchronization with node $j$. Then, node $i$ sends a HELLO message to node $j$ (as if node $i$ were a new neighbor of node $j$). Then, node $j$ (since it is in Listen mode after its data transmission) will hear this HELLO message from node $i$, and will re-schedule its wake-up time for the next cycle.

In Figure 2.7(b), when node $i$ fails (e.g. node $i$ runs out of energy), node $j$ will announce $i$ in its Missing list. However, since node $i$ is no longer available in the network, node $j$ will not be able to hear back from node $i$ in cycle $t_{m+1}$. Then, node $j$ will remove node $i$’s wake-up time from its table of wake-up times, starting at cycle $t_{m+2}$. In other words, in this figure, if a node does not hear from a neighbor for two consecutive cycles of the neighbor, it deletes its wake-up schedule for the neighbor to save energy. In general, node $j$ can count the number of cycles for which no transmissions from node $i$ have been heard, and this number of cycles may be set as a configuration parameter to trade off energy consumption versus protocol robustness.
Figure 2.7: (a) Handling loss of synchronization, (b) Handling a node failure.
2.3 Addressing Channel Variations

So far, we have assumed that the propagation delay remains the same between cycles, but the propagation delay in reality varies due to channel fluctuations caused by the relative motion of the transmitter, receiver, or significant scattering surfaces, and platform motion. In this section, we model propagation delays in the network as random variables. However, arbitrarily large, random propagation delays are difficult to address in design, and do not occur in practice. Hence, we will also assume that the nodes also know the value (or the order of magnitude) of the maximum propagation delay between nodes in the network. For example, for densely deployed underwater networks, the maximum propagation delay between nodes is about 70 ms for distances of up to 100 meters. The basic protocol that we presented in the previous section is robust in the sense that it works without any knowledge of the propagation delays, and this section may be viewed as an enhancement of the basic protocol to address more realistic channel models. When we discuss simulation results in the next section, we will report the results of the protocol both with and without these additional assumptions on the channel model.

We now describe the improvements that we propose that use this more realistic channel model. First, in order to handle random propagation delays, we let each node place listen durations: $\tau_e$ for the early arrival, and $\tau_l$ for the late arrival of data from a neighbor. For example, in Figure 2.8, we show the Listen mode duration of node $B$ for any existing neighbor $A$. Node $B$ can catch node $A$’s data, even when the propagation delay from node $A$ to node $B$ does not remain exactly
the same as in the previous cycle. In this figure, although node A’s data in the first cycle arrives earlier and its data in the second cycle arrives later than the scheduled wake-up time of node B for node A, node B can still receive them.

Second, by exploiting the maximum propagation delay information, we present an improvement of our protocol in order to avoid receive-receive collisions: Each node $i$ places a certain “guard time” on both sides of its transmission duration. This guard time is different from the guard times employed in TDMA systems, which are allocated by the base station depending on the multipath channel conditions. The guard times in this section are chosen by each node in a completely localized manner in order to reduce the collision rate. For example, Part (a) of Figure 2.9 shows an example of a receive-receive collision in the presence of propagation delays. In order to avoid this collision, if $\tau_1$ or $\tau_2$ is less than a predefined guard time $\tau_g$, then node A or B re-selects its transmission start times satisfying that its transmission duration ($\tau$) plus $2\tau_g$ is not overlapping with any other wake-up times. Part (b) of this figure suggests one possible solution for the case of $\tau_1 < \tau_g < \tau_2$. No collision occurs as node B’s newly selected transmission start
(a) A receive-receive collision due to the propagation delays

(b) A guard time solution for the case: $\tau_2 < \tau_g < \tau_1$

Figure 2.9: A scenario of a receive-receive collision with propagation delays and a solution of the collision using the guard time policy.

We now build an explicit algorithm to carry out this guard time policy. Let node $i$ follow its current transmission schedule, which consists of the transmission start time and the cycle period. Then, if the duration between a node’s selected transmission start time (or end time) and its wake-up times is less than the guard time, then the node reselects its new data transmission time which does not
overlap with the existing wake-up time schedules for its neighbor nodes. Stated precisely, let $X_i$ denote node $i$’s current transmission start time (which can be fixed as 0); $T_i$ node $i$’s current cycle period; $W_{i,k}$ the wake-up time for node $i$’s neighbor node $k$; $\tau$ the data transmission duration, $\tau_g$ the guard time duration; and $X'_i$ node $i$’s new transmission start time. Each node sets its new schedule to satisfy the following: If $X_i - W_{i,k} < \tau_g$ or $W_{i,k} - (X_i + \tau) < \tau_g \forall k$, then select $X'_i \in (X_i, X_i + T_i)$ subject to: $X'_i - W_{i,k} \geq \tau_g$ and $W_{i,k} - (X'_i + \tau) \geq \tau_g$ for every $k$.

The simplest choice of the guard time duration is twice the maximum propagation delay, which can be the same for all nodes. (More optimized schemes are also possible when more information on each link is available.) A node announces this new transmission schedule to its neighbors using the SYNC header, and follows the new schedule starting with the following cycle.

### 2.4 Simulation Results

As our performance metric, we define the fraction of energy wasted due to collisions as the energy wasted on collisions divided by the total energy used. When two packets collide, we say that the two nodes failed in their transmissions and that they wasted their transmission energies. Even though the same packets may have been successfully delivered to the other neighbors, we still assume that the two nodes wasted their energies. This provides us with an upper bound on the fraction of energy wasted. Below, we examine two cases: the case with no propagation delays and the case with long propagation delays, and quantify the
2.4.1 No Propagation Delays

In this subsection, we assume that all nodes share the same medium so that all of them can hear from each other. (In Section 2.4.3, we remove this assumption under a realistic network scenario.) In the absence of propagation delays, we compute the probability of that a collision occurs. Each node generates its transmission start time uniformly in the duration \((0, T)\) in an i.i.d fashion, and its data transmission duration is \(\tau\). Fig 2.10 shows the two cases in which two nodes’ transmissions collide in the duration \((0, T)\). Taking node A’s transmission as a fixed reference in this figure, we see that node B’s transmission in the first cycle collides with A’s either with A’s transmission in the first cycle shown, or with A’s transmission in the next cycle. Hence, the probability that node A’s transmission collides with the node B’s is \(\frac{2\tau}{T}\), since the transmission start times are uniformly distributed. Thus, the probability that a node’s transmission collides with at least
another node’s transmission over 1 cycle is as follows:

$$
P[A \text{ node collides with at least one other}] = 1 - P[A \text{ node does not collide with any other node}] = 1 - \left( 1 - \frac{2 \tau}{T} \right)^{N-1}, \quad (2.1)$$

where $N$ is the number of nodes in this network. Assuming that each node uses the same energy for transmission and that each transmission is independent of other transmissions, the expected value of the total energy wasted due to collisions is $N \left( 1 - \left( 1 - \frac{2 \tau}{T} \right)^{N-1} \right) E$, where $E$ is the value of the energy for each transmission, and the total value of the energy used for all nodes is $NE$. Thus, the fraction of energy wasted due to collisions is $\left( 1 - \left( 1 - \frac{2 \tau}{T} \right)^{N-1} \right)$.

The simulation results displayed in Figure 2.11 under the scenario with no propagation delay are very close to the analytical result above. In this simulation, every node is a neighbor of every other node in the network, and the result is averaged over 10000 runs. As the number of nodes increases from 2 to 10 nodes, the fraction of energy wasted due to collisions is shown for different values of the duty cycle $\tau/T$. For the data transmission duration, $\tau = 200$ ms is used for all simulations throughout this section. In this regime of network operation with a low duty cycle, writing down the Taylor expansion of (2.1) for low duty cycles shows that the fraction of energy wasted is a linear function of the number of nodes, as can also be seen in this plot. For 6 nodes, which is a typical number of nodes for a node if topology control were applied, the figure shows that 96% of the energy is spent in successful transmission for a duty cycle $\tau/T = 0.004$. 

28
2.4.2 Long Propagation Delays

Now, we show the impact of long propagation delays on the energy efficiency of our protocol. In the simulations of this section, propagation delays between any two nodes are independent and uniformly distributed between 5 ms and 75 ms, corresponding to internodal distances up to 100 meters, assuming a propagation delay of 0.67 ms per meter.

Since there was no delay between any two nodes in Section 2.4.1, the transmissions and receptions could be represented on the same (global) time axis, and transmit-receive collisions and the receive-receive collisions by the hidden terminal

Figure 2.11: Fraction of energy wasted due to collisions when ignoring propagation delay.
Figure 2.12: Fraction of energy wasted due to collisions with propagation delays.

A different kind of receive-receive packet collisions could be added in the presence of propagation delays (under our protocol). An example of this collision is illustrated in Part (a) of Figure 2.9. Due to the different propagation delays on the links, two packets transmitted from node A and B might collide at node C even though neither of the two transmissions collide at nodes A and B. Thus, propagation delays worsen the energy efficiency of the protocol over the case without propagation delays. The resulting Figure 2.12 displays this impact of long propagation delays on the energy efficiency of our protocol as the number of nodes grows. There is as much as 20% loss in energy efficiency due to propagation delays (for the duty cycles considered).
2.4.3 A Realistic Network Scenario

So far in the simulations, we have assumed that all nodes are neighbors of each other and every node could reach another. However, in a large-scale network, topology control limits the transmission range of each node as in Fig 2.2 in order to improve performance. In this section, we assume that the topology of the network (channel gain matrix) is known for all nodes and that each node determines its neighbors using localized neighbor search.

In the simulation set-up, 20 – 30 nodes are randomly deployed over 500 by 500 $m^2$ area (for 25 nodes, roughly 100 meter apart). In order to compute the channel gain matrix, we fix the reference received power. Then, the channel gain is equivalent to the required minimum transmission power. We used the energy consumption model from [22] in which the transmit energy consumption model is given as

$$E = P_0 T_p A(r) = P_0 T_p r^k a^r,$$  \hspace{1cm} (2.2)

where $E$ denotes the transmit energy, $P_0$ the reference received energy required to decode the received packet, $T_p$ the packet duration, $r$ the distance from the source, and $k$ the spreading factor which is 1 for cylindrical, 1.5 for practical, and 2 for spherical spreading. In the above expression, $a = 10^{<\alpha(f)>>}$ is the frequency-dependent term obtained from the absorption coefficient $\alpha(f)$. By Thorp’s expression [44] in [dB/km],

$$\alpha(f) = \frac{0.11 \cdot f^2}{1 + f^2} + \frac{44 \cdot f^2}{4100 + f^2} + 2.75 \cdot 10^{-4} f^2 + 0.03,$$  \hspace{1cm} (2.3)

where $f$ is in kHz. For our simulations, we chose the parameters as $f = 25$ kHz and $k = 1.5$.
Each node maintains its neighbor table corresponding to the list of neighbors within its transmission range. This list is used to schedule the receiver node to wake up to listen to its neighbors. Assume that the maximum transmission power is the same for all nodes and that the generation of propagation delay is the same as in Section 2.4.2. We set a maximum transmission power to enable a node to have 4 to 6 neighbors on average in the simulations. Each node generates its transmission start time uniformly in the duration \((0, T)\) in an i.i.d fashion. For the control packet durations, we set \(\tau_s = 0.1\tau\) for the SYNC packets and \(\tau_h = 0.1\tau\) for the HELLO packets, where \(\tau\) is the data transmission duration. The average duty cycle \(\tau/T\) is 0.004. Each transmit energy for different types of packets is proportional to the corresponding packet length.

Figure 2.13 shows the fraction of transmit energy wasted due to collisions per cycle. In this figure, every node is in its initialization phase in cycle 1. In this initialization, each node broadcasts only SYNC packets to its neighbor nodes within its transmission range. We see that the fraction of energy wasted is not significant because the duration of control packets is relatively short compared to the data packets. Due to the longer packet durations for data transmissions, the fraction of the energy wasted becomes much larger in cycle 2.

As an enhancement of our protocol, nodes refine the transmission/listen schedules using a guard time, as explained in Section 2.3. A node modifies its transmission schedule if the time difference between its own transmission and its receptions from the neighbors is less than the guard time. Now, in cycle 3 of Figure 2.13, energy waste due to collisions is significantly reduced by the guard time policy. In order to show the overall fraction of energy wasted due to collisions in our
Sleep Cycle: $T$, Tx duration: $\tau = 200$ ms, $\tau/T = 0.004$

Figure 2.13: Fraction of energy wasted due to collisions under a realistic network scenario.
protocol, we plot the average of the fraction over 10 cycles in Figure 2.13. We see that our MAC protocol wastes less than 3% of the transmit energy due to collisions when the average number of neighbors is 4.

2.4.4 Self-configuration of the Protocol

In order to test the robustness of our protocol, we change the network topology by deploying new nodes in the existing network and show how our protocol adapts to change. The result is shown in Figure 2.14.

In this simulation, a network operates with $N_o$ nodes (in Figure 2.14, $N_o \in \{15, 20, 25\}$) in the beginning. In the first cycle, the nodes initialize their transmission cycles and synchronize the schedules with their neighbors, and then refine the transmission/listen schedules by using a guard time. After initialization, a node modifies its transmission schedule if the time difference between its own transmission and its receptions from the neighbors is less than the guard time. As shown in Figure 2.14, the number of collisions is significantly reduced in cycle 3 after adopting this guard time policy.

After cycle 3, we add 5 new nodes to the network. These new nodes send the HELLO packets whenever they hear from the existing nodes, and they also send their SYNC packets for initializing the transmission/wake-up schedules. Regardless of this change, the existing nodes keep sending their DATA packets at the scheduled transmission times. However, because of this addition of control packets (HELLO and SYNC packets) of new nodes, more collisions occur in cycle 4. Therefore, the fraction of energy wasted due to collisions increases. In cycle 5, the new nodes start sending their own DATA. The existing nodes apply their
Figure 2.14: Fraction of energy wasted due to collisions when 5 new nodes are added.
guard times with the newly issued schedules for the newcomers, and this reduces the fraction of the energy due to collisions. From cycle 6 onwards, all nodes have refined their transmission schedules with guard times, and our protocol eventually gives the similar (or slightly better) performance over the original network for $N = 20, 25, 30$ nodes (which was shown in Figure 2.13). Hence, the protocol is self-configuring in the sense that it converges a high energy efficiency even after the addition of new nodes.

### 2.5 Summary

In this chapter, we have proposed an energy-efficient MAC protocol for underwater eco-sensing networks. Most energy-efficient MAC protocols that are currently available for terrestrial sensor networks would not perform well due to the adverse communication environment caused by the long, unknown propagation delays. This propagation delay when not accounted for, would cause a network to lose the coordination of transmission schedules among its nodes. In our proposed protocol, the use of relative time stamps and the sleep mode of sensor units allow the nodes to operate in a synchronized environment. Our protocol not only operates despite the very long, unknown propagation delays of the underlying medium, but is also distributed and scalable. Further, our MAC protocol is also robust in that it adapts to the changes of a network such as channel variations, new node deployment, loss of synchronization among the nodes, and node failures. The simulation results show that, under a realistic underwater sensor network scenario, our protocol wasted only 3% of the transmit energy due to
collisions, when the average number of neighbors is 4, and the duty cycle \( (\tau/T) \) is 0.004, where \( \tau = 200 \) ms.
Chapter 3

Energy Maps for Large-scale, Mobile Networks

In the second chapter of this thesis, we focused on underwater ecological sensing which is one of the emerging applications for delay-tolerant networks. The nodes in these networks are assumed to be stationary, which is very common in current underwater network deployments. When we turn our interest to energy-limited, terrestrial sensor networks as another application of delay-tolerant networking, we find that they are still large-size networks (in terms of the number of nodes), but the nodes are no longer stationary in most wireless environments. In these networks, energy-aware routing constitutes one of the most significant challenges. For high-mobility networks, this problem has been deemed infeasible due to the high amount of node mobility that was believed to limit the performance of protocols. A key problem in high-mobility, large-scale networks is that the paths quickly become obsolete, rendering end-to-end energy-based routing decisions very difficult.
In the rest of this thesis, we seek to find methodologies for handling high mobility in order to manage the limited node energy supplies in large-scale, mobile wireless networks. Our main plot is as follows: First, in this chapter, we establish a novel framework and methodology to make end-to-end information on energy metrics available in high-mobility networks. Second, in Chapter 4, we show that, with the information on energy metrics, the energy optimization and planning problem becomes feasible in high-mobility networks, and we present quantitative results of the performance of energy planning schemes that employ the energy maps developed in this chapter.

In this chapter, we propose a novel framework to share, retain and refine end-to-end energy metrics in the joint memory of the nodes, over time scales over which this information can be spread to the network and utilized to make energy-aware decisions. We construct maps of end-to-end energy metrics for high-mobility networks by exploiting the underlying stationary distributions in node density that exist over relevant time scales. In particular, we show how to (1) construct the energy potential function over space, (2) compute the spatial derivatives of these potentials in high-mobility networks, (3) allow the nodes to construct these energy maps on-demand via path integration methods, and (4) distribute, share, fuse, and refine the maps of end-to-end energy metrics over time by information exchange during encounters. The methodology and results are developed to enable energy-aware routing and energy optimization in high-mobility networks.
3.1 Introduction

Energy-aware routing is one of the most significant challenges for mobile wireless sensor networks. A key problem in high-mobility, large-scale networks is that the paths quickly become obsolete, rendering end-to-end energy-based routing decisions very difficult. Localized routing decisions (such as in geographic routing) are efficient; however, they do not rely on end-to-end energy metrics. Localized routing protocols that work well for stationary sensor networks (e.g. directed diffusion [45], and GLIDER [46]) are currently difficult to adapt to high-mobility networks because the end-to-end information that they rely on would become obsolete, due to the current lack of methods to enable QoS routing in highly mobile networks. Our main goal is to establish a novel framework and methodology to make end-to-end information on energy metrics available in high-mobility networks.

Our main strategy is to keep track of spatial, end-to-end energy metrics between fixed physical locations over time. Two factors determine whether the spatial, end-to-end energy metrics can be utilized: first, the collected information must remain valid for the duration that it takes for a packet to reach the destination, using the path with the associated energy metric. Second, the changes in patterns must be spread fast enough through the network such that the energy information becomes available at the nodes to make routing decisions.

Our first insight is that even though the network might be highly mobile, and the nodes between two physical locations in space might change at a very high rate, the end-to-end energy metric between two physical points in space might still
be roughly constant. The reason is that if the main goal is to transmit information within a certain total energy consumption, the relay nodes in between that carry the information are *interchangeable*. Hence, in this chapter, paths are conceived of not as fixed sequences of nodes, but rather as “paths through space”. This view has already been utilized in trajectory-based routing [17], where a physical trajectory is embedded in the packet to be routed. However, there, no claims have been made as to whether the end-to-end metrics may stay roughly constant over time between two physical end points, and this is the contribution that our work makes in that regard. That is, even though the network might be highly mobile, it is possible (as we shall show) for end-to-end metrics to be roughly constant. This view reduces a considerable amount of uncertainty that appears when one focuses on the motions of individual nodes; however, surprising stationarity may be present when we graph the end-to-end energy metrics for these high-mobility networks.

Our second insight is the relationship between how fast information can be spread through a high-mobility network (via exchanges upon encounters) and how the end-to-end energy information metrics can be refined. Through this chapter, we find that control information can be spread extremely fast in high-mobility networks, and we shall quantify this spread through the “spreading period” that we introduce later in this chapter. Now, we connect this notion of spreading the control information with the notions of end-to-end energy metrics through space: if the end-to-end energy metrics remain roughly invariant through paths in space for durations on the order of the time that is required for the delivery of data, then this energy information can be spread surprisingly fast through the network,
and hence become useful for energy-aware routing protocols.

At first, this might seem surprising and counterintuitive. It might seem that higher mobility would lead to a higher rate of dissemination of energy metrics, but that the energy metrics themselves would be changing at a higher pace as well. However, this is not the case because there are two fundamental aspects of variation that results from mobility. The first aspect is the variation that results from individual node motion, and this aspect is used to spread the energy metric information. The second aspect is the rate at which the end-to-end energy metrics themselves are changing. In the simplest example, for the Random Waypoint (RWP) Model, due to the fact that there is a stationary node density function, the end-to-end transmit energy consumption, when averaged over many measurements, settles down to an invariant function over space (Figure 3.1), if it is sampled by the nodes at a suitable rate. Hence, one can separate the two components of mobility with respect to a given energy metric of interest: the mobility of the individual nodes determines how fast control information can be spread, and a separate “coherence time” of the average energy metric over space determines how fast the energy metric is changing. If the coherence time of the average energy metric is much larger than the spreading period, we have a high-mobility network for which energy-aware routing is possible. (For the RWP model, the coherence time is infinite, since there is a stationary underlying distribution over the deployment region.) In order to make our methods applicable to the case in which the end-to-end energy metrics do not remain invariant over time in Section 3.4, we explore a time-varying network scenario using the contraction model introduced in [47] and show how our methods can be handled in this case.
The rest of the chapter is organized as follows: in Section 3.2, we describe the relationship of this work to the current state of knowledge. Section 3.3 is devoted to the construction of the energy map for time-invariant energy potentials. In Section 3.3.1, we describe our assumptions and the system model. In Section 3.3.2, we discuss the global end-to-end measurements to obtain the energy potential function. In Section 3.3.3, we describe a localized routing strategy that we use to make localized measurements of the spatial derivative of the energy potential. In Section 3.3.4, we develop a path integration method that allows each node to construct the energy potential from these localized measurements. In Section 3.3.5, we develop new methods for the mobile nodes to disseminate the localized measurements efficiently among themselves for fast construction of the energy potential. In Section 3.3.6, we describe how this work can enable energy-aware routing decisions in high-mobility networks, and introduce a system
model which can provide higher accuracy of energy metrics. Section 3.4 is devoted to the application of the framework to time-varying energy potentials. In Section 3.4, we develop further our path integration method for the case of fast-changing end-to-end energy metrics. In Section 3.5, we conclude with a summary of this chapter.

3.2 Relationship to Current State of Knowledge

QoS routing in mobile wireless ad hoc networks has attracted much attention in the past. The QoS issues were discussed extensively in [48]. The QoS metrics (average delay, and average number of hops) were compared by simulations in [49] for on-demand routing, and hierarchical routing methods. In [50], a QoS routing protocol was developed, which can establish QoS routes with reserved bandwidth on a per flow basis in a TDMA network. Distributed QoS routing was developed in [51] to select a path with sufficient resources to satisfy QoS requirements in the presence of imprecise state information. In all of these past approaches, the underlying assumption was that a path was constituted by a fixed sequence of nodes, and if the nodes changed, then the path was no longer there (and a path update was required). In contrast, in our work, we focus on paths through space, and the nodes are taken to be interchangeable, and we aim to associate a QoS metric (the energy metric in our case) with the physical path through space.

Much of the literature on mobility has focused on how to track the nodes in mobile networks in order to estimate node locations [52]. Both the proactive [53][54][55][56][57] and reactive [58][59][60] routing protocols conceive paths as
fixed sequences of nodes. The localized routing protocols such as directed diffusion [45] and gradient routing [46] rely on the establishment of a gradient, but usually assume that the sensors have fixed positions in space for such gradients to be established. Geographic routing [61][62] approaches use the position information to route towards the destination; however, geographic routing by itself usually sidesteps consideration of QoS metrics (such as end-to-end energy consumption and end-to-end delay). Recent work [63] has proposed to combine geographic and energy-aware routing approaches by a weighted combination of the two metrics and demonstrated high performance. In addition, TTDD [64] and SEAD [65] have emerged as two promising approaches to address routing in the presence of sink mobility in large-scale sensor networks.

In [66], Fei and Gerla developed “Smart Forwarding”, which builds a table that contains QoS metrics of the network and uses this pre-computed table to decide the next hop to minimize the cost objective subject to its QoS constraints. The idea of utilizing the pre-computed table is similar to our work, but there are two important differences: (1) In [66], the table is constructed and updated periodically either by link-state information flooding followed by the Dijkstra algorithm, or by the distributed Bellman-Ford algorithm. Between the periodic updates, the QoS metrics might have drastically changed over the time because of node mobility. The update period would be very short for highly mobile networks, and this short period would cause a considerable overhead to construct the table. In contrast, our work does not use the global state information and instead constructs the table fully based on local neighbors. (2) In [66], there is a scalability issue when newcomers join the network and the pre-computed table does not
include the assigned metrics for the newcomers. However, in our method, nodes share the energy map with any other nodes who have been encountered. Hence, any newcomer can obtain the energy map from its neighbors, immediately after it joins the network.

Localized decisions based on the distance metric are used in both geographic routing [67][61][62] and EFR [68]. In distance-based routing methods (for example in greedy distance routing [69], or compass routing [70]), the local decisions are made based on geometric considerations, and in EFR, they are made to follow constructed electric field lines through space in order to achieve load balancing. Such localized methods can still be used in the framework that we develop for localized routing decisions; however, our aim, in contrast, is to show whether and how the network itself can maintain over time, some notion of end-to-end energy metrics. These end-to-end metrics are assumed to be unavailable in current approaches and are believed to be intractable due to high mobility. We aim to show that they can in fact be constructed, and later be utilized by energy-aware routing protocols as an information base through high-mobility networks.

In the last three years, the long-term node position densities over rectangular simulation areas were derived [71][72][73][74] for commonly-used simulation models such as the Random Waypoint Model and others [75][76]. The key result of these papers is that there is an underlying invariant probability density function for the position of nodes over space, which is useful for simulation purposes. Real networks do not admit such simple descriptions; however, recent work [77][47][78] has shown the spatial regularity patterns in mobile networks, which can partially be modelled using gravitational and fluid models. These recent studies and asso-
associated spatial models are different from previous investigations into time-domain
mobility prediction \cite{79,80,81,82,83,84,52} in that the spatial models focus on
the invariant, spatial characteristics of mobile networks as a whole, rather than
attempting to predict the mobility patterns of individual nodes or groups of nodes.
We are motivated by these characterizations of mobility in the spatial domain, and
seek to identify invariants (in our case, the energy consumption) between fixed
locations in space, at least over a time duration during which this information
can be spread to the network and utilized.

The dependence of the construction of energy and delay fields on the position
(location) information appears, at first, to place a significant constraint on these
constructions. However, \cite{61} has shown, for example, that even with limited
destination information (e.g. quadrant routing) or erroneous angular information,
the routing delays show order-wise robustness, and \cite{62} has indicated how a sensor
network may construct a coordinate system of its own without reliance on GPS
information. Hence, these works may indicate the way in the future to construct
the underlying energy potentials without exact position information.

Sensor data compression, aggregation and fusion methods \cite{85,86,87,88,89}
have also made great strides in the last four years. Many of the challenges that
arise in stationary sensor networks that motivate compression are readily resolved
by use of mobility in highly mobile networks, but they are replaced by new chal-
lenges that arise from very short encounter times between the nodes in highly
mobile networks, which place a cap on the amount of data that can shared upon
each encounter. Hence, the need for data compression arises with a different set
of constraints (see Section 3.3.5).
3.3 Construction of the Energy Map for

Time-invariant Energy Potentials

3.3.1 System Description

In this section, we use a specialized scenario where there is a single destination (which we refer to as a “base station”) in the middle of the deployment region, and a high-mobility network around it that moves according to the Random Waypoint Model. We define the “energy potential” as the average amount of total energy per bit incurred to travel from a fixed point in space to the base station, through a mobile network. The average is computed over many measurements taken from the same physical location to the base station (as explained in the next section).

For simplicity, only the transmit energy consumption is modelled, and it depends only on the path loss model given by \( d^q \) where \( d \) stands for the distance over which the transmission is made, and \( q \) is the path loss exponent. On the data channels, we assume that the aim of each node is to continually send small amounts of data to the base station, using the high-mobility network in between. We do not model the data channels, but rather concentrate on the control channels that will enable the information on the amount of energy it takes to reach the base station, to be available to all of the nodes, for all the locations on the deployment region.

We make the following assumptions: (1) the network transfers small amounts of critical data over energy-limited, mobile devices. Hence, the network operates in the energy-limited regime with no data-rate-intensive traffic, (2) the forwarding
operation at each node is fast enough that from the time that a packet arrives at a node to the time that the forwarded packet is put out on an outgoing port, the node moves negligibly. This assumption allows us to construct paths without modelling the physical transport of the packet with the node at which it is waiting to be forwarded.

Throughout this section, we shall refer to results to illustrate the main ideas that we develop. The results we refer to were all generated using the following simulation set-up: 100 mobile nodes are deployed on a square deployment region that is 1000 meters on each side. Each node moves according to the Random Waypoint Model; however, with a fixed velocity $v = 10$ m/s. The pause times are zero. The path loss exponent $q = 2$ in our simulations. In this section, we do not address the optimal selection of the transmission radius for a high-mobility network. In our simulations, the transmission radius is set to 100 meters (and thus is not adaptive to node density). However, in Sections 3.3.3 and 3.3.5, we select only subsets of these neighbors within the transmission radius for different purposes.

### 3.3.2 End-to-end Measurement of the Energy Potential

One possible method to measure the end-to-end energy consumption is for the high-mobility nodes to sample this energy potential function. In this case, the base station can continually initiate a cost distribution (based on the energy metric), and the mobile nodes can spread out this cost metric to the rest of the network, adding their own link energy costs to it. When each node receives this information, it stamps it with the position that it is currently at. If the nodes
could exchange these cumulative measurements that they made, each node may be able to construct the overall energy potential function over space by averaging around each location, the measurements obtained by itself and by others, when they were passing through that location.

The problem with the above method is that there might be differences between the energy consumptions of the forward physical routes (to the base station) and reverse physical routes (from the base station) through the network. When the base station initiates the energy cost distribution, the end-to-end costs reflect the energy of the reverse routes. In our data channel model, every node wishes to send data from a physical location to the base station. Hence, the energy consumption values of the forward routes are the correct measures. However, if the forward routes are used for cost distribution, these cumulative costs can then be collected only by the base station, and there remains the question of how such information could be distributed back to the nodes. Eventually, we shall solve the entire problem by not employing the global end-to-end measurements and by showing how they can be constructed from only local measurements. However, in this section, we collect the end-to-end measurements in order to use them as a benchmark against which to measure the performance of the algorithms we shall develop. Hence, we collect the end-to-end measurements on the forward routes in this section, to allow a correct comparison later.

Hence, in our end-to-end energy measurement procedure, each node sends a short packet to the base station, and the total energy consumption required for the packet to reach the base station is recorded and then attributed to the cell where the packet originated. This constitutes a sample of the energy potential
function. As the nodes move through the network, they send a packet from each new cell that they move into, and these measurements are then globally collected, and averaged for each cell. The important point is that for the Random Waypoint Model, when the average for each cell is plotted, it converges to the energy potential function, as shown in Figure 3.1. (In this figure, we have normalized the energy values by assigning 1 to the highest energy in the plot.) The fact that the average of the measurements converges to a cup-shaped energy potential, which remains invariant over time is due to the fact that there is an underlying stationary node position density of the Random Waypoint Model.

This energy potential has been obtained by averaging global end-to-end measurements of energy consumption, and for real networks, we must propose a method for how to estimate this potential using only local measurements (of the “field” associated with this potential). The link energy costs can be obtained by measuring the transmit energy (per bit) to transmit useful data to one’s neighbor; however, this presumes that a certain localized routing algorithm is in place to decide which neighbor to send the data to. The choice of a neighbor must be done in a completely localized fashion, since the network is assumed to be highly mobile. We explore this in the next section, and turn to the construction of the energy potential from link cost measurements in Section 3.3.4.

### 3.3.3 Localized Link Energy Cost Measurements

In the last five years, localized routing has emerged as the routing method of choice for sensor networks, in the form of energy-aware routing, position-based routing, as well as hybrid routing protocols. In localized routing, the key problem
is the choice of a neighbor node (since the routing tables are obviated). For example, in GEAR[63], the neighbor choice is based on the learned cost metric which is a function of the distance metric towards the destination and the remaining energy deposits of the neighbor nodes; however, this assumes that a stationary sensor network topology is in place. In [23], fast, local topology control decisions are made while guaranteeing global properties such as network connectivity and minimum energy. However, there, there is no notion of direction towards the destination, and thus, even though the enclosure graph is obtained via only localized methods, a global distribution of energy costs is still necessary and requires global information passing. In contrast, we do not use such global information passing: we base our decisions on the minimum Joules per bit-meter that a node can achieve at every hop. That is, the next-hop neighbor in a mobile network is chosen as the node that minimizes the Joules per bit-meter towards the destination. Note that the “meter” in this metric is the number of useful meters towards the destination, not the number of meters in a hop (hence, one must project the distance onto the line towards the base station). For example, in Figure 3.2, the “Joules per bit-meter” metric $M_1$ for node 1 is as follows:

$$M_1 = \frac{|\vec{d}_1|^{q/2}}{|\vec{d}_1^p|},$$  \hspace{1cm} (3.1)

where $\vec{d}_1$ is the vector displacement from the source to the node 1, and $\vec{d}_1^p$ is the projected vector onto the line toward the base station. This minimization of Joules per bit-meter at every hop can be shown to be a suboptimal scheme, and counterexamples are easy to construct. However, in randomly deployed mobile networks with a single base station as the destination, most of the true minimum
energy routes to the destination are indeed along these greedy Joules per bit-meter neighbors at each step.

We say that a point \((x, y)\) in space is dominated by \((x', y')\) if the Joules per bit-meter metric to transmit to \((x, y)\) is more than that to \((x', y')\). Hence, we can exclude the dominated region from consideration when searching for optimal neighbors at each hop. In Figure 3.2, we plot the region dominated by a neighbor located at point \((50,75)\) [the source is at \((0,50)\)], when the path loss exponent is \(q = 4\). For \(q = 2\), the boundary of the dominated region is circular and coincident with the source node and the relay node (node 1). The figure shows how the boundary becomes distorted for environments with a larger path loss exponent, but that it retains its essential property of being bounded.
In the figure, the base station is much farther out in the direction of the \( +x \) axis. (Note that the \( +x \) axis in this figure is a locally constructed \( x \) axis in the direction of the base station. Hence, the coordinate system is different from that in Figure 3.1.) Note that in the figure, the nearby nodes to which it would take a small amount of energy to reach are penalized if that hop does not make much headway toward the base station. The advantage of this approach is that it gives us a geometric basis by which to eliminate most of the nodes from consideration in neighbor selection.

We assume that when energy-aware routing protocols are developed that utilize our framework, these protocols will decide when to initiate information transfer, based on the value of the energy metric (that our framework computes) for that location. However, a key assumption of our framework is that once the delivery of the packet is initiated, the delivery is continued along the minimum Joules per bit-meter path, with the minimum delay possible, until it reaches the base station. In other words, the current framework does not allow the relay nodes to optimize further by holding onto the relay traffic until they themselves find better locations. Such considerations might become possible in future research. In such cases, the definition of the energy potential must be considerably generalized to account for such decisions suitable for delay-tolerant networks.

Now, to connect these results with the aims of Section 3.3.2, we see that the localized link energy cost measurements that we were seeking, can be made based on these optimal neighbors with the minimum Joules per bit-meter toward the base station. Hence, each node records its minimum Joules-per-bit measurement and collects this over time. The nodes will share these localized link costs mea-
surements with each other, in a manner which we shall discuss in Section 3.3.5.

3.3.4 Construction of Energy Potential from Local “Field” Measurements

We divide the deployment region into a uniform grid of square “cells”. We assume that each node knows its own position via a position device (such as a GPS receiver), but not the instantaneous positions of the other nodes in the network (except when explicitly communicated, as we discuss in Section 3.3.5). In our simulation set-up, the deployment region is divided into a uniform, 20 × 20 grid; hence, each cell is 50 meters on each side. (We will divide the same grid with adaptively selected cell sizes for the performance improvement later in Section 3.3.6.)

We define the “sampling interval” as the time that passes from one measurement of the energy potential to the next measurement. Every node uses the same sampling interval in our design, and in our simulations, the sampling interval is 5 seconds; that is, every node records a measurement of the link energy costs every 5 seconds. During this interval, since $v = 10 \text{ m/s}$ in our simulations, every node moves 50 meters, which is the same as one side of a cell in our design. Hence, a node will typically have moved to a new cell when it is ready to record a new sample of the link costs based on Joules per bit-meter.

The link energy costs are measured using the minimum Joules per bit-meter neighbor (as explained in Section 3.3.3) and serve as the spatial derivative of the energy potential function. Hence, this method does not rely on any cost
distribution from any physical location. Any node that wishes to get the energy metric required to reach the base station, will be able to compute the value of the energy potential for its current location, on demand, by using the local measurements made available to it via exchanges with other nodes (which we shall describe in Section 3.3.5), and using a “path integration method”, which we develop next.

In the path integration method, we assume that the nodes have recorded local link cost measurements that they have made based on the minimum Joules per bit-meter neighbor choice, for as many points in space \((x, y)\) as is affordable to measure. By mechanisms that we shall describe in Section 3.3.5, the nodes have also been able to share these local link cost measurements with other nodes with which they come into contact as they move around. If the link cost measurement was based on a transmission from \((x, y)\) to some point \((x', y')\), we record this as a “transmission vector”. Assume, for now, that the data set of these local link costs measurements and their transmission vectors are globally available to us. Starting at any point \((x, y)\) in space, we take the vector average of all of the transmission vectors recorded for the cell in which \((x, y)\) falls. This gives us the average vector displacement of the useful distance travelled toward the base station in the first hop from \((x, y)\). (The energy field constructed this way is shown in Figure 3.3.)

The average transmission vector is less jagged and more aligned with the direction toward the base station, than one would find in a typical hop (which would show more variations perpendicular to the useful direction). This is fine because we also average separately the energy consumption of the measured link energy costs recorded for the transmission vectors from that cell, and associate this
average energy with that useful displacement. The average energy consumption of a link itself accounts for variations perpendicular to the vector toward the base station. After we take the useful average displacement using the average transmission vector, we apply the same procedure to the new point at which we have arrived. Hence, this procedure is repeated for subsequent points (as shown in Figure 3.4), and a “projected” path is virtually constructed from the point \((x, y)\) to the base station. This projection path does not represent an actual, typical path, but it has associated with it an energy that approximates well that obtained as the end-to-end energy metric in Section 3.3.2. Examples of projection paths obtained via the path integration method are shown in Figure 3.4. The sum of the averaged per-hop energy metrics on this path from the point \((x, y)\) to the base station...
Figure 3.4: Virtual, projection paths obtained via the path integration method

station, constitutes the end-to-end energy potential at the point \((x, y)\).

In Figure 3.5, we plot the percentage error of this method with respect to the end-to-end measurements obtained in Section 3.3.2. We find that the average error is about 5% over the deployment region (this is not shown in the figure, but computed from the data), with a 10% peak error as we approach the base station. (The sharp drop that occurs very close to the base station is due to the fact that if the algorithm detects the base station, then it sends directly to the base station, not relying on the energy potential computed. Hence, those values are not represented in the figure.) This peak error as we approach the base station arises from the resolution of the uniform grid. Because the node density is large in the middle of the deployment region for the Random Waypoint Model, the averaged transmission vectors for a fixed cell size fail to approximate well the
globally measured end-to-end energies. This calls for adaptive approaches to cell resolution selection; the size of the cells used to record and lump measurements must get smaller with increased node density. These inaccuracies that occur for high node density areas are also responsible for pushing up the error farther from the base station, since the path integration method uses those inaccurate paths in high node density regions. Hence, the error percentages across the entire deployment region are expected to decrease if adaptive cell-sizing based on node density is utilized, which we shall show later in Section 3.3.6.

The end result of this procedure is that it permits us to estimate the potential at every cell (albeit with some error), where the potential is an estimate of the average total energy that a bit would consume to reach the base station. Note that this potential is constructed from completely local measurements of link
costs, and no end-to-end measurement of total energy is necessary. However, these local link cost measurements made by different nodes must somehow be exchanged, distributed and fused to create a picture of this energy potential, which we address in the next section.

3.3.5 Collecting and Distributing the Spatial Energy Measurements

In this section, we show how the nodes can exchange the locally made measurements of the spatial derivative so that they can form the averages in question. Significant problems arise for two reasons: (1) at first glance, there is an immense number of local measurements that need to be exchanged, and (2) the nodes have very short encounter times with each other, which limits the amount of information exchanged upon every encounter. The spatial derivative information is constructed in the joint memory of the nodes in the network, and we show how this locally measured energy information can be exchanged, fused, and refined to aid in energy-aware routing decisions in high-mobility networks.

In our design, the nodes measure the magnitude and direction of their transmission vectors and the energy per bit that it takes to transmit along a link. Each such measurement is stamped with the following: (1) the position \((x, y)\) that the node was at when the measurement was made, (2) the time at which the measurement was made (the absolute time is provided by the same GPS receiver), and (3) node ID. Over time, each node accumulates a large cache of its own measurements. Since the node density over the deployment region is stationary, the
energy potential is also constant. This means that the measurements made do not become stale over time, since they are associated with particular positions. In fact, as these local measurements made over time increase, the accuracy of the construction increases as well, since they are related to the same, stationary underlying energy potential over space. As a result, the nodes build and carry with them their link cost data, which if properly shared with other nodes, can lead to the construction of the energy potential function.

An important issue is how these measurements can be fused, distributed and used to compute the energy potential. There are complex problems here: first, it appears expensive to share raw data of link energy cost (and associated transmission vector, generation time stamp, and node ID) of the measurements. For example, since a node moves on average across one cell of the grid in one sampling interval of 5 seconds, it will have 100 measurements of its own to share in 500 seconds. Further, it can share all the measurements that it received from the other nodes it has encountered. If it encounters 2 new nodes every time it moves to a new cell, then after 500 seconds, it will have at least 20100 measurements. Much of this would be redundant measurement data over time, as many nodes would have copies of the same raw data. A second approach might be to share a fused version of the data, such as the average link energy cost and the vector average of transmission vector, computed for every cell in the grid over time. However, some data points might be repeated many times in these averages. A solution that would allow new, collected link cost data to be weighted equally with previously collected measurements is to include the number of points in the average, when reporting the average to the other nodes. However, even this does
not eliminate the fact that a lot of replicated data, known to the other nodes, is still being communicated to them, now in fused form.

Fusing data collected from sensors has been widely investigated in the literature [85][86][87], however, mainly for non-mobile networks. In this section, our goal is to develop novel spatial energy metric dissemination algorithms for highly mobile networks. First, we introduce the following ideas and definitions: In the beginning, the nodes start out with no spatial energy data. Hence, if a node attempts to construct the value of the energy potential from its current position to the base station, it will fail. Then, we say that an “outage” occurs when a node is not able to construct the required potential from its current position. In the beginning, we expect most of the nodes to be in outage, but as the nodes exchange (by the methods that we shall describe) the measurements that they collect, the number of outages should decrease.

We would like to construct a quantitative measure of how fast the data spread through the network. Although the probability of outage might appear as a reasonable metric to quantify the performance of data dissemination, it does not give any information on the accuracy of the constructed energy potential. As a more accurate measure, we use the percentage error between the constructed energy potential and the optimal (global) one, as our measure. Whenever the energy potential cannot be constructed (due to a missing value for a cell: that is, due to outage), we penalize the percentage error by assigning 100% error to such situation of outage.

We define the “spreading period” to be the duration required for the percentage error to drop below a certain value (e.g. 20%), on average, for the nodes in
the network. The spreading period is very important because it measures how long it takes for information to spread through the network; however, note that the information here is the relevant information required to construct the energy potential. That is, it is possible to construct some energy metrics even when no measurement data is available about the cells next to the edge of the network because those cells are typically not used by the nodes to construct the energy potential. The key benefit of measuring the spreading period for a stationary energy distribution is that it also tells us how long it would take for time variations in the energy function to be transmitted to the other nodes. Hence, the spreading period is a key indicator of how well this spatial energy memories framework can be applied when the energy potentials are time-varying.

Compared with the gossip-based (or epidemic) methods [90][91][92][93], we note the following difference: Gossip-based methods typically address one token of information, and examine how fast for a single token of information it takes to spread throughout the network. In contrast, we must transmit and refine an entire “field” of measurements, which is more like an image, that must be constructed from local measurements. Hence, both the amount and the type of data are different. In our system, we aim at making the spreading period as small as possible, in order to address the time-varying energy functions for real networks. Hence, the overhead reduction methods that we use are designed such that they will not increase the spreading period, as the gossip-based algorithms might. The main overhead reduction method we use is the judicious choice of the transmission radius based on node density, which the nodes can adjust on the fly (or based on node density statistics collected over time).
One of the key observations in reducing the amount of data exchanged is the following: as the new measurements spread throughout the network, it soon becomes no longer necessary to exchange those measurements since almost every node will have received them. Hence, we define a parameter $W$, that we use as a window size, to limit the number of measurements by the generation times of data. The nodes report only those measurements that were made with generation stamps that are within the window of $W$ discrete-time steps of the current time. (Each discrete-time step is 5 seconds in our simulations.) As $W$ becomes larger, the spreading period is expected to decrease; however, the amount of data exchanged also grows with $W$. Since in a high-mobility network, each encounter will have a limit on the number of measurements exchanged before those neighbors disappear, we would like to empirically find a low enough $W$ that strikes a good trade-off between the number of measurements broadcast in each sampling interval and the percentage error in the energy field construction. (Each node makes one broadcast of measurement data per sampling interval for design simplicity.) Note that the measurements before $W$ time units are not being discarded; rather, they are no longer being exchanged because every node is assumed to have received them and integrated these into its calculation of the average of its localized measurement for every cell.

With the above intuitions, Figure 3.6 displays the algorithm that every node uses. Every node makes one measurement at every discrete-time increment. Every node keeps a table of the measurements it has heard or generated with the generation times that are at most $W$ discrete-time increments into the past. At each sampling interval, it reports (i.e. broadcasts) only this data to its neigh-
bors. (Recall that each measurement of the link energy cost is reported along with the position and time that this data was generated at, as well as the node ID that generated the data.) During this time unit, the node also receives data from its neighbors. Since the neighbors follow the same algorithm, that data also have generation time stamps within the last $W$ time intervals. The received data is checked against the data reported, and any received data that is a replica of what has been reported is eliminated in the “Filtering” block shown in the figure. The Figure shows an example where two blocks of measurement data have been received, one from neighbor node $j$, and another from neighbor node $k$. In the middle, there is a replica where both the node ID (4 in the figure) and the generation times ($t_{-3}$ in the figure) match; hence, the replica of this measurement is eliminated. Such replicas occur for various reasons: a node may have not moved sufficiently within the last interval, and the nodes with which it exchanges data may be at least partly the same, leading to replication in the table entries. A second reason is that the link energy cost measurements that have travelled through different paths and arrived at the nodes’ measurement tables might be intersecting. After filtering, the remaining data is used in two ways: first, the new measurements are used to update the average link cost vector assigned to each cell (shown in the lower right hand corner of Figure 3.6). Second, the next report is created by incorporating the new data into the current table, and also shifting the window of size $W$ by one interval (so that the older data disappears from the window, and the new data are added to the window, at each time unit).

The performance of this energy metric dissemination algorithm can be measured along three dimensions: first, the percentage difference between the energy
REPORT:
Report the raw measurements with generating node ID and time-tag
\[ t_W \quad \cdots \quad t_1 \quad t_0 \]

RECEPTION:
Receive the raw measurements reported by its neighbors
\[ t_W \quad \cdots \quad t_1 \quad t_0 \]

FILTERING:
Check the generating node ID and time-tag of the measurements and remove the replica
From node j, \{ m(1, t_w), m(4, t_3), \cdots, m(9, t_1) \}
From node k, \{ m(3, t_w), m(4, t_3), \cdots, m(9, t_2) \}

UPDATE NEXT REPORT:
Generate next reporting data by shifting the timing window and including new link cost measured using Joules per meter metric at \( t_1 \)
\[ t_{W+1} \quad \cdots \quad t_0 \quad t_1 \]

UPDATE FIELD INFORMATION:
Another filtering out the replica already used for the previous update up to \( t_1 \) and update the field information at \( t_0 \)

Figure 3.6: Algorithm for energy metric dissemination and filtering
potential that is obtained via data dissemination (and path integral construction) must be compared with the actual end-to-end measurements that were obtained using the method described in Section 3.3.2. Second, the percentage of error of the energy map with respect to the energy map constructed via the path integration method using the global end-to-end metrics as a function of continuous time determines the spreading period required for the information to spread. Third, the average amount of measurement data broadcast by a node (once per sampling interval) indicates the amount of bandwidth consumed by measurement exchanges. In actual design, this amount must be compared with the cap on the amount of measurements that can be exchanged upon the short encounters that nodes experience in high-mobility networks. Such control information must consume only a small fraction of this cap.

The neighbors to which a node disseminates this energy metric data are selected as follows: in our simulations, the 3 neighbors with the lowest Joules per bit-meter costs are selected. This can be achieved, in practice, by transmitting the energy metric on lower power control channels that reach only those neighbors. First, the next set of graphs shows a percentage error of the energy potential constructed by data dissemination.

Figure 3.7 shows the percentage error of energy constructed via the data dissemination method versus the energy obtained by end-to-end measurements (described in Section 3.3.2). The graph reports the percentage error averaged over all the nodes in the network; further, it is parameterized by the window size \( W \). Note that \( W = 0 \) corresponds to the case in which every node reports only its own new measurement in the current time interval. (When collecting this perfor-
Figure 3.7: Percentage error of data dissemination with respect to end-to-end global measurements of the energy potential.

mance data, whenever outage occurs, an error of 100% is assigned to the error percentage in that occurrence.) We see that a window size of \( W = 5 \) produces an error percentage of roughly \( 8 - 9\% \) in the limit. This error has two components. First, an irreducible floor component comes from the fact that data dissemination methods used the path integration method of Section 3.3.4, and our path integration method’s accuracy is limited by the large cell size around the base station that causes an average of 5% error on the deployment region (see the discussion on Figure 3.5). This 5% error can only be reduced by adaptive cell sizing, based on node density, but we do not use this technique in this section; therefore, it constitutes a bound on the performance of the data dissemination methods. We expect that a difference of \( 3 - 4\% \) arise (that is, \( 8 - 9\% \) minus 5%) when the data dissemination is compared directly with the path integration with access to all of
the link cost measurements (as in Section 3.3.4).

Figure 3.8 displays the error percentage (averaged over all the nodes) of the data dissemination versus the path integration method with access to all of the obtained measurements of link costs. The fact that the error percentage approaches $3-4\%$ confirms our hypothesis. This difference arises due to the effects of a finite $W$ (where $W = 5$ in our discussion). Because $W$ places limits on what can be disseminated before it is considered obsolete, it is as expected that this causes errors when compared to a scheme that has global access to all of the link cost measurements when computing the average link cost vector for each cell. However, the difference is small.

Based on Figure 3.8, we define the “spreading period” as the time that it takes to reach the point where a node can construct the energy potential with
80% accuracy, that is, with a target of 20% percentage error. (This definition can be modified to suit the application, by varying the target percentage of error.) We see that $W = 0$ leads to a spreading period that is roughly 1400 seconds, and the spreading period is down to 65 seconds for $W = 5$. (This has been calculated based on the exact data set.) In order to interpret this result, we note that it takes roughly 50 seconds (in our simulation set-up) for a node to reach from a starting point to an end-point (where it picks a new destination to travel to) in the Random Waypoint Model. Hence, for $W = 5$, after a typical node has travelled roughly 1.3 such lengths, we arrive at a point in the network where a node can, on average, construct the end-to-end energies with 80% of accuracy, for all the positions it needs. This spreading period might seem long for practical purposes; however, note that this value is for constructing the entire energy function for any locations the node encounters.

Second, Figure 3.9 displays the average number of measurements broadcast per node at each sampling interval, as a function of continuous time. The average is computed by averaging both over time that has passed so far, and over the entire set of nodes. The plot is parameterized by the window size $W$. We see from this graph that the average number of broadcast measurements increases as a function of $W$. The reason for the initial sharp rise is that in the beginning, no nodes have any measurements. As the measurements are made, exchanged and accumulate, the tables of measurement data (obtained as in Figure 3.6) grow very fast, and this fast growth results in a large number of broadcast measurements (even though the replicas are continuously being eliminated). However, the key aspect of this graph is that the average number of measurements broadcast per
node per sampling interval settles down to a steady-steady value. An upper bound for this number is $NW + 1$ (where $N$ is the number of nodes) since a total of $NW$ measurements are generated in the entire network in the last $W$ intervals (and 1 is added for the reporting node’s current measurement in that time interval). However, the steady-state values are much lower than this bound.

The steady-state values allow us to quantify the overhead involved in disseminating QoS information. For example, when $W = 5$, the average number of measurements that a node broadcasts in steady-state is 75 per broadcast (which happens every 5 seconds in the simulations). Since a node ID, the position and the time are also transmitted with each measurement, the following calculation gives an idea of the overhead involved: there are 7 bits required to address 100 nodes uniquely. Since there are 400 cells in our design, 9 bits are required to address the
positions. Using a round-robin sequencing method for the times, and assuming that we can wrap around after 4 spreading periods, 7 bits are required to stamp the times (in discrete-time steps) for $W = 5$. Each measurement reports a transmission vector, which we describe as a displacement from the center of the cell to an angular quantization of 10 degrees, leading to 18 such useful displacements in the relevant half-plane, hence representable by 5 bits. Finally, the measured value of the link cost can be quantized to 8 bits, representing 256 such values. (Although the link costs are expected to scale with node density, they are limited by the resolution of the cell in a uniform grid in our design.) Therefore, each measurement entails a total of 36 bits; hence, 75 measurements per discrete-time unit entails 338 bytes of information. For $W = 3$, this is reduced to 112 bytes (25 measurements), and the percentage error performance from Figure 3.8 is still within reasonable range. Whether this is a lot depends on the link speeds. If the link speed is 100 kbps (whenever a transmission occurs), 901 bits for the $W = 3$ design would be transmitted in roughly 0.01 seconds, and this case, would not consume a large fraction of the overall bandwidth. However, the Doppler shift that is induced due to mobility will also place bounds on the link speeds that can be achieved. A real network design must carefully examine these trade-offs.

### 3.3.6 Adaptive Cell-sizing Approach

In the development so far, we have used uniform cell sizing; that is, we have divided the deployment region into a square grid of equal-size cells. Since the nodes are assumed to have position information throughout this chapter (via positioning methods such as GPS), it is possible to hardwire a uniform grid into
the nodes’ memories in order for them to record the collected information for each cell. This does not require any knowledge of the node density, which arises from the network itself, and thus can be used for any mobile network.

However, as the percentage error in Figure 3.5 indicates, the percentage error is largest in the middle of the deployment region, for the model that we have been using so far. This arises due to the higher node density (due to transit traffic) in the middle of the deployment region. A method to reduce this peak in the percentage error is to use “adaptive cell sizing”, that is, using cell sizes that are chosen adaptively to the stationary node density. For the model at hand, smaller cell sizes would be used for regions with a higher node density.

The purpose of this section is to explore how much we lose by using a uniform grid, over a near-optimal approach that uses cell sizes that are adapted to the node density. We will show that the loss is in fact not significant; hence, this justifies using the more practical uniform grid in the rest of this thesis. Conversely, the results of this section indicate how much gain is possible when adaptive cell sizes are used, if for example, node density information can be collected and built from the network (along with energy metrics). The results of this section show that this performance difference is not significant and that most practical designs can use a uniform grid.

In Sections 3.3.1-3.3.5, we concentrated on the average energy consumption to reach from one point to another point; however, it is possible that large deviations might occur around the average that we computed. The standard deviation of

---

1This approach is near-optimal because we consider only rectangular tessellations in our adaptive cell-size designs. It is possible that other (such as hexagonal) tessellations provide even better performance; however, they are not pursued in this section. We expect that the adaptive rectangular tessellations shall perform close to optimal.
Figure 3.10: The ratio of the standard deviation of energy measurements to the average energy with uniform cells

the energy divided by the average computed is an indication of the lack of accuracy if the end-to-end energy metric averages were used to make energy-aware routing decisions. The ratio of the standard deviation of energy measurements to the average energy with uniform cell sizing is displayed in Figure 3.10. From this figure, we see that when a node sends a bit from a point close to the periphery of the deployment region to the base station, it can be assured that the bit will be delivered within roughly $5 - 10\%$ error. The reason that the results are more accurate for farther points is that the incorrect routes picked near the base station (due to the uniform cell size) matter less in the deviation from optimality, for farther points. Overall, this $\sigma / E_{avg}$ graph shows that the framework developed in Sections 3.3.1-3.3.5 on the average energy is applicable to giving energy consumption guarantees, within the confines of the Random Waypoint Model, where
\( \sigma \) denotes the standard deviation of end-to-end energy metrics and \( E_{avg} \) denotes the average of the energy metrics.

The implication of this result is that when we perform the energy-aware routing, we can specify a maximum energy guarantee (for example, \( E_{avg} + 2\sigma \)) as the energy cost of transmitting traffic to the base station. However, in Figure 3.10, there are large peak errors around the base station due to the uniform cell sizing. The lack of accuracy around the base station places bounds on the accuracy of the constructions of paths around the base station. We shall show that the adaptive cell sizing approach can reduce this inaccuracy close to the base station.

Our adaptive cell sizing selects the resolution of each cell resolution according to a given node density. This approach scales the cell size inversely with the node density such that the average number of nodes per cell is roughly the same. Using the node density of the Random Waypoint model in [74], the corresponding cell tessellation is illustrated in Figure 3.11. In this figure, the grid is divided into 20 by 20 cells with adaptively selected sizes. That is, we have kept the total number of cells at 400 (as in Sections 3.3.1-3.3.5), but now adapt their sizes to the node density of the RWP model. Note that due to the adaptation, the average number of nodes per cell is 0.25; that is 100 nodes divided by 400 cells with the adaptive size.

This adaptive cell sizing reduces the larger errors near the base station. We showed the ratio of the standard deviation of the energy measurements to the average energy with adaptive cell sizing in Figure 3.12. The peak in the percentage error around the base station still remains but has been significantly reduced with compared to the uniform cell sizing. A node transmits directly to the base
station when it gets very close to the base station. Thus, this peak does not have much impact on the energy-aware routing performance, because any node will use a direct transmission to the base station on a direct link regardless of what kind of routing protocol being used, when it is in the same cell as that of the base station. Further, from Figure 3.10 and 3.12, it is shown that the ratio of the standard deviation to the average with adaptive cell sizing remains roughly constant and small except for the cells very close to the base station, while the ratio with uniform cell sizing keeps growing and exhibiting large errors toward the base station. The adaptive cell sizing provides less than around 5% increase in accuracy except for the region of direct transmission.

In Figure 3.13, we compare the percentage errors of two approaches with respect to the end-to-end global measurements of the energy potential. In order
Figure 3.12: The ratio of the standard deviation of energy measurements to the average energy with adaptive cells.

Figure 3.13: Performance comparison between with uniform and adaptive cells: Percentage error of data dissemination with respect to end-to-end global measurements of the energy potential.
to make the graph legible, we illustrated the results for only three window sizes (W=1, 3, and 5). The dotted and solid lines indicate the uniform cell sizing and the adaptive cell sizing methods, respectively. In Figure 3.13, the uniform cell sizing performances are copied from those in Figure 3.7. It is seen that the performance of the percentage error around the base station is reduced by using the adaptive cell sizing approach. The reason is that, by using the adaptive cells, we can reduce the large errors which were incurred due to the large cell size with a uniform cell sizing. Further, since the adaptive approach used larger cell sizes for the farther points from the base station than in the uniform cell sizing approach, there is a slight increase in error as a node goes farther away from the base station. However, overall, the adaptive cell sizing method outperforms the uniform cell sizing method. In Figure 3.13, for the window size $W = 5$, the uniform cell sizing produces the percentage error of 9.08% in the limit, and the adaptive cell sizing produces about 6.15%. The difference is 2.93%, which is small.

The adaptive cell sizing requires that the node density function over the deployment region be known. If adaptive cell sizing is pursued, the following method can be used to estimate the node density function: In the beginning, we assume that nodes do not have any knowledge of the node density. Thus, nodes start out their cell-sizing with the uniform grid. After the nodes have spent enough time (for example, on the order of the spreading period), they start using the adaptive cell sizing with the node density information collected from the network. A way to collect the node density information over the deployment region is that each node collects the number of nodes in the cell where it is currently located, and broadcasts its node density information using the dissemination algorithm.
described in Section 3.3.5. However, even though the node density information can be achievable for the nodes in the deployment region, this broadcasting of extra data would increase the overhead over the network. In contrast, the uniform cell sizing can be easily implemented without the need of knowledge of the node density function. Thus, there is a trade-off between the error performance and the ease of system design in cell-sizing for the construction of energy maps. In the design of actual systems, the uniform cell sizing would be preferable due to its simplicity at the cost of more percentage error roughly of only 3%, which is small. Hence, we use the uniform cell-sizing in the rest of this thesis.

3.4 Construction and Dissemination of the Energy Map for Time-varying Energy Potentials

In this section, we address how the time-varying energy potentials that model more realistic networks can be disseminated to the nodes. In order to address how our path integration method and data dissemination algorithm can handle the time-varying energy potentials, we consider a network scenario with the contraction of node density in two sample networks: First, a campus network may be a good example of a network having time-varying energy potentials. In this scenario, there are mainly two kinds of student behaviors during the regular hours: attending classes and moving to their next destinations or wandering around the campus. When classes start, many of students would head to the lecture halls and
stay there for a long time. Outside the lecture halls, only a few students move around. As the number of people significantly decreases outside the lecture halls, the number of relay nodes also decreases in the deployment region. As a result, the energy metrics become larger due to the lack of relay nodes and the end-to-end energy potential in the deployment region is scaled up in all of the cells. When students come out of the lecture halls, the network conditions can then be approximated in this case by a time-invariant energy potential again under the random movements such as the RWP mobility model. Although this incidence of movements does not cover all kinds of behaviors, we will focus on these simplified characteristics of student behaviors on campus, because the overall composition can exhibit time-varying energy potentials of the campus network.

The second example is a resort area. In this kind of area, there will be a few big stages for events which attract people in the region. People visit these stages, stay for a while, and leave then. They convene when the events start and leave at the almost same time when the events end. Between the event stages, people wander around the area before making decisions on their next visiting points. We expect that this example also shows a drastically changing network conditions over time. Beside the above two examples, we can observe this kind of behavior in many places such as big malls in a city and tourist spots. In order to take advantage of these scenarios, we now try to imitate this behavior by using the following mobility models.

In our simulations, we use two mobility modes: Random Waypoint model only, and the combination of the RWP model and modified contraction model [47]. We use the Random Waypoint model only mode to generate a time-invariant energy
metric as previously used in Section 3.3.1, which corresponds to the case when people wander around in the area. We use the combination of the RWP and the modified contraction models in order to generate a time-varying energy metric, which corresponds to the case when some people conglomerate at the attractors, and others remain and keep moving around outside the attraction areas. We will call the case of using the RWP model only, a “time-invariant mode”, and the case of using the combined mobility models, a “time-varying mode”.

The combination of two mobility models for the time-varying mode is illustrated in Fig. 3.14. In the deployment region, we place four attractors at positions (0.275, 0.275), (0.275, 0.725), (0.725, 0.275) and (0.725, 0.725) km with attraction regions of 0.0225 km² around each. The base station is located at (0.5, 0.5) km. Outside the attraction areas, a node follows the RWP model. When a node’s destination is inside the attraction area, it moves according to the modified contraction
model after it reaches that destination. In the modified contraction model which was introduced in [47], a node chooses a destination by keeping itself directed to the center of attraction area. A node strictly follows this contraction model until it arrives at the cell where the attractor is located. This modified contraction model is very useful to simulate the networks having several attractors such as famous visiting points, the market areas in a city, and the lecture halls on campus.

In our simulation set-up, we set the time-varying mode for the duration only from 1000 seconds to 1500 seconds and time-invariant modes for the other durations over the whole simulation time of 2500 seconds starting from 0 seconds. During a time-varying mode, if a node arrives at the cell with an attraction point, it stays there until the time-varying mode ends. For consistency of system evaluation with respect to the time-invariant case, all other set-up parameters are the same as in the previous sections.

In order to show how our scenario using the contraction model affects the energy potential over space, we use Figure 3.15 and Figure 3.16. In Figure 3.15, we display the node density corresponding to four different time sequences. In the beginning for the time-invariant mode, the node density is an increasing function toward the center of the grid and there is a peak around the center, which is similar to the analytical node density of RWP mobility model. After 1000 seconds, the system goes into the time-varying mode. Some nodes start converging to the cells where the attractors are located. As time passes under the time-varying mode, more nodes reach the attractors, and only a few nodes wander around outside the attraction areas by RWP model. (In the simulations, we observed that 50 to 60 out of 100 nodes are gathered near the four attractors, and the rest are distributed
outside.) At 1500 seconds, the time-varying mode is over and the time-invariant mode is re-activated. As shown in the figure, at 1600 seconds, the corresponding node density is then going back to that of the RWP model only.

The energy maps highly depend on the node density of the deployment region. As the channel becomes sparse due to lack of the number of intermediate nodes in a region, the end-to-end energy costs to send a traffic to the base station increase. That is, The energy metric in a cell is highly variable depending on the existence of intermediate nodes between the cell and the base station. The results of this dependence are displayed in Figure 3.16. We see that the energy metrics have

Figure 3.15: Time-varying node densities at different time sequences
been scaled up when applying the time-varying mode. At 1500 seconds, it has the highest energy cost due to the scarcity of intermediate nodes in the network. It should be noted that although the energy metric has increased, the shape of energy potential over space still resembles a cup, albeit with some distortion. This is an important property that we shall exploit in Chapter 4. It is obvious that the farther the cells are located from the base station, the more energy would be necessary.

Now, we discuss how our methods in the previous sections, used with a time-invariant energy potential, work for the time-varying mode. In fact, the same
methods can be applied except for two modifications. First, a modification concerns how many measurements are used for the average of a node’s local link costs that correspond to each cell, which will be used for the path integration to estimate the end-to-end energy metric. The difference is that the local link measurements gathered is now averaged over a certain number ($S$) of most recent measurements, instead of being done over the all measurements achieved so far through the entire time horizon. The reason is that, if we do not limit the number of measurements to be averaged, measurements become obsolete in the time-varying mode, and we avoid the use of these obsolete data which would reduce the accuracy on the construction of energy maps. For example, in Figure 3.16, we have seen that there is a large gap between the values of energy potentials for the different modes. Under the time-varying mode, since the energy metric varies so fast, any underlining average of the energy potential for each cell does not exist. The data achieved during the time-invariant mode would become obsolete after the system enters the time-varying mode, because the difference between the energy metrics might become very large. In order to avoid this harmful effect, we use only $S$ most recent measurements for the average of the transmission vector and the link energy cost of each cell, where $S$ is a parameter to be chosen. Note that this parameter $S$ should be considered separately from the window size $W$ in Section 3.3.5. The window size $W$ is not related to the time variance of energy potentials, but rather to the spreading period of data dissemination over the network. In fact, $S$ can be considered as the coherence time for a time-varying channel. (Hence, $S = \infty$ is the optimal choice for a time-invariant energy potential.) We will discuss the effect of the limited window size of $S$ with the simulation results later in this
Second, when the contraction model is applied to the network, only a few nodes remain outside the attraction areas. Due to a decrease in node density, the number of measurements per cell also decreases. No measurements might be available for some cells, if nodes do not visit those cells for a while. As a result, the number of outages will be increasing when doing the path integration over the network, since there are not enough data. In order to avoid these outage problems, we use an improvement over our path integration method: For the cell which causes an outage of path integration, we use an interpolation of the transmission vectors and the link energy costs by looking up the information of its adjacent cells. In order to explain this interpolation, we use Figure 3.17. Let the cell which has no measurements be cell $k$. We interpolate by averaging over any pair of neighboring cells of cell $k$, in which a pair of cells are symmetric with respect to cell $k$, and such that one of the pair is the further cell from the base station, and the other is the cell closer to the base station. In this figure, a node is constructing a path from its current point to the base station, but there are no measurements for cell $k$ which lies on the path to the base station. The transmission vector and the link energy cost for cell $k$ need to be estimated from the adjacent cells of cell $k$. In the figure, there are 8 adjacent cells but there are only two pairs of cells (i.e., $(1a, 1b)$ and $(3a, 3b)$) which currently have collected measurements and are symmetric with respect to cell $k$. In our interpolation method, the transmission vector and the link energy cost are obtained by respectively averaging the metrics of those four cells, and these averaged estimates ($E_k$ and $\vec{v}_k$ in the figure) are used by the node to keep constructing its path from the point in cell $k$ toward the base.
station. However, for the worst case, if such a pair of adjacent cells with data does not exist, an outage would still occur. (Note that a node does not report the interpolated measurements obtained by using the adjacent cells, but rather broadcasts only the raw measurements to the other nodes when disseminating the data.)

Now, with the two improvements, we show the results for the simulations that includes the time-varying case in Figure 3.18. Again, the RWP-only indicates the time-invariant mode, and the combination of the RWP model and the modified contraction model indicates the time-varying mode. As a performance metric, we compute an instantaneous percentage error of the constructed energy value by using our path integration via the dissemination algorithm with respect to the instantaneous, global energy measurements (not the averaged, global energy measurements over the entire time horizon, which we used in the previous sections).
The conclusion of this graph is two-fold: First, a path from any point over the deployment region to the base station can be constructed with the percentage error of less than 27%. For the time-invariant mode, the percentage error is roughly 17% on average and, for the time-varying mode, its percentage errors are between 20 – 27%. The reason of this worse performance for the time-varying mode is that the channel is extremely fast changing and its performance is severely affected by the scarcity of the network. Despite of this degradation of the percentage error performance, overall, a node is able to catch up the channel variance at the cost of percentage errors roughly of only 3 – 10 % more. Furthermore, it should be noted that if we can detect which mode of the network is currently in, such a large percentage error for the time-invariant mode can be reduced by expanding the window $S$ (namely, by using more relevant measurements) to obtain the reliable average energy metric.
Second, we show the effect of the window size $S$ on the average energy potential in this figure. For the time-invariant case in which an underlying average exists, the larger the value of $S$, the better the methods perform. In contrast, for such a fast changing energy potential in the time-varying case, increasing $S$ (using more older measurements) would be detrimental. This result implies that the coherence time of the time-invariant mode is infinite (hence the optimal $S = \infty$), while the coherence time should be shorter for the time-varying channel. From Fig. 3.18, we see that the case of $S = 50$, which is the largest $S$ in our simulations, performs better for the time-invariant mode, while for the time-varying mode the case of $S = 1$ does better. This is because the coherence time of the energy potential in the time-invariant mode is infinite (i.e. the optimal $S = \infty$), while the coherence time of the every potential in the time-varying mode in this simulation is only 5 seconds (equal to the sampling interval), because the mobility models we used induce an extremely fast varying energy potential. The implication of this result for the actual system design is that the coherence time $S$ could be obtained by measurements from the network conditions. One might perform a field study over the deployment regions such as campuses, malls and tourist spots. In the study, the collected energy maps will play the role of the indicators of the energy potential characteristics. Then, a look-up table of $S$ corresponding to the different network conditions can be built, and each node can use this table for its path integration to construct the energy maps.
3.5 Summary

In this chapter, we developed a novel framework and methodology to make end-to-end information on energy metrics available in high-mobility networks. We showed that mobile networks may be more tractable than commonly believed since their node densities remain stationary over relevant time durations whereas individual nodes change positions relatively fast. By exploiting this high-mobility, nodes can exchange the local link measurements very fast, and the energy maps based on these local measurements can be constructed via the proposed path integration method. Further, we showed that our energy map construction and dissemination methods can still work even for a fast changing network which does not have a stationarity of node density.
Chapter 4

Energy-efficient Transmission

Utilizing an Energy Map

In the previous chapter, we explained how to construct the energy maps which are the maps of the end-to-end energy metrics on the deployment region, and how to share this information among nodes during their encounters, via the path integration method and our information dissemination algorithm. In this chapter, given that a node can develop knowledge of the entire energy map over the deployment region, we analyze the problem of minimizing the energy required to send a fixed amount of data subject to delay constraints in mobile networks. We develop a smart transmission strategy by which a node decides when to send its bits; a decision is made based on dynamic programming using a finite time horizon while meeting the delay constraints imposed by applications.
4.1 Introduction

For most wireless networks, energy management and adaptivity to mobility have been two key concerns, since the devices are severely energy limited and the network topology changes. The wireless networks of the next 10 years are expected to be made up of a plethora of tiny microprocessor-sensor units embedded in clothes, shoes, cars, and buses. We envision that these mobile sensor networks will be extensions of the wired and wireless networks today (which include conventional devices such as laptops and PDAs). For these battery-operated applications, it is more suitable to view the networks as energy-constrained networks, rather than the bandwidth-constrained which is the traditional regime of networks. In order to measure the throughput of the networks, we use the “bits-per-Joule” concept introduced in [23], which is roughly defined as the maximum number of bits that a network can deliver per Joule of energy that is deployed into the network. For bandwidth-constrained networks, researchers have worked on the maximization of data rate (in bits-per-second), but we will seek to minimize the energy consumption of the network for a given number of bits that network needs to deliver.

We expect that the future networks will have a wider variety of data traffic, including delay-tolerant and error-tolerant data. In this chapter, we focus on delay constraints that are on the order of seconds to minutes rather than milliseconds for the delay-tolerant data applications. Since the data traffic is delay-tolerant, the reliable delivery of small amounts of sensor data under severe energy constraints is more important than the maximization of the data rate. Thus, in this
chapter, we develop a traffic allocation algorithm which minimizes the energy consumption subject to a fixed number of bits to be transmitted, while meeting delay constraints.

The energy maps that we developed in Chapter 3 are very valuable in designing energy optimization and planning protocols in high-mobility networks. In actual wireless mobile networks, there would be good or bad positions with respect to the distance to the destination for each node, as illustrated in Figure 4.1. In this figure, the index $t$ denotes time, the index $k$ denotes a cell number in the deployment region, and the index $\bar{e}$ denotes the estimated end-to-end energy metric from an energy map for this network. The energy map is projected onto
a two-dimensional space with the contours for different energy levels. The darker each shaded region is, the less end-to-end energy from the point to the base station is necessary. In the figure, a mobile node travels around the deployment region, and its near future trajectory is known. If the node has to send its traffic to the base station within a certain delay deadline (which gives a truncated future trajectory that is relevant), then it has to decide when to transmit along this truncated trajectory. The node senses from its energy map that when it gets into cell $k_3$ in time $t_3$, it will have the minimum end-to-end energy $\bar{e}_3$. Hence, the decision of this node at time $t_0$ is to plan to send its data when it gets into cell $k_3$. This is the main idea behind energy planning.

Intuitively, based on Figure 4.1, if mobile nodes know their future trajectories, then they can reduce their energy consumptions by leveraging their movements around the network, since they are able to estimate the expected energy consumption for the future locations by using our energy map (that is achievable via the methods in Chapter 3). Here, even though the end-to-end energy metrics are available, an energy-efficient traffic allocation decision still has to be developed, subject to a delay constraint imposed by the application. Hence, the main goal of this chapter is two-fold: (1) to develop an optimal algorithm to allocate traffic over the relevant future trajectory by utilizing an energy map of the network, and compare with a competing algorithm which allocates the traffic at a constant rate without utilizing the energy map, and (2) to compare the total energy consumption of energy planning based on our energy maps, with that of the energy optimization based on global end-to-end energy metrics (which are not available in practice but constitute an upper bound on the performance).
The rest of this chapter is organized as follows: In Section 4.2, we describe the relationship of this work to the current state of knowledge. In Section 4.3, we discuss the key assumptions and how an energy map can be utilized in this chapter. In Section 4.4, we describe our traffic allocation algorithm with an energy map on a finite time horizon. In Section 4.5, we present our simulation results. In Section 4.6, we conclude with a summary of this chapter.

### 4.2 Relationship to Current State of Knowledge

Energy-limited wireless networks have been an active research area in the past decade. The related works are given by the following papers, classified by topic: the capacity of wireless networks [94][95][96], the capacity of wireless networks under mobility [97][98], delay-limited capacity of wireless networks [99][100], energy efficient transmission algorithms [101][102][103], throughput maximization with energy constraints [104][105], energy efficient routing algorithms [23][106][107][108][109][110], information theoretic approaches to wireless relaying [111][112][113][114].

The energy allocation subject to a finite time constraint has been investigated in [104] and [105]. Our work differs from these works in that we seek to minimize the energy consumption in a network of multiple nodes, whereas [104] and [105] maximize the throughput by using only a single transmitter. Further, our framework involves device portability and takes the single transmitter as a special case. In [102], the authors propose an optimal transmission strategy to minimize the energy consumption by using a method called “lazy scheduling”,

95
and in [110], the authors find a rate allocation algorithm in wireless sensor networks subject to a network lifetime constraint. However, neither of these works considers the delay constraints of traffic streams and the portability of networks, both of which constitute a major emphasis of our work\textsuperscript{1}. In [108][109][106][107], the authors propose an efficient routing and scheduling algorithm with limited resources including energy, bandwidth, and the number of transceivers in wireless networks, assuming that the networks are stationary. In contrast, in this work, we emphasize networks in which the devices can move around. To the best of our knowledge, no past work has addressed the energy-efficient allocation problem based on the end-to-end energy measurements in high-mobility networks, and this is the contribution that our work makes in that regard.

Mobility has been shown to increase the throughput capacity of wireless networks [99][97][98][100]. Our work differs from these previous works in the following respects: (1) We model the limited energy supply and delay constraints of each node whereas [99][98], and [103] model only the bandwidth constraints, and [100] models the bandwidth and delay constraints. (2) We focus on the transfer of delay-constrained traffic, whereas in [97], a mobile user transmits delay-insensitive data directly to the destination when it achieves the closest position.

\textsuperscript{1}However, these two works employ a packet-based framework which is more realistic than our flow-based framework. We hope to extend our results to a packet-based framework in the future.
4.3 Basic Assumptions

In this section, we discuss the key concepts and the basic assumptions of this chapter.

We assume that each traffic stream has to be transferred to its destination within a certain amount of time, called “a communication session”, which sets a delay constraint on the traffic stream. Figure 4.2 illustrates the sessions of two nodes in a network. In this example, the deadline for the first session (denoted by \( \Omega_1^{(1)} \)) of node 1 is \( 4T_s \), where \( T_s \) stands for the sampling interval at which we take snapshots of the network. (\( T_s = 5 \) seconds in our simulations as equal to that in Chapter 3.) Further, we assume that each communication session lasts an integer multiple of \( T_s \). We call each discrete-time increment in this picture a “stage” which lasts for \( T_s \) seconds. For each communication session, we assign a fixed number of bits that a node needs to transfer to its destination in a given number of stages that is determined by the communication session duration. We refer
to the amount of traffic transmitted end-to-end from \( i \) to \( BS \), as the “demand” of node \( i \) for the base station \( BS \). Below, the parameter \( D^{(i,BS)} \) denotes the demand of node \( i \) for the destination \( BS \) for the given session. A session of node \( i \) is composed of a fixed number of stages. The index \( m \) denotes a stage in a set of stages \( \Omega_{l}^{(i)} \) in the given session \( l \) of node \( i \). The parameter \( \Omega_{l}^{(i)} \) is given by the duration of session \( l \) of node \( i \) and can vary depending on the session distribution over the time horizon. We assume that each node has the capability to partition its energy supply as well as the total number of bits for a given session, between different network configurations that the node is likely to encounter.

In our framework, a set of nodes is assumed to be randomly deployed over a deployment region. Using the path integration and the information dissemination methods described in Chapter 3, each node can construct an energy map of the entire field over the deployment region. We assume that each node can predict the trajectory that it will take in the near future. The index \( t \) denotes time which is an integer multiple of sampling interval \( T_{s} \), and the index \( k \) denotes a cell number in a set of cells, denoted by \( K \), over the deployment region.

Each node has an associated energy-per-bit cost required to transmit from a fixed cell to the base station. (When an energy map is used, this is the best estimate of the energy cost.) We denote the end-to-end energy cost as \( c_{k}(t) \) for the cell index \( k \) in time \( t \). Each node records its \( L \) most recent energy cost measurements for each cell, where \( L \) is a finite number to avoid recording stale measurements. Note that \( L \) is different from the parameter \( S \) which we used to limit the number of local measurements in Chapter 3. In Chapter 3, \( S \) local measurements per cell were averaged to obtain the averaged transmission vector.
and the averaged link energy for each cell. Based on these estimated local link information of each cell, our path integration constituted an end-to-end energy metric for the cell. However, the new parameter $L$ is the number of end-to-end energy metrics computed via the path integration method. For the purpose of energy planning and optimization, each node keeps the $L$ most recent number of end-to-end energy costs per cell separately from the $S$ raw measurements. In Figure 4.3, we show a cost distribution of the end-to-end energy cost from a cell to the base station for a given sample cell. This distribution shows how the end-to-end energy metric changes on the finite time horizon for each cell. Thus, each node can exploit it to keep track of the end-to-end energy consumption over time. Using the histogram of measurements, we quantize the end-to-end “channel states” for a cell into a number of levels. For example, in Figure 4.3, we can assign three levels: “good”, “medium”, and “bad” to the histogram. (Note
For each channel state of the cell, a node computes the probability \( p_k(t) \) that the node falls into the state, and the average energy cost of the state in the cell \( k \) in time \( t \), given the cost distribution of \( L \) samples. We show how to choose these parameters in Figure 4.4. (We denote these three channel states as \( A \), \( B \) and \( C \), for simplicity throughout this chapter.) In this example, \( \gamma_1 \) and \( \gamma_2 \) are the boundaries between the three states. These parameters are selected by making the areas under the histogram for each state equal. Then, the probability of each state will be same. In this figure, \( \bar{c}_A, \bar{c}_B, \) and \( \bar{c}_C \) are the representative energy costs for the three states, respectively. Intuitively, we can see that when scheduling a transmission, if a node is in a state currently that is far below average (i.e. in a good state) or that is far above average (i.e. in a bad state), then it will either send bits or wait to see how the channel state changes in the future. Or, if it is in a medium state, then the node runs a dynamic program to figure out the optimal way to transmit. In the following section, we will explain how this...
optimal transmission method works.

4.4 Traffic Allocation Algorithm using Dynamic Programming

In the previous section, we explained how a node can extract the statistics of the end-to-end energy costs from the energy maps over time. These statistics will be useful for the node to make decisions for its energy planning and optimization. In this section, based on these statistics, we seek to develop an optimal traffic allocation algorithm for the delay-constrained traffic when future channel states are uncertain. Our algorithm, which is based on Dynamic Programming (DP), is adaptive to the changes in the network and dynamically allocates the optimal amount of bits at each stage. Note that as the network moves through the stages, a new channel state is revealed to the nodes at each stage and the algorithm determines the optimal allocation for that stage in light of this new knowledge as well as the statistics for the remaining stages (until the deadline of a given communication session).

We assume that the single mobile node has many serial sessions to send to the base station (but with possible gaps in between as shown in Figure 4.2) subject to a finite time horizon for each session. For example in Figure 4.2, for the session \( s_1^{(1)} \), node 1 has a demand of \( D_1^{(1,BS)} \) with a delay constraint of \( 4T_s \), where \( T_s \) stands for the stage duration.

Now, we find an optimal strategy (i.e. an optimal traffic allocation) in each stage for this node to transfer its traffic to the base station subject to the delay
Energy-efficient Transmission Utilizing an Energy Map

Chapter 4

constraint. Let a session of the mobile node have a session duration that lasts for $M$ stages with an initial demand of $D$ bits to send to the destination. Then, the problem can be modeled as a stochastic dynamic program with the following recursive optimal-value function [115]: For stage $m \in \{1, 2, \ldots, M - 1\}$ and state $q \in Q$,

$$
e^{(m)}(t) = c_{km}(t) \cdot d^{(m)} + E(Q) \left\{ e \left( z^{(m+1)}, \bar{c}_{km+1,q}(t) \right) \right\}
= c_{km}(t) \cdot d^{(m)} + \sum_{q} p_{km+1,q}(t) \cdot e \left( z^{(m+1)}, \bar{c}_{km+1,q}(t) \right),
$$

$$
e^{(M)}(t) = c_{kM}(t) \cdot z^{(M)}, \quad (4.1)
$$

where $z^{(m)}$ is the state variable at stage $m$, which corresponds to the residual number of bits from the previous stages; the $d^{(m)}$ is the decision variable at stage $m$, which is the number of bits allocated at stage $m$, $\forall m \in \{1, 2, \ldots, M - 1\}$; $z^{(m+1)} = z^{(m)} - d^{(m)}$ for the state variable for the future stage $m + 1$; $c_{km}(t)$ is the cost corresponding to the cell in which the node is currently located in the stage $m$ and in the current time $t$; and $\bar{c}_{km+1,q}(t)$ is the expected cost value for the state $j$ of the cell corresponding to the stage $m + 1$; and $p_{km+1,q}(t)$ is the probability that the state $q$ occurs for the stage $m + 1$ in the current time $t$. Above, for simplicity, we have let $e^{(m)}(t) \overset{\text{def}}{=} e \left( z^{(m)}, c_{km}(t) \right)$.

Figure 4.5 shows an example of a trellis diagram with three stages. Let a node be in state ‘B’ in cell $k_1$ in time $t$. A node decides an optimum solution of $d^{(1)}_{\text{opt}}$ that minimizes the energy consumption function, $e^{(1)}(t)$. In the next stage (time is also incremented as $t + 1$), a node reveals the current energy cost and updates the probabilities and the cost values corresponding to the current time $t + 1$. For example, in Figure 4.5, the probability $p_{k3,A}(t)$ and $\bar{c}_{k3,A}(t)$ are replaced.
by $p_{k3,A}(t+1)$ and $\bar{c}_{k3,A}(t+1)$ in stage 2, respectively. With newly updated values, the node repeats its optimization procedure until the final stage.

We now build an explicit algorithm to carry out the above recursion. Note that the above optimal-value function is the sum of the energy consumption at the present stage and the expected value of the energy consumption at future stages. The aim is to find the optimal number of bits to be allocated at the present stage that minimizes the optimal value function, namely, the expected value of total energy consumption for the given session. That is, the aim is to find the optimal $d_{\text{opt}}^{(m)}$ that minimizes (4.1) at stage $m$. Thus, $d_{\text{opt}}^{(1)}$ is allocated to the first stage from the result of stage 1.

This program is a linear program, and the solution can be found by the graphical method illustrated in Figure 4.6. In this figure, $a$ is the current energy cost and $b$ is the expected cost value for the future stage. For stage 1, the optimal solution is either $d_{\text{opt}}^{(1)} = 0$ or $d_{\text{opt}}^{(1)} = D$. That is, if the current energy cost is less
\[ d^R = D - d^{(1)} \]

**minimize** \[ e^{(1)} = a \cdot d^{(1)} + b \cdot d^R \]

*case 1: \( a < b \)*

*case 2: \( a > b \)*

\[ d^{(1)} + d^R = D \]

**Figure 4.6:** Finding an optimal solution via the graphical method at stage 1.

than the expected energy value of the future stages, then the node assigns all residual bits in the current stage (if the bandwidth is available enough to send the amount). After deciding \( d^{(1)}_{\text{opt}} \), the procedure is repeated and selects \( d^{(2)}_{\text{opt}} \) for a given configuration at stage 2. Continuing in this fashion, the algorithm determines \( d^{(m)}_{\text{opt}} \) at the corresponding stage \( m, \forall m \in \{1, 2, \ldots, M - 1\} \), and sends the remaining bits at the final stage \( S \) with subject to the session demand constraint: \( \sum_{m \in \{1, 2, \ldots, M\}} d^{(m)} = D \). Since the computational complexity is \( O(\alpha \beta) \) where \( \alpha = M - 1 - m \) and \( \beta = 3 \), this algorithm has polynomial complexity and can thus be performed efficiently.

**Structure of the Optimization Program:** We now interpret the above recursive optimization in a simpler way, using a \( \nu \) function which is defined as the minimum cost for each stage:

\[ \nu^{(m)}(t) = \min \{ c_{km}(t), \mu^{(m+1)}(t) \} , \text{ where } \mu^{(m+1)}(t) = E_{(Q)}\{\nu^{(m+1)}(t)\} \]

\[ \nu^{(M)}(t) = c_{kM}(t). \]  \hspace{1cm} (4.2)
Energy-efficient Transmission Utilizing an Energy Map

Figure 4.7: An energy trellis diagram with 3 stages in time \( t \), using \( \nu \) function

<table>
<thead>
<tr>
<th>( \nu )</th>
<th>state A</th>
<th>state B</th>
<th>state C</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \nu_1(t) )</td>
<td>( \bar{c}_{k_3,A}(t) )</td>
<td>( \bar{c}_{k_3,A}(t) )</td>
<td>( \bar{c}_{k_3,C}(t) )</td>
</tr>
<tr>
<td>( \nu_2(t) )</td>
<td>( \min{\mu_3(t), \bar{c}_{k_3,A}(t)} )</td>
<td>( \min{\mu_3(t), \bar{c}_{k_3,B}(t)} )</td>
<td>( \min{\mu_3(t), \bar{c}_{k_3,C}(t)} )</td>
</tr>
<tr>
<td>( \nu_3(t) )</td>
<td>( \min{\mu_2(t), c_{k_3,A}(t)} )</td>
<td>( \min{\mu_2(t), c_{k_3,B}(t)} )</td>
<td>( \min{\mu_2(t), c_{k_3,C}(t)} )</td>
</tr>
<tr>
<td>( \nu_1(t+1) )</td>
<td>( \bar{c}_{k_3,A}(t+1) )</td>
<td>( \bar{c}_{k_3,B}(t+1) )</td>
<td>( \bar{c}_{k_3,C}(t+1) )</td>
</tr>
<tr>
<td>( \nu_2(t+1) )</td>
<td>( \min{\mu_3(t+1), c_{k_3,A}(t+1)} )</td>
<td>( \min{\mu_3(t+1), c_{k_3,B}(t+1)} )</td>
<td>( \min{\mu_3(t+1), c_{k_3,C}(t+1)} )</td>
</tr>
<tr>
<td>( \nu_1(t+1) )</td>
<td>( \min{\mu_2(t+1), c_{k_3,A}(t+1)} )</td>
<td>( \min{\mu_2(t+1), c_{k_3,B}(t+1)} )</td>
<td>( \min{\mu_2(t+1), c_{k_3,C}(t+1)} )</td>
</tr>
<tr>
<td>( \nu_3(t+2) )</td>
<td>( c_{k_3,A}(t+2) )</td>
<td>( c_{k_3,B}(t+2) )</td>
<td>( c_{k_3,B}(t+2) )</td>
</tr>
<tr>
<td>( \nu_1(t+2) )</td>
<td>( \min{\mu_2(t+2), c_{k_3,A}(t+2)} )</td>
<td>( \min{\mu_2(t+2), c_{k_3,B}(t+2)} )</td>
<td>( \min{\mu_2(t+2), c_{k_3,C}(t+2)} )</td>
</tr>
<tr>
<td>( \nu_1(t+2) )</td>
<td>( \min{\mu_2(t+2), c_{k_3,A}(t+2)} )</td>
<td>( \min{\mu_2(t+2), c_{k_3,B}(t+2)} )</td>
<td>( \min{\mu_2(t+2), c_{k_3,C}(t+2)} )</td>
</tr>
</tbody>
</table>

Table 4.1: Computation of \( \nu \) values from \( t \) to \( (t+2) \) for Figure 4.7
Using this $\nu$ function, we illustrate in Figure 4.7 an interpretation of the trellis diagram of Figure 4.5. Further, Table 4.1 shows the computation of $\nu$ values for each stage corresponding to this $\nu$-based trellis diagram. Thus, the optimum strategy in each stage is that a node explicitly compares the current cost with the expected $\nu$ values for the future states, subject to the currently remaining number of bits, and chooses the stage having the minimum energy. If the current cost is higher than the expected $\nu$ value, it simply waits without sending data with subject to its delay constraint. Otherwise, it assigns all residual data to the current stage. Then, the algorithm in (4.1) can be re-written as: For stage $m \in \{1, 2, \ldots, M\}$ and state $q \in Q$,

$$\min \{e^{(m)}(t)\} = z^{(m)} \cdot \nu^{(m)}(t) \quad (4.3)$$

The benefit of this $\nu$-based allocation algorithm is that a node can easily make a decision of transmission via a comparison between the current cost and the $\nu$ value. (We use this $\nu$-based algorithm in our simulations later in this section.) In order to show an example of this $\nu$ function based allocation algorithm, we use Figure 4.8, in which $\nu$ values are computed with sample energy costs and the transition probabilities. In this figure, if a user is currently in state ‘B’ in stage 1, then it compares its current energy cost $c_{k_{1},B} = 1.5$ with the expected cost $\nu_{k_{1},B}^{(1)} = 1.42$ of next stage. This user will wait until the next stage because the expected energy cost in its next stage is less than the current cost. If a user is currently in state ‘B’ but now in stage 2, then the node does not wait but transmits its traffic. We can see that as the delay constraint becomes looser (i.e. the deadline is further away in time), the nodes can wait longer until they obtain a better channel. Such an effect of delay constraints will be will be shown in
There are various session durations and demands for each user in the network. A user needs to relay the traffic from the other node (as an intermediate node). For example, the session distributions of two users can be as in Figure 4.2. In order to implement our algorithm for the multiple users having different session durations (demands), it is necessary to set up the session durations and the demand constraints for each user. Our traffic allocation algorithm on a finite time horizon can be performed using the following procedure:

1. Session setup: for any time $t$:

$$[S, i] = \max_{u \in U(t)} \{l_u\},$$

where $l_u$ is the number of stages left for its current session of node $u$ which
is a multiple of $T_s$; a set $U(t)$ is the set of users having requested relaying their data to the base station at time $t$; and $i$ denotes the node having the maximum number of stages left.

2. Demand Constraints: $\forall u \in U(t) \setminus \{i\}$:

$$d^{(u,BS)}(l_u + 1) = d^{(u,BS)}(l_u + 2) = \cdots = d^{(u,BS)}(l_i) = 0. \tag{4.5}$$

3. Traffic allocation using DP: Each node runs the DP algorithm separately for each session of the multiple number of sessions subject to the demand constraints set-up in step 2.

Assuming that the spatial energy map is available for any node in the network, our traffic allocation is a distributed algorithm which can be performed at each node. In addition, our transmission allocation algorithm can be applied after the construction phase of the energy map.

### 4.5 Simulation Results

In this section, we show the quantitative results of the proposed energy optimization based on dynamic programming described in the previous section, when we allow a node to utilize the energy maps via the methods explained in Chapter 3. We show the quantitative results of the performance improvement when the proposed traffic allocation algorithm is used, and we compare the total energy consumption of our energy planning based on the energy maps with that of an optimal method that uses the actual, optimal end-to-end energy measurements.
As a scheme competing with ours, we introduce a reasonable traffic allocation algorithm for the purpose of performance comparison. This competing scheme is called “Constant Traffic Allocation (CTA)” and divides the session demand of a node by the number of stages and allocates the same number of bits to each stage without utilizing an energy map. We compare the total energies spent in a given session for these two algorithms. In order to show the percentage performance advantage of our DP-based algorithm over the constant traffic allocation algorithm, we define the amount of energy saved as,

\[
\text{Energy Saving \[%\]} = 100 \times \frac{\text{Amount of Energy Saved using our Algorithm}}{\text{Energy Consumed using CTA algorithm}}.
\]

(4.6)

For our simulation scenario description, we again use Figure 4.1. In this figure, a node moves around the deployment region, and it needs to send its traffic to the base station with a delay deadline of \(4T_s\), where \(T_s\) denotes the sampling interval. Assuming that the node can predict its trajectory in the near future, end-to-end energy metrics from the cells where the node will be located can be known via its current energy map of the network. Thus, the main goal of our simulations is two-fold: First, in the presence of energy maps both for the time-invariant case and for the time-varying case, we show how our traffic algorithm can perform over the constant traffic allocation algorithm. As we showed in Section 3.4, the change in the ordering of energy metrics over the geometric surface occurs only for a small fraction of the cells. Due to this roughly same ordering of energy metrics, we expect that the energy optimization over the deployment region is still feasible even for the case of the time-varying energy metrics. Second, by applying the same DP-based allocation algorithm, we compare the total energy consumptions
of our energy maps and an optimal method in which the total energy consumption is computed by using the shortest path algorithm with global information of the network.

In our simulations, 100 mobile nodes are deployed in a square grid divided into 20 by 20 uniform cells. Each side of square grid is 1000 meters. Each node moves according to the Random Waypoint Model for the time-invariant mode, and the Random Waypoint model and the modified contraction model for the time-varying mode; however, with fixed velocity $v = 10$ m/s. The pause times are zero. The path loss exponent $q = 2$ in our simulations. (For consistency, we use the same simulation parameters as in Section 3.4.) Further, we use the most recent raw measurement (i.e., $S = 1$) for constructing the energy maps of our deployment region and a window size $W = 5$ for the dissemination algorithm of the constructed energy maps. For all simulation results in this section, we will show the performance of our traffic allocation algorithm not only with time-invariant energy maps but also with time-varying energy maps of the network.

First, we consider a single session of each user, since we assumed that a single type of traffic stream is transferred for the given session and each session of multiple users can be superposed. The only given parameter of this simulation is the duration of the single session ($M$). The number of measurements ($L$) used for computing the statistics of the cells is 10. (We will also show the effect of this parameter ($L$) later in this section.) The other parameters such as energy costs and the probabilities of three states in a cell are obtained from the dissemination simulations presented in Section 3.4. We compare the performances of our DP-based algorithm for both the cases of the time-invariant energy metrics, and the
time-varying energy metric. Further, we vary the parameter $M$ to observe how the proposed traffic allocation algorithm is affected by the duration of a session. The results are shown in Figure 4.9 and 4.10. Part (a) of each figure displays the normalized energy consumption for the two traffic allocation algorithms, and Part (b) of each figure displays the percentage of energy saving of the DP-based algorithm over the CTA algorithm.

The conclusion of these results is two-fold: First, in both cases, our DP-based transmission algorithm performs better than the constant traffic allocation
Figure 4.10: Energy performance for the time-varying case.
algorithm. Further, we can see that the performance of the DP-based algorithm for the time-varying case is decreased roughly by $10 - 15\%$ with compared to that of the time-invariant case. The reason is that the gap between the energy maps at different time sequences becomes large as the channel is varying (as the energy metric in each cell becomes time-varying). This large gap causes a harmful effect when a node performs the dynamic programming based on the current statistics of the costs for cells in which the node will be located at its future. For example, let a node determine to wait without sending its data in the current stage, according to the current statistics of the future cells. As it arrives in the next cell (of its future trajectories) in next stage, the node updates the statistics based on the newly updated energy map. If the end-to-end energy metrics of future cells in this stage become much worse than those expected at the previous stage, then it could have saved some energy consumption by sending the traffic in the previous stage. In contrast, the CTA algorithm will have sent some amount of traffic in both the previous and the current stage, so that it performs better in this special case. However, the overall performance of our scheme is remarkably better. The energy saving of our scheme over the CTA algorithm is up to $45\%$ for the time-invariant case and $30\%$ for the time-varying case.

Second, the longer the session duration is, the more the DP-based optimal allocation algorithm outperforms the constant traffic allocation algorithm. Subject to a time deadline of $10T_s$, the DP-based traffic allocation algorithm saves the total energy consumption by $45\%$ over the constant traffic allocation method. As the number of stages for a given session increases, both the total energy used and the energy saving of the DP-based algorithm increase. The implication of this
result is that, for traffic that is less delay-sensitive, the user uses the statistics of the network configurations (and can wait until it has a better channel) and allocates the number of bits to the promising stages, while the constant traffic allocation algorithm allocates bits too early, even when it has bad channels. The DP-based traffic allocation algorithm optimizes the traffic allocation for the delay-constrained traffic by exploiting the statistics of the entire energy map obtained by the dissemination algorithm, while the constant traffic allocation algorithm does not benefit from the energy map.

Now, we discuss how the performance of our transmission algorithm is affected by the two parameters: $L$ (which is the number of measurements used), and the number of states. Figure 4.11 and Figure 4.12 show the effects of these parameters for the both time-invariant and time-varying cases. First, for the time-invariant case, as the number of measurements ($L$) and the number of states per stage increase, the performance also increases. Since the variance of energy metrics is small, old measurements may also be useful. Also, by using more number of states, our dynamic programming can be made more accurate. However, we can intuitively see that this effect does not hold for the time-varying case. Due to the fast changing network status, the old data should not be applied. As seen in Figure 4.12, the performance increases as $L$ increases up to $L = 10$, while when $L = 50$, the performance is degraded because of some obsolete measurements. Further, it should be noted that the amount of performance increase is not significant. When we discussed the characteristics of our energy map in Section 3.4, we saw that the shape of energy cup was almost the same over time. Even though there might be a large gap between the magnitudes of energy maps at different
Figure 4.11: Performance with varying the parameters $L$ and the number of states for a given session: time-invariant case.
Figure 4.12: Performance with varying the parameters \( L \) and the number of states for a given session: time-varying case.
sampling times, the cup-shape still remains roughly the same. In other words, when we sort the energy metrics not by the cells but by the magnitude of energy, the order of cells with respect to the magnitudes of energy is not significantly changed. Thus, our dynamic program has similar solutions (when to send) for the different parameter \( L \) since this order does not change much even though the energy potential is fast-varying.

Finally, in order to see the effect of inaccuracy of energy maps on the transmission algorithm, we compare our energy-aware planning based on our path integration and local dissemination algorithm, with an optimized allocation that uses an actual energy map which is obtained by the global, optimal end-to-end measurements. Here, note that we use the same DP-based traffic allocation algorithm for both cases. The simulation has the two modes: time-invariant and time-varying energy metrics as in Section 3.4. We display the results in Figure 4.13. Part (a) of this figure shows the normalized energy expenditures of two methods over the entire simulation time, and Part (b) shows the percentage error of our energy-aware planning over the optimal energy expenditure. According to this figure, the percentage error is roughly between +15% and −15%. The positive errors come from the over-estimation of the energy value of our path integration, while the negative ones come from under-estimated values. This resulting error results from all the errors induced by the path integration and the local dissemination. The implication of this result is that, in an actual system design, we can use an energy map to do the energy-aware planning of data transmission with an accuracy of more than 85%. This energy-aware planning can provide us significant energy savings.
Figure 4.13: Performance comparison of energy planning: optimization based on disseminated energy maps versus that based on actual energy maps with global information on the energy metrics.
4.6 Summary

We presented an energy planning scheme using the DP-based transmission by combining with a path integration and the dissemination method of the energy map. We analyzed this algorithm with relatively large delay constraints suitable for network applications such as delay-tolerant networks. With the available energy map, our DP-based traffic allocation algorithm is aimed at reducing the total energy consumption of a network. The simulation results show that our DP-based scheme reduces the energy consumption by up to 45% over the constant traffic allocation algorithm. Overall, our energy-efficient transmission with a energy map has a percentage error of less than 15% over the optimal transmission which uses the global, minimum end-to-end energy measurements.
Chapter 5

Conclusions

In this thesis, we have addressed how to efficiently manage the energy resource for two types of delay-tolerant networks: underwater ecological sensing networks, and terrestrial, energy-limited, mobile sensor networks.

In Chapter 2, we proposed an energy-efficient MAC protocol that serves as a primer for the development of underwater sensor network MAC protocols that have to operate under long, unknown propagation delays. Most energy-efficient MAC protocols that are currently available for terrestrial sensor networks would not perform well due to the adverse communication environments caused by the long, unknown propagation delays; however, in our proposed protocol, the use of relative time stamps, not only in the transmission of data but also in the establishment of communication with newcomers allows the nodes to operate in a synchronized environment where they can sleep and save energy in the remainder of the time when they are not communicating. The development of such energy-efficient MAC protocols for underwater sensor networks will facilitate the
deployment of hundreds of nodes in this harsh communication medium.

This MAC protocol serves both the development of underwater sensor networks for marine science applications (as pursued for example in Long Term Ecological Research sites [116] in the U.S.), and the development of robust communication protocols for underwater surveillance applications [117][118] where underwater vehicles must operate with high energy efficiency while remaining connected with other vehicles and sensors in their vicinity.

The directions for future research in this area include the development of integrated protocol stacks for underwater communications, that take into account the long, unknown propagation delays in the medium. Such work must successfully integrate the decades of accomplishments in the physical layer underwater acoustic communications with advances at the network layer and is likely to lead to the development of novel protocol stacks that work under large, unknown delays.

In Chapters 3 and 4, we developed methodologies for handling high mobility in delay-tolerant, energy-limited terrestrial networks in order to manage the limited node energy supplies in large-scale, mobile wireless networks.

The novel network design framework in Chapter 3 enables energy-aware planning and optimization in large-scale, highly mobile networks. In the past, it was believed that mobility makes it very difficult to keep track of nodes’ movements, network state, link costs, and in particular, end-to-end energy consumption; however, we have shown that stationarity may be present when we graph the end-to-end energy metrics for a high-mobility network. This stationarity of node mobility over space is represented in our work by the end-to-end “energy potential”. In Chapter 3, we showed how the spatial derivative information is constructed in a
Conclusions

Chapter 5

network by using the path integration method. The speed of individual nodes’
movements determines how fast the information can be spread through a net-
work. In fact, the faster the individual nodes move in the network, the faster
the energy metric information can be shared among the nodes. We can exploit
this node mobility to share and refine energy maps via exchanges upon encoun-
ters. We also showed how this locally measured end-to-end energy information
can be exchanged, fused, and refined to aid in energy-aware planning decisions,
when a high-mobility network exhibits time-varying energy potentials as well as
time-invariant energy potentials.

In Chapter 4, we showed that a node can make smart decisions to transmit its
traffic based on an understanding of the end-to-end energy cost to the base station.
The approach that we sought of developing spatial fields of energy metrics for
highly mobile networks clearly has implications for the design of energy planning
protocols for such networks: Based on these field representations, a node’s energy
planning can be greatly simplified. For the optimality of the decision algorithm,
we used a dynamic program, which in fact turned out to be reduced to the problem
of choosing a minimum value between the current energy cost and the future
expected energy cost at each stage. We showed that our DP-based algorithm
can save energy up to 45% in the time-invariant mode, compared with the traffic
allocation algorithm which sends traffic at a constant rate in the same mode. For
the overall performance of our proposed transmission algorithm under the use of
energy maps, we also showed that our energy-aware planning had a percentage
error of at most 15% when compared with the optimal method which uses global
energy information.
Based on the above results, we would like to give some broader perspectives on the network design paradigm that we have developed. First, we shall develop an analogy between the quality of a wireless point-to-point link and an end-to-end network QoS metric.

In the domain of a physical layer communication problem, a point-to-point wireless communication channel is classified as either a fast-fading or a slow-fading channel, depending on how rapidly the transmitted signal changes, compared to the rate of change of the channel. In a fast-fading channel, the channel impulse response changes rapidly within the symbol duration. That is, the coherence time of the channel is smaller than the symbol period of the transmitted signal. This causes signal distortion within one symbol period, and the channel cannot be estimated accurately in this case. In contrast, in a slow-fading channel, the channel impulse response changes at a rate that is much slower than that of the baseband equivalent of the transmitted signal. In this case, the channel parameters can be tracked and estimated, since the channel can be assumed to be static over subsequent symbol periods.

The coherence time of QoS metrics in a network is similar to the coherence time of the point-to-point channel. This analogy is shown in Table 5. In a network, the end-to-end channel is obtained via a multi-hop communication between a node and its destination. In our case, the exploited channel metric was the end-to-end

<table>
<thead>
<tr>
<th>Point-to-point link concepts</th>
<th>Analogous network concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>coherence time of channel</td>
<td>coherence time of QoS potential metric</td>
</tr>
<tr>
<td>symbol duration</td>
<td>spreading period</td>
</tr>
</tbody>
</table>

Table 5.1: Analogy between point-to-point and network QoS concepts developed in this thesis.
energy potential in space, and we showed that an energy map, which is aimed at energy planning and optimization, could be constructed and distributed fast enough through the network. In this case, the concept of “coherence time” is the duration over which the energy map remains roughly invariant such that it does not become obsolete over time (for the purposes at hand), and the concept of “spreading period” is the duration over which accurate information on energy metrics can be spread through the network. If the coherence time of the energy map is much larger than the spreading period of the network, the end-to-end energy map can be assumed to be invariant over several spreading periods of the network, and hence utilized for energy planning and optimization protocols. This is analogous to a “slow-fading” point-to-point link where the coherence time of the channel is much larger than the symbol duration.

Besides the end-to-end energy metric that we have considered in this thesis, we can potentially extend our results to end-to-end delay which is another important QoS metric. We believe that an end-to-end “delay potential” might also exist over space, since node delays such as buffer delay and processing delay, depend on the node density as well. For a denser area, the node delay would increase since there are normally more requests of relays among nodes. This may lead us to the construction of a “delay map”.

For example, it may be possible for us to construct a delay map via our path integration method described in Chapter 3. In our path integration method for a point \((x, y)\), the sum of the averaged per-hop transmission vector constituted a path from the point \((x, y)\) to the base station, and the sum of the associated per-hop energy metrics on this path constituted the end-to-end energy potential at
The point \((x, y)\). In order to construct a delay potential, we compute an associated delay metric per hop by averaging the local per-hop delay measurements for the point \((x, y)\). In order to explain how to measure these per-hop delays, we use Figure 5.1. In this figure, node \(i\) sends a short packet with its transmission time stamp of \(t_i\) to the neighbor node \(j\). Node \(j\) replies with an acknowledgement which tells the absolute time difference from the time \(t_i\) to the node \(j\)'s transmission time \(t_j\). This time difference corresponds to a local link delay per bit measurement, assuming that timing synchronization of the network is provided by GPS devices. Note that this measurement takes into account not only the transmission delay but also the node delays such as buffer delay and processing delay. Now, the same strategy that we used for the end-to-end energy metric is applied: the local delay measurements are averaged for the cell in which the point \((x, y)\) is located, and the sum of the associated average delay metrics on the path from the point \((x, y)\) to the base station, constitutes the end-to-end delay potential at the point \((x, y)\).

\(^1\)Here, we assume terrestrial RF networks with negligible propagation delays. The methods of Chapter 2 may be applied for underwater scenarios.
In the dissemination method, each node adds the delay measurements into the measurement bank. Then, we are able to construct a delay map which is a map of end-to-end delay per bit metrics over space, by using this extended version of the path integration method and the same dissemination algorithm including the delay measurements. Note that for the delivery of data, the data packet length would be considered as well, since the length of packet will increase all types of delays. A careful exploration would be necessary for this data delivery, because delay does not necessarily increase as a linear function of the packet length. This analysis assumes that steady-state congestion characteristics would exist and have a stationary distribution over space under the RWP model. Time-varying congestion can also be explored and its effects quantified by the methods in Chapter 3.

For the applications of delay maps, the same analogy of the relationship between the coherence time of the end-to-end energy metric and the spreading period can be applied. If the coherence time of the end-to-end delay metric is much larger than the spreading period of the network, then we have a high-mobility network for which delay-aware routing is possible.

We can connect the notion of spreading the information with the notions of QoS metrics such as the end-to-end energy metric and the end-to-end delay metric through space: if the end-to-end QoS metrics remain roughly invariant through paths in space for durations on the order of the time that is required for the delivery of data, then this QoS information can be spread fast enough through the network, and hence enables improved QoS-aware protocols in future wireless networks.
In order to describe another broader impact of our network design paradigm, we would like to discuss how our energy map can be exploited and can enable efficient energy planning decisions for actual networks such as sensor networks having several base stations and mobile ad hoc networks where no fixed base station is deployed. For a sensor network having several collection sites (a.k.a base stations), the application of our energy map is straightforward. Each sensor node would construct an energy map for each collection site, and for the data delivery at hand, it would utilize the map for the closest collection site to do its energy planning by the methods in Chapter 4.

Now, we turn our attention to ad hoc networks and discuss how the framework that we have developed can be utilized. For ad hoc networks, there are mainly two challenges to perform energy planning using energy maps. In these networks, the key problem is that the destination might be mobile. Hence, a source node might need to estimate the location of the destination node, which induces a considerable control overhead. Even though the source can build an energy map for a point where the destination is currently located, the energy map for that point would not be valid any longer if the destination moved far away from that point. This is a problem in case a node cannot construct another energy map fast enough for the new position.

As a possible solution to this problem in ad hoc networks, we consider a design strategy that we call, a “coarse energy planning”. In coarse energy planning, when a source node cannot obtain the exact position of its mobile destination, it uses a coarse energy map with an estimate of the location of the destination. We expect that this coarse energy map might still be very useful to save energy via
energy planning for data delivery. Here, we shall utilize the correlation between
the encounter times and the distance developed for EASE [119], FRESH [120],
and MAID [121]. These works showed that the use of this node encounter his-
tory can significantly reduce control overhead that is induced to estimate nodes’
locations, compared to other on-demand based methods. A node can estimate
the location of node $D$ as follows: Every node builds its “encounter history” ta-
ble and exchanges the encounter time and location information with other nodes.
The node that has encountered node $D$ the latest (among all the nodes) will have
more accurate information on the location of node $D$. Hence, by disseminating
the encounter table, a node sorts out the encounter time history and picks the
location information obtained from the latest encounter node. This way, a node
keeps track of the position of its destination node.

In order to explain our coarse energy planning for ad hoc networks, we use
Figure 5.2. Let node $D$ be the destination of a source node $S$. In this figure, there
are 16 reference points in the center of each square region in a 4 by 4 uniform grid.
Note that these reference points are not fixed nodes that are deployed but rather
the physical locations for the purpose of collecting field information toward these
locations. Now, all the nodes make their local measurements with respect to each
of these fixed locations by choosing the link that minimizes the Joules per bit-
meter metric for each reference region, and spread the measurements. Then, each
node will have all the measurements for each reference point. When node $S$ has
an estimate of the location of the destination by looking at the encounter history
table, it chooses the measurements that corresponds to the reference region in
which the estimated location falls and constructs its coarse energy map via the
path integration for that location. In the figure, let node $S$ know that node $D$ was recently in a cell near the reference point $R3$ (via its encounter history table). Then, node $S$ refers to all the measurements corresponding to the reference region and constructs a coarse energy map via path integration. As node $D$ moves around the network, let node $S$ have the information that node $D$ moved to other position near the reference point $R7$. In this case, node $S$ can refer to the available raw measurements for the reference point $R7$ and simply performs path integration for this new position. Utilizing this currently available energy map, the node performs its energy planning. This energy planning might not be optimal, because the estimate of the location of the destination might not be exact when the decisions are made. In order to improve the performance of energy planning, we would allow the intermediate nodes to do their own energy
planning based on their best estimates for the delivered packet subject to its delay constraint. The reason is that a closer intermediate node to the destination might perform energy planning better than the further ones, because a closer node (due to more recent encounters) would have more accurate information on the position of the destination.

Using the above method, even though the destination moves around the network, a node is able to construct a coarse energy map on demand based on its best estimate of the location of the destination and does its energy planning by utilizing this energy map. However, note that since each node updates the raw measurements for each reference point, the overhead will be proportional to the number of reference points. A rigorous evaluation will be needed to measure how much accuracy can be obtained at the cost of overhead increases due to the number of reference points. In addition, there is a limitation of the application of the energy maps to mobile ad hoc networks. If the destination node moves faster than what can be estimated, then the available energy map would be obsolete. An extension would be necessary in order to develop a new methodology of measuring the link costs which can be re-used for any point in a network.
Bibliography


[66] A. Fei and M. Gerla, “Smart forwarding technique for routing with multiple


[96] V. Rodoplu and T. H. Meng, “Computation of core capacity of wireless


