

ADAPTIVE SCALABLE PROTOCOLS FOR HETEROGENEOUS WIRELESS NETWORKS

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Computer Science

by

Vamsi Paruchuri

B.S., Sri Venkateswara University, 2001

M.S., The Ohio State University, 2003

August 2006

To my parents,
Kusuma and Ramana.

Acknowledgments

First and foremost, I would like to earnestly thank my advisor, Dr. Arjan Durrezi, for taking me on as a student about five years ago, even though he knew little about me at the time. It was an extraordinary piece of good fortune that led to my becoming his student. He has been an ideal advisor in every respect, both in terms of technical advice on my research and in terms of professional advice. The immense trust he placed in my abilities was always a great source of motivation. I hope that I can live up to his high standards.

I would like to express my gratitude to Dr. Jianhua Chen, Dr. Bijaya Karki and Dr. J Ramanujam for being on my Ph.D. committee and helping me improve my thesis with their profound and inspiring comments.

I benefited greatly from the technical and career advice given to me by Dr. Sitarama Iyengar, Dr. Raj Jain and Dr. Hsiao-Chun Wu. I am grateful to them for this and look forward to interacting with them in the future. I would also like to thank the entire faculty, the staff, and friends in the CS Department, at LSU, who have made my stay a memorable one.

The more important thanks are reserved for the last. Thanks to my love Chandu for everything, and everything that is beyond words. I always thank her and her sister, Radhika, for bringing so many joys into my life.

I would like to express my utmost gratitude to my brother and his wife, Dileep and Madhuri. I would not have achieved this goal without their constant support, concern and motivation in the past, the present, and the future.

I owe a special debt of gratitude to my mom and dad, Kusuma and Ramana. They have, more than anyone else, been the reason I have been able to get this far. Words cannot express my gratitude to my parents, who give me their support and love from across the seas. They instilled in me the value of hard work and taught me how to overcome life's disappointments. Their selfless support and love always make me want to excel.

Finally, and most importantly, I thank God for all the incredible blessings I am receiving in my life.

Table of Contents

Acknowledgments	iii
Abstract	viii
1 Introduction	1
1.1 Some Wireless Scenarios	1
1.1.1 Constellation of Wireless Devices	2
1.1.2 Pervasive Systems and Sensor Networks	2
1.1.3 Emergency Ad hoc Cellular Networks	3
1.2 Network Requirements and Protocol Design Issues	4
1.2.1 Cross Layer Design Principle	6
1.3 Research Objectives and Solutions	8
1.3.1 Analytical Modeling for Transmission Power Control	8
1.3.2 Broadcasting	9
1.3.3 Routing and Energy Management	9
1.3.4 Efficient Topology Control	10
1.3.5 Adaptive Clustering	11
1.3.6 Anonymous Communication	12
1.3.7 Lightweight Data Integrity	13
2 Models to Adapt Protocols to Network Conditions	14
2.1 Related Work	15
2.2 Modeling Impact of Collisions on Broadcast Messages	16
2.2.1 Assumptions	17
2.2.2 Optimal Range in Presence of Collisions	18
2.2.3 Estimation of Probability of Collision	21
2.3 Modeling Impact of Collisions on Unicast Messages	22
2.3.1 Back off Characterization	22
2.3.2 Queuing Delay	23
2.3.3 Total Service Time	24
2.4 Energy Model	24
3 Optimized Flooding Protocol	25
3.1 Related Work	26
3.2 Background	28
3.2.1 The Covering Problem	28
3.2.2 The Modified-Covering Problem	29
3.2.3 Number of Transmissions in Ideal Scenario	30
3.3 Optimized Flooding Protocol	32
3.3.1 Our Approach	33

3.3.2	OFP without Neighborhood Knowledge	34
3.3.3	OFP with Neighborhood Knowledge	35
3.4	Analysis of OFP	36
3.4.1	Bounds on the Performance of OFP	36
3.4.2	Effect of Threshold Th	37
3.4.3	Forwarding Distance - Effective Range	38
3.4.4	Time Taken for Broadcasting the Entire Network	39
3.5	Ensuring Broadcasting Reliability	40
3.5.1	Number of Messages Received by a Node	40
3.5.2	Relation between Desired Reliability and Range	41
3.6	Modeling Impact of Transmission Losses	41
3.6.1	Forwarding Distance - Effective Range in Presence of Trans- mission Errors	41
3.6.2	Expected Increase in Number of Transmissions due to Errors	42
3.7	Energy Consumption and Transmission Range	43
3.8	Reliable Broadcast	44
3.8.1	OFP with Global Adaptation (OFP-GA)	45
3.8.2	OFP with Local Adaptation (OFP-LA)	45
3.8.3	Energy Balancing	46
3.9	Experimental Results	47
3.9.1	Effect of Threshold Th	48
3.9.2	OFP Efficiency	48
3.9.3	Mobile Networks	51
3.9.4	Effect of Non-Uniform Radio Propagation	52
3.9.5	Adapting to the Network Conditions	54
3.9.6	Energy Balancing	57
3.10	Broadcasting in Three Dimensional Networks	60
3.10.1	Background	62
3.10.2	Three Dimensional Broadcast Protocol - 3DB	64
3.10.3	Simulation Results	66
3.11	Summary	71
4	Adaptive Routing and Energy Management for Heterogeneous Wireless Networks	72
4.1	Related Work	74
4.2	Adaptive Routing and Energy Management (AREM)	77
4.2.1	Adaptive Energy Management (AEM)	78
4.2.2	Adaptive Routing Mechanism Based on Forwarding Sets (ARM)	78
4.2.3	The AREM Protocol	83
4.2.4	AREM Adaptation to Energy Levels (AREM-E)	84
4.3	Analytical Model	85
4.3.1	Average Path Length	85
4.3.2	Average Packets in the Network	86
4.3.3	Sleep Delay Characterization	86

4.3.4	Total Service Time	87
4.3.5	Average Duty Cycle	87
4.3.6	Energy Consumption	88
4.4	AREM - Load Sensitivity	88
4.4.1	AREM with Global Adaptation (AREM-GA)	88
4.4.2	AREM with Local Adaptation (AREM-LA)	89
4.5	Experimental Results	90
4.5.1	Effect of Sleep Duration	90
4.5.2	Performance with Varying Loads	91
4.5.3	Validation of Analytical Model	92
4.5.4	Performance Study of AREM with Range Adaptation	94
4.5.5	Performance in Presence of Heterogeneous Energy Levels	97
4.6	Summary	98
5	An Efficient Coordination Protocol for Heterogeneous Wireless Networks	100
5.1	Related Work	102
5.2	Problem Statement and Background	104
5.2.1	Problem Statement	104
5.2.2	Background	105
5.3	Efficient Coordination Protocol	105
5.4	The Protocol	107
5.4.1	Selection of Th	108
5.4.2	Energy Balancing through Rotation	109
5.4.3	Load Adaptive Backbone Formation (ECP-A)	109
5.5	Analysis of ECP	111
5.5.1	Backbone Structure	111
5.5.2	Arrival Rate	113
5.5.3	Average Number of Hops	113
5.5.4	End-to-End Delay	114
5.5.5	Time Taken for Backbone Formation	114
5.6	Performance Evaluation	116
5.7	Summary	123
6	Adaptive Clustering Protocol for Wireless Networks	124
6.1	Related Work	125
6.2	Adaptive Clustering Protocol	127
6.2.1	Hexagonal Clustering Protocol	128
6.2.2	Cluster Reconfiguration	130
6.2.3	Adaptive Clustering Protocol	131
6.2.4	Adaptive Clustering Protocol (ACP)	131
6.3	Performance Evaluation	133
6.3.1	Ideal Case Scenario	133
6.3.2	Effect of Threshold Th	133

6.3.3	ACP Efficiency	134
6.3.4	Distortion	136
6.3.5	Average Delay per Hop	137
6.3.6	Performance Comparison	138
6.3.7	Energy Balancing	139
6.3.8	Adaptation to Network Conditions	139
6.4	Summary	141
7	A Hierarchical Anonymous Communication Protocol for Heterogeneous Wireless Networks	143
7.1	Related Work	144
7.2	Design Goals and Network Model	145
7.2.1	Design Goals	145
7.2.2	Network Model	146
7.3	Hierarchical Anonymous Communication Protocol (HACP)	148
7.3.1	Anonymous Communication with in a Cluster	148
7.3.2	Anonymous Communication between Cluster Heads	148
7.3.3	Multiple Rings	149
7.4	Performance of HACP	151
7.4.1	Metrics	151
7.4.2	Communication Overhead	152
7.4.3	Data Exposure Index	153
7.4.4	Mean Waiting Time	154
7.5	Summary	155
8	Lightweight Data Integrity Protocol for Wireless Networks	157
8.1	Related Work	159
8.2	Data Integrity-Lightweight Network Layer Security	160
8.2.1	Assumptions	160
8.2.2	The Protocol	161
8.3	Analysis and Results	162
8.3.1	Bandwidth Overhead	162
8.3.2	Computational Overhead	163
8.4	Summary	163
9	Conclusions	165
	References	167
	Vita	179

Abstract

The focus of this dissertation is to propose analytical models to study the impact of collisions and interference in heterogeneous wireless networks and propose simple scalable and lightweight protocols that use these models to adapt to network conditions thus increasing efficiency, decreasing energy consumption and prolonging network lifetime.

The contributions of this dissertation are multifold and are summarized as follows:

- Analytical models to study the impact of collisions and interference on both broadcast and unicast messages. These analytical models are incorporated into the proposed protocols to adapt to the prevailing network conditions to improve their performance.
- Optimized Flooding Protocol (OFP) a geometric approach to achieve network wide broadcast of messages. The key advantages are - simple and stateless, minimizes the number of retransmissions and more importantly ability to adapt to network conditions to guarantee required reliability criteria. OFP is also extended to 3D networks and the performance is verified through rigorous simulations.
- Adaptive Routing and Energy Management (AREM), an integrated routing and MAC protocol that uses the concept of random wakeup and forwarding set based routing to simultaneously conserve energy and achieve low latencies. Nodes adapt their transmission power to the prevailing network conditions to operate at optimal conditions, thus further improving the network lifetime and reducing latencies.
- Efficient Co-ordination Protocol (ECP) that exploits high node redundancy to elect a small subset of nodes to perform network tasks. The subset of nodes is periodically rotated and each node is active for a duration proportional to its capabilities. The load is uniformly distributed among all nodes.

- Adaptive Clustering Protocol (ACP), an efficient stateless scalable clustering protocol that adapts to network conditions and balances load among nodes.
- Hierarchical Anonymous Communication Protocol novel protocol that prevents traffic analysis from revealing node information including its location.
- Lightweight security protocol to preserve the integrity of messages in a wireless network even in presence of compromised nodes.

Chapter 1

Introduction

The best way to have a good idea is to have lots of ideas.

- *Linus Pauling*

If I have seen further it is by standing on the shoulders of Giants.

- *Sir Isaac Newton*

Wireless and mobile networks represent an increasingly important segment of networking research as a whole, driven by the rapid growth of portable computing, communication and embedded devices connected to the Internet. Overall, it is clear that mobile, wireless and sensor devices will certainly outnumber wired end-user terminals on the Internet in the near future, strongly motivating consideration of fundamentally new network architectures and services to meet changing needs.

Over the next 10-15 years, it is anticipated that significant qualitative changes to the Internet will be driven by the rapid proliferation of mobile and wireless devices, which may be expected to outnumber wired PC's as early as 2010. The potential impact of the future wireless Internet is very significant because the network combines the power of computation, search engines and databases in the background with the immediacy of information from mobile users and sensors in the foreground.

Wireless networks are of a fundamentally different character: To begin with, wireless connections are by nature significantly less stable than wired connections. Effects influencing the propagation of radio signals, such as shielding, reflection, scattering, and interference, inevitably require routing systems in ad hoc networks to be able to cope with comparatively low link communication reliability. Also, many scenarios for ad hoc networks assume that nodes are potentially mobile.

1.1 Some Wireless Scenarios

The revolutionary advances in the wireless communication technologies are enabling the realization of a wide range of heterogeneous wireless systems. This technological development is further inspiring the researchers to envision several

scenarios: Constellation of Wireless Devices (Mobile Ad hoc Networks), Pervasive Systems and Sensor Networks, and Emergency Ad hoc Cellular Networks.

1.1.1 Constellation of Wireless Devices

A Mobile Ad hoc Network consists of wireless Mobile Nodes (MNs) that cooperatively communicate with each other without the existence of fixed network infrastructure. Depending on different geographical topologies, the MNs are dynamically located and continuously changing their positions. The fast-changing characteristics in ad hoc networks make it difficult to discover routes between MNs. It becomes important to design efficient and reliable multihop routing protocols to discover, organize, and maintain the routes in ad hoc networks.

An area where there is much potential for wireless technologies to make a tremendous impact is the area of vehicular ad hoc networks (VANET). There are numerous emerging applications that are unique to the vehicular setting. For example, safety applications would make driving safer; driver information services could intelligently inform drivers about congestion, businesses and services in the vicinity of the vehicle, and other news. Mobile commerce could extend to the realm of vehicles. Existing forms of entertainment may penetrate the vehicular domain, and new forms of entertainment may emerge.

Ad hoc radio constellations also apply to civilian disaster recovery and in tactical defense environments. These applications usually involve communications between a number of first responders or soldiers who work within close proximity of each other. The response team may need to exchange text messages, streaming media (e.g. voice or video), and use collaborative computing to address a shared task such as target recognition or identification of a spectral jammer. Individual nodes may also need to access the Internet for command and control purposes or for information retrieval. This application has similarities with the ad hoc mesh network for suburban or rural broadband access mentioned earlier.

1.1.2 Pervasive Systems and Sensor Networks

Recent advances in wireless communications and microelectro-mechanical systems have enabled the development of extremely small, low-cost sensors that possess sensing, signal processing, and wireless communication capabilities. These sensors can be deployed at a much lower cost than that of traditional wired sensor

systems. An ad hoc wireless network of large numbers of such inexpensive but less reliable and accurate sensors can be used in a wide variety of commercial and military applications such as target tracking, security, environment monitoring, and system control.

Wireless sensor networks are expected to be the basic building block of pervasive computing environments. Aggregating sensor nodes into sophisticated sensing, computational and communication infrastructures to form wireless sensor networks will have a significant impact on a wide array of applications ranging from military, to scientific, to industrial, to health-care, to domestic, establishing ubiquitous computing that will pervade society redefining the way in which we live and work.

1.1.3 Emergency Ad hoc Cellular Networks

A cell phone is essentially a battery-powered microprocessor with one or more wireless transmitters and receivers optimized for voice I/O. Even a bare-bones model provides a keyboard, an LCD screen, and a general-purpose computing platform, typically supporting Java2 Mobile Edition (J2ME) or .NET Compact APIs. More sophisticated models provide a camera, 1MB-5GB of local storage, a full-color screen, multiple wireless interfaces, and even a QWERTY keypad.

Today's cellular networks use fixed infrastructures, which are vulnerable to the disaster effects like hurricanes and terrorist attacks. One scenario is that cellular phones switch to an ad hoc mode when their fixed infrastructure is no longer functioning. The advantage of using cellular phones in disaster/emergency conditions is that everyone has one; therefore, the communication tools will be always ready, even when the unexpected happens.

It is very important to consider conditions and restrictions created by emergency and disaster situations. For example, in disaster conditions, which duration is unpredictable, saving energy becomes an important goal, as it may be impossible to charge cellular phones. Another critical issue during natural or man made disasters is that the situations changes rapidly, in most of the cases in unpredictable ways, and it is almost impossible, using the normal channels of communication, to avert and direct the population. For example people trying to escape from flooding caused by hurricanes, may choose damaged roads, bridges, or tunnels that could become mortal traps. Another scenario is that of a terrorist attack in a subway or

a building. People trying to escape via a more obvious way, may go toward closed exits, even more dangerous locations such as fire and poison. Terrorists may plan their attacks by taking into account the victims' most likely reaction. Therefore, we need protocols that enable quick and efficient delivery of information to people. The source of information could be other users, officials, or generated by sensing devices. Finally, we need ways to guarantee communication and interoperability between the area under disaster and the unaffected areas.

1.2 Network Requirements and Protocol Design Issues

Wireless communication is much more difficult to achieve than wired communication because the surrounding environment interacts with the signal, blocking signal paths and introducing noise and echoes. As a result wireless connections have a lower quality than wired connections: lower bandwidth, less connection stability, higher error rates, and, moreover, with a highly varying quality. These factors can in turn increase communication latency due to retransmissions, can give largely varying throughput, and incur high energy consumption. In this section, we discuss a set of protocol design issues related to the networking requirements of the representative wireless scenarios identified earlier.

- **Quality of Service**

Since wireless networks deal with the real world processes, it is often necessary for communication to meet real-time constraints. In battle surveillance systems, for example, communication delays within sensing and actuating loops directly affect the quality of enemy tracking. Due to the nature of the wireless communication and unpredictable traffic pattern, it is infeasible to guarantee hard real-time constraints, however, research that provides probabilistic guarantee for timing constraints is quite achievable and essential.

- **Heterogeneity**

In contrast to most stationary computers, mobile device encounter more heterogeneous network connections. As they leave the range of one network transceiver they switch to another. In different places they may experience different network qualities. There may be places where they can access multiple transceivers, or even may concurrently use wired access. The interface may also need to change access protocols for different networks, for example when switching from wireless LAN

coverage in an office to cellular coverage in a city. This heterogeneity makes mobile computing more complex than traditional networking.

- **Large Scale**

Smart hospitals, battlefields and earthquake response systems are applicable sensor network systems. Such systems require a large geographic coverage. At the same time, a high density is required to work against the high failure rate of sensor nodes, the low confidence in individual sensor readings, the limited communication range and low capability of single sensor nodes. Due to these reasons, sensor networks are expected to scale up to thousands and millions of nodes, two orders of magnitude larger than traditional ad hoc networks.

- **High Unpredictability**

Sensor network applications are driven by environmental events, such as the earthquake and fire, anywhere anytime following an unpredictable pattern. Sensor node failures are common due to the sheer number of sensor nodes and the hostile environment. The radio media shared by densely deployed nodes is subject to heavy congestion and jamming. High bit error ratio, low bandwidth and asymmetric channel make the communication highly unpredictable. Such unpredictability usually prevents off-line design of system parameters. Online monitoring and feedback control are required to provide a certain degree of QoS guarantee under such situations.

- **Robust Data Delivery under Failure and Mobility**

Sensor networks are faulty networks where failures should be treated as normal phenomena. Unreliable nodes, constrained energy, high channel bit error ratio, interference and jamming, multi-path-fading, asymmetric channel and weak security make the communication highly unreliable. At same time, sensor networks are highly dynamic networks where network topologies are constantly changing due to a high rate of node failure, changes of power modes, and nodes' mobility. It is a challenging research problem to provide a robust data delivery under such a situation.

- **Energy-efficiency**

The wireless network interface of a mobile computer consumes a significant fraction of the total energy of a mobile computer. More extensive and continuous use

of network services will aggravate this problem. Energy efficiency can be improved at various layers of the communication protocol stack.

- **Adaptiveness**

Wireless networks is challenging because of the unpredictable behavior of the medium and the proactive effect of interference. Compared to the wired networks the degree of variability of the state of wireless networks is quite high. Also the performance of the network, in terms of delay and throughput, is highly dependent upon the state of the network. The effects of the state of a wireless network are spread across several layers. Thus in order to meet the requirements of the application despite variable link state, network topology and power levels, it is important that the layers coordinate and adapt to the change in network state.

To deal with the dynamic variations in networking and computing resources gracefully, both the mobile computing environment and the applications also need to adapt their behavior depending on the available resources including the batteries.

1.2.1 Cross Layer Design Principle

One of the major components in the success of the Internet is the layered open system interconnection (OSI) architecture. The modularity achieved through layering leads to better understanding of the abstract functionality of layers and thus enables better understanding of the overall system. The interfaces between the layers are static and independent of individual network constraints and applications. But, layering is inflexible because the developer of a new application has to rely solely on the functionality of the lower layers.

Ad hoc networks are inherently more dynamic than wired networks. Traditional protocols designed for wired networks therefore generally fail to satisfy the requirements of wireless ad hoc networks. The layers in a wireless network must coordinate and adapt with the change in the state of the wireless network. This is the motivation behind the cross layer paradigm for protocol design in wireless networks. The cross layer approach is perceived as one of the efficient solutions for designing protocols for the wireless networks. The cross layer design aims to achieve adaptivity and optimal performance by allowing sharing of information across several layers.

The cross layer design of protocol stack enables layers to exchange state information in order to adapt and optimize the performance of the network. The sharing of information enables each layer to have a global picture of the constraints and characteristics of the network, leads to better coordination and enables them to take decisions that would jointly optimize the performance of the network. The cross layer principle further requires that the protocols must not be developed in isolation but in an integrated and hierarchical framework so as to take advantage of the interdependencies between the protocols. These interdependencies are related to the adaptivity at each layer, system constraints and requirements of the application.

In cross layer architecture, the MAC layer may adapt its scheduling based on the link quality and interference such that the performance constraints of the application are satisfied. Thus the MAC layer needs to have information about the link characteristics from the link (lower) layer and the performance constraints from the application (upper) layer. Similarly a adaptive cross layer routing protocol may choose the routes based on the information about the link characteristics and the MAC scheduling policy in order to meet performance requirements.

It is important to understand that in order to adapt to a change in the network a layer must first try local adaptation and inform the upper layer about the change only if the local adaptation does not work. This is because the time-scale of changes at lower levels is much lower than the time-scale of changes at the upper layers. For example, the SINR of a link may change much more rapidly than the position of a node. So when the quality of a link degrades the link layer must first try to adapt to the change, possibly by increasing the transmit power or using better coding. This would temporarily solve the problem if the change in SINR is due to a random fluctuation and the SINR of the link would later be restored. However if the SINR of the link does not improve for a long time then the link layer realizes that this degradation may be due to a change of the topology, so it informs the network layer that something has gone wrong with the link. The network layer then recalculates the routes using this information.

1.3 Research Objectives and Solutions

In order to realize these next generation heterogeneous wireless networks introduced in previous sections, the communication challenges posed by each of these environments must be effectively addressed. In this thesis, new advanced transport protocols are developed for the next generation heterogeneous wireless network architectures. The approach made in our research was to study practical solutions to the inherent problems of handheld multimedia terminals. In this field too often, system architectures, protocols, and applications are developed with a theoretical background only and with a limited scope covering one horizontal layer in a system. In contrast, this research is characterized by a strategy that traverses vertically through various layers of the system architecture.

The chapters are largely based on papers presented at conferences and published in journals. The structure of the thesis is guided along these papers.

The following six areas are investigated under this research and each of them is described in the following subsections.

1.3.1 Analytical Modeling for Transmission Power Control

Transmit power control refers to the problem of selecting appropriate power level for transmission of each packet. Transmit power control is an important problem in wireless ad hoc networks because of various reasons. Most of the mobile ad hoc networks have battery powered nodes, so lifetime of network depends on the power that a node consumes for transmitting packets. Also the SNR at a node depends upon the transmit power levels of the neighboring nodes. Low transmission power might also reduce interference, thus by reducing the collisions the latency can be reduced.

We develop analytical models to derive optimal transmission power/range for both broadcast and unicast messages. In case of broadcast messages, the objective is to ensure a given ratio of nodes receive the broadcast packet, while in case of unicast messages, the objective is to minimize the latency. We also present the energy trade-offs and let the network administrator to choose the transmission range depending on the network requirements. We then incorporate these models into other protocols we propose to enhance the performance and to adapt to the prevailing network conditions.

1.3.2 Broadcasting

Broadcasting is the process in which one node sends a packet to all other nodes in the network. Many applications as well as various unicast routing protocols use broadcasting or a derivation of it. Applications of broadcasting include location discovery, establishing routes and querying. Broadcasting can also be used to discover multiple paths between a given pair of nodes. Considering its wide use as a building block for other network layer protocols, the broadcast methodology should deliver a packet from one node to all other network nodes using as few messages as possible.

The simplest method for broadcast service is flooding. Its advantages are its simplicity and reachability. However, for a single broadcast, flooding generates abundant retransmissions resulting in battery power and bandwidth waste.

In this thesis, a new broadcast protocol, Optimized Flooding Protocol (OFP) for Heterogeneous Wireless Networks is presented. OFP requires minimal neighborhood information; neither the neighboring node addresses nor their locations are needed. The nodes need to know only their own position. OFP is based on a geometric approach and adapts itself to local radio propagation conditions. OFP is fully distributed and very scalable to the change in network size, node type, node density and topology. OFP accommodates seamlessly such network changes.

An analytical framework that analyzes the performance of OFP and optimizes OFP depending on the use requirements, network resources and environment conditions, is also developed. The framework considers the radio channel characteristics. In particular we will optimize OFP by using non-isotropic radio models. Performance evaluation via simulation experiments validates the analytical model.

We also propose Three Dimensional Broadcast Protocol (3DB) an extension of OFP for three-dimensional networks. The protocol is performed in an asynchronous and distributed manner by each node in the network. The efficiency of 3DB remains very high even in large networks and 3DB scales with density.

1.3.3 Routing and Energy Management

Routing in a communication network is the process of forwarding a message from a source host to a destination host via intermediate nodes. As elaborated earlier, wireless ad hoc networks are fundamentally different from wired networks: To

begin with, wireless connections are by nature significantly less stable than wired connections. Effects influencing the propagation of radio signals, such as shielding, reflection, scattering, and interference, inevitably require routing systems in ad hoc networks to be able to cope with comparatively low link communication reliability. Also, many scenarios for ad hoc networks assume that nodes are potentially mobile.

Apart from the above factors, more importantly, nodes might not participate in routing all the times (primarily to save energy). Another critical challenge is that the protocols need to adapt to the ever changing wireless environment including the traffic loads, energy levels and node failures for efficient performance and prolonging network lifetime. The cross layer approach is perceived as one of the efficient solutions for designing protocols for the wireless networks. The cross layer design aims to achieve adaptivity and optimal performance by allowing sharing of information across several layers.

We present Adaptive Routing and Energy Management (AREM), a novel power management and routing protocol for heterogeneous wireless networks. While reducing energy consumption is the primary goal in our design, AREM protocol also achieves good scalability and low latency. To achieve the primary goal of energy efficiency, we reduce idle listening by making the nodes operate at low duty cycle modes. To reduce latency, AREM uses the concept of forwarding sets.

We also develop an analytical model to deduce the optimal transmission range taking into consideration the node density and transmission rates. The optimization criterion is the end-to-end latency. AREM enables nodes to adapt the optimal transmission ranges to the prevailing network conditions, thus yielding better performance results. AREM also evenly balances the load among the nodes based on their residual energy levels, thus simultaneously prolonging both individual node lifetime and the network lifetime.

1.3.4 Efficient Topology Control

In wireless networks channel is usually shared among many hosts. Sharing increases the complexity of route discovery, reduces the network performance, and increases energy consumption due to aggravated radio interference. Topology control addresses these problems. Topology control optimizes network topology and reduces routing cost by restricting the connections among pairs of hosts.

One approach of topology control is to exploit the node redundancy in wireless networks. A subset of nodes can be selected to serve as the coordinators through which all nodes can, directly or indirectly, communicate with each other. The coordinators form the backbone of the network. The nodes that are not in the backbone have at least one neighboring node that is in the backbone. The non-backbone nodes that do not have active communication can safely go to sleep to save energy.

We present Efficient Coordination Protocol (ECP), an algorithm of constructing backbone in ad hoc wireless network for energy conservation. ECP employs a geometric approach and extends the Covering problem for this purpose. Also, ECP uses a simple technique to rotate the backbone nodes in order to balance the energy across the whole network. ECP constructs backbones that are smaller, it results in energy savings that translate into extended network lifetimes, and at the same time ECP does not deteriorate network performance. We have validated these results through both analytical and simulation results.

1.3.5 Adaptive Clustering

Efficiently organizing nodes into clusters is an important application in wireless networks. Clustering divides the network into disjoint subsets, wherein a node from each subset is elected to represent that cluster. Many proposed protocols for both sensor networks and ad-hoc networks rely on the creation of clusters of nodes to establish a regular logical structure on top of which efficient functions can be performed. For example, clustering can be used to perform data aggregation to reduce communications energy overhead [1, 2]; or to facilitate queries [3]; to form an infrastructure for scalable routing [4, 5]; clustering also can be used for efficient network-wide broadcast [6]. Clustering also facilitates in resolving other aspects like MAC layer contention resolution [7], coverage, security [8, 9] and in-network processing. The efficiency of many higher level applications and network functions is pertinent on the regular and efficient structure attained in clustering.

We propose Adaptive Clustering Protocol (ACP), a simple but efficient clustering protocol. The key advantages of our protocol are: a) With ACP the number of clusters required scales with density of the network; i.e., the number of clusters required does not increase with the density; b) ACP has very low communication

overhead while performance is comparable to other protocols; c) In ACP, a node does not need to know locations/ addresses of all its neighbors and hence ACP does not impose any bandwidth overhead such as *hello* messages; d) Behavior of ACP in large networks has been presented and it is shown that ACP performs well even in very large networks. Because of the above-mentioned advantages, ACP is very well suited as an efficient clustering protocol for Heterogeneous Wireless Networks.

1.3.6 Anonymous Communication

With the growth and acceptance of the wireless networks, there has been increased interest in maintaining anonymity in the network. The mere fact that a node has sent some information to the base station can reveal extremely important information. For instance, consider a sensor network deployed for intruder detection in which a sensor keeps sensing for intruders. Thus, when an intruder, once in the network area, sees a transmission from a sensor close to his location, can rightly assume that the his presence is sensed and might pursue evasive actions immediately.

Privacy International [10] defines four categories of privacy: information privacy, bodily privacy, communication privacy, and territorial privacy. Location privacy is a particular case of information privacy and can be defined as the ability to prevent other parties from learning one's current and past locations [11]. Anonymity can be defined as the state of being not identifiable within a set of subjects called the anonymity set [12].

Conventional protocols [13, 14, 15] proposed to ensure user anonymity in the Internet are based on the communication model in which high traffic conditions and high processing power is assumed, which might not be true with respect to wireless networks.

We present a novel Hierarchical Anonymous Communication Protocol (HACP) that hides the location of nodes and obscure the correlation between event zones and data flow from snooping adversaries. We use token ring approach for achieving anonymity of communication between cluster heads. Routes are chosen and frames are scheduled to traverse these routes. Each frame is assigned a token and a node can send a message through a frame only if the token is free.

We quantify the anonymity strength of our protocol by introducing a new anonymity metric: *Degree of Exposure Index*. Our protocol is designed to offer flexible trade offs between degree of anonymity and communication-delay overhead. We also present the trade offs between the overhead imposed and ring sizes. We show that higher anonymity comes at a cost - either higher communication/energy overhead or at higher latency. The choice of the parameters is left to the network administrator and depends on level of security needed and the type of traffic in the network.

1.3.7 Lightweight Data Integrity

Wireless networks, in general, are more vulnerable to security attacks than wired networks, due to the broadcast nature of the transmission medium. Furthermore, wireless sensor networks have an additional vulnerability because nodes are often placed in a hostile or dangerous environment where they are not physically protected. Security solutions for ad-hoc networks based on symmetric key cryptography are too expensive in terms of node state overhead and are designed to find and establish routes between any pair of nodes-a mode of communication

A key technical challenge is to detect malicious activity by distinguishing fake/ altered data from the correct one and identifying the malicious nodes. Since, wireless networks are highly unstructured, it is extremely difficult to identify vulnerable nodes/network zones a priori. Therefore there is a need to develop a broad spectrum of dynamic defense mechanisms for detecting such malicious behavior.

We present a novel lightweight protocol for data integrity in wireless networks. Data integrity is the assurance that the data received by the destination is the same as generated by the source. Data Integrity ensures that data is unchanged from its source and has not been accidentally or maliciously altered.

Our protocol is based on a simple leapfrog strategy in which each cluster head verifies if its previous node has preserved the integrity of the packet using the secret key it shares with two hop *up tree* node. The analysis and simulation results show that the protocol needs very few header bits, as low as three bits, thus resulting in negligible bandwidth overhead; the protocol poses very low computational overhead, it needs to compute just a hash as compared to multiple complex operations required by any cryptographic implementation for verifying authenticity.

Chapter 2

Models to Adapt Protocols to Network Conditions

Models are to be used, not believed.

- *Henri Theil* (1924-2000),

In mathematics you don't understand things. You just get used to them.

- *Johann von Neumann* (1903 - 1957)

Transmit power control refers to the problem of selecting appropriate power level for transmission of each packet. Transmit power control is an important problem in wireless ad hoc networks because of various reasons. Most of the mobile ad hoc networks have battery powered nodes, so lifetime of network depends on the power that a node consumes for transmitting packets. Also the SNR at a node depends upon the transmit power levels of the neighboring nodes. Low transmission power might also reduce interference, thus by reducing the collisions the latency can be reduced.

In order to meet the requirements of the applications despite variable link state and network topology, it is important that the protocols coordinate and adapt to the change in network state. To deal with the dynamic variations in networking and computing resources gracefully, both the mobile computing environment and the applications also need to adapt their behavior depending on the available resources including the batteries.

In this chapter, we develop such analytical models to enable protocols to adapt to the local network conditions: load, energy levels and number of neighbors. We develop analytical models to derive optimal transmission power/range for both broadcast and unicast messages. In case of broadcast messages, the objective is to ensure a given ratio of nodes receive the broadcast packet, while in case of unicast messages, the objective is to minimize the latency. We also present the energy trade-offs and let the network administrator to choose the transmission range depending on the network requirements. We then incorporate these models

into other protocols we propose to enhance the performance and to adapt to the prevailing network conditions.

We first discuss related works that consider the problem of finding an optimal range depending on the network conditions. Then we derive a geometric based, probabilistic model that describes the expected coverage of a one hop broadcast as a function of range, sending rate and density. We present an analytic model to predict the optimal range for maximizing 1-hop broadcast coverage in wireless networks, using information like network density and node sending rate. Finally, we present some preliminaries on deriving an optimal transmission range for transmitting unicast messages to minimize the latency.

2.1 Related Work

The concept of optimizing the radio transmission range of wireless networks is well studied. In [16], the optimal transmission radii that maximize the expected progress of packets in desired directions were determined for different transmission protocols in multihop packet radio networks with randomly distributed terminals. The optimal transmission radii were expressed in terms of the number of terminals in range. The study concentrated on limiting transmission interference to improve throughput performance in wireless networks under heavy traffic condition. Energy consumption, however, was not considered in the paper.

Similar assumptions were made in [17], which further allowed all nodes to adjust their transmission radii independently at any time. It was found that higher throughput and progress could be obtained by transmitting packets to the nearest neighbor in the forward direction and using the lowest possible transmission power for each transmission.

In addition to draining the battery of the node, since a wireless link is a broadcast mechanism, increasing the power used to transmit a packet might cause other side-effects such as interference with other nodes in the network. Therefore, it is important to determine the minimum power necessary to route a packet, and some works in ad-hoc network have focused on the problem of optimized routing that minimize the total path power consumption, see e.g. [18].

In [19] the critical power a node in an ad-hoc network needs to transmit at to ensure that the network is connected with probability one is computed. The

problem of adjusting the transmission power to control network connectivity is addressed in [20]. The problem is formulated as a constrained optimization problem with the connectivity as its constraint and the power used as its objective function. As the transmission range reduces, nodes contending for the channel reduce, thus minimizing the MAC layer contention.

Gobriel et al. [21] study the trade off between the low transmission power and the high probability of collision per message arising from increasing the number of hops on the path from source to destination. They come to the conclusion that sending the data packet to the nearest neighbor is not always optimal. They do not, however, account for the required latency when selecting the transmission power level.

The work in [22] presents a strict analytic model that predicts the optimal range for maximizing 1-hop broadcast coverage given information like network density and sending rate. The approach is very conservative especially at high transmission rates, while our model is more accurate.

While the current works focused on deriving optimal transmission range either to reduce energy consumption or end-to-end latency and to have the network connected, most of them do not consider the impact of interference and delays introduced due to the underlying MAC layer.

2.2 Modeling Impact of Collisions on Broadcast Messages

Broadcasting is a process by which a source node sends a message to all the other nodes in the entire network. The broadcast operation is the most fundamental role in wireless networks because of the broadcasting nature of radio transmission: When a sender transmits a packet, all nodes within the senders transmission range will be affected by this transmission. The advantage is that one packet can be received by all neighbors; the disadvantage is that it interferes with the sending and receiving of other transmissions, creating *exposed terminal problem*, that is, an outgoing transmission collides with an incoming transmission, and *hidden terminal problem*, that is, two incoming transmissions collide with each other.

Given the high densities of these future sensor networks, a resulting challenge for applications using broadcast will be how to manage channel capacity to ensure good performance in terms of throughput, fairness and broadcast coverage. This

challenge arises because if all nodes act greedily, using the maximum range, the channel will collapse; that is, the likelihood of any neighbors receiving the message correctly quickly approaches zero in dense networks due to collisions.

Also, broadcasts in wireless networks are unreliable; it is possible for rebroadcasts to be lost due to interference, transmission errors or collisions. The loss rate can be considerable if high interference exists or if link quality is bad as has been observed in wireless testbeds. Algorithms that control redundancy to reduce overhead have increased vulnerability to this problem; redundancy provides some protection against losses.

Providing reliable broadcast in a wireless environment is a very challenging task. One way to know for sure that a broadcast has reached all the neighbors is to get an acknowledgment from each of the neighbors. But by having all the neighboring nodes to send acknowledgments to all the receiving packets will result in a bottleneck at the sender. This is called the ACK implosion problem.

We are thus motivated to consider how devices in these networks can maximize the number of 1-hop receivers of a broadcast message. Our approach centers on the spatial reuse of wireless resources. Specifically, given the surrounding sending rate, node density, and a simple geometric model of wireless communication to compute the radio range, each node can just set its range to the optimal value, which probabilistically maximizes the 1-hop coverage for a broadcast packet. We use radio range as a parameter because it is more tractable to analyze than output power directly.

We develop an analytic model to predict the optimal range for maximizing 1-hop broadcast coverage in dense ad-hoc wireless networks, using information like network density and node sending rate. We derive a geometric based, probabilistic model that describes the expected coverage as a function of range, sending rate and density.

2.2.1 Assumptions

We use a set of assumptions to make an analytic solution tractable: (1) nodes are uniformly distributed with an average density of ρ nodes per $R * R$ region; (2) applications running on each node transmit packets according to a Poisson distribution with average rate λ_p ; and, (3) all packets are of the same length (size)

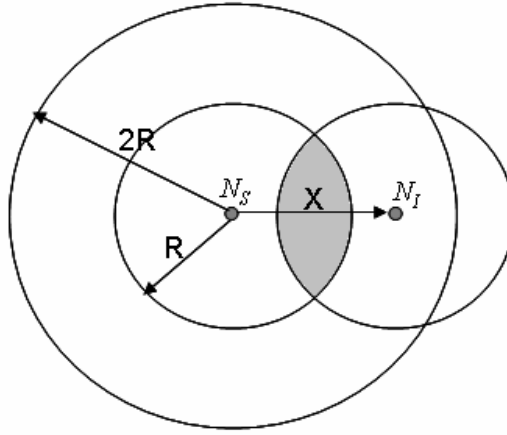


Figure 2.1. Computing the expected distance to the Interfering node

and take time T to transmit. We also use a fairly simple wireless communication model similar to the one in [23]: all nodes have the same radio range, where nodes within R distance from a transmitter will detect the packet transmission while those further away will not. More than one packet transmission within distance R to a receiver will cause collision and all overlapped packets at the receiver are corrupted.

2.2.2 Optimal Range in Presence of Collisions

Let X ($R < X < 2R$) be a random variable that represents the distance from an interfering node to the transmitting node N_s . Expected value of X can be expressed as (refer Figure 2.1)

$$E[X] = \frac{1}{(\pi(2^2) - \pi(1^2))} \int_1^2 (2\pi x) (x) dx = 1.556R \quad (2.1)$$

Consider two circles of radii R and the centers d ($d < 2R$) apart. The area of intersection is given as

$$A_{int} = 2R^2 \cos^{-1} \left(\frac{d}{2R} \right) - \frac{d}{2} \sqrt{(2R-d)(2R+d)} \quad (2.2)$$

As can be seen, the task of modeling coverage really becomes a task of deriving the number of failed nodes. An exact derivation, however, would be quite challenging because in the general case, we would have to account for multiple overlapping circular interference regions caused by colliding transmissions from the interference torus.

Consider a neighbor N_n , at a distance $d_n(0 < d_n < R)$ from the transmitting node N_s . Also let N_i be the interfering node located at a distance $d_i(R < d_i < 2R)$ from N_s . Since, the source node is able to detect a collision and retransmit if N_i is its neighbor and since, there would not be any interference if $N_i > 2R$, it follows that $(R < d_i < 2R)$. The probability that N_n is in the collision region is equivalent to the probability that N_n is also the neighbor of N_i i.e., $P(d_n < d_i - R)$.

Thus, the probability that a node is affected when an interfering node transmits can be computed as follows:

For simplicity, we use $R = 1$. But, R could be scaled to the actual transmission range and similar expressions could be obtained. Consider N_i to be at a distance x from transmitting node. Then, the probability is equivalent to the probability that neighbor lies in the intersection area. Hence

$$P = \int_1^2 P(\text{neighbor is in area of intersection}) \quad (2.3)$$

$$* P(N_i \text{ is at a distance } x)$$

$$\Rightarrow P = \int_1^2 \left[\frac{1}{\pi} \left(2 \cos^{-1} \left(\frac{x}{2} \right) - \frac{x}{2} \sqrt{4 - x^2} \right) \right] \left[\frac{2\pi x dx}{3\pi} \right]$$

$$\Rightarrow P = \frac{4}{3\pi} \int_1^2 x \cos^{-1} \left(\frac{x}{2} \right) dx - \frac{1}{3\pi} \int_1^2 x^2 \sqrt{4 - x^2} dx$$

$$\Rightarrow P = \frac{4}{3\pi} \left| \frac{-1}{2} x \sqrt{1 - \frac{x^2}{4}} + \frac{1}{2} x^2 \cos^{-1} (x/2) + \sin^{-1} (1/2) \right|_1^2$$

$$- \frac{1}{3\pi} \left| \frac{x^3 (4 - 1.x^2)^{0.5}}{3} \frac{{}_2F_1(1.5, -0.5; 2.5; 0.25x^2)}{(1 - 0.25x^2)^{0.5}} \right|_1^2 \quad (2.4)$$

where ${}_2F_1[a, b, c, x]$ is the *Hypergeometric* ${}_2F_1[a, b, c, x]$ function.

Thus, $P \approx 0.137832$.

Let P_i be the probability that a node i transmits a packet in a given time slot of duration T . Thus, expected number of interfering transmissions when a node transmits a broadcast packet is given by

$$T_i = [\pi ((2R)^2 - R^2) \rho] P_i \quad (2.5)$$

Now the probability that a node receives a message successfully in spite of the k interferences/collisions can be computed as

$$\begin{aligned}
P_s &= \sum_{k=0}^{\infty} \left[(P_{k\text{-interferences}}) (1 - P)^k \right] \\
&= \sum_{k=0}^{\infty} \left[\left(\frac{T_i^k e^{-T_i}}{k!} \right) (1 - P)^k \right] \\
&= e^{-T_i} \sum_{k=0}^{\infty} \frac{(T_i (1 - P))^k}{k!} \\
&= e^{-T_i} e^{T_i(1-P)}
\end{aligned} \tag{2.6}$$

Thus,

$$P_s = e^{-T_i P} \tag{2.7}$$

In scenarios, where transmission errors are present, probability of successful reception can be expressed as

$$\bar{P}_s = e^{-T_i P} (1 - \tau) \tag{2.8}$$

where, τ is the transmission error rate, i.e., the probability that a packet is received in error.

We note that our estimation of \bar{P}_s is conservative, as we assume that the probabilities that a node is unaffected by an interfering transmission are independent. But, when the interfering transmissions have overlapping area(s), then actual affected area is lesser and the independence does not hold. But, with this assumption, the worst-case probability that a node is affected is easily tractable. Also, this only results in more nodes receiving a broadcast successfully than estimated.

Figure 2.2 presents the expected reachability for varying transmission ranges with different loads. When transmission range is high, number of nodes in the neighborhood is high and hence the probability that an interference can occur is also high. This results in a decrease in the reachability. Similarly, at higher loads since the probability of an interference occurring is higher, the reachability is lower.

Figure 2.3 presents the required transmission range for various loads to obtain a desired reliability. Depending on the required reliability, an appropriate range can be selected. The trade off is that higher transmission range implies higher energy consumption as presented in Section 2.4.

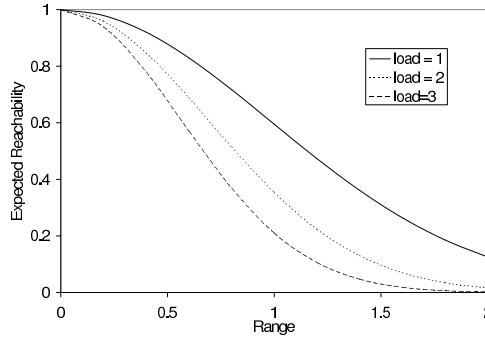


Figure 2.2. Expected reachability for varying transmission ranges with different loads

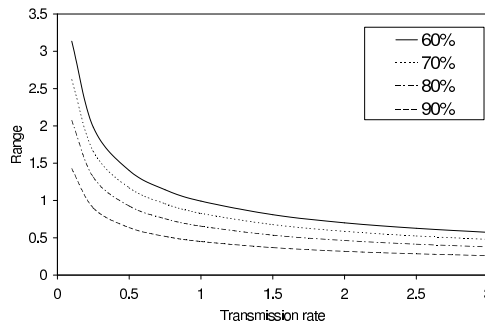


Figure 2.3. Transmission ranges for various loads to obtain a desired reachability

2.2.3 Estimation of Probability of Collision

We observe that all nodes might not have equal packet transmission rates and thus for accurate computation of transmission range to achieve a given delivery ratio requires the transmission rates of all two-hop neighbors. One simple and direct mechanism for nodes to acquire this information is periodic *hello messages*. Each node would periodically transmit a hello message constituting the transmission rates of all of its neighbors. This approach has several drawbacks. The biggest drawback is the communication overhead. Also, by the time a node receives transmission rate of its two-hop neighbor, the information would be already outdated by up to one hello interval. During computation of transmission range, the information could be outdated by up to two hello intervals. Thus, the transmission range computed might not be accurate, especially when the hello interval is very high, which might be the case so as to keep communication overhead low.

We propose an alternative mechanism in order to eliminate any communication overhead. Instead of computing the transmission range using the transmission rates, we propose to use *observed idle time* (t_i) - the duration during which the

channel is sensed idle in a time interval T_p . Thus, each node observes the channel for the idle state and computes the total idle duration during a time interval T_p . At the end of the interval, t_i is updated. Thus, $t_b = (T_p - t_i)$ gives the channel busy duration. The average transmission rate of the nodes can be thus estimated as $t_b/(\pi R^2 \rho T)$, where ρ is the node density and T is the packet transmission time. Again, the estimation of average transmission rate is conservative, since t_b also includes the collision time. But, this only leads to more nodes successfully receiving the broadcast message than estimated. We also assume that the network load is approximately same for one-hop and two-hop neighbors of a node and use the estimated one-hop node transmission rate for computing the optimal range.

2.3 Modeling Impact of Collisions on Unicast Messages

The modeling of the 802.11 MAC layer has been well studied [24, 25, 26, 27]. In particular, the particular problem of obtaining a queuing model for a wireless node is extremely difficult to treat since the queuing arrivals at subsequent nodes become dependent on each other in not-trivial manner. Our analytical model is based on [24], which provides accurate analysis for the average and variance for the total time a packet spends in back off.

2.3.1 Back off Characterization

Let s be the time used when the channel is sensed idle (i.e., one back off slot), t_s , the average time the channel is sensed busy due to a successful transmission, and t_c the average time the channel is sensed busy due to a collision in the channel. The three possible events a node can sense during its back off are $E_s = \{\text{successful transmission}\}$, $E_i = \{\text{idle channel}\}$, and $E_c = \{\text{collision}\}$. Each of the time intervals between two consecutive back off counter decrements, which we call *back off steps*, will contain one of these three mutually exclusive events. In other words, during a node's back off, the j^{th} *back off step* will result in either a collision, a transmission, or the channel being sensed idle. Let E_i , E_s , and E_c have probabilities $p_s = P\{E_s\}$, $p_i = P\{E_i\}$, and $p_c = P\{E_c\}$, respectively, and assume that these events are independent and mutually exclusive at each back off step.

The average back off time can be expressed as [24]:

$$T_B = \frac{\alpha (W_{\min} \beta - 1)}{2q} + \frac{(1 - q)}{q} t_c \quad (2.9)$$

where

$$\beta = \frac{q-2^m(1-q)^{m+1}}{1-2(1-q)}$$

q is the probability of success that a packet experiences when transmitted

W_{\min} is minimum (initial) contention window size

m is the maximum back off stage i.e., $W_{\max} = 2^m W_{\min}$

$$\alpha = \sigma p_i + t_c p_c + t_s p_s$$

It is shown in [24] that in both DSSS and FHSS, the fewer back off stages, the better is the performance, especially for large networks. It is also concluded that it is more effective to keep a constant, large contention window size W^* than to increase the size of the contention window exponentially. This way, nodes will be more aggressive in acquiring the floor, providing lower delays. Based on the above conclusions from now on we consider $m = 0$, although the results can be easily extended to other cases.

For $m = 0$, the average back off time and variance can be expressed as:

$$\bar{T}_B = \frac{\alpha(W^* - 1)}{2q} + \frac{(1 - q)}{q} t_c \quad (2.10)$$

$$Var \{ \bar{T}_B(k) \} = \left[\frac{\alpha(W^* - 1)}{2} + t_c \right]^2 \frac{(1 - q)}{q^2} \quad (2.11)$$

2.3.2 Queuing Delay

A node N_i can be modeled as a G/G/1 system with average arrival rate γ_{N_i} and uniform service time distribution. Then, the average waiting time in the queue at N_i can be shown to satisfy

$$W_{N_i} \leq \frac{(\sigma_a^2 + \sigma_b^2)}{2\gamma_{N_i}(1 - u)} - \frac{(1 - u)\sigma_a^2}{2\gamma_{N_i}} \quad (2.12)$$

where

σ_a^2 = variance of the inter-arrival times

σ_b^2 = variance of the service time

u = utilization factor given by (S_{N_i}/γ_{N_i})

The upper bound becomes exact asymptotically as $u \rightarrow 1$, that is, as the system becomes heavily loaded. Computation of the variance of inter-arrival times is very hard as it is interdependent on the queuing delays and varied transmission delays

at each node. We approximate the variance of the inter-arrival times by assuming that the packet arrival times are uniformly distributed.

2.3.3 Total Service Time

Note that the service time distribution can be derived from the distributions of back-off delay and sleep delay. For further explanation and channel state probabilities, we refer the reader to [25]. Finally, the time intervals t_s and t_c , can be expressed as follows [24]

$$t_s = RTS + SIFS + \tau + CTS + SIFS + \tau + H + E\{P\} + SIFS + \tau + ACK + DIFS + \tau \quad (2.13)$$

where, $E\{P\} = P$ for fixed packet sizes and $t_c = RTS + DIFS + \tau$

Given the back off time characterization, the *average service time* can be expressed as

$$\bar{T} = \bar{T}_B + t_s \quad (2.14)$$

where t_s is the time to successfully transmit a packet.

2.4 Energy Model

In the most commonly used energy model [28, 29, 30], the measurement of the energy consumption of network interfaces when transmitting a fixed size message depends on the range of the emitter u :

$$E(u) = \begin{cases} r(u)^\alpha + c_e & \text{if } r(u) \neq 0, \\ 0 & \text{otherwise} \end{cases} \quad (2.15)$$

$r(u)$ being the transmitting range of u and c_e , a constant that represents an overhead due to signal processing. The model $\alpha = 4$, $c_e = 10^8$ is derived from a work by Rodoplu and Meng [31], and it seems realistic enough to be used as a reference. These values are expressed in arbitrary units, and can be converted into any given units by using the corresponding multiplication factor.

Nodes also consume some energy upon the reception of a message. This consumption c_r is constant, regardless of the distance between the emitter and the receiver. The reference value generally used is one third of the energy consumed by a 100-meter emission, that is, $c_r = \frac{1}{3}(100^\alpha + c_e)$. In the above model, this gives $c_r = \frac{2}{3} \times 10^8$.

Chapter 3

Optimized Flooding Protocol

There is a certain majesty in simplicity which is far above all the quaintness of
wit.

- *Alexander Pope* (1688-1744)

Simplicity is the ultimate sophistication.

- *Leonardo da Vinci*

MAC broadcasts are unreliable; it is possible for rebroadcasts to be lost due to interference, transmission errors or collisions. The loss rate can be considerable if high interference exists or if link quality is bad as has been observed in wireless testbeds. Algorithms that control redundancy to reduce overhead have increased vulnerability to this problem; redundancy provides some protection against losses. This is especially true for virtual backbone based broadcast algorithms that statically determine the set of forwarding nodes: if a transmission to one of these nodes is lost, the broadcast message is lost to the remainder of the backbone and the nodes they cover.

We first derive a geometric based, probabilistic model that describes the expected coverage of a one hop broadcast as a function of range, sending rate and density. We first present an analytic model to predict the optimal range for maximizing 1-hop broadcast coverage in wireless networks, using information like network density and node sending rate.

Next, introduces the Covering Problem and a modification of the Covering Problem, we present Optimized Flooding Protocol (OFP), a novel protocol for broadcasting. In the process we present a solution to a variation of the Covering Problem [32]. The geometric approach makes functionality of OFP independent of network topology. OFP is performed in an asynchronous and distributed manner by each node in the network and OFP does not require a node to have any neighborhood information. We also study the reachability of OFP by using the analytical model

presented in section and also analyze the impact of transmission errors and losses on the performance of OFP.

Next, we adapt the transmission range of nodes to the local network conditions so as to meet required reliability conditions by using the analytical model developed in section 2.2. For instance, by observing the local collision probability, a node would be able to adapt its transmission range to make sure the coverage meets the reachability requirements. We propose two Reliable Broadcast algorithms: (i) OFP with Global Adaptation (OFP-GA) that forces every node to use same transmission range, and (ii) OFP with Local Adaptation (OFP-LA) that allows every node to independently decide its transmit range.

3.1 Related Work

Network-wide broadcast is an essential feature for wireless networks. The simplest method for broadcast service is flooding. Its advantages are its simplicity and reachability. However, for a single broadcast, flooding generates abundant retransmissions resulting in battery power and bandwidth waste. Also, the retransmissions of close nodes are likely to happen at the same time. As a result, flooding quickly leads to message collisions and channel contention. This is known as the broadcast storm problem [33].

Due to several inherent characteristics common between sensor networks and MANETs, all the broadcast protocols proposed for MANETs can be extended for sensor networks. Hence, in this section we even consider the broadcast protocols presented for MANETs.

The broadcast problem has been extensively studied for multihop networks. Optimal solutions to compute Minimum Connected Domination Set (MCDS) [34] were obtained for the case when each node knows the topology of the entire network (centralized broadcast). The broadcast protocol introduced in [35] completes the broadcast of a message in $O(D \log^2 n)$ steps, where 'D' is the diameter of the network and 'n' is the number of nodes in the network. From the result proved in [35], this protocol is optimal for networks with constant diameter. For networks with a larger diameter, a protocol by Gaber et al. [36] completes the broadcast within $O(D + \log^5 n)$ time slots, and it is optimal for networks with $D \in \Omega(\log^5 n)$. These solutions are deterministic and guarantee a bounded delay on message delivery,

but the requirement that each node must know the entire network topology is a strong condition, impractical to maintain in wireless networks.

Several broadcast protocols that do not require the knowledge of the entire network topology have been proposed. In a counter-based scheme [33], a node does not retransmit if it overhears the same message from its neighbors for more than a prefixed number of times and in a distance-based scheme [33], a node discards its retransmission if it overhears a neighbor within a distance threshold retransmitting the same message.

Source Based Algorithm [37], Dominant Pruning [38], Multipoint Relaying [39], Ad Hoc Broadcast Protocol [40], Lightweight and Efficient Network Wide Broadcast Protocol [41] utilize 2-hop neighbor knowledge to reduce number of transmissions.

A good classification and comparison of most of the proposed protocols is presented in [42]. It is also concluded that Scalable Broadcast algorithm (SBA) [37] and Ad Hoc Broadcast Protocol (AHBP) [40] perform very well as the number of nodes in the network is increased. Both these techniques are based on two-hop neighbor knowledge.

The Scalable Broadcast Algorithm requires that all nodes have knowledge of their neighbors within a two-hop radius. This neighbor knowledge coupled with the identity of the node from which a packet is received allows a receiving node to determine if it would reach additional nodes by rebroadcasting. Two-hop neighbor knowledge is achievable via periodic hello messages; each hello messages contains the node's identifier and the list of known neighbors. After a node receives hello messages from its neighbors, it has two-hop topology information centered at itself.

AHBP also requires that all nodes have knowledge of their neighbors within a two-hop radius. In AHBP, only nodes that are designated as a Broadcast Relay Gateway (BRG) within a broadcast packet header are allowed to rebroadcast the packet. BRGs are proactively chosen from each upstream sender, which is a BRG itself. A BRG selects set of 1-hop neighbors that most efficiently reach all nodes within the two-hop neighborhood as subsequent BRGs. Location Aided Broadcast [43] presents three location aided broadcast protocols to improve communication overhead and shortcomings of various protocols are also summarized.

In self-pruning methods [37, 44, 45], each node makes its local decision on forwarding status: forwarding or nonforwarding. Although these algorithms are based on similar ideas mentioned above, this similarity is not recognized or discussed in depth. Fair comparison of these algorithms is complicated by the lack of in depth understanding of the effect of the underlying mechanisms.

The drawback of the above Neighbor Knowledge methods is the need to store 2-hop neighborhood information at each node. In large networks, especially with high densities, this might impose very high communication/memory overhead.

In Gossip based routing [46], a node probabilistically forwards a packet so as to control the spreading of the packet through the network; the probability typically being around 0.65. Though, this simple mechanism reduces the number of redundant transmissions, there is still a lot of scope for improvement.

Several data dissemination protocols [47, 48, 49] have been proposed for sensor networks to disseminate data to interested sensors rather than all sensors. A broadcast protocol is presented in [50] for regular grid like sensor networks.

Lou and Wei identified the vulnerability of these approaches as being not reliable and proposed a solution for addressing it (Double Covered Broadcast, or DCB) [51]. DCB works by constructing virtual backbone graphs that provide double coverage of all nodes - every node in the graph is in range of two different nodes in the CDS. Therefore, two retransmissions would need to be lost before a node is not covered. However, in [52], it is shown that static CDS based approaches perform worse than dynamic/adaptive approaches in terms of coverage in lossy environments.

Selective Additional Rebroadcast (SAR) [53] proposes an approach where broadcast packets are selectively rebroadcast an additional time if they are suspected to have been lost. Experimental results show that the number of retransmissions is very high and in some cases more than the actual transmissions. [54] proposes a single source reliable broadcasting algorithm for linear grid-based networks.

3.2 Background

3.2.1 The Covering Problem

The Covering Problem can be stated as follows:

”What is the minimum number of circles required to completely cover a given 2-dimensional space.”

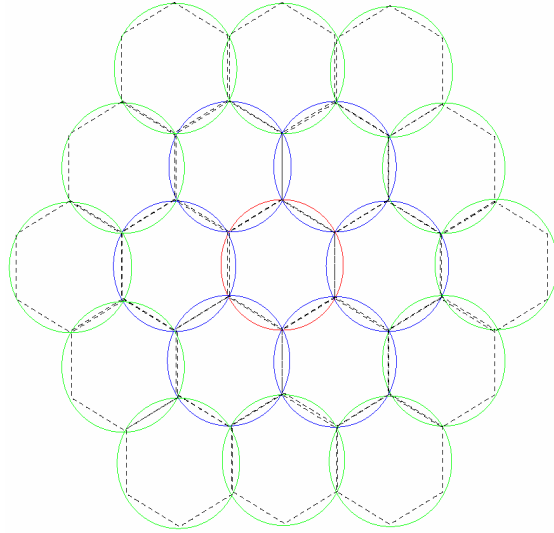


Figure 3.1. Covering a plane with circles in an efficient way

Kershner [32] showed that no arrangement of circles could cover the plane more efficiently than the hexagonal lattice arrangement shown in Figure 3.1. Initially, the whole space is covered with regular hexagons, whose each side is R and then, circles are drawn to circumscribe them.

3.2.2 The Modified-Covering Problem

Here, we state a modified version of *The Covering Problem* that finds its application in wireless networks. The solution we present here is to put forward the intuition behind our protocol and the solution is just for an ideal case scenario. A more practical solution is presented in section 3.3.

The modified version of *the Covering Problem* can be stated as follows:

”What is the minimum number of circles of Radius R required to entirely cover a 2-dimensional space with the condition that the center of each circle being placed lies on the circumference of at least one other circle.”

If the range of a node is considered to be R , then the reason behind the condition that the center of a circle should lie on the center of another circle is that a node has to receive a message for it to retransmit the message. A possible solution for the *Modified-Covering Problem* is shown in Figure 3.2. As done for covering problem, initially the whole region is covered with regular hexagons whose each side is R . Then, with each of the vertices as a center, circles of radius R are drawn.

The following properties of the vertices in Figure 3.2 should be noted:

Property-1: Each vertex v is joined to three other vertices.

Property-2: The lines joining these three vertices to vertex v make an angle of $120^0(2\pi/3\text{radians})$ with each other.

Property-3: Each vertex is at a distance of R from each of its neighboring vertices.

Thus, given a vertex v and one of its neighboring vertices, using these properties, it is possible to determine the other two neighboring vertices of vertex v .

The approach followed here to solve *the Modified-Covering problem* is for an ideal case scenario. We use the same approach to achieve broadcasting in a more general case, where there need not be any node at the optimal locations. In this case Figure 3.2 can get *deformed* a lot. For illustration, two such deformed figures are presented in Section 3.9. Even when the *deformation* is very large, the number of transmissions required to cover the whole region remains very low.

Though we do not claim that the solution we presented for the *Modified-Covering problem* is the best, through simulations we show that our protocol implemented using this solution outperforms other broadcasting protocols. We note that the source is not a vertex. Thus, one straight forward improvement is to have the source itself to be a vertex, thus reducing the number of retransmitting nodes by three when compared with the solution presented. One particular reason for choosing the source not to be a vertex is to ensure symmetry which would be lost otherwise.

3.2.3 Number of Transmissions in Ideal Scenario

In this section, we present the number of transmissions required to cover the whole network assuming ideal conditions. For this purpose, we see the network as hexagonal lattice and each vertex being a node that retransmits the packet.

Let N_H be the number of hexagons required to cover the entire network of area A . Each regular hexagon's arm length is R and area is $3\sqrt{3}R^2/2$. When area of the network is large when compared to the area of one hexagon, then N_H can be approximated as

$$N_H \approx \frac{A}{3\sqrt{3}R^2/2} \quad \text{when } A \gg \pi R^2 \quad (3.1)$$

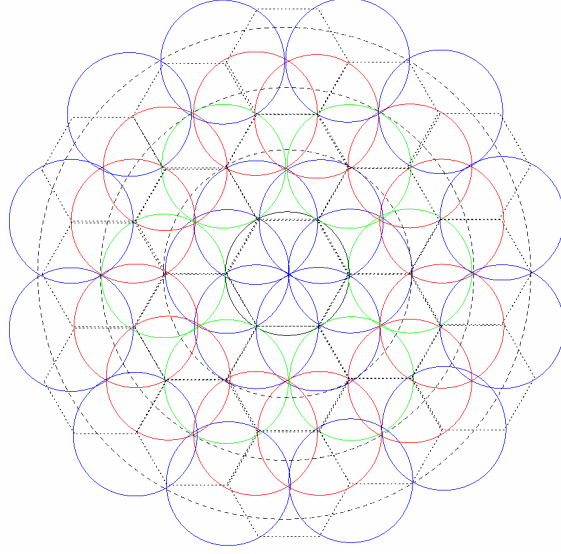


Figure 3.2. Our Solution for the Modified-Covering Problem

Additional hexagons might be needed to cover the gaps at the boundaries, but this number will be small for very networks. Also, the network topology will have an effect on this number. But, for regular large networks, equation 3.1 provides a very good approximation.

To compute number of transmissions, N_T , required to cover an entire area in ideal case, it should be observed that one transmission occurs at each vertex. Also, each vertex of a hexagon belongs to two other hexagons. Thus, when area of the network is large when compared to the area of one circle, total number of transmissions can be approximated as

$$N_T \approx \frac{2 * A}{3\sqrt{3}R^2/2} \quad \text{when } A \gg \pi R^2 \quad (3.2)$$

Defining efficiency as ratio of the area of the network to the total areas that each broadcast message has covered,

$$Efficiency = \frac{A}{2 * N_C \pi R^2} = 0.413 \quad (3.3)$$

From above equation, it can be observed that a node receives on the average 2.4 ($=1/Efficiency$) messages per node. Also, the above expression shows that the efficiency does not depend on the total number of transmitting nodes. Unlike the previous broadcast protocols that either select the retransmitting nodes with

TABLE 3.1. Number of transmissions required to cover a circular area in an Ideal Case

Radius of Circular region	Number of transmissions
2R	12
3R	24
4R	42
5R	60
6R	90
7R	126
8R	168

TABLE 3.2. Number of transmissions required to cover a rectangular area in an Ideal Case

Size of the rectangular region	Number of transmissions
3R*3R	8
4R*4R	10
5R*5R	16
6R*6R	26
8R*8R	42
10R*10R	74
4R*6R	18
6R*8R	36
8R*10R	54

help of neighbor knowledge or probabilistically, OFP selects the retransmitting nodes based on the above geometric solution. This makes OFP functionality to be independent of the network topology and hence, this solution is scalable as the number of nodes increases in the region.

The number of transmissions required to cover small circular and rectangular regions in the ideal case scenario are observed and are as presented in Table 3.1 and Table 3.2. The number of transmissions required in the Ideal case present a lower bound on the number of transmissions required. As the density of the network increases the number of transmissions required approaches the lower bound.

3.3 Optimized Flooding Protocol

In this section, we present the Optimized Flooding Protocol (OFP). Flooding achieves the goal of location discovery by letting all the nodes that receive the message, retransmit it again. The intuition behind our protocol is that in order to achieve the goal, there is no need for all nodes to transmit/retransmit the message. Instead, the goal can be achieved by allowing only a few strategically selected nodes

to retransmit the message. The strategy to select such nodes is same as the strategy to solve the Modified Covering Problem presented in the previous section.

3.3.1 Our Approach

Let S be the Source node that sends the route request. As seen in Figure 3.2, after the first circle centered on the center of region (location of S), six more circles whose centers are located on circumference of the first circle are drawn. These can be considered as first stage retransmissions of the request. In the next stage again six more circles are drawn whose centers lie on the circumference of the circles drawn in the first stage. From now on using the properties 1, 2 and 3 presented in previous section, it is very easy to predict the centers of the circles to be drawn in the next stage.

In real life, though, it is impractical to assume nodes to be located at the strategically selected locations. Thus, if the neighbor nodes are not in the optimal strategy locations, the coverage figure will get *deformed*; moreover, the *deformation* effect may propagate. Our goal is to extend the Modified Covering Problem to meet this restriction. A simple solution is to select the nearest node to the point selected and that received the message to retransmit.

It should also be observed that a node could receive a message more than once - from different directions and from different nodes, each node specifying different optimal strategy location (because of the *deformation*). This may cause two nodes very close to each other retransmit. We propose to avoid these transmissions by having a node keep track of its distance to the nearest node that has retransmitted the packet and to have a node retransmit only when its distance to the nearest transmitting node is greater than a threshold Th .

To elaborate for every broadcast packet, each node M stores the distance dm to the nearest node that has already transmitted the packet. A node does not retransmit, if dm for that broadcast message is less than a threshold Th . The choice of a right *threshold* will be the key for the success of the proposed algorithm. Later, we study the performance of OFP with different threshold values and show that a Th value of $0.4 \cdot R$ is a good choice to ensure high delivery ratio while keeping the number of transmissions very low. R is the transmission range.

3.3.2 OFP without Neighborhood Knowledge

Each broadcast packet contains two location fields, L1 and L2 in its header. Whenever a node transmits a broadcast packet, it sets L1 to the location of the node from which it received the packet and sets L2 to its own location.

The Optimized Flooding Protocol is as follows:

The Source Node S sets both L1 and L2 to its location (S_X, S_Y) and transmits the packet.

1. A node M , upon receiving a broadcast packet, first determines if the packet can be discarded. A packet can be discarded under any of the following conditions:
 - If the node has transmitted the packet earlier.
 - If a node which is very close has already transmitted this packet, i.e., if $dn < Th$.
2. If the packet is not discarded, M determines if it received the packet directly from the broadcast Source S .
 - If yes, M finds the nearest vertex V of a hexagon with (S_X, S_Y) as its center and with $(S_X + R, S_Y)$ as one of its vertices. It computes its distance l from V and then delays the packet rebroadcast by a delay d given by $d = l/R$.
 - Else, if M has not received the packet directly from the source S , but from some other node K , then using properties 1, 2 and 3 mentioned in the previous section and with the nearest strategic location. The packet transmission is delayed by $d = l/R$.
3. After delay d , M again determines if it has received the same packet again and if the packet can be discarded (for the same reasons mentioned above). Thus, delaying enables a node to decide if it is the nearest node to the strategic location. M updates L1 to location of the node from which it received the packet and L2 to its location, sets dn to zero and transmits, if the packet cannot be discarded.

The *delaying* is used to make a node decide if it is the nearest node to the strategic location. Low delay values decrease the time needed to broadcast a message all over the network, while high delay values help reduce redundant transmissions in instances where two nodes are of about same distance from the strategic location. The delay function we used causes a packet to be delayed a maximum of 50ms per retransmission, though typically this value lies around 10ms. In dense networks, the delay values are much less than 10ms.

The computational complexity of OFP is negligible; when compared to flooding, the major additional computation is finding the node's distance to the nearest optimal point according to *the modified covering problem*, which can be easily computed. The bandwidth overhead is just new header fields in the broadcast packet to carry location information of two nodes which is not significant.

3.3.3 OFP with Neighborhood Knowledge

We observe that OFP does not need any neighborhood information. Instead we utilize the concept of *delaying* retransmission of a message. In some mission critical networks, where any sort of delay needs to be eliminated, OFP can use the neighborhood information rather than delaying retransmissions. The protocol is then much simpler and is as follows:

1. A node M, upon receiving a broadcast packet, first determines if the packet can be discarded. A packet can be discarded under any of the following conditions:
 - If the node has transmitted the packet earlier.
 - If a node which is very close has already transmitted this packet, i.e., if $dn < Th$.
2. If the packet is not discarded, M determines if it received the packet directly from the broadcast Source S.
 - If yes, M finds the nearest vertex V of a hexagon with (S_X, S_Y) as its center and with $(S_X + R, S_Y)$ as one of its vertices.

- Else, if M has not received the packet directly from the source S, but from some other node K, then using properties 1, 2 and 3 mentioned in the previous section, M computes V, its nearest strategic location.
3. Using the location information of its neighbors, M checks if it is the nearest node to V. If M is the nearest neighbor, it retransmits; else, it does not.

We note that, though this version is much simpler and faster, it has an additional overhead in terms of requirement of location information of neighbors. To quantify, *hello messages* would result in each node transmitting one message per hello interval. In networks, where nodes are fairly stationary, the hello interval could be high. But, for networks with mobile nodes, the hello interval needs to be small. To be precise, one solution for selecting an appropriate hello interval would be such that $speed_{max} * T_{hello} < 0.1 * R$. $speed_{max}$ is the maximum speed of the nodes and T_{hello} is the hello interval. R is the transmission range of the nodes. This ensures that the maximum error in location information is 10% of the range. For protocols that are more sensitive to location inaccuracies, smaller hello intervals ensuring smaller errors can be implemented.

3.4 Analysis of OFP

3.4.1 Bounds on the Performance of OFP

In this section we obtain the analytical bounds on the performance of OFP. The best case performance of OFP is equivalent to the ideal case. We show that the worst case performance of OFP is bounded by a constant multiple of number of transmissions required in an ideal case. The constant is a multiple of Th , system parameter. Later in the section, we present the trade offs involved in fixing Th .

To derive the worst case performance bounds for OFP, we present the worst case scenario in which maximum number of transmissions occurs. First, it should be noted that minimum distance between any two transmitting nodes is controlled by Th . Thus, we claim that when every transmitting node is at a distance of $Th * R$ from some node that has transmitted, such a scenario would result in maximum number of transmissions. Again, this scenario is no different from the ideal case scenario as shown in Figure 3.2, except the transmission range of each node is Th instead of R.

Now, we compute the worst case bound on the performance of OFP. First, we would like to observe that the number of transmissions needed to cover an area is inversely proportional to the area one single transmission can cover. Let n_{ideal} and n_{worst} be the number of transmissions in ideal case and worst case scenarios respectively. Then, it should be observed that from the above argument that

$$\frac{n_{ideal}}{n_{worst}} = \frac{(Th * R)^2}{R^2} \Rightarrow n_{worst} = n_{ideal}/Th^2 \quad (3.4)$$

From equation 3.4, we can see that, the number of transmissions is upper bounded by a constant multiple of number of transmissions needed in ideal case. The constant is determined by Th . In the following section, we present the aspects governing the value of Th .

3.4.2 Effect of Threshold Th

The purpose of having Threshold is to prevent two nodes that are very close to each other from transmitting, thus reducing the redundancy. The key factors affecting Th are number of transmissions and delivery ratio.

Number of transmissions: As Th increases, the number of transmissions decreases. This is because, at high Th values, the minimum distance between any two transmitting nodes is more. This in turn implies that additional area covered is higher and hence number of transmissions needed for covering the entire network is lesser.

Delivery Ratio: It is the percentage of nodes that received the broadcast. More the number of transmissions more is the redundancy and hence more is the probability that a node receives the broadcast. So, for higher delivery ratios, lower Th are preferred.

To elaborate, consider Figure 3.3. For simplicity, consider transmission range as unity. For a given Th , the additional area covered due to a transmission by a neighbor of S is at least $\Delta_{ILL'L'}$, area of $ILL'L'$.

$$\Delta_{ILL'L'} = \pi - 2 * \Delta_{JILL'} = \pi - 2\theta + Th * \sin \theta \quad (3.5)$$

where, $\theta = \cos^{-1}(Th/2)$

Now, for higher Th values, $\Delta_{ILL'L'}$ is high and hence lesser transmissions are enough to cover the region. But, at the same time, if Th is high, the number of

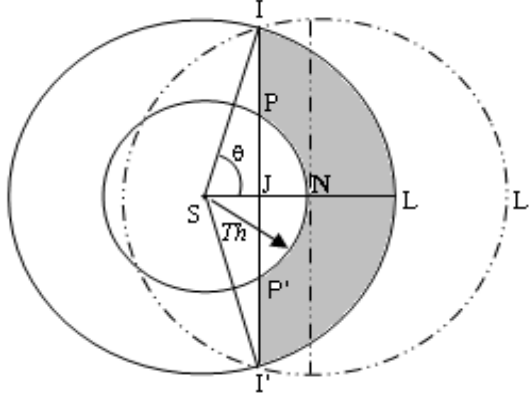


Figure 3.3. A scenario illustrating effect of Th

potential neighbors that could retransmit the message is less. To illustrate, consider the shaded region $IPNP'I'L$. At high Th values, this area is less and hence the probability that some node exists in this area is also less. Thus, at high Th values there might not be any transmission corresponding to the strategical location L . This might result in some nodes not receiving the broadcast. Section 3.9 using simulation results, illustrates the trade offs.

3.4.3 Forwarding Distance - Effective Range

Let ϵ be the distance of the nearest node to the strategic location selected. The probability distribution of ϵ can be calculated by finding the area of intersection of two circles with centers (S and S') at distance unity and with radii 1 and ϵ . More specifically the probability that the distance to the strategic location is at least ϵ is the probability that the area of intersection does not contain any nodes. If A_ϵ is the area, then we have

$$P[x \geq \epsilon] = e^{-\rho A_\epsilon} \quad (3.6)$$

The area under consideration is

$$A_\epsilon = 2 \int_0^\epsilon x \cos^{-1} \left(\frac{x}{2} \right) dx = 2 \left[\left(\frac{x^2}{2} - 1 \right) \cos^{-1} \left(\frac{x}{2} \right) - \frac{x}{2} \sqrt{4 - x^2} \right]_0^\epsilon$$

$$A_\epsilon = \pi + 2 \left(\frac{\epsilon^2}{2} - 1 \right) \cos^{-1} \left(\frac{\epsilon}{2} \right) - \frac{\epsilon}{2} \sqrt{4 - \epsilon^2} \quad (3.7)$$

We approximate the area of intersection to a sector S'PP' as shown in figure 3.3. Now the area under consideration is

$$A_\varepsilon = \theta \varepsilon^2 \quad (3.8)$$

$$\theta = \frac{1}{2} \cos^{-1} \left(\frac{Th}{2} \right) \quad (3.9)$$

The distance from the strategic location is x with probability distribution

$$P[\zeta \geq \lambda] = \begin{cases} e^{-\rho A_\varepsilon}, Th \leq \lambda \leq 1 \\ 0, \lambda < Th \\ 1, \lambda > 1 \end{cases} \quad (3.10)$$

Let

$$f_\zeta(\lambda) = f_\zeta^c(\lambda) + P[\zeta = Th] \delta(\lambda)$$

be the pdf of the advancement where $f_\zeta(\lambda)$ is the derivative of $P[\zeta \geq \lambda]$ in $\lambda \in (Th, 1)$.

The average advancement is then found as

$$E[\zeta] = \int_{Th}^1 \lambda f_\zeta(\lambda) d\lambda = \int_{Th}^1 \lambda f_\zeta^c(\lambda) d\lambda \quad (3.11)$$

Notice that ζ depends on the density of the network.

3.4.4 Time Taken for Broadcasting the Entire Network

We first estimate the number of hops h_k necessary to reach the farthest node k in the network from the source node. Let the distance between these two nodes be d_k . Let, \bar{d}_k be the distance from the source to the nearest strategic location to node k . Now, if \bar{h}_k is the number of hops to this strategic location then $\bar{h}_k \leq h_k \leq \bar{h}_k + 1$

Thus, once \bar{h}_k is computed, it is straightforward to derive bounds on h_k . For computing \bar{h}_k , it should be observed every odd hop¹ in the ideal scenario results in a progress of $\sqrt{3}R/2$, while every even hop results in a progress of R . In a practical scenario, the average progress with each odd hop is $(\sqrt{3}R/2) * E[\zeta]$ and with each even hop is $E[\zeta] * R$. Thus, the average progress made per hop toward a given destination can be approximated to

$$\bar{d} = \left(1 + \frac{\sqrt{3}}{2} \right) * \frac{E[\zeta]}{2} \quad (3.12)$$

¹The first hop is an exception; the progress is R . In large networks, when $d_k \gg 1$, this fact can be ignored.

Thus, the number of hops to a strategic location can be expressed as

$$\bar{h}_k = \frac{\bar{d}_k}{\bar{d}} \quad (3.13)$$

Also, note that each node waits for duration of d before retransmitting a broadcast message. Duration d is proportional to the node's distance from the strategic location. Thus, average delay before each retransmission can be expressed as

$$E[d] = c * (1 - E[\zeta]) \quad (3.14)$$

where, c is the proportionality constant. Assuming that transmission delay is negligible when compared to d , the total time taken for broadcasting a message throughout the network is equivalent to the time taken for the farthest node k to receive the message. Thus,

$$T_{Broadcast} = \bar{\mu}E[d] \quad (3.15)$$

Thus, for a given network, time taken for broadcasting scales as $O(\text{network diameter})$ and also reduces as node density increases.

3.5 Ensuring Broadcasting Reliability

3.5.1 Number of Messages Received by a Node

We observe that in an ideal scenario, each node receives a broadcast at least twice, while several nodes receive even three times. To be precise consider Figure 3.2. Expected number of nodes receiving a message twice and thrice can be computed as follows:

Area covered by three nodes can be calculated as

$$\begin{aligned} & 3 * (\text{Area of intersection between two circles}) - \pi R^2 \\ & = 3 * 1.228R - pR^2 \\ & = 0.544R^2 \end{aligned} \quad (3.16)$$

Thus, around 17.3% of nodes receive a message three times while around 82.7% nodes receive a message twice in an ideal scenario. In practice, the number of times a node receives a message more than twice is significantly higher (as high as 80% at low densities). The reason is that two transmitting nodes are closer to each other ($<R$) than in an ideal scenario ($= R$).

3.5.2 Relation between Desired Reliability and Range

Consider a required reliability of λ i.e., the broadcast needs to guarantee that at least $\lambda\%$ of the nodes receive each broadcast message. From Equation 2.8 combined from the observation that at least 17.3% of nodes receive a message at least three times while remaining nodes receive a message at least (refer section 3.5.1), we can obtain the following relationship:

$$\lambda = 1 - \left[0.173 * (1 - \bar{P}_s)^3 + 0.827 * (1 - \bar{P}_s)^2 \right] \quad (3.17)$$

In fact, the above setting of \bar{P}_s is conservative, since, several nodes receive a broadcast message three times.

3.6 Modeling Impact of Transmission Losses

3.6.1 Forwarding Distance - Effective Range in Presence of Transmission Errors

In presence of transmission errors, all nodes might not receive the broadcast packet. In this section we analyze the effect of transmission errors on the forwarding distance in such scenarios. Let t be the transmission error rate, i.e., the probability that a packet is received in error.

Now, the probability that the distance to the strategic location is at least ϵ is the probability that the area of intersection does not contain any nodes or none of the nodes present in the area of intersection receive the packet due to transmission errors. Thus

$$P[x \geq \epsilon] = e^{-\rho A_\epsilon} + \sum_{i=1}^N \left(\frac{e^{-\rho A_\epsilon} (\rho A_\epsilon)^i}{i!} \right) (\tau^i) \quad (3.18)$$

The first term in the summation is the probability that i nodes exist in the area of intersection and the second term is the probability that none of the i nodes receive the packet. The above equation can be simplified as follows:

$$P[x \geq \epsilon] = e^{-\rho A_\epsilon} \left[\sum_{i=0}^N \frac{(\rho A_\epsilon \tau)^i}{i!} \right] \quad (3.19)$$

As the number of nodes in the network, N , can be safely assumed to be very large, the equation can be further simplified and expressed as

$$P[x \geq \epsilon] = e^{-\rho A_\epsilon} \cdot e^{\rho A_\epsilon \tau} = e^{-(1-\tau)\rho A_\epsilon} \quad (3.20)$$

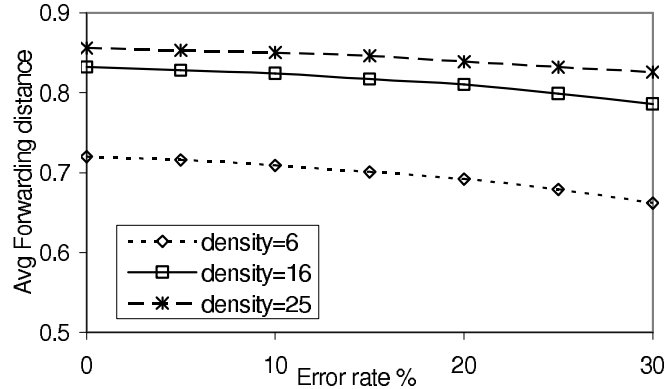


Figure 3.4. Observed values of the average forwarding distance in presence of transmission errors

The extreme cases can be easily be verified; $t = 0$ yields the case when there are no transmission losses and $t = 1$ (all packets are lost) yields a probability of 1, implying that independent of ε , no node can be found.

The distance from the strategic location can be easily computed similar to the no transmission error case and is x with probability distribution

$$P[\zeta \geq \lambda] = \begin{cases} e^{-(1-\tau)\rho A_\delta}, Th \leq \lambda \leq 1 \\ 0, \lambda < Th \\ 1, \lambda > 1 \end{cases} \quad (3.21)$$

The average advancement $E[\bar{\zeta}]$, in presence of transmission errors can be derived similar to equation 3.11.

Figure 3.4 presents the observed values of the average forwarding distance in our simulations. The average forwarding distance remains quite high in spite of high transmission errors. This can be explained from the fact that, there are multiple nodes that could retransmit a broadcast packet. Hence, higher the node density, lower is the impact on the average forwarding distance. This explains the resilience of OFP toward transmission errors.

3.6.2 Expected Increase in Number of Transmissions due to Errors

Now, we compute the impact of transmission errors on number of broadcast transmissions of CAB. First, we would like to observe that the number of transmissions needed to cover an area is inversely proportional to the square of transmission range of the nodes. Let n_i and n_l be the number of transmissions in ideal case (no transmission errors) and lossy transmission scenarios respectively. Also,

we observe that the effective transmission range of a node while considering broadcast message is equivalent to the average advancement $E[\zeta]$ and $E[\bar{\zeta}]$, in ideal and lossy scenarios respectively. Then, it should be observed that from the above argument that

$$\frac{n_l}{n_i} = \frac{(E[\zeta])^2}{(E[\bar{\zeta}])^2} \quad (3.22)$$

Again, we observe that at a node density of 16, 20% transmission loss rate results in a marginal decrease of around 3%. Thus, the increase in number of transmissions is just around 6%.

3.7 Energy Consumption and Transmission Range

We assume that all packets are of the same size (number of bits). In the most commonly used energy model, the measurement of the energy consumption of network interfaces when transmitting a fixed size message depends on the range of the emitter u :

$$E(u) = \begin{cases} r(u)^\alpha + c_e & \text{if } r(u) \neq 0, \\ 0 & \text{otherwise} \end{cases} \quad (3.23)$$

$r(u)$ is the transmitting range of u and c_e is a constant that represents an overhead due to signal processing. The model $\alpha = 4$, $c_e = 10^8$ is derived from a work by Rodoplu and Meng [31], and it seems realistic enough to be used as a reference. These values are expressed in arbitrary units, and can be converted into any given units by using the corresponding multiplication factor.

Nodes also consume some energy upon the reception of a message. This consumption c_r is constant, regardless of the distance between the emitter and the receiver. The reference value generally used is one third of the energy consumed by a 100-meter emission, that is, $c_r = \frac{1}{3}(100^\alpha + c_e)$. In the above model, this gives $c_r = \frac{2}{3} \times 10^8$.

For OFP, we can express the energy consumption in an ideal scenario as

$$PC(r) \approx \left(\frac{2A}{3\sqrt{3}r^2/2} \right) * (r^\alpha + c_e) \quad (3.24)$$

In the above expression, we do not account for energy spent by nodes when they receive messages. The reason is that this energy is constant irrespective of the chosen transmission range. There exists a very simple explanation for this

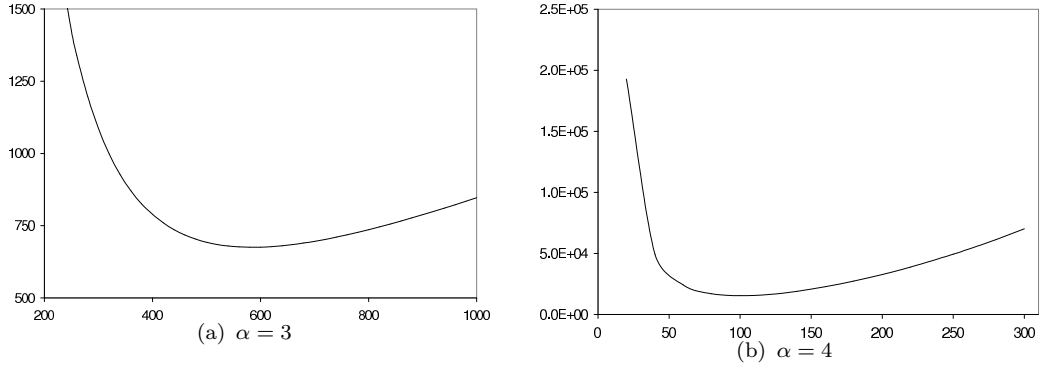


Figure 3.5. Power consumption per unit square region

outcome. For any r , the percentage of nodes receiving the broadcast message twice or thrice is same and is 82.7% and 17.3%, respectively. Hence, receiving energy would not affect the choice of optimal transmission range.

Consider the following two cases:

- $a=2$: $PC(r)$ has no minimum, but greater the r , lesser the consumption.
- $a>2$: In this case, the minimum power consumption occurs at $r = \sqrt[\alpha]{\frac{2c_e}{\alpha-2}}$

Figure 3.5 gives the power consumption per unit square region for $c_e = 10^8$ and two different values of α : 3 and 4. Below the minimum value, there are too many emitting nodes, making the constant c_e a problem while a greater radius makes the constant a a problem. It can be observed, however, that the function has small slope around the optimal radius, and deviation of up to 20 percent from the optimal radius does not have significant impact on the optimality ($<4\%$ when $a=3$ and $<15\%$ when $a=4$). This is an encouraging observation, since in reality we do not have nodes at ideal hexagonal tiling, but selecting existing nodes nearby gives satisfactory approximations.

3.8 Reliable Broadcast

In this section we propose two Reliable Broadcast algorithms: (i) OFP with Global Adaptation (OFP-GA) that forces every node to use same transmission range, and (ii) OFP with Local Adaptation (OFP-LA) that allows every node to independently decide its transmit range.

3.8.1 OFP with Global Adaptation (OFP-GA)

General wireless networks assume symmetric links and routes. In such scenarios, we propose to use OFP-GA, where all nodes in the network use the same transmission range. The optimal transmission range can be computed as presented in Section 3.5.2 and can be advertised to the entire network. In case when the transmission rates change, the base station can compute the new optimal transmission range and advertise to the network.

Each node would thus delay the transmission of a broadcast message by $1/d_t$, where d_t is the distance to the nearest vertex.

3.8.2 OFP with Local Adaptation (OFP-LA)

In several scenarios, traffic rates might be different in different regions of the network. Hence, it is desirable for the ideal power control scheme to support distributed coordination among nodes. OFP-LA allows each node to adapt its own transmission range to its neighborhood conditions. However, because two neighboring nodes may use different transmission powers, some links will be asymmetric. While several recently proposed protocols tackle the presence of asymmetric links at the routing layer, the possibility of wide-spread proliferation of asymmetric links will also necessitate changes at the MAC layer.

One extension for IEEE 802.11 to support asymmetric links could be as follows: In the conventional IEEE 802.11 MAC, a sender transmits an RTS, and DATA packets to a receiver, and the receiver responds with CTS and ACK packets to the sender. Because the MAC layer uses the same power for all packets, asymmetric links can induce link failures. If the receiver, however, uses the power notified by the sender (say piggybacked on the RTS packet) to transmit CTS and ACK packets, asymmetric links can be supported successfully. While this will increase the header overhead by about one byte, it is a negligible increase.

Each node could estimate the *approximate* network load in its neighborhood based on the observed channel idle time. Thus, each node could periodically compute the approximate load in its neighborhood and adjust its transmission range accordingly.

In such scenarios, each node would delay the transmission of broadcast message based on the additional area it would cover. For instance, consider two nodes

separated by distance d and with transmission ranges R and r respectively. Then the additional area covered can be computed as follows:

The area of intersection is given as

$$A_{int} = r^2 \cos^{-1} \left(\frac{d^2 + r^2 - R^2}{2dr} \right) + R^2 \cos^{-1} \left(\frac{d^2 + R^2 - r^2}{2dR} \right) - \frac{1}{2} \sqrt{(r + R - d)(-r + R + d)(r - R + d)(r + R + d)} \quad (3.25)$$

Thus, the additional area covered by a node with transmission range R and located at a distance d from the strategic location can be obtained as

$$A_{additional} = \pi R^2 - A_{int} \quad (3.26)$$

Thus, the delay function at each node is given by

$$delay_k = \frac{c}{A_{additional}(k)} \quad (3.27)$$

3.8.3 Energy Balancing

Heterogeneous Wireless Networks are envisioned to comprise of nodes with different capabilities leading to different energy levels of nodes. Even in Homogeneous Wireless Networks, where all nodes have same energy levels during the bootstrapping stage, because of different roles/tasks each node would be performing, node energy levels vary from one another.

To simultaneously prolong the network lifetime as well as each nodes lifetime, it is required that nodes with higher energy levels forward more packets than nodes with lower energy levels. We propose to achieve this by setting the delay d as follows:

$$delay_i = c \left(\frac{1}{d_t} + \frac{Avg_energy_i}{Energy_i} \right) \quad (3.28)$$

where,

d_t is the distance to the nearest vertex.

$Energy_i$ is the energy level of node i .

Avg_energy_i is the average energy level of the neighbors of node i .

The intuition behind this is as follows: The lower the energy level of a node, the lesser it should participate in broadcasting. Thus, by having an energy component in the delay, a node with low energy will delay retransmitting a broadcast for a

longer duration than a node with higher energy levels. We note that, though this mechanism requires energy levels of the neighbors, the information need not be accurate as we observe that the changes in the energy levels is not drastic.

In case of OFP-LA, the delay is computed as follows:

$$delay_k = c \left(\frac{1}{A_{additional}} + \frac{Avg_energy_i}{Energy_i} \right) \quad (3.29)$$

where, $A_{additional}$ is explained in the previous section.

3.9 Experimental Results

To evaluate the performance of OFP, we have used ns-2 simulator [55]. The nodes were uniformly distributed all over the region with a density of 30 nodes per RXR unless stated otherwise. We compare our protocol with blind flooding. We also compare our protocol with Ad Hoc Broadcast Protocol (AHBP) [40] as AHBP is one of the protocols (SBA [37] being the other) that best approximates MCDS fairly [42]. A wireless network of different physical areas and different shapes with different number of nodes were simulated. To be more specific, circular regions of radius varying from R to $10R$ and rectangular/square regions of size varying from $3R * 3R$ to $10R * 10R$ have been simulated, where R is the transmission range of each node, which is 100 in all our simulations, unless specified.

The nodes were uniformly distributed all over the region with the density varying from 4 nodes per $R * R$ region to as high as 100 nodes per $R * R$ region. Every simulation is repeated until the 95% confidence intervals of all average results are within $\pm 5\%$.

The simulations are aimed at studying the performance of OFP in networks of different sizes and densities. Initially, to study the performance of OFP and for comparing with existing works, we do not consider any data packets in the network and thus there are no collisions. We studied the effect of different threshold values on the performance of OFP. Then, we concentrated on the algorithm efficiency by studying the performance of OFP in static networks and also in highly mobile networks. We study the performance of OFP in networks where the coverage area of a node is not circular.

We then focus on studying the reachability and reliability of OFP in presence of varying loads and transmission losses. The effectiveness of OFP in balancing the

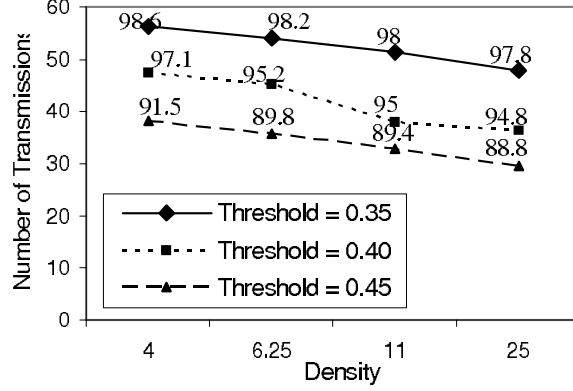


Figure 3.6. Effect of Th on performance of OFP. Network size = $6R * 6R$

load among nodes according to their energy levels is also presented. The simulation results under each network study are presented in a subsection below.

3.9.1 Effect of Threshold Th

The purpose of this study is to evaluate the effect of different threshold values on the performance of OFP. Figures 3.6 and 3.7 show the simulation results for threshold values of $0.35R$, $0.40R$ and $0.45R$. Apart from the number of transmissions in each case, the delivery ratio in percentage for each case is indicated at each data point. Delivery Ratio is the average number of nodes that receive the message to the total number of nodes in the network. Figure 3.6 is for a network size of $6R * 6R$ and Figure 3.7 is for a network of size $4R * 4R$.

For a threshold value of $Th = 0.35R$, a delivery ratio of around 98% is achieved and for $Th = 0.4R$, the delivery ratio is around 95%. But, for $Th = 0.45R$, the delivery ratio falls to around 90%. This is understandable, because with the increase in threshold value, number of retransmitting nodes decrease.

For all further simulations, we use threshold value of $Th = 0.4R$ and for each simulation case, we present the minimum and maximum delivery ratio, instead of presenting the delivery ratio for each for each data point.

3.9.2 OFP Efficiency

The purpose of this study is to evaluate the performance of OFP in networks of different sizes and different densities. We include a "best case" bound provided by the simulation results in ideal case scenarios. It is impossible for any algorithm to perform better than the performance in ideal case scenario and unlikely to perform

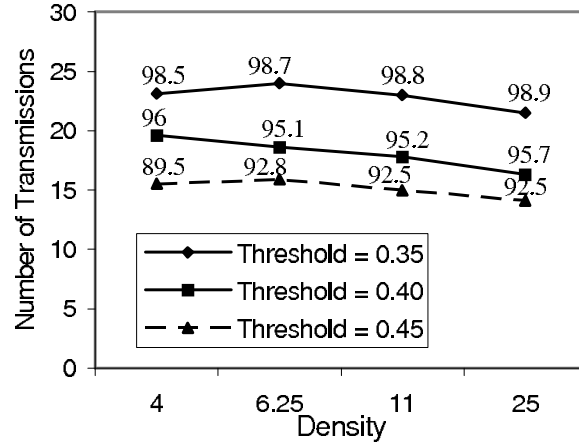


Figure 3.7. Effect of Th on performance of OFP. Network size = $4R * 4R$

worse than simple flooding. Thus, these two bounds provide a useful spectrum to gauge the performance of our protocol. OFP, when compared to flooding, uses up to 65% to 90% fewer messages depending on the density of the network.

For this study we varied the network size from $3R * 3R$ to $10R * 10R$, while keeping the transmission radius of each node fixed to 100. We also varied the network density from 4-nodes/ $R * R$ region to 100-nodes/ $R * R$ region.

First, fixing the density of the MNs in the region, we simulated the number of transmissions needed to cover a square/rectangular region completely. The coverage figure gets deformed a lot as in most of the cases no node exists at the strategic location. Figure 3.8 shows two such cases - one for $4R * 4R$ and another for $6R * 4R$ regions, both with a density of 4 nodes per $R * R$ region.

Figure 3.9 is a plot between the number of transmissions required to cover entire region for varying densities and for different areas of the region. Network areas up to $10R * 10R$ have been considered. It gives a plot between the number of transmissions and density of the network for different network sizes. It can be seen that the number of transmissions required decreases as the number of nodes (density) increases. The number of transmissions at a density of 100 is very near to the number needed in an Ideal case. The minimum delivery ratio achieved by OFP was 94.3% for the case with network size of $6R * 8R$ and with a density of 6.25. In all other cases, the delivery ratio was close to 95% with the maximum being

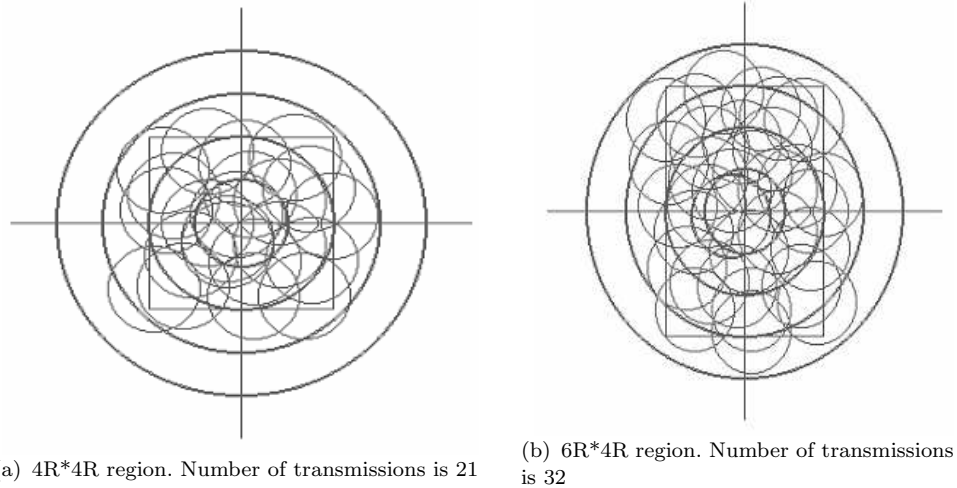


Figure 3.8. Example *deformed* figures with uniform node deployments

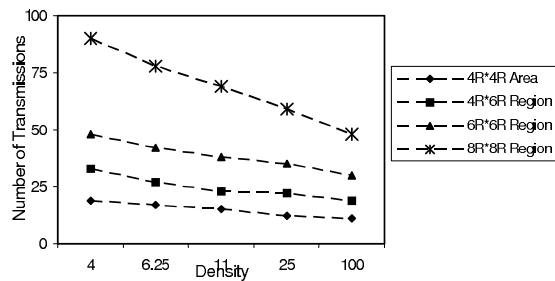


Figure 3.9. Number of Transmissions for varying node densities and areas

97.3%. The results show that the performance of OFP remains very efficient even in large networks; network size does not seem to affect the performance of OFP.

Figure 3.10 shows the percentage of nodes in the network retransmitting a broadcast message. The simulations were done in networks of sizes up to $10R \times 10R$ with different node densities. For a given network density, the percentage of retransmissions remains almost a constant for all network sizes. This reflects that OFP performance is not hindered in large networks.

Figure 3.11 presents the performance of OFP against flooding. Even at low network sizes, OFP reduces number of transmissions by 66% to 90%. Next, we compare OFP with Ad Hoc Broadcast protocol (AHBP) [37]. Networks of $4R \times 4R$, $6R \times 6R$ and $8R \times 8R$ were considered. As shown by Figure 3.12, the performance of both OFP and AHBP is very similar, though OFP performs slightly better than AHBP especially at high network densities. Here, we considered only static

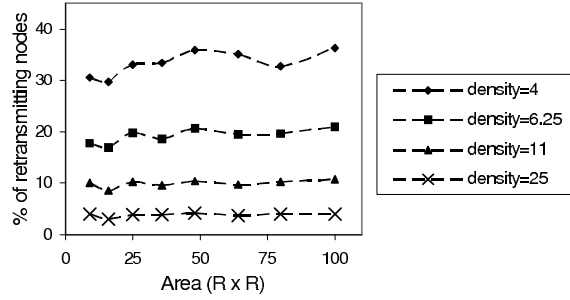


Figure 3.10. Percentage of retransmitting nodes for different networks

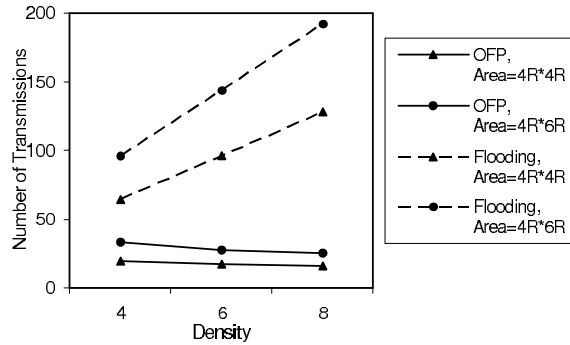


Figure 3.11. Performance of OFP and Flooding in static networks

networks and in the next section, we present results for mobile networks where OFP clearly performs much better than AHBP.

3.9.3 Mobile Networks

This section presents the simulation results of OFP and AHBP in mobile networks. We use the Random Walk mobility model [56] with zero pause time. The range of mean speeds of the nodes is varied from 1 to 20 meters per second. The upper bound corresponds to around 50 miles per hour, which we assume to be a realistic maximum speed of any mobile node.

Figure 3.13 presents the effect of mobility on each of the protocols. The simulation is in a network of 144 nodes and with the network size being $8R * 8R$. The performance of OFP remains unaffected, as OFP algorithm uses minimal neighborhood information. But, the performance of AHBP rapidly deteriorates with increase in speed and its performance is also affected by the hello interval.

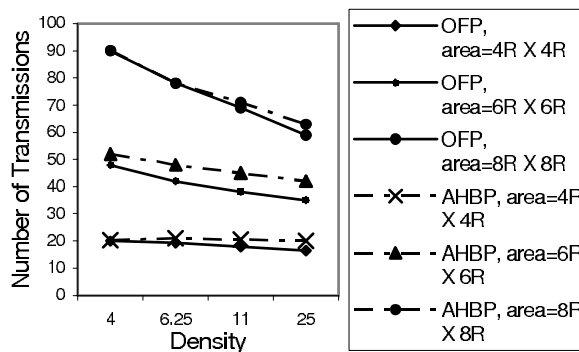


Figure 3.12. Performance of OFP and AHBP in static networks

The two-hop neighbor knowledge based protocols use hello messages to gather the neighborhood information. With a hello interval of t seconds, the two-hop neighbor information (that is obtained through the hello messages of one-hop neighbors) would always be outdated by an average of t seconds. For instance, if $t = 10$ seconds and a nodes speed is 36mph, then the node would have moved up to 100m before its information has been conveyed to one of its 2-hop neighbors. Also, once a node gets this information, it is not updated again till 10 sec. Thus, a node could have moved up to 200m before its information is updated at its neighbors. Also, the average time by which a node's information at 2-hop neighbor is outdated is 15 seconds ($t + (0 + t)/2$), which corresponds to a displacement up to 150m. This shows the intensity of the effect mobility has on these protocols. Thus, the hello interval t should be very small for efficient performance of two-hop neighbor knowledge based protocols, which in turn means that the bandwidth overhead due to hello messages is very high.

3.9.4 Effect of Non-Uniform Radio Propagation

In this section, we study the performance of OFP in wireless networks where wireless propagation is noncircular. We use the term noncircularity to mean that the range of a node might be different in each direction, the maximum being R , which is the range in an ideal case. Contours of the terrain and obstructions like large buildings contribute in creating such nonuniform radio propagation. We think this sort of study is necessary, especially as our protocol is an extension of the Modified Covering Problem solution developed for an ideal case.

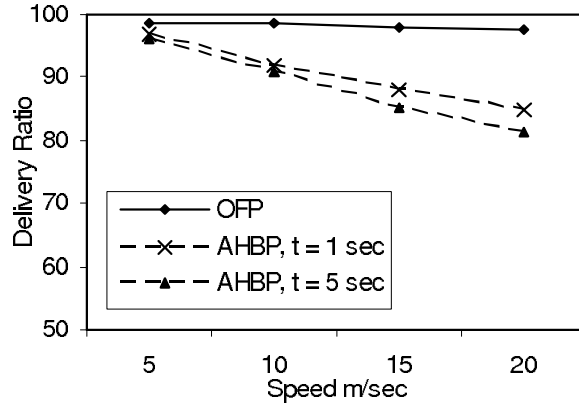


Figure 3.13. Effect of Mobility on different protocols. Network size = $6R * 6R$. Number of nodes = 144

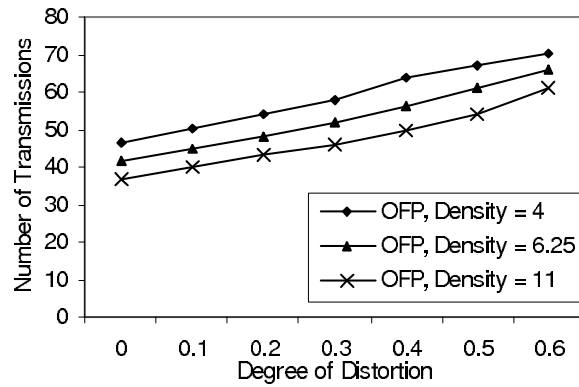


Figure 3.14. Effect of nonuniform propagation on OFP. Network size is $6R * 6R$

In the simulations, for each node, we generated the coverage area by setting the transmission range in different directions to a random value between $[D * R, R]$, where D is the Degree of Distortion and R is the range of a node in an ideal scenario. The simulations were for static networks.

The performance of OFP in case of noncircularity is presented in Figure 3.14. Figure 3.14 is for a network area of $6R * 6R$. It can be observed that the number of transmissions needed grow linearly with the degree of distortion. The delivery ratio in all cases was above 94% with the least being around 94.3%.

The performance comparison of OFP and AHBP is presented in Figure 3.15. The figure is a plot between the number of transmissions and Degree of Distortion for network sizes of $6R * 6R$ and $8R * 8R$ and for a network density of 6.25. Performance of both the protocols is similar. In both protocols, the number of

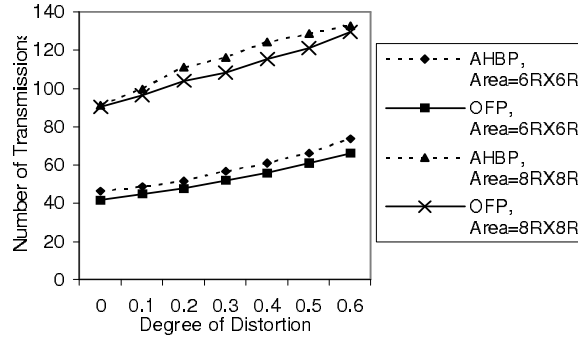


Figure 3.15. Performance comparison of OFP and AHBP under nonuniform propagation. Network density = 6.25

transmissions increases almost linearly with respect to the Degree of Distortion. The effect of mobility is not considered in these simulations

The purpose of the study was to see the performance of OFP in networks with nonuniform transmission ranges. As shown by figures 3.14 and 3.15, OFP’s performance remains efficient even under such conditions. This can be attributed to fact that in OFP, the decision if a node retransmits or not is made locally at each node that receives the packet. Thus, even if a node very close to the strategic location does not get the packet, the reachability is not affected as some other node that received the packet retransmits.

3.9.5 Adapting to the Network Conditions

The focus of this section is to study how effectively OFP can adapt to the prevailing network conditions. We study both OFP-GA and OFP-LA. The simulations are over a network area of $500 * 500$ with 100 and 250 nodes. We also introduce background traffic where nodes transmit data packets at a rate depending on *the term Average Load, L_{avg}* i.e., each node selects a data transmission rate randomly from the interval $(0, L_{avg})$ and transmits packets at this rate through out the simulation. Data has been collected from 10 simulation runs, each simulation run being for 100 seconds and consisting of 50 broadcasts.

We study three metrics:

- *Number of retransmitting nodes* - The average number of nodes that retransmit a broadcast message is computed. We note that the number of times a broadcast message has been retransmitted is different from average number

of retransmitting nodes, because of collisions occurring at the retransmitting node itself, a node might transmit the broadcast message more than once.

- *Delivery Ratio* - The average number of nodes receiving a broadcast message is computed. A node might receive a broadcast message more than once from different nodes.
- *Energy consumed/broadcast* - The average energy consumed by the entire network to complete one broadcast operation is computed. This also includes the energy consumed for unsuccessful transmissions. However, to effectively carry out the study, we do not consider the energy consumed for transmitting the data packets.

The performance of OFP, OFP-GA and OFP-LA is presented in Figure 3.17. For baseline comparison, we choose OFP with transmission rate set to 100. In case of OFP-GA, we computed required transmission range to guarantee that 90% of the nodes receive the broadcast, assuming all nodes are transmitting at the *average load*. This transmission range is then assigned to all the nodes in the network. Also, for the no-load case, we set transmission range to 200 so as to have reasonable energy consumption. For OFP-LA, as elaborated in Section 3.8, each node computes its transmission range independently. Figure 3.16 is one snapshot of retransmissions for a broadcast. The selection of different transmission ranges by different nodes based on the conditions in their neighborhood and their energy levels could be observed.

First we present the number of retransmitting nodes per broadcast in Figures 3.17(a) and 3.17(b) for networks with 100 and 250 nodes, respectively. At low loads, since the transmission range of nodes can be high, the number of transmission in case of OFP-GA and OFP-LA are much lower than that of base OFP. But, as load increases, the nodes adapt to lower transmission ranges and hence we observe higher number of transmissions. Also, OFP-LA performs slightly better than OFP-GA (around 5% lesser transmissions) especially at higher loads, as nodes can better adapt to local conditions. Also, as load increases, the number of retransmitting nodes initially increases and then decreases. The initial increase is because, as load increases, some nodes do not receive the message due to collision and hence the

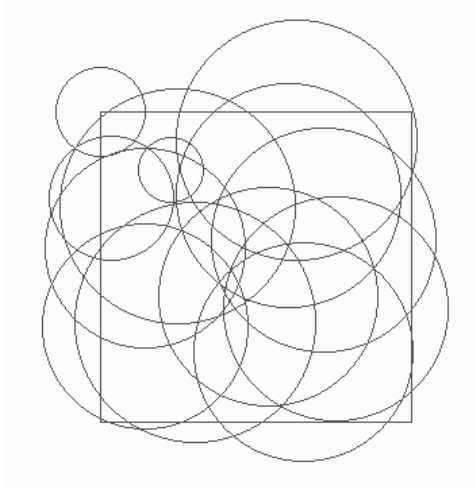


Figure 3.16. OFP-LA enables nodes to select different transmission ranges

effective forwarding decreases, resulting in increased retransmitting nodes. But, as load further increases, the number of nodes not receiving a message is significant and cases where there are no nodes further than *Threshold Th* that receive the message begin to show up thus leading to lower retransmitting nodes and lower delivery ratio.

The efficiency with which the broadcast message can be delivered to the nodes in the entire network is presented in Figures 3.17(c) and 3.17(d). For OFP-GA and OFP-LA, the transmission range is selected so as to achieve at least 90% reachability. We note that, by adapting, OFP can achieve much better reachability. The reason is that, nodes receive a broadcast more number of times than assumed by our analytical model. Also, we observed that the unreachable nodes, especially at higher loads are located at the network border, where a node might receive lesser number of broadcast messages than predicted. Nevertheless, both OFP-GA and OFP-LA can guarantee the reliability/reachability requirements.

Finally, we present the energy consumed to complete a network wide broadcast operation is presented in Figures 3.17(e) and 3.17(f). We only considered the case when $\alpha = 4$. An interesting observation regarding energy consumption of base OFP is that in spite of lower number of transmitting at higher loads, the energy consumption is higher because a node might have to retransmit the same broadcast message multiple times as a result of collisions at the transmitting node itself.

In case of OFP-GA, the energy consumption is high at low loads due to high transmission ranges. It reduces as load increases as transmission range also reduces. But, then with further increase in the load, the increases number of retransmitting nodes plays a bigger role and leads to an increase in the energy consumption. Again, OFP-LA performs slightly better than OFP-GA especially at higher loads, due to better adaptation to local conditions.

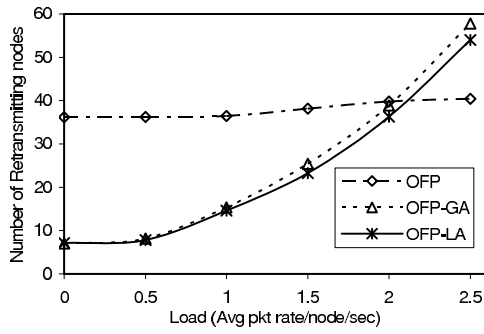
3.9.6 Energy Balancing

In this section we study the effectiveness of OFP in distributing the load among the nodes according to their energy levels. For this purpose, the initial energy of each node is set to a value between 5 and 20 power units. The energy level is a function of node address, i.e., a node i has an initial energy of $5 + \frac{(20-5)*i}{N}$, where N is the total number of nodes in the network. Thus ideally, we expect the number of transmissions a node is involved in to be proportional to its address.

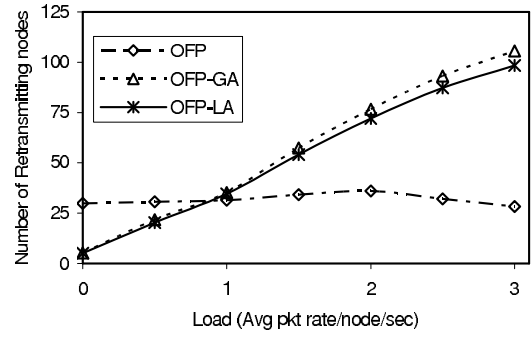
The nodes are uniformly distributed in a network of size $400 * 400$. There are 256 nodes in total. We collect the data from 500 broadcast messages, each message generated by a source randomly selected.

The number of times a node is selected to retransmit by OFP without any adaptation is shown in Figure 3.18. The transmission range of each node is 100. As observed, there does not seem any relation between the address of a node (i.e., the energy level) and the number of times it broadcasts. Each broadcast required an average of 34.55 transmissions, the standard deviation is 7.62 and 90th percentile being 45.

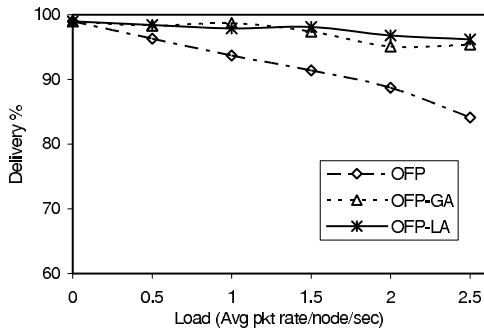
Figure 3.19 presents the number of times a node is selected to retransmit by OFP with energy consideration (OFP-E). The transmission range of each node is 100. We observe that there does exist a relation between the address of a node (i.e., the energy level) and the number of times it retransmits. While there is not complete dependency on the energy level, OFP-E does significantly balance the load. Each broadcast required an average of 37.19 transmissions, slightly higher than the case without any adaptation. This is because a retransmitting node need not be the closest to the strategic location; nevertheless the number of additional messages is less than 10%. Also, the standard deviation is 13.48 and 90th percentile is 55.4.



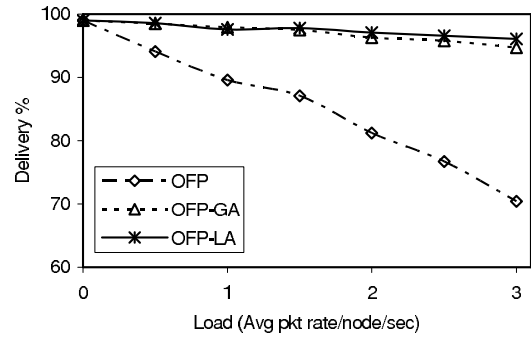
(a) Average number of Retransmitting nodes, N=100



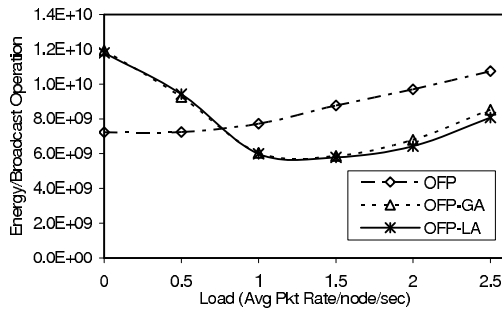
(b) Average number of Retransmitting nodes, N=250



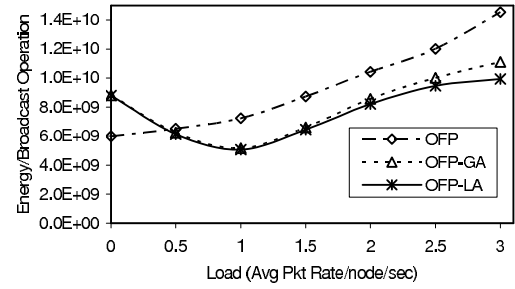
(c) Delivery Ratio, N=100



(d) Delivery Ratio, N=250



(e) Energy Consumed per broadcast operation, N=100



(f) Energy Consumed per broadcast operation, N=250

Figure 3.17. Performance study of adaptability of OFP for two network scenarios: with 100 nodes and 250 nodes

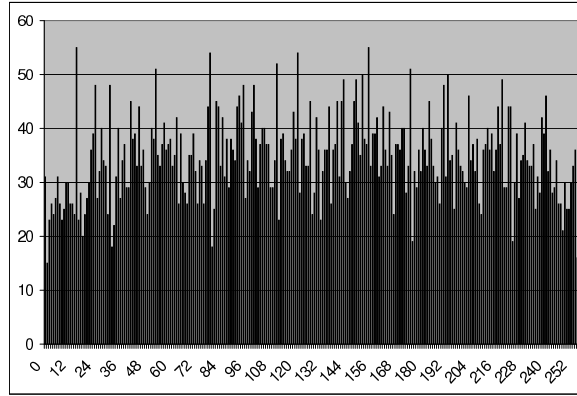


Figure 3.18. Performance of OFP without Energy consideration

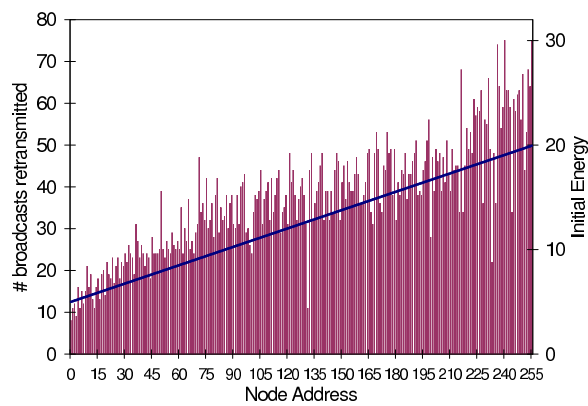


Figure 3.19. Performance of OFP with Energy consideration

Figure 3.20 presents the variance in the energy levels of the nodes for both OFP and OFP-E. For OFP, the variance slightly increases as a node with low energy level is as likely to be selected to retransmit as a node with higher energy levels. But, some nodes might be selected to retransmit more frequently based on their location and hence the slight increase in the variance. By explicitly considering the energy levels of the nodes, OFP-E is able to effectively select nodes with higher energy levels to retransmit causing a decrease in the variance. We also observe that the rate of decrease in the variance is higher than after several broadcasts. We attribute this to the factor that initially nodes with highest energy levels retransmit with highest frequency. After some broadcasts, their energy levels drop and there are some other nodes with similar energy levels that also participate in retransmitting. After a while, there are very few nodes with very high energy levels causing a further decrease in variance due to those nodes less probable.

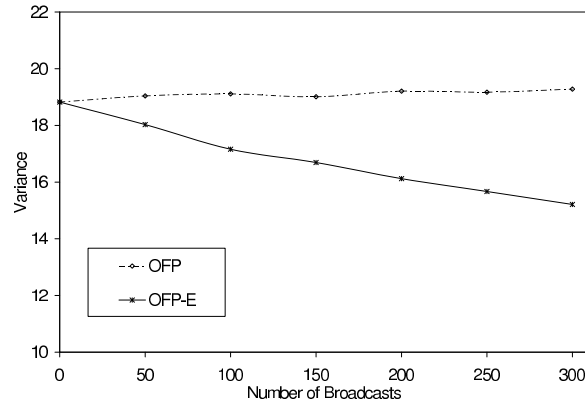


Figure 3.20. Variance in the residual energy levels with and without adaptation to energy

3.10 Broadcasting in Three Dimensional Networks

Communication among airplanes is becoming an important tool that can improve significantly the safety of flights. The increase of airplanes density, accompanied with more freedom in choosing the flight path, requires airplanes to be able to communicate with each other. Furthermore the threat of combined terrorist attacks, which could target more than one airplane as well as their communication capabilities, requires the development of efficient and robust protocols able to function in adverse conditions.

While in the air, the pilots communicate with the ground controllers or other airplanes (satellites etc.) using wireless channels. For this purpose, the U.S. government has allocated certain frequency and bandwidth for air-ground and air-air communications. These existing communications are highly dynamic i.e., the airplane has to search for the closest point of contact for effective communication, while today's airplanes move at supersonic speeds. The bandwidth of the available channels is very limited which allows only the radiotelephones. But the major problem of existing communications systems is that they are not very suitable for emergency situations, when the whole communication system might be under multiple attacks, such as jamming of links to ground stations, disruption caused by enemy flying objects, etc. as shown in Fig. 3.21.

We think that broadcast is the most suitable type of protocol for emergency situations. Ad Hoc Broadcast does not rely on any infrastructure such as specific ground station, which can become vulnerable to enemy attacks. On the other hand

Broadcast is the fastest way to spread emergency information to all interested airplanes.

We present Three Dimensional Broadcast (3DB) that enables optimized broadcast among ad hoc networks of airplanes. 3DB is an extension of OFP, presented in the previous sections. 3DB minimizes the number of transmissions needed for broadcasting by doing selective forwarding, where only a few selected nodes in the network do the broadcasting. It is assumed that each node knows its location. To "select" the transmitting nodes, we extend *the Covering Problem* [32], which deals with covering a region completely using minimum number of circles.



Figure 3.21. Use of 3DB for emergency communication in adverse conditions among airplanes as well as airplanes and ground stations

The key advantages of our protocol are: a) 3DB scales with network size; in fact the number of transmissions required decreases as the density of the network increases; b) 3DB minimizes the number of unnecessary transmissions and outperforms other variations of flooding; c) 3DB does not require any neighborhood information and hence, 3DB does not impose any bandwidth overhead in terms of hello messages; d) 3DB is able to reach a large fraction of nodes even when the nodes are moving at high speeds; e) In 3DB, a node independently decides whether to retransmit a broadcast message or not; hence, 3DB is robust to transmission errors as shown by our simulation results. Because of the above mentioned advantages, 3DB can also be used as an efficient broadcast protocol for Wireless Networks that operate even in adverse conditions. At the best of our knowledge, this is the first broadcast protocol designed for air to air communications.

The rest of this section is organized as follows: Section 3.10.1 introduces a modification of the Covering Problem for three Dimensional spaces, Section 3.10.2 our approach for broadcasting, Section 3.10.3 presents the simulation results of 3DB.

3.10.1 Background

- **Modified Covering problem for 3D Spaces** The Covering Problem can be stated as ” *What is the minimum number of circles required to completely cover a given 2-dimensional space.*” Kershner [32] showed that no arrangement of circles could cover the plane more efficiently than the hexagonal lattice arrangement.

For a three dimensional space, the modified covering problem can be stated as ” *What is the minimum number of spheres of Radius R required to entirely cover a three-dimensional space with the condition that the center of each sphere being placed lies on the surface of at least one other sphere.*”

If the range of a mobile node is considered to be R , then the reason behind the condition that the center of a sphere should lie on the surface of another sphere is that a node has to receive a message for it to retransmit the message. We develop a possible solution for the Modified-Covering Problem in 3-D networks. It is known that that: ”To completely cover the surface of a sphere S of radius R four spheres of radii R are needed such that the center of each covering sphere lies on or within the covered sphere.” (See Figure 3.22 and Figure 3.23.)

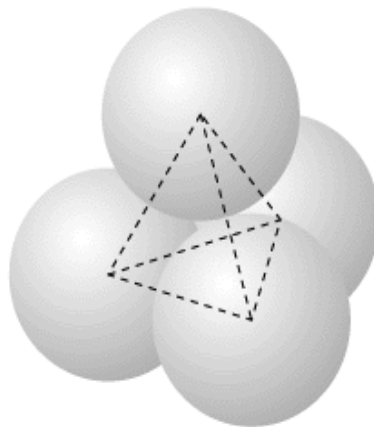


Figure 3.22. Arrangement of four spheres to cover a sphere completely

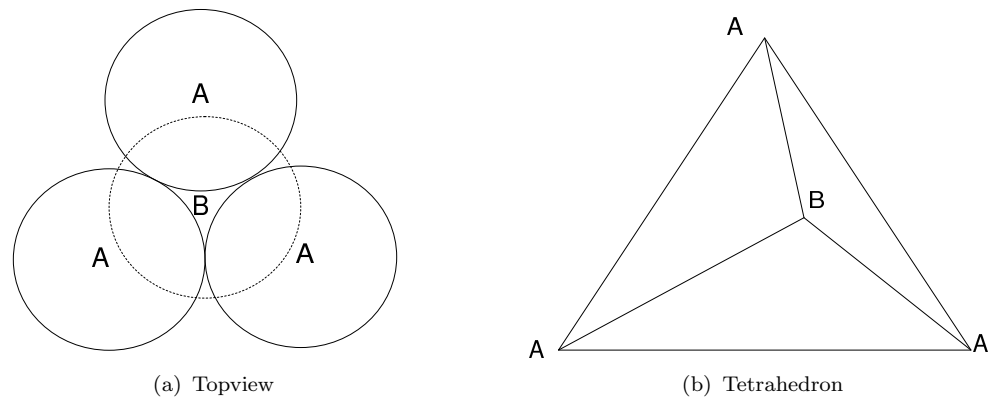


Figure 3.23. Completely covering a sphere

Now, in this arrangement four equal spheres are placed so that each touches the other three, thus all the four spheres covering the entire sphere at the center. The centers of the four spheres form a regular tetrahedron. We would also observe that the vertices of a tetrahedron of side length R can be given in a particularly simple form when the vertices are taken as corners of a cube of side length $R/\sqrt{2}$ as shown in Figure 3.24.

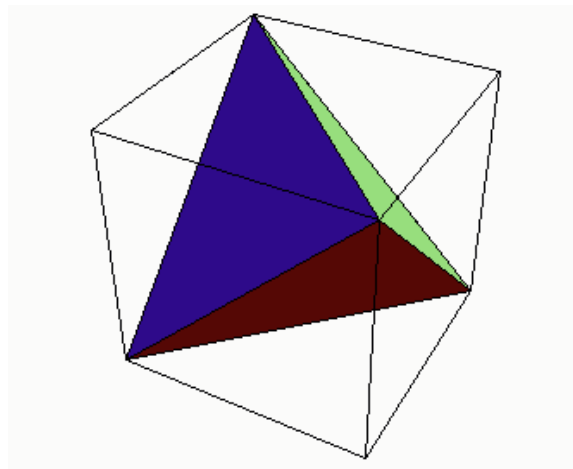


Figure 3.24. The vertices of a tetrahedron of side length R are also the corners of a cube of side length $R/\sqrt{2}$

The regular tetrahedron whose vertices are the sphere centers then has 0.7796 of its volume occupied by portions of the spheres. This gives the Rogers bound [57], an absolute upper bound to the density of any possible packing of equal spheres in Euclidean 3D-space. It is unattainable because regular tetrahedra do not pack

together without gaps. The densest packing of equal spheres is the well known hexagonal close packing, with density 0.7405.

Frank and Kasper [58, 59] investigated the possibilities of space filling by almost regular tetrahedra. The dihedral angle of a regular tetrahedron is about 70.50, so that five of them can share a single edge, leaving only a small gap which can be closed by reducing the length of the common edge until the dihedral angle at this edge becomes 720. We then have a pentagonal bipyramid with ten equilateral triangular faces. Similarly, twelve regular tetrahedra can share a single vertex; the gaps between them can be closed by a slight deformation. Another way of filling the entire space is with Octahedra and Tetrahedra [60] and a nice illustration can be found at [61].

We just use the approach of filling the space with regular tetrahedra, even though it results in some small gaps, mainly for following reasons: the gaps are very small; due to the very fact that in a real network, there might not be a node present at the strategic location, resulting in deformation of the ideal structure. Also, each point in space might be covered by multiple transmissions means that the chance that a node does not receive a message is very small. Simulation results indeed show that the performance based on this approach is very efficient and more than 95% of nodes receive the broadcast message. Another key factor is that this is the simplest approach while other approaches might be computationally more difficult and need more information to be carried in the packet header.

3.10.2 Three Dimensional Broadcast Protocol - 3DB

In this section, we present the Three Dimensional Broadcast Protocol (3DB). The intuition behind our protocol is that in order to achieve the goal, there is no need for all nodes to transmit/retransmit the message. Instead, the goal can be achieved by allowing only a few strategically selected nodes to retransmit the message. The strategy to select such nodes is same as the strategy to solve the Modified Covering Problem for three dimensional networks presented in Section 3.10.1.

- **Approach**

As noted earlier, 3DB is very similar to OFP. Let S be the Source node that sends the route request. The source also includes the centers of four spheres that

could entirely cover its spherical transmission region, in the broadcast message. In a real network, a node might not exist at these specified points. Thus, nodes closest to these points are selected to retransmit. Each node decides by itself whether to retransmit or not using a delay mechanism (described later). These can be considered as first stage retransmissions of the request. From here on, each of the retransmitting nodes includes three vertices of the tetrahedron with it as the center of the tetrahedron and the node from which it received the message as the fourth vertex.

- **The Algorithm** Each broadcast packet contains the location fields in which the (re)transmitting node stores the next strategic locations (four in case of the source and three in case of all other retransmitting nodes).

The Three Dimensional Broadcast Protocol is as follows:

1. The Source Node S stores the information of the four vertices of the regular tetrahedron with it as the center in the broadcast packet header and transmits the packet.
2. A node M , upon receiving a broadcast packet, first determines if the packet can be discarded. A packet can be discarded under any of the following conditions:
 - If the node has transmitted the packet earlier.
 - If a node which is very close has already transmitted this packet, i.e., if $d_n < Th$.
3. If the packet is not discarded, finds the nearest location the header, say V . It computes its distance l from V and then delays the packet rebroadcast by a delay d given by $d = l/R$.
4. After delay d , M aborts retransmission if it has received the same broadcast packet again from a node closer to V than itself or if $d_n < Th$ (Thus, delaying enables a node to decide if it is the nearest node to the strategic location.). Else, M updates the location fields in the broadcast packet with the next strategic locations and retransmits the packet.

The computational complexity of 3DB is negligible; when compared to flooding, the major additional computation is finding the strategic locations or the vertices of a tetrahedron, which can be computed easily by using the observation shown in Figure 3.24. The only bandwidth overhead due to 3DB is because of addition of new header fields to carry location information.

3.10.3 Simulation Results

We have used *ns-2* simulator [55] to evaluate the performance of 3DB. The radio range of each node is considered as unity. The nodes were uniformly distributed all over the region with densities varying from 5 nodes per cubic unit to 20 nodes per cubic unit. We use this value for all further simulations. Every simulation is repeated until the 95% confidence intervals of all average results are within $\pm 5\%$.

- **Contention** We analyze the performance of various protocols in terms of contention caused by the protocol. To address the contention problem, consider the situation where a node i broadcasts a message and there are n nodes hearing this message. If all these nodes try to rebroadcast the message, contention may occur because two or more nodes are likely to be close and thus contend with each other on the wireless medium.

We studied the probability of contention through simulations by randomly placing n nodes in node i 's transmission range. We observed the probability that all n nodes experience contention and probability of having one contention-free node. The results are shown in Figs. 3.25 and 3.26. We considered flooding protocols in which the broadcasting nodes proactively chose neighbors to rebroadcast.

From figure 3.25, we can see that with flooding, probability that all n nodes experience contention increases rapidly and is more than 0.8 even in presence of just six neighbors. For neighbor protocols, the probability increases at a much slower pace because number of retransmitting nodes (depends on network topology) might not increase with the increase of number of neighbors. With 3DB, the probability is almost zero. This is essentially because of two reasons - first, in 3DB at most three neighbors (four in case of the source) would be retransmitting irrespective of number of neighbors; second, because of delay based self-selection of retransmitting nodes, probability that two nodes experience the same delay is very low, thus reducing the probability further.

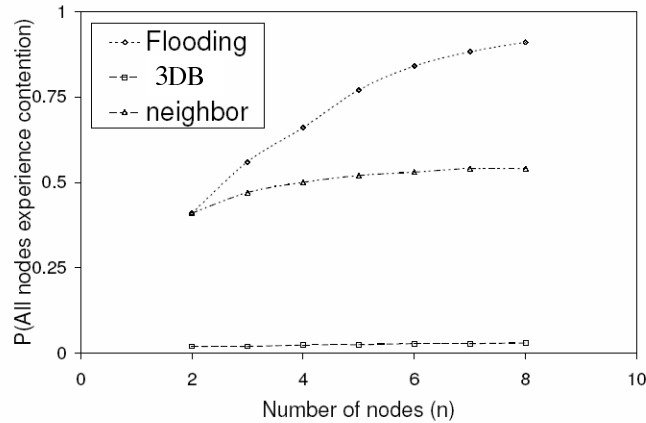


Figure 3.25. The probability that all n nodes experience contention

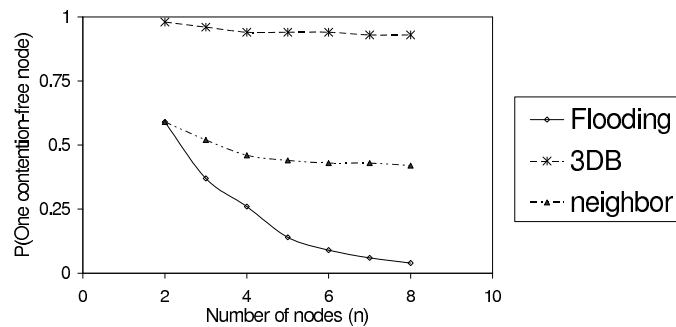


Figure 3.26. The probability of having one contention-free node

Probability of having one contention-free node is shown in figure 3.26. Understandably, with flooding this probability drops sharply as n increases. Further it is more unlikely to have more contention-free hosts. With neighbor protocols, because only few nodes retransmit, the probability does not decrease as rapidly as with flooding. Again for the same reasons mention above, with 3DB, the probability remains close to one.

- **Efficiency of 3DB**

We study the number of transmissions required by 3DB for different network sizes. figure 3.27 shows the performance results for various networks of sizes $6 \times 3 \times 3$, $8 \times 4 \times 3$, $5 \times 5 \times 3$ and $10 \times 3 \times 3$. The number of nodes is varied from 400 to 1000. We observe that the number of transmissions increases linearly with network size, which implies scalability with respect to network size. We observed a delivery ratio greater than 95% in all scenarios. Also, as the number of nodes increases, the performance improves. This can be understood from the observation that the

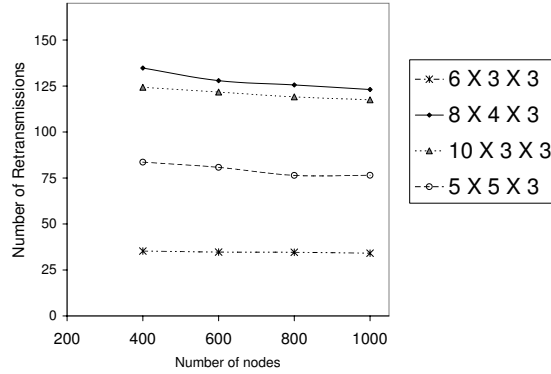


Figure 3.27. Number of transmissions required for different networks

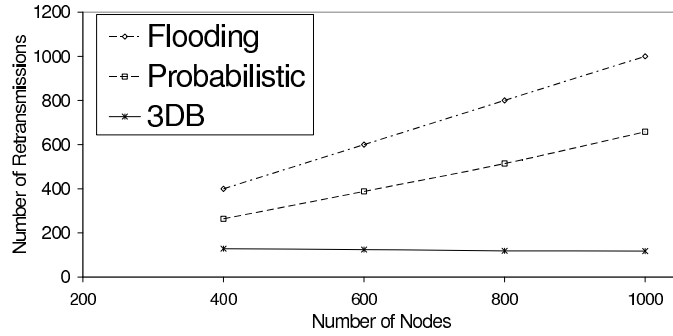


Figure 3.28. Performance comparison of various schemes

higher the density, the higher is the probability of finding a node close to the strategic location.

Figure 3.27 presents the performance comparison with pure flooding and Probabilistic [46] protocols in a network of size $10 \times 3 \times 3$. In the probabilistic scheme, each node retransmits the query with a probability of 0.65. It should be observed that 3DB outperforms the other schemes. In 3DB less than 12% of the nodes retransmit when there are 1000 nodes in the network.

- **Average delay per hop**

Assuming low load, in which there are no collisions, we observed the average delay a node has to wait at each hop before retransmitting. The network density is varied from 5 nodes per cubic unit to 25 nodes per cubic unit. The results are presented in the Table 3.3. The maximum allowed delay is 50 ms. We observe that

even at low densities, the delay is as low as 15.3 ms; while at high densities the delay is very low and nearly negligible.

TABLE 3.3. Delay observed for various densities

Density	Delay per hop (<i>msec</i>)
5	15.3
10	12.6
15	11.2
20	10.7
25	10.2

- **Mobile Networks**

This section presents the simulation results of 3DB in mobile networks. We use the Random Walk mobility model [56] with zero pause time. The range of mean speeds of the nodes is varied from 0 to 0.1 units per second. For instance, if the radio range of a node is 4000m, then the maximum speed simulated corresponds close to 900*mph* (1440 km per hour), which we assume to be a realistic maximum speed of mobile nodes.

Figure 3.29 presents the effect of mobility on each of the protocols. The simulation is for a network with 400 and 800 nodes, with the network size being $5 * 5 * 3$. The performance of 3DB remains almost unaffected, as 3DB does not need any neighborhood information. At the maximum simulated speed, the number of re-transmissions increases slightly (less than 5%), while the delivery ratio remains unaffected.

We would like to point out that any broadcast protocol that is based on neighborhood information either suffer in mobile networks as the topology keeps changing very frequently or would need very frequent location updates by all nodes leading to unacceptable communication overhead. Since, 3DB does not require any neighborhood information, 3DB does not have any of these drawbacks.

- **Effect of Transmission Errors**

Wireless networks are characterized by losses due to transmission errors. We simulated the performance of 3DB in networks with errors in transmission. Figure 3.30 presents the performance of 3DB in a network of size $5 * 5 * 3$. In the simulations, the number of nodes is varied from 400 and 800. These simulations

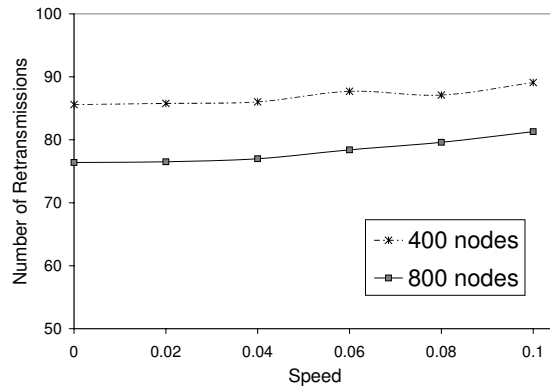


Figure 3.29. Effect of Mobility on different protocols. Network size = $5 * 5 * 3$

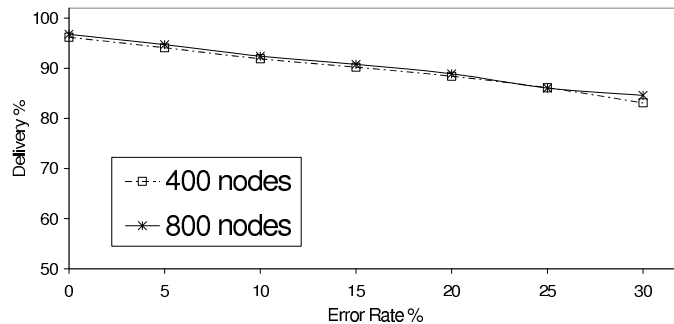


Figure 3.30. Performance of 3DB in presence of transmission errors. Network size = $5 * 5 * 3$

were for static networks. Transmission error rates up to 30% were simulated and we simulated Uniform transmission error model. The performance in both the networks is identical. It can be seen that the performance of 3DB degrades gracefully with increase in transmission errors and 3DB was able to achieve a delivery ratio of 84% even at a transmission error rate of 30%. The results show the robustness and resilience of the protocol.

This makes 3DB a good choice for wireless networks that operate in adverse conditions. The high delivery ratio of 3DB can be attributed to the fact that each node decides on its own whether to retransmit a packet or not and the decision is not based any neighborhood information. In presence of transmission errors, the closest node to the strategic location that has received the packet properly will retransmit. Also, one might expect that at a transmission error rate of 30%, on an

average around 30% of the nodes would not be able to get the packet error free and the delivery ratio should be less than 70%. But, it should be noted that most of the nodes in the network receive a packet more than once and from different directions and hence, delivery ratio is significantly better than 70%.

3.11 Summary

Building efficient broadcast protocols for wireless networks is challenging due to the energy, communication, and computation constraints. In this chapter, we proposed a novel Optimized Flooding Protocol(OFP). The adaptive-geometric approach makes OFP very scalable and energy efficient due to the minimum number of transmissions. OFP maximizes each hop length while getting the best out of the existing radio propagation situation. OFP is performed in an asynchronous and distributed manner by each node in the network. Nodes do not need any neighborhood information, therefore the communication and memory overhead is low. The efficiency of OFP remains very high even in large networks and OFP scales with density. Its efficiency in mobile networks and its robustness even in the presence of transmission errors make it an ideal choice for mobile ad hoc and sensor networks. OFP has been extended to three dimensional networks and its performance has been studied.

Chapter 4

Adaptive Routing and Energy Management for Heterogeneous Wireless Networks

If our behavior is strict, we do not need fun!

- *Zippy*, the Pinhead

Creativity is the ability to introduce order into the randomness of
nature.

- *Eric Hoffer* (1902-1983)

Routing in a communication network is the process of forwarding a message from a source host to a destination host via intermediate nodes. In wired networks, routing is commonly a task performed by routers, special fail-safe network hosts particularly designed for the purpose of forwarding messages with high performance. In ideal wireless ad hoc networks, in contrast, every network node may act as a router, as a relay node forwarding a message on its way from its source node to its destination node. This process is particularly important in ad hoc networks, as network nodes are assumed to have restricted power resources and therefore try to transmit messages at low transmission power, leading to the effect that the destination of a message can typically not be reached directly from the source. The importance of this task also becomes manifest in the popular term multihop routing, expressing the essential role of network nodes as relay stations.

In wired networks, routing almost always takes place in relatively stable conditions. The main focus of routing in wired networks is on high-performance forwarding of messages; reaction latency in the face of network topology changes, caused by failing hosts or connections, is generally of secondary importance. Considering the stability of wired networks, prompt reaction to topology changes or rapid propagation of according information is often not required; as such events are relatively rare.

Wireless ad hoc networks are of a fundamentally different character: To begin with, wireless connections are by nature significantly less stable than wired connections. Effects influencing the propagation of radio signals, such as shielding,

reflection, scattering, and interference, inevitably require routing systems in ad hoc networks to be able to cope with comparatively low link communication reliability. Also, many scenarios for ad hoc networks assume that nodes are potentially mobile.

Apart from the above two factors, more importantly, for various reasons nodes might not participate in routing all the times. In several wireless architectures, the nodes continuously expend energy even during idle and listening modes. Hence, to increase the lifetime of power-limited wireless devices, much research has focused on energy efficiency by having a node turn off the radio whenever possible. An important question to be addressed is providing periodic energy-efficient radio sleep cycles while minimizing the end-to-end communication delays.

Another critical challenge is that the protocols need to adapt to the ever changing wireless environment including the traffic loads, energy levels and node failures for efficient performance and prolonging network lifetime. This chapter presents adaptive protocols to efficiently route and manage the energy of the nodes to reduce delay and prolong the network lifetime. These factors cause ad hoc networks to be inherently more dynamic than wired networks. Traditional routing protocols designed for wired networks therefore generally fail to satisfy the requirements of wireless ad hoc networks. Routing in wireless network is challenging because of the unpredictable behavior of the medium and the proactive effect of interference. Compared to the wired networks the degree of variability of the state of wireless networks is quite high. Also the performance of the network, in terms of delay and throughput, is highly dependent upon the state of the network. The effects of the state of a wireless network are spread across several layers. Thus in order to meet the requirements of the application despite variable link state, network topology and power levels, it is important that the layers coordinate and adapt to the changes in network state.

The cross layer approach is perceived as an efficient solutions for designing protocols for the wireless networks. The cross layer design aims to achieve adaptivity and optimal performance by allowing sharing of information across several layers.

We present Adaptive Routing and Energy Management (AREM), a novel power management and routing protocol for heterogeneous wireless networks. While re-

ducing energy consumption is the primary goal in our design, AREM protocol achieves good scalability and low latency. To achieve the primary goal of energy efficiency, we reduce idle listening by making the nodes operate at low duty cycle modes. Low duty cycle increases latency and reduces throughput. To reduce latency, AREM uses the concept of forwarding sets. Unlike in geographic routing where a packet is forwarded to a node that is closest to the destination, a packet can be forwarded to any node in the forwarding set as detailed further in this chapter. The design reduces the energy consumption due to idle listening and reduces latency because of the presence of multiple forwarding nodes.

AREM design also considers adapting the transmission ranges to the prevailing network conditions like load and node energy levels. We develop a model to derive the optimal transmission range in order to minimize the end-to-end delay to satisfy the delay constraints and incorporate into the design of AREM. Also, AREM ensures fairness in energy consumption. AREM distributes the load of forwarding messages among the nodes according to their remaining energy. Therefore, AREM handles seamlessly the presence of heterogeneous nodes, by using resources of high-capability nodes to the advantage of other nodes with less energy.

The rest of this chapter is organized as follows: Section 4.1 discusses related work, Section 4.2 discusses the Adaptive Routing and Energy Management mechanisms. In Section 4.3 we present the analytical model and in Section 4.4, we introduce Load Adaptive Sleep Scheduling mechanisms. In Section 4.5 we validate our analytical model through simulations and study the performance of AREM.

4.1 Related Work

- **Energy Efficiency**

There are several solutions addressing the problem of energy wastage due to idle listening. Energy conservation is of paramount importance in wireless networks. The main sources of energy wastage are collisions; idle listening, over hearing and control packet overhead. All MAC protocols, contention based (like CSMA) or scheduled protocols (like TDMA) try to avoid collisions. Next major energy wastage source is idle listening, which occurs when radio is listening to the channel to receive possible data. The energy spent during idle listening is comparable to energy spent during transmitting or receiving, even though data is sent or received

during this period. Overhearing occurs when a node receives packets that are destined to other nodes. Lastly sending, receiving and listening for control packets consume energy, which reduces the effective throughput.

Controlling the node radio by setting radio to sleep mode when no data is expected and wake up when communication is expected (wakeup schemes) to reduce save energy has been proposed and studied by several researchers. Wakeup schemes can be classified as synchronous and asynchronous. Synchronous wakeup approach is used by the IEEE 802.11 [62] ad hoc power save (PS) mode. This method requires time synchronization all hosts. Time synchronization in a large scale distributed networks is generally very costly. Many proposals exist for asynchronous wakeup schemes, wherein each node follows a certain schedule of periodic wakeup and sleep. The final objective of all the schemes is to guarantee the overlap of wakeup times of neighboring nodes within finite time.

An asynchronous wakeup scheme for mobile ad hoc networks by Zheng et al [63], builds on the block design problem in combinatorics. The energy savings and wakeup delay can be improved by an additional wakeup or signaling radio. The PAMAS (Power Aware Multi-Access) protocol [64] is an adaptation of the basic mechanisms of IEEE 802.11 to two-radio architecture. Since the power consumption of the wakeup radio is significantly low compared to the data radio, it can be awake for entire period, consuming little energy. PAMAS however ignores the idle listening problem. The main drawback is that low power wakeup radio has lower transmission range than the data radio. This causes limitations where two nodes are within data radio range and not in wakeup radio range. Also, two-radio architecture is expensive to implement on wireless nodes.

STEM (Sparse Topology and Energy Management) [65] also uses two radios, one is used as a wakeup radio and other is used for data transmission. In STEM, each node periodically turns on their wakeup radio for T_{wake} every T duration, where T_{wake}/T is defined as duty cycle. Low power consumption of wakeup radio is achieved by having high duty cycle ratio instead of low power wakeup radio, thus avoiding some problems discussed above.

S-MAC [66] is a protocol developed to address the energy issue in the wireless networks, building on contention-based protocols like IEEE 802.11. S-MAC follows

a simple scheduling scheme to allow neighbors to sleep for long periods and synchronize wakeups. A complete sleep/wake cycle constitutes a frame. Each frame begins with a listen period for nodes that have data to send to coordinate. A sleep period follows, during which nodes sleep for a certain period if they have no data to send or receive and nodes remain awake and exchange data if they have data to communicate. All nodes independently choose their listen/sleep schedules and share their schedules with neighbors. S-MAC needs synchronization to some extent, but that is not as critical as in TDMA-based protocols. Also, S-MAC uses a fixed sleep interval regardless of traffic load. T-MAC [67] extends S-MAC by adjusting the length of the time nodes are awake between sleep intervals based on communication of neighbors. Thus, less energy is wasted due to idle listening when traffic is light.

GeRaF [68, 69] also proposes and analysis the performance of routing based on forwarding sets for networks that are based on two-radio architecture. Also, the analysis and tuning of the parameters are directed toward networks with low or no loads. GeRaf could be considered as the closest work to AREM. GeRaF proposes and analysis the performance of routing based on forwarding sets for networks that are based on two-radio architecture. Rigorous analysis and tuning of the parameters are presented for networks with very low loads. The major drawback of having a fixed and uniform duty cycle might lead to inefficiency; at high loads, the duty cycle has to be high (to permit for congestion and contention) which leads to inefficiency when the load is low. Also, GeRaF does not consider adapting the transmission ranges to the network conditions.

- **Optimal Range Allocation**

Several researchers have studied the problem of optimal range allocation in wireless networks.

A power-controlled multiple access wireless MAC protocol (PCMA) which generalizes the existing collision avoidance protocols is proposed in [70]. [71] proposes a transmit power control (TPC) mechanism to address the trade off between the MAC TPC and the physical layer (PHY) transmission rate.

To achieve the minimal interference at non-involved nodes, several iterative power control algorithms have been developed. The power control scheme in [72]

provides protection for links that are currently operational, that is, their signal-to-interference ratios (SIRs) are maintained above a certain threshold at all times. However, these protocols require prior knowledge or perfect estimates of quantities such as the SIR. To overcome this challenge, [73] presents a new power control algorithm that makes use of available measurements, and then converges stochastically to the optimum power.

Gobriel et al. [21] study the trade off between the low transmission power and the high probability of collision per message arising from increasing the number of hops on the path from source to destination. They come to the conclusion that sending the data packet to the nearest neighbor is not always optimal. They do not, however, account for the required latency when selecting the transmission power level.

To the best of our knowledge, we are not aware of any study that studies the impact of sleep scheduling mechanisms on assigning the optimal range of wireless nodes.

4.2 Adaptive Routing and Energy Management (AREM)

AREM mainly consists of two main components - Adaptive Energy Management (AEM) and Adaptive Routing Mechanism based on forwarding sets (ARM).

AEM is a random wakeup scheme that allows a node to be active for a randomly chosen fixed interval during each time frame. This removes the necessity of time synchronization and makes the protocol implementation very simple and practical.

The routing methodology in ARM is designed to take advantage of the fact that wireless networks are densely deployed. In conventional routing protocols, shortest path between two nodes is computed proactively or reactively and a node forwards a packet only to the next node in the shortest path computed. High node density results in existence of several paths between two given nodes, whose path lengths are very close to the shortest path. Thus, a packet can be forwarded to any of several nodes in order to be delivered to the destination with out affecting the path length and delay experienced by the packet when compared to forwarding through the shortest path.

In this section, we first describe the energy management methodology of AREM. Then we elaborate our routing methodology that is based on forwarding sets and

then study the random wake up scheme, followed by description of AEM. Finally, we describe how AEM adapts to the energy levels of the nodes so as to efficiently balance the network load among the nodes.

4.2.1 Adaptive Energy Management (AEM)

AEM uses the idea of random wakeup/active. Each node is active random for a given duration of time. In this section, we first describe random wakeup scheme and then describe the protocol implementation.

The idea is to have each node wake up once in every slot, be awake for a predetermined time, and then sleep again. To elaborate, consider time slots of fixed interval T and the active time of T_a for each wireless node in each time slot ($T_a < T$). Thus, if there are m neighbors in the forwarding set of node S to which a packet destined to D can be transmitted to, then the probability that at least one of those nodes is awake, when S is awake is given by:

$$P = 1 - \left(1 - \frac{T_a}{T}\right)^m \quad (4.1)$$

Figure 4.1 shows the probability that at least one node in the forwarding set is active for different T_a values. It should be noted that even for a T_a as low as 15%, at a node density of 10, a node could find an active neighbor to whom it can forward the packet with high probability ($> 82\%$). For higher densities, the probability is even higher. Thus, even if a node is active for a randomly selected duration of T_a , there is a high probability that a packet can be forwarded to the destination. This is used as the basis of design in AEM. The protocol is detailed in the following section.

4.2.2 Adaptive Routing Mechanism Based on Forwarding Sets (ARM)

We use routing based on forwarding sets to reduce the latency drastically. A wireless network is densely deployed for several reasons, some being to increase reliability, redundancy, accuracy and lifetime. Thus, instead of routing through a particular best neighbor that leads to the shortest path, a node can forward a packet to any one of a set of neighbors that lead to short paths. Though this leads to an increase in path lengths, we show that the increase in path lengths/energy expended is not significant at the same time significantly reducing the latency.

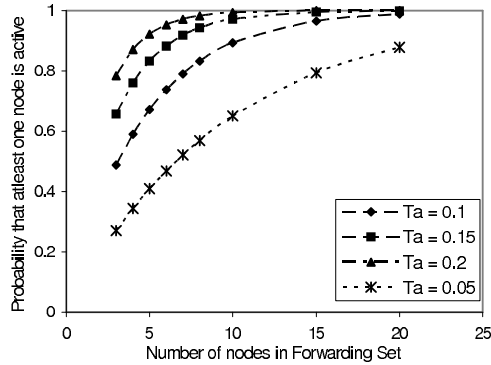


Figure 4.1. Probability that at least one node in the forwarding set is active for different active times

- **Routing Based on Forwarding Sets**

In the geographic routing protocol, a packet is forwarded to a neighboring node that is closest to the destination. However, in a wireless network, in which not all nodes might be active at a given point of time, a packet can be forwarded to the active neighbor that is closest to the destination, or the packet can be queued until the closest neighbor among the rest becomes active, and the packet can then be forwarded to this neighbor.

ARM is a modification of the geographic routing protocol such that a packet is sent to any of the active neighbors that meet a forwarding criterion (discussed later). We define Neighboring Set and Forwarding Candidate Set as follows:

- The Neighbor Set of node i : This is the set of nodes that are inside the radio range R of node i .

$$NS_i = \{node \mid distance(node, node \ i) \leq R\}$$

- The Forwarding Candidate Set of node i : For a given destination, this is the set of potential neighboring nodes to which node i can forward a packet.

We consider two criteria for defining the Forwarding Candidate Set - one is based on path lengths and the other is based on geographic distance to the destination.

- **Hop based Forwarding Candidate Set (h-FCS)**

Forwarding criterion: For a given source s and destination d , a neighbor k of s is a node in FCS if

$$H(k, d) < H(s, d) + \Delta$$

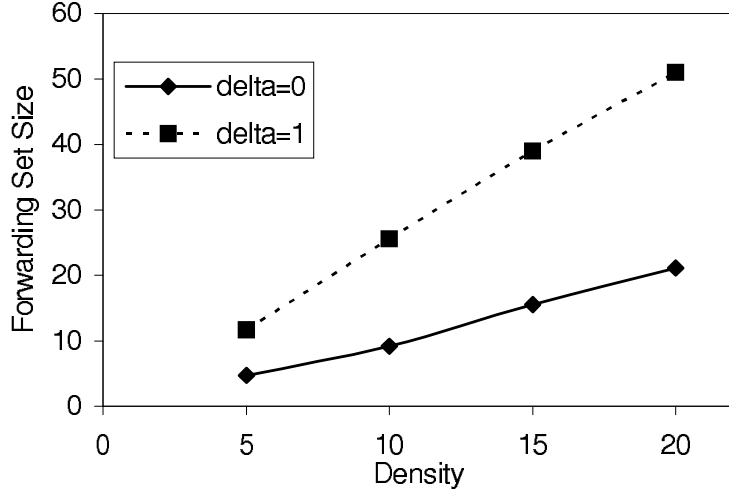


Figure 4.2. The size of Forwarding Candidate Set for $\Delta = 0$ and $\Delta = 1$

where, $H(i, j)$ is the hop length of the shortest path between nodes i and j .

When $\Delta = 0$, it implies that a shortest path between s and d exists through node k . When $\Delta > 2$, every neighbor of s belongs h-FCS. This is because, for a given neighbor k , there always exists a path $s \rightarrow k \rightarrow s \dots \rightarrow d$ whose length is $H(s, d) + 2$, thus satisfying the forwarding criterion. Also, it should be noted that unless $\Delta = 0$, selecting a forwarding node based on this forwarding criterion does not guarantee that a packet reaches the destination. This is because the path length to the destination from any two neighbors in the path can be same. Figure 4.2 shows the number of nodes in h-FCS for $\Delta = 0$ and 1 for different densities.

Computing h-FCS requires each node to know the shortest path length to all other nodes in the network. Thus, this criterion of selecting FCS might not be very appealing owing to the computational overhead involved. To overcome this overhead, in the following section, we propose selection of FCS based on the geographic distances between the nodes.

- **Distance based Forwarding Candidate Set (d-FCS)**

Forwarding criterion: For a given source s and destination d , a neighbor k of s is a node in FCS if:

$$D(k, d) < D(s, d) - Th$$

where, $D(i, j)$ is the geographic distance between nodes i and j .

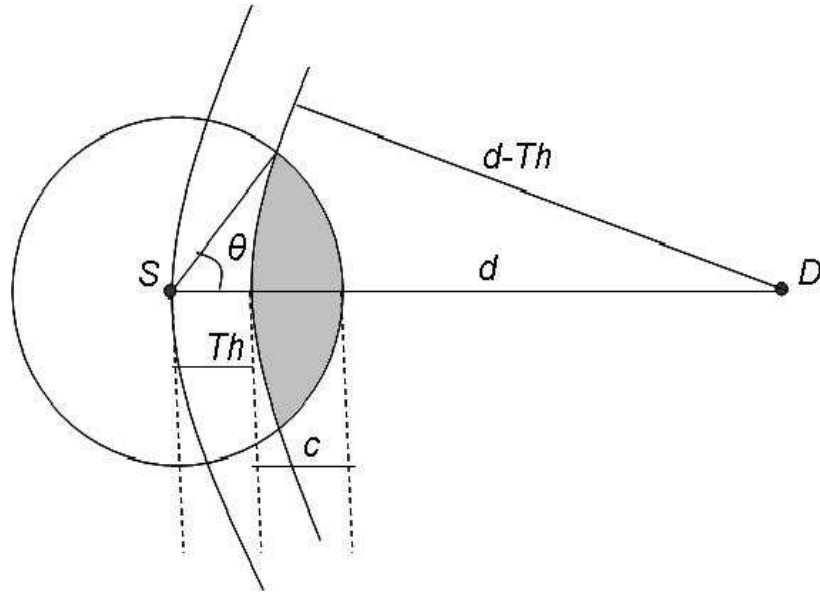


Figure 4.3. Forwarding Candidate Set is set of all nodes lying in the shaded region

Thus, if a neighbor k is closer to the destination by at least Th than the node s itself, then k belongs to the *Forwarding Candidate Set* (see figure 4.3). The d-FCS selection criterion guarantees that there would be no loops in the path. This is because a node always forwards a packet to a node that is closer to the destination than itself. At the same time, this simple criterion cannot guarantee the delivery of a packet to the destination in presence of holes. At high network densities, it can be safely assumed that holes would not exist. In case holes are present, the criteria for selection has to be extended based on the ideas presented in [74]. We assume that no holes are present in the network.

Routing based on forwarding sets increases the path length. The Th value limits the maximum path length, as with each transmission a packet traverses at least a distance of Th towards the destination. Intuitively, because of increased path lengths, it might seem that Forwarding Set based routing adds additional overhead in terms of energy consumption. However, when combined with the random wake up scheme the total energy consumed by a node with AREM is lower.

- **Optimal value of Th** In this work, we use a modified geographic routing protocol in which a packet is sent to an active neighbor closer to the destination by at least Th than the forwarding node itself. In this section we derive the optimal

value of Th in order to minimize the expected end-to-end delay experienced by a packet.

Let S be location of the sender and D the location of the destination. A packet is forwarded by a node to its neighbor, if the neighbor is closer to the destination by at least Th . In this section, we show that the optimal value of $Th = 0$.

Consider the Figure 4.3. A neighbor is an eligible candidate to receive the packet if it lies in the region of intersection of two circles drawn with centers as S and D and with radii r and d . Without loss of generality, for simplicity, we assume the radio range of each node is $r = 1$.

The expressions for various values can be derived as follows:

$$\theta = \cos^{-1}(Th) \quad (4.2)$$

The area of intersection can be expressed as:

$$A_{int} = r^2 \cos^{-1} \left(\frac{d^2 + r^2 - R^2}{2dr} \right) + R^2 \cos^{-1} \left(\frac{d^2 + R^2 - r^2}{2dR} \right) - \frac{1}{2} \sqrt{(-d + r + R)(d + r - R)(d - r + R)(d + r + R)} \quad (4.3)$$

where $r = 1$ and $R = d - Th$. The area of intersection takes the minimum when $d = 1$ and the maximum when $d = \infty$, the maximum being

$$A_{int} = \cos^{-1}(Th) - d\sqrt{1 - Th^2} \quad (4.4)$$

For simplicity, we consider the case when $d = \infty$. Also, when $d \gg 1$, $A_{int} \approx \max(A_{int})$. The procedure can be easily extended to other cases to obtain the same result.

Now, the distance of the geometric centroid from the center of the circle, \bar{x} , can be expressed as:

$$\bar{x} = \frac{4 \sin^3 \theta}{3(2\theta - \sin 2\theta)} \quad (4.5)$$

In Section 4.3.3, we show that the expected delay due to the random sleeping is $\bar{T}_s = T/(n + 1)$, where n is the number of nodes in the forwarding set. Thus, $n = A_{int} * \rho$, where ρ is the density of network. In order to minimize the delay experienced at each hop, we need to maximize \bar{x}/\bar{T}_s . Thus, the optimal value of Th is the one that maximizes (\bar{x} / \bar{T}_s) and can be computed as $Th=0$.

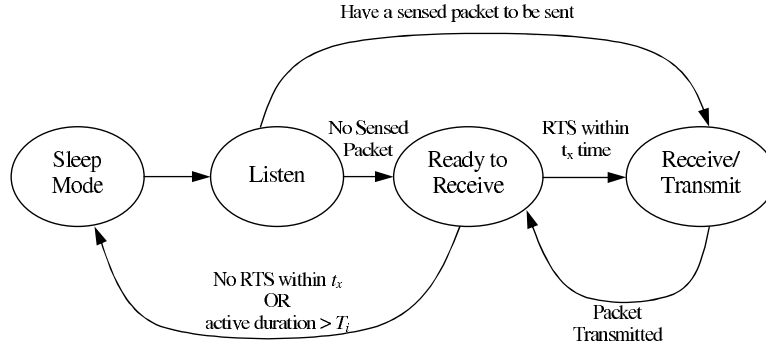


Figure 4.4. State-transition diagram of Wakeup Scheme

4.2.3 The AREM Protocol

Every node gets up periodically, transmits a beacon message indicating that it is ready to receive/forward a message on control channel. It waits for duration t_x for a reply. If it gets an RTS from any of its neighbor in that duration, it receives the message. Then it checks if it can forward the message to any of its neighbor. If no neighbor in the forwarding set is awake it waits until a neighbor is awake. Then it forwards the message to that node and goes to sleep again.

Let T_{setup} be the time taken by a node to send a beacon message once it is awake and receive a reply consisting of its neighbor information. t_x is the duration that the node waits for an RTS. In order to better explain the protocol, we make use of the state-transition diagram shown in Figure 4.4.

- **Sleep State** The node is in the sleep mode. Also, in order to conserve energy, it is desirable to maximize the time a node spends in sleep mode. Each node i selects its own schedule period T_i based on different parameters (described below) and sleeps for a duration of $T'_i = random(0, T_i)$ before it wakes up again. Thus, on an average, the node sleeps for $T_i/2$ duration between two active periods. Thus, if Γ_i is the average duration of active interval of node i , then its average duty cycle is $\frac{\Gamma_i}{\Gamma_i + T_i/2}$.

The value of T_i depends on the application, density of the network. Another factor that influences T_i is the energy level of node i and is further discussed in Section 4.2.4.

- **Awake/Setup State** When ever a node becomes active, it broadcasts a beacon message through the control channel, advertising to its neighbors that it is awake. Then, it checks if there is any packet generated/sensed by it to be

transmitted. In that case, the node goes directly to the Receive/Transmit state, else it goes to the Ready to Receive state.

- **Listen State** In this state, a node i keeps its receiver antenna active and listens to channel to see if any of its neighboring nodes are forwarding a packet. If it receives an RTS addressed to it, it goes to Receive/Transmit state. If the node does not receive any RTS with in t_x it goes to sleep. Also, to avoid excessive drainage of energy at node i , we put a constraint that at anytime a node cannot be continuously active for a duration more than T_i . The value of t_x should be set such that once a node into this state, if there is packet transmission is going on, the node has to be awake till the transmission is over and then still should be awake for some more duration to see if any node is sending an RTS to it.

- **Receive/Transmit State** A node in this state performs the tasks of receiving and transmitting packets. It should be observed that a node will be awake until it could forward the packet, after which it goes back to Ready to Receive state.

4.2.4 AREM Adaptation to Energy Levels (AREM-E)

Heterogeneous Wireless Networks are envisioned to comprise of nodes with different capabilities leading to different energy levels of nodes. Even in Homogeneous Wireless Networks, where all nodes have same energy levels during the bootstrapping stage, because of different roles/tasks each node would be performing, node energy levels vary from one another.

To simultaneously prolong the network lifetime as well as each nodes lifetime, it is required that nodes with higher energy levels forward more packets than nodes with lower energy levels. This can be achieved by setting T_i as follows:

$$T_i = T \cdot \frac{Avg_energy_i}{Energy_i} \quad (4.6)$$

where,

T is the duration value in case when all nodes have same energy levels.

$Energy_i$ is the energy level of node i .

Avg_energy_i is the average energy level of the neighbors of node i .

The intuition behind this is as follows: Total sleep duration of a node i can be expressed as $O(T_i)$. Since all nodes (when awake) have same probability of being

selected to forward packets, assume that all nodes are active for same duration. Thus, the duty cycle of a node is $1 - \frac{T_i/2}{T_w + T_i/2}$. As $T_w \ll T_i$, the duty cycle can be approximated to $\frac{2T_w}{T_i}$. Thus, rate of energy consumption is roughly inversely proportional to T_i . Hence, by having T_i in turn inversely proportional to the residual energy level, we have rate of energy consumption proportional to the residual energy level and thus meet our objective.

To compute the average energy level of neighbors, each node includes its energy level in the beacon message it broadcasts. Though each node would not be able to obtain the energy levels of all the neighbors, a node is able to approximate the average energy level fairly accurately with in few wake up cycles.

4.3 Analytical Model

In this section, we develop an approximate analytical model to study the effect of transmission range of nodes and obtain a relation between network load and transmission range to minimize the delay.

4.3.1 Average Path Length

Zorzi and Rao [69] derive the bounds for the average number of hops to reach a destination at distance D from source. They quantize the whole range of possible distances from the destination, from 0 through D . Let v be the number of quantization intervals per unit distance, so that the total number of intervals considered is Dv . More specifically, let $\Delta_i = (\frac{i-1}{v}, \frac{i}{v}]$ be the i^{th} quantization interval.

Consider the case in which the transmitter is at distance $\gamma = i/v > 1$. Then, $w(i, k)$, the probability that the advancement will lead to a remaining distance in interval $i - v + k$ is

$$w(i, k) = \begin{cases} e^{-A(\frac{i-v+k-1}{v}, \frac{i}{v})} - e^{-A(\frac{i-v+k}{v}, \frac{i}{v})} & k = 1, \dots, v \\ 0 & \text{otherwise} \end{cases} \quad (4.7)$$

In addition, with probability $w_0(i)$, there is no advancement, where

$$w_0(i) = e^{-A(\frac{i}{v}, \frac{i}{v})} \quad (4.8)$$

We would like to observe that the maximum advancement can be to the interval Δ_v , thus the range of a node can be expressed as $R = v$. Now, the average number of hops $n(D)$ to reach a destination at distance D can then be bounded as follows

$$n_2(D_v) \leq n(D) \leq n_1(D_v) \quad (4.9)$$

where,

$$n_1(i) = \frac{1 + \sum_{k=1}^{v-1} w(i, k) n_1(i - v + k)}{1 - w_0(i) - w(i, v)} \quad (4.10)$$

with initial condition $n_1(i) = 1; i = 1, \dots, v$

$$n_2(i) = \frac{1 + \sum_{k=1}^v w(i, k) n_2(i - v + k - 1)}{1 - w_0(i)} \quad (4.11)$$

with initial condition $n_2(i) = 1; i = 1, \dots, v$

4.3.2 Average Packets in the Network

Given that source-destination pairs are random, the expected distance between a S-D pair can be computed as follows: Let (x_1, y_1) and (x_2, y_2) be the coordinates of the source and destination. Then, the expected distance is the integral of $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ over all four variables between 0 and L, L being the network dimension. The integral can be evaluated to $(2 + \sqrt{2} + 5 * \log(1 + \sqrt{2})) * L/15$ which is approximately $0.521405 * L$.

Thus, for a given square network, with density of network being ρ and the radio range of each node being R, the expected distance length between a source-destination pair is

$$E[distance] = 0.521405 * L \quad (4.12)$$

Let each node in the network generate γ messages each second. Then the total number of messages entering the network can be computed as $\gamma\rho L^2$, with each message traversing a path of expected length $E[distance]$. Assuming, no packets are dropped, using Equation 4.9, average number of transmissions for each node can be bounded as

$$\gamma n_2(E[distance]) \leq \gamma_{node} \leq \gamma n_1(E[distance]) \quad (4.13)$$

4.3.3 Sleep Delay Characterization

Apart from the average virtual transmission time, time to transmit also includes the time a packet has to wait until a forwarding neighbor is awake. Let \bar{T}_s be the delay induced due to the scheduling algorithm. Then, $E[\bar{T}_s]$ can be expressed as

follows:

$$\begin{aligned}
E [\bar{T}_s] &= \int_0^{T-t_x} \left(1 - \frac{y}{T}\right)^n dy \\
&= T \cdot \frac{(-1)}{(n+1)} \cdot \left(1 - \frac{y}{T}\right)^{n+1} \Big|_0^{T-t_x} \\
&= \frac{T}{(n+1)}
\end{aligned} \tag{4.14}$$

n is the number of neighbors in the forwarding set.

The derivation is based on the observation that if a node is to be awake for at least t_x duration in every schedule period of length T , then it has to become active once in the next duration of T .

4.3.4 Total Service Time

Note that the service time distribution can be derived from the distributions of back-off delay and sleep delay.

Given the back off time characterization, the *average service time* can be expressed as

$$\bar{T}' = \bar{T} + \bar{T}_s \tag{4.15}$$

where \bar{T} is given by Equation 2.14 is the time to successfully transmit a packet. Thus, the probability that a node is transmitting is given by

$$P_{tx} = \min(\gamma n(D) \bar{T}', 1) \tag{4.16}$$

where \bar{T}' is the average service time which in turn depends on number of neighbors transmitting i.e., $\pi R^2 P_{tx}$. Equations 4.15 and 4.16 represent a nonlinear system in the two unknowns \bar{T} and P_{tx} , which can be solved using numerical techniques.

4.3.5 Average Duty Cycle

Let γ_i be the number of transmissions by node i during one awake period. Then, the average duration a node is awake can be expressed as $t_{awake} = (T_{setup} + \gamma_i (\frac{t_x}{2} + \bar{T}'))$. \bar{T}' is the average time take to transmit a packet and $\frac{t_x}{2}$ is the average duration node i waits before it receives an RTS. Thus, the average duty cycle for node i can be expressed as

$$duty cycle_i = \frac{t_{awake}}{t_{awake} + T/2} \tag{4.17}$$

4.3.6 Energy Consumption

In this section, we derive an approximate model to compute the energy consumed by nodes. Again we use the energy model in which the energy consumption depends on the range of the emitter u :

$$E(u) = \begin{cases} r(u)^\alpha + c_e & \text{if } r(u) \neq 0, \\ 0 & \text{otherwise} \end{cases} \quad (4.18)$$

As mentioned earlier, the model $\alpha = 4$, $c_e = 10^8$ is derived from a work by Rodoplu and Meng [31]. Nodes also consume some energy upon the reception of a message. This consumption c_r is constant, regardless of the distance between the emitter and the receiver. In the above model, this gives $c_r = \frac{2}{3} \times 10^8$. For simplicity, we assume that the energy consumed by a node while sleeping is negligible.

For each packet transmission, the duration of time a node spends in transmitting (either RTS, CTS, data or ACK) can be expressed as

$$T_{tx} = RTS + CTS + P + ACK \quad (4.19)$$

Thus, during one wakeup-sleep cycle, during which γ_i packets were transmitted, the energy consumption can be expressed as

$$E_i = \gamma_i T_{tx} (r(i)^\alpha + c_e) + (duty_cycle_i - \gamma_i T_{tx}) c_r \quad (4.20)$$

$r(i)$ is the transmission range of i .

4.4 AREM - Load Sensitivity

In this section we propose two Load Sensitive AREM algorithms: (i) AREM with Global Adaptation (AREM-GA) that forces every node to use same transmission range, and (ii) AREM with Local Adaptation (AREM-LA) that allows every node to independently decide its transmit range.

4.4.1 AREM with Global Adaptation (AREM-GA)

General wireless networks assume symmetric links and routes. Also, each node might be sensing its environment and periodically sending messages at pre-determined intervals. This leads to uniform transmission rates across the entire network. In such scenarios, we propose to use AREM-GA, where all nodes in the network use the same transmission range. The optimal transmission range can be computed as

presented in the previous section and can be advertised to the entire network. In case when the transmission rates change, the base station can compute the new optimal transmission range and advertise to the network.

4.4.2 AREM with Local Adaptation (AREM-LA)

In several scenarios, traffic rates might be different in different regions of the network. Hence, it is desirable for the ideal power control scheme to support distributed coordination among nodes. AREM-LA allows each node to adapt its own transmission range to its neighborhood conditions. However, because two neighboring nodes may use different transmission powers, some links will be asymmetric. While several recently proposed protocols tackle the presence of asymmetric links, the possibility of wide-spread proliferation of asymmetric links will also necessitate changes at the MAC layer.

One extension for IEEE 802.11 to support asymmetric links could be as follows: In the conventional IEEE 802.11 MAC, a sender transmits an RTS, and DATA packets to a receiver, and the receiver responds with CTS and ACK packets to the sender. Because the MAC layer uses the same power for all packets, asymmetric links can induce link failures. If the receiver, however, uses the power notified by the sender (say piggybacked on the RTS packet) to transmit CTS and ACK packets, asymmetric links can be supported successfully. While this will increase the header overhead by about one byte, it is a negligible increase. Since, our scheme uses geographic routing to forward packets, asymmetric links/routes would not pose severe challenges at the routing layer.

AREM-LA enables each node to independently adapt to its local environment. Whenever each node awakes, it computes the optimal range to minimize the delay at its hop and uses this range for all further transmissions during that interval. Thus, by optimizing the transmission range for each node and thus minimizing the delay at each hop, AREM-LA tends to minimize the end-to-end delay.

In AREM-LA, each node would piggy back its rate of transmission on the DATA packets. As nodes could be sleeping for most of time, a node might not be able to overhear packets from all of its neighbors. But, each node could estimate the approximate network load in its neighborhood based on the average traffic rate of over-heard DATA packets and the network density. As we observe, the average

TABLE 4.1. Physical layer parameters

Simulation parameters of 802.11	
W^*	64
MAC Header	34 bytes
ACK	38 bytes
CTS	38 bytes
RTS	44 bytes
Slot Time	20 μ sec
SIFS	10 μ sec
DIFS	50 μ sec
ACK Timeout	212 μ sec
CTS Timeout	348 μ sec
Raw Bit Rate	2 mbps
Packet Size	512 bytes

delay does not deviate significantly with in 20% of the optimal transmission range. Each node thus could periodically compute the approximate load in its neighborhood and adjust its transmission range accordingly.

4.5 Experimental Results

In this section we evaluate the accuracy of our model. For this purpose, we use the simulator *ns-2* to run simulations. All nodes transmit to some other node in the network according to the same CBR source rate with fixed packet sizes. We pick different source rates. Nodes are randomly placed in an area of $5 * 5$ and have no mobility. The transmission range is varied from 0.5 to 2. Each run corresponds to 15 minutes of data traffic. We trace each node in the network and compute both the mean and variance of its service time. We repeat the experiment for 20 different seeds for statistical reasons. Table 4.1 summarizes the parameters used for our simulations.

4.5.1 Effect of Sleep Duration

We observe that each node *sleeps* for a random duration $\in (0, T)$ between every two *wake up* durations, T being the sleep length. Figure 4.5 and 4.6 present the performance of LWP for different sleep lengths. For these simulations, the data rate is kept low at 1 packet/sec. As observed from Figure 4.5, the end-to-end latency significantly depends on T . As observed earlier, higher the sleep length, larger the delay due to node sleeping and hence higher the end to end latency. Figure

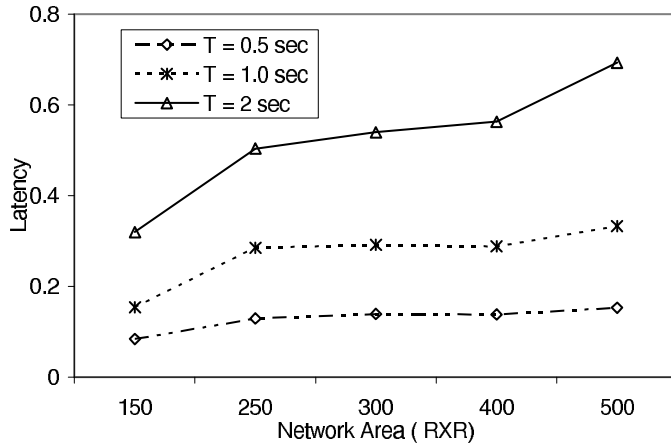


Figure 4.5. Effect of *sleep* duration on latency

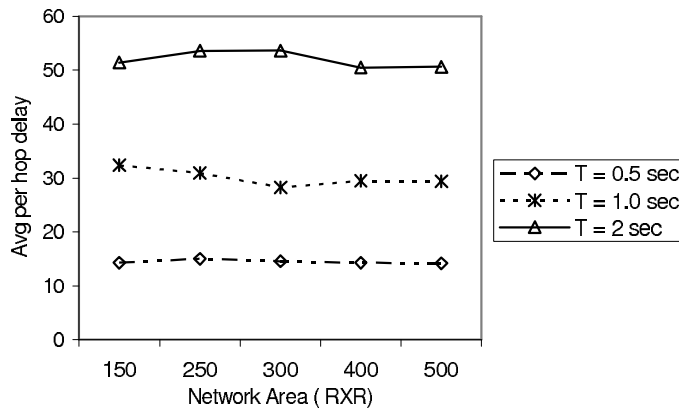


Figure 4.6. Effect of Network Size duration on per-hop latency

4.6 shows the average per hop delay for various networks. Per hop delay remains almost a constant in networks of different sizes.

4.5.2 Performance with Varying Loads

In this section, we present results on the performance of LWP as load varies. For this purpose, we choose networks with 150, 300 and 500 nodes corresponding to a density of 6, 12 and 20 respectively. The time period T is set to 1 sec and each node upon getting active will be awake for a minimum duration of 40 msec. This corresponds to average back off delay of eight nodes actively contending for the channel. There are 10 connections between randomly chosen source-destination pairs and the data rate of each connection is varied from 0 till the delivery ratio drops below 50%.

The average end-to-end latency is presented in Figure 4.7(a). As expected the latency increases with load. A significant observation is that as density increases,

TABLE 4.2. Path length Vs Transmission Range

Tx range	0.5	0.75	1.0	1.25	1.5	1.75	2.0
Path Length	3.8	3.22	3.0	2.9	2.83	2.78	2.74

there is a significant improvement with respect to the latency. This is because of the fact that, higher the density, lower is the delay factor due to nodes sleeping. In fact this also affects the capacity of the network as depicted by figure 4.7(b). At a network density of six, the delivery ratio is less than 50% for a packet rate of just five. At a node density of 20, data rates up to 9 packets/sec have a delivery ratio higher than 50%. For 80% delivery ratios, the data rates are 2.5, 4.5 and 6 packets/sec for densities 6, 12 and 20 respectively.

Figure 4.7(c) plots the average path length for different loads. We make two observations: first, path length decreases as density increases and second, path length decreases as load increases. The first observation can be seen from the fact that, higher the density, higher is the probability that a forwarding node closer to the destination can be found. The second observation can be explained as follows: as load increases, the duration a node stays active for longer durations. Again, the increased active durations lead to decrease in path lengths, as the probability that a forwarding node closer to the destination can be found. The active durations for various loads is shown in figure 4.7(d). At low data rates, the average active duration is same for various densities and close to the minimum active duration. As load increases, for low densities, the active duration increases more rapidly than for high densities, as at higher densities more nodes share the forwarding responsibilities.

4.5.3 Validation of Analytical Model

We initially present the numerical results we obtained for various networks obtained from our analytical model. Table 4.2 presents the average path length for different transmission ranges. As expected, the higher the transmission range, lesser the average number of hops to reach a node.

Figure 4.8 presents the analytical results for sleep duration of 1 second. Both average per hop delay and end-to-end delay are computed. We use the term *Load* to indicate the number of packets a node would be transmitting per second. At low loads (load = 0.1), sleep delay is the major component of delay, while at higher

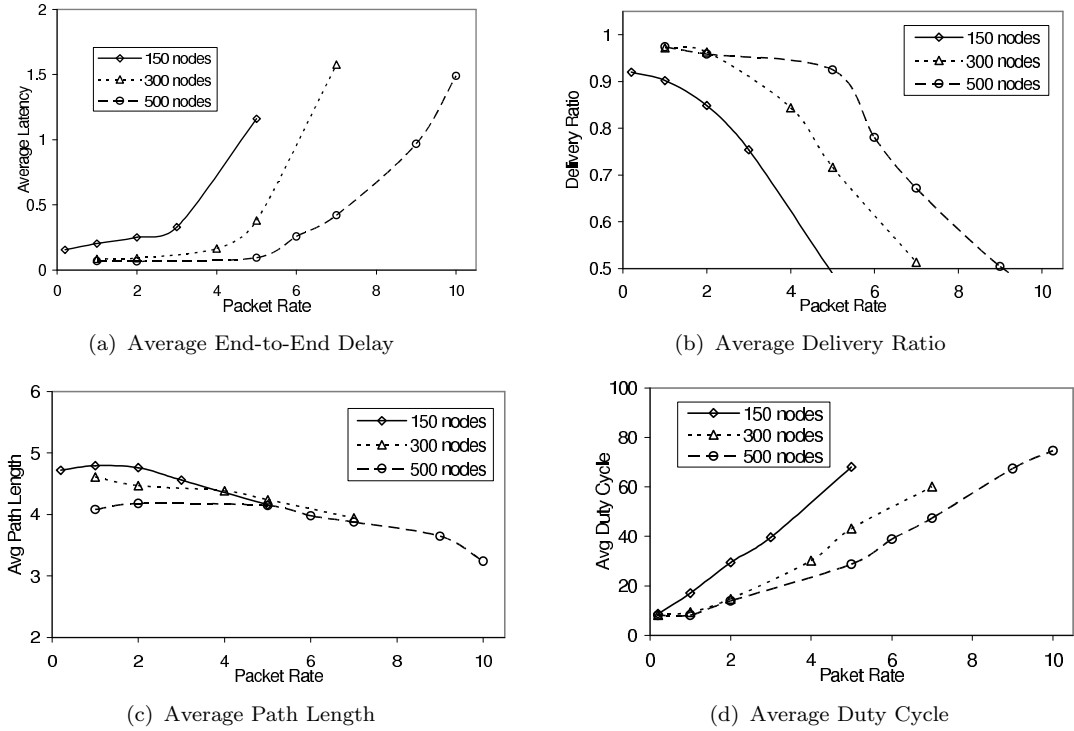


Figure 4.7. Performance of AREM with varying loads

loads queuing and back off delays play a major role. Similarly, at low transmission ranges, the probability that collision occurs is less. Hence, major component is sleep delay. At higher transmission ranges, back off delay might play an important role. These observations are reflected by figure 4.8.

An important observation is regarding the minimum delay point. We observe as reasoned above, the transmission range resulting in least delay keeps decreasing towards zero as load increases. This is because, at lower ranges, the contention delay is lower hence resulting in lower service time.

Figure 4.9 validates the analytical results through simulation results for two load scenarios. At very low loads, both simulation and analytical results agree to great extent. This is because, at low delays, the major delay component is sleep delay which can be computed fairly easily and accurately. At higher delays, the analytical results give a tight upper bound. But, as transmission range increases, the values deviate to a larger extent. We attribute this to dropped packets, which the analytical model does not capture. Nevertheless, the analytical model forms

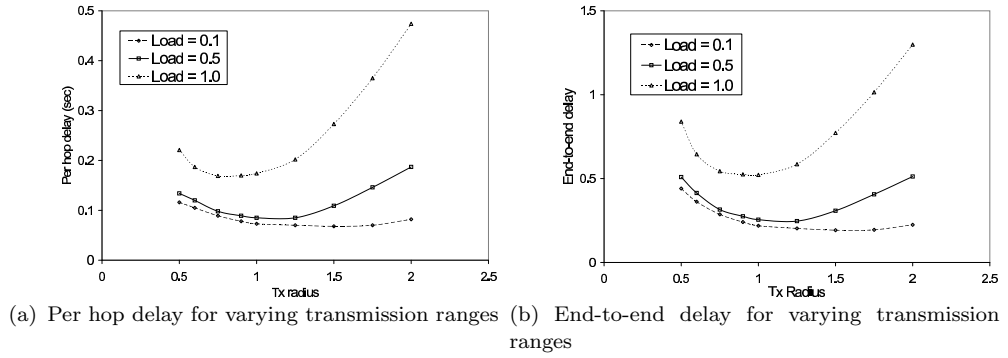


Figure 4.8. Analytical results for varying transmission ranges

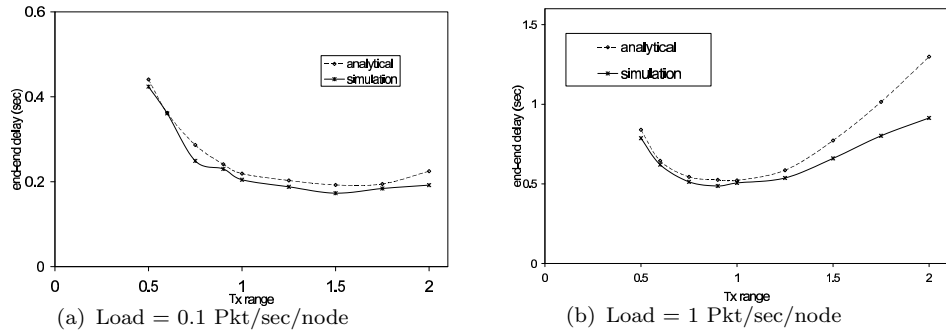


Figure 4.9. End-to-end load for varying ranges: analytical and experimental

a tight upper bound especially at lower loads. We would like to note that this is the case (low to moderate loads) of typical operation of wireless networks, so as to limit the number of packets dropped.

4.5.4 Performance Study of AREM with Range Adaptation

The performance of AREM is shown in Figure 4.10. We consider the metrics: average end-to-end packet delay, delivery ratio (percentage of packets delivered to the destination) and the duty cycle (percentage of duration, nodes are active on an average). We study both uniform and non-uniform transmission ranges and compare the performance with non-adaptive versions of the protocol with transmission

radii, $R = 0.75$ and $R = 1.0$. Different load scenarios are considered. In particular, we consider three scenarios:

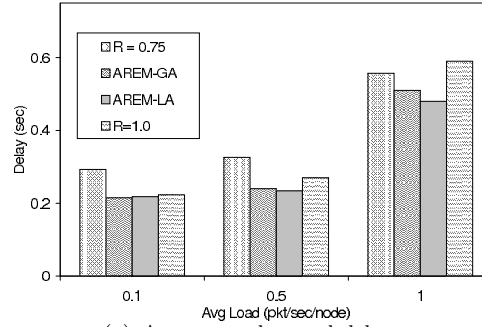
- Avg load = 0.1: This case is study the performance in low load conditions. Nodes are randomly assigned a packet transmission rate based on a uniform distribution from 0 to 0.2 packets/second.
- Avg load = 0.5: This case is study the performance in moderate load conditions. Nodes are randomly assigned a packet transmission rate based on a uniform distribution between 0 and 1 packets/second.
- Avg load = 1: This case is study the performance in heavy load conditions. Nodes are randomly assigned a packet transmission rate based on a uniform distribution between 0 and 2 packets/second.

Figure 4.10(a) shows the performance in terms of delay for all three load scenarios. Under low load conditions, the performance of all mechanisms except $R = 0.75$ is very similar. The reason as seen from analytical results, the optimal performance is close to the performance at $R = 1.0$. At $R = 0.75$, packets need to be transmitted more times, leading to an additional delay. As load increases, the delay with AREM-LA is the lowest, while AREM-GA yields more than 15% reduction in end-to-end delay compared to non-adaptive scenario, $R = 1.0$. AREM-LA performs slightly better than AREM-LA, because as each node is able to locally optimize the delay while in AREM-LA all nodes are assigned same transmission range. Nevertheless, the improvement is around 5%. An interesting observation regarding non-adaptive versions is that at high loads, $R = 0.75$ performs better than $R = 1.0$. This shows that an optimal choice in one scenario might not be optimal in another scenario.

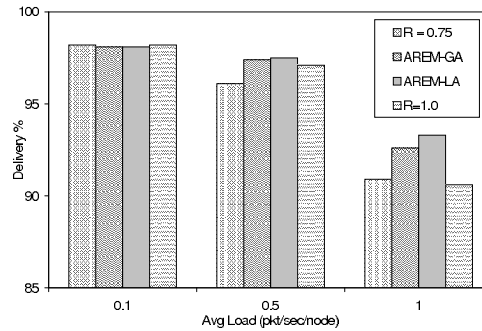
The performance in terms of delivery ratio is presented in Figure 4.10(b). Again, at low load scenarios, the performance of all scenarios is similar. But, as load increases, AREM-GA and AREM-LA perform better. We can reason this through the fact that, as delay decreases, throughput and hence the capacity increases, thus yielding a higher delivery ratio.

Finally, figure 4.10(c) presents the average duty cycle in all three scenarios. At high load, AREM-GA reduces the duty cycle by more than 10% while AREM-LA

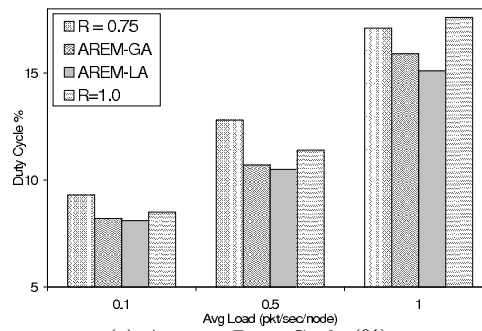
performs better - a reduction by around 16%. Again, the improvement is because adaptation results in decrease in transmission delays, and hence nodes are able to sleep longer. Also, a reduction of duty cycle by 10% will result in an increase of network longevity by around 10%. Thus, AREM-GA and AREM-LA are able to increase network lifetime up to 10% and 15%, respectively.



(a) Average end-to-end delay



(b) Delivery Ratio (%)



(c) Average Duty Cycle (%)

Figure 4.10. Performance comparison of adaptive and non-adaptive mechanisms

While Figure 4.10 considers scenarios where all nodes are generating packets, we also consider scenarios where there are only a few connections in the network.

In particular, Figure 4.11 shows the performance of AREM in a network with six randomly selected Source-Destination pairs. All sources generate packets at a given rate, and the results are shown for different packet rates.

Figure 4.11(a) shows the average end-to-end delay for three different packet rates. The performance improvement with AREM-LA is more apparent in this scenario: around 30% when compared to $R=1.0$ and over 20% when compared to $R=0.75$ at a packet rate of 15 pkt/sec. Figure 4.11(b) shows the average duty cycle. Again compared to $R=1.0$ and $R=0.75$ schemes, AREM-LA reduces the duty cycle by around 31% and 23%, respectively.

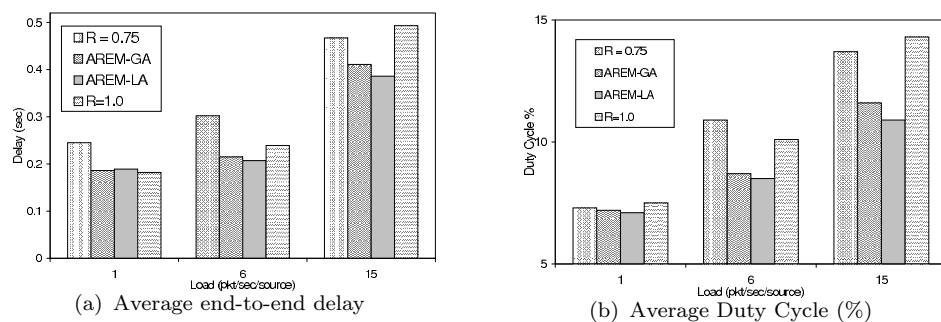


Figure 4.11. Performance comparison of adaptive and non-adaptive mechanisms with six source-destination pairs

4.5.5 Performance in Presence of Heterogeneous Energy Levels

Power consumption in the model is based on the amount of the current draw that Crossbow MICA2 node's radio transceiver uses. The typical current draw values are: $2\mu\text{A}$ while sleeping, 8mA while receiving, 15mA when transmitting and 7mA when idle (not transmitting nor receiving)[75]. The initial energy of each node is randomly set to a value between 5 and 20 power units.

We primarily focus on the reduction of the energy disparity among the nodes. We use variance of the energy levels of all the nodes as the primary measure of dispersion. A high variance indicates higher energy consumption at some of the nodes compared to others. Figure 4.12(a) presents the variance of energy levels as a function of time for HWSNs for 150 and 250 nodes. As it can be observed, the variance in energy aware wake up scheme (AREM-E), the variance keeps decreasing. In the case when energy is not considered, the variance slightly increases. This can be attributed to the fact that each node is active for equal durations as

other nodes and some nodes might be generating/transmitting data packets thus expending slightly more energy than nodes that are not.

We also study the performance of AREM-E in improving the network lifetime of the network. For this we measure the number of nodes whose energy level falls below a threshold as a function of time. Fig. 4.12(b) presents the results. As it can be noticed, the rate at which number of nodes that fall below the threshold energy level is considerably less with AREM-E. We considered a threshold of 5 power units.

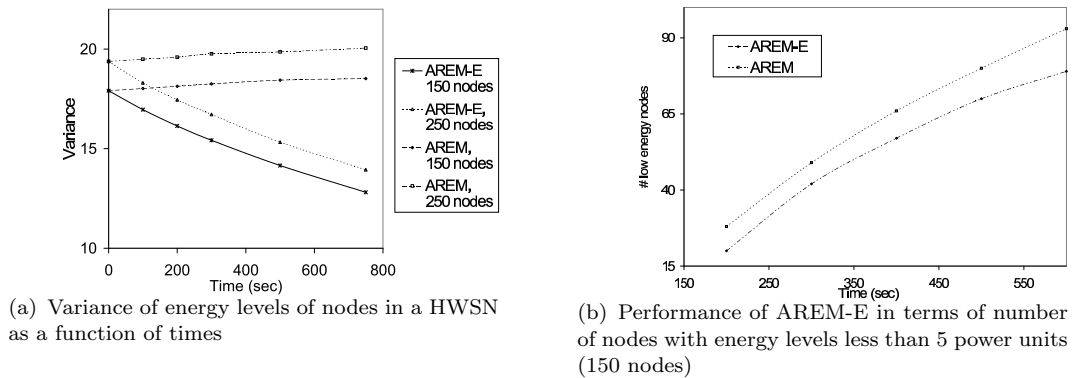


Figure 4.12. Performance comparison of energy adaptive and non-adaptive mechanisms

4.6 Summary

Protocols and applications for future wireless networks must be designed for efficient operation under the resource constraints and behavioral dynamics of wireless networks. A critical challenge is that the protocols need to adapt to the ever changing wireless environment including the traffic loads, energy levels and node failures for efficient performance and prolonging network lifetime. This chapter proposes adaptive protocols to efficiently route and manage the energy of the nodes to reduce delay and prolong the network lifetime.

We build an analytical model to have nodes in a mostly sleeping network transmitting optimal transmission range and also validate it through extensive simulations. Using the analytical model as a basis, we present two adaptive power control schemes that adapt the transmission power based on the dynamic network conditions. In the first scheme, all nodes are forced to use the same transmission

power and hence can be easily adopted in tandem with existing network protocols that assume common power usage. On the other hand, the second approach allows nodes to use independent transmission powers and operates in a purely distributed fashion. We also propose to distribute the load based on the residual energy of the nodes, thus prolonging both network and individual node lifetimes. We show that through energy management the lifetime could be extended upto 1300%. Our evaluation shows that adaptation leads to a further improvement of up to 15% both in end-to-end delay and duty cycles of the nodes.

Chapter 5

An Efficient Coordination Protocol for Heterogeneous Wireless Networks

Light is the task where many share the toil.

- *Homer* (9th-8th Century BC)

In wireless networks a channel is usually shared among many hosts. The sharing increases the complexity of route discovery, reduces the network performance, and increases energy consumption due to aggravated radio interference. Topology control is a technique used in wireless networks to address these problems. Topology control optimizes network topology and reduces routing cost by restricting the connections among pairs of hosts. If we view a wireless network as a graph $G = (V, E)$, where V is the host set and E is the set of links between hosts, the initial graph is heavily connected. Topology control removes unnecessary links from the initial graph and derives a connected sub-graph with fewer links, which enables efficient routing. Another important application of topology control is to save energy in wireless networks. Because nodes usually rely on power supplies of limited capacity, such as batteries, energy conservation is critical to the operational lifetime of wireless networks.

One approach of topology control is to exploit the node redundancy in wireless networks. Wireless networks have high level of node redundancy because of the high density. Each node can reach a number of neighboring nodes. Therefore a subset of nodes can be selected to serve as the coordinators through which all nodes can, directly or indirectly, communicate with each other.

The coordinators form the backbone of the network. The nodes that are not in the backbone have at least one neighboring node that is in the backbone, i.e. all the nodes in the network are connected through the backbone. The non-backbone nodes that do not have active communication can safely go to sleep to save energy. The duration of sleep time depends on how long the backbone can be maintained - which is usually dozens of seconds.

It is desirable to form a small backbone to save more energy. The problem of constructing a minimum backbone is equivalent to finding the *Minimum Connected Dominating Set* of a graph. This problem has been proven to be NP-complete even when the complete network topology is available. In wireless networks, node unreliability and high cost of transferring information across the whole network make it impractical to use a centralized backbone algorithm. Thus, many distributed algorithms have been proposed.

Several works have been proposed to form the backbone by non-deterministic negotiations in *wireless ad hoc networks*. In these approaches nodes decide to join or quit backbone mostly based on their observation of the nearby topology change.

Several characteristics of a good power-saving coordination technique for a *wireless ad hoc network* have been identified and are as follows: It should allow as many nodes as possible to turn their radio receivers off most of the time. On the other hand, it should forward packets between any source and destination with minimally more delay than if all nodes were awake. This implies that enough nodes must stay awake to form a connected backbone. Furthermore, the backbone formed by the active nodes should provide about as much total capacity as the original network, since otherwise congestion may increase. Also it is desired that the coordination technique does not make many assumptions about the link layer's facilities for sleeping and it inter-operates correctly with whatever routing system the wireless network uses.

For wireless networks, in addition to the above, we identify the following characteristics:

Energy balancing: Nodes in the connected dominating set consume more energy to handle various bypass traffic than nodes outside the set. Therefore, *static* selection of dominating nodes will result in a shorter life span for certain nodes, which in turn results in a shorter life span of the whole network. To prolong the life span of each node and, hence, the network by balancing the energy consumption in the system, nodes should be alternately chosen to form a connected dominating set. We strongly feel that the protocol needs to explicitly provide balancing of energy among nodes and choose nodes with higher energy levels as backbone nodes with higher probability.

We present Efficient Coordination Protocol (ECP) that achieves all the above objectives. We propose to save overall energy consumption by selecting a few nodes that form a connected dominating backbone and keeping them awake. Selection of dominating backbone network is based on the *extended Covering Problem* [76]. We allow only backbone nodes to participate in routing. In case of a broadcast message, only the backbone nodes retransmit the broadcast packet, to reduce broadcast redundancy. In addition, to maximize the lifespan of all nodes, ECP periodically selects nearly disjoint subset of nodes to form connected dominating sets.

The effectiveness of ECP is studied through simulations. Our simulation results also show that system lifetime with ECP is significantly better than without ECP, for a range of node densities, without much reduction in overall forwarding capacity.

Our contributions can be summarized as follows:

- We propose ECP, a lightweight protocol for distributed formation of a backbone network. ECP does not require any neighborhood information and consequently scales well with the number of nodes and network size.
- We present an analytical framework to study the performance of ECP. The factors studied include size of the backbone network, average path length. We also present a queuing model to compute an upper bound on the end-to-end delay.
- By making the nodes rotate the role of backbone nodes, we extend the life of the network. To the best of our knowledge, this is first such work that comprehensively studies the effectiveness of such mechanism.

The rest of this chapter is organized as follows: Section 5.1 reviews the related work. Section 5.2 presents a formal definition of the problem and reviews the *extended Covering Problem*. It also states the network assumptions made by ECP. Section 5.3 describes our protocol ECP. Section 5.5 presents our analytical model and section 5.6 presents the simulation results.

5.1 Related Work

Routing based on a connected dominating set is a promising approach and is a well studied aspect for wireless ad hoc networks. A wide range of heuristic algorithms have been proposed to construct a Minimum Connected Dominating Set

(MCDS) [77, 78, 79, 80, 81, 82] for a graph G . Such algorithms require typically global knowledge of the layout of the network graph G . Optimal solutions to compute Minimum Connected Domination Set (MCDS) were obtained for the case when each node knows the topology of the entire network (centralized broadcast). These solutions are deterministic and guarantee a bounded delay on message delivery, but the requirement that each node must know the entire network topology is a strong condition, impractical to maintain in wireless networks.

Several works have been proposed to form the backbone by non-deterministic negotiations in wireless ad hoc networks. In SPAN [83], a node joins the backbone if it has two neighbors that are not connected either directly or through a third node. GAF [84] constructs the backbone based upon the geographic location of nodes. It divides space into grids of equal size and elects one coordinator in each grid. The size of grid is chosen in such a way that a node any where in one grid can reach another node any where in an adjacent grid.

The PAMAS power-saving medium access protocol [64] turns off a node's radio when it is overhearing a packet not addressed to it. This approach is suitable for radios in which processing a received packet is expensive compared to listening to an idle radio medium. Kravets and Krishnan [85] present a system in which nodes wake up periodically and poll a base station for newly arrived packets.

Selective Backbone Construction (SBC) [86] starts by electing a small number of seed nodes in the backbone and then completes its construction by making a sweep of the network spreading outwards from the seed nodes. Topology information is transferred to allow better coordinator selection decisions. But, this protocol incurs high overhead because of the necessity of two-hop neighborhood information.

In Mobile Backbone Network Topology Synthesis Algorithm (MBN-TSA) [87], every backbone capable node examines its conversion criteria periodically and independently, using 2-hop Backbone Node neighborhood information and 1-hop overall neighborhood information, makes its own decision.

To the best of our knowledge, though some protocols propose to balance the energy consumption that occurs because of backbone network by periodically rotating the role of backbone node among all the nodes, none of them comprehensively study the effectiveness of such rotation.

5.2 Problem Statement and Background

5.2.1 Problem Statement

We study the problem of designing an efficient and distributed algorithm that partitions the nodes in a wireless network into k covers such that each cover forms a connected dominating set, thus yielding a virtual backbone. The problem of choosing which cover a node will belong to is abstracted into *SET-K-CDS* problem and can be stated as follows:

Problem SET-K-CDS

Given: A graph $G(V, E)$

To find: A partition ζ of the graph into k subsets $S_1, S_2 \dots S_k$, where each subset is a dominating set.

Criteria:

Maximize $|S|$

Minimize = $\sum_{\substack{\forall i, j \\ i \neq j}} |DS_i \cap DS_j|$.

Informally, we are maximizing the number of dominating sets while making the dominating sets themselves as independent as possible. Ideally, there should be no overlapping between any two dominating sets ($= 0$). This implies that each node belongs to one and only one dominating set. Also, it is desired that all nodes are part of some dominating set, as this results in perfect load balancing. Additional criteria can be added. Once the partition is obtained, in an effort to increase the longevity of the network and conserve battery power, it would be beneficial to activate groups of nodes in rounds, so that the battery life of a node is prolonged and the same time connectivity is maintained. Rather than using all the nodes all the time to forward packets and maintain connectivity for events, SET-K-CDS solutions provide a simple way for nodes to share in the tasks, so that their energy resources can be conserved.

Computing an optimal partition ζ meeting the above criteria is NP-hard. Also, achieving the complete independence among dominating sets might be too strict and impractical. In this chapter, we present ECP that to a great extent balances the load among the nodes. ECP also considers the residual energy levels of nodes.

5.2.2 Background

The Covering Problem was stated as follows:

”What is the minimum number of circles required to completely cover a given 2-dimensional space.”

We stated a modified version of *The Covering Problem* that finds its application in wireless networks. The objective of *minimal Backbone Network formation* is to minimize the number of nodes in the backbone network and can be stated as:

”What is the minimum number of nodes transmission range R required to entirely cover a 2-dimensional space.”

If the range of a node is considered to be R , then the reason behind the condition that the center of a circle should lay on the center of another circle is that a node has to receive a message for it to retransmit the message. A possible solution for the *Modified-Covering Problem* is shown in Figure 3.2 and reproduced again here with slight modifications Figure 5.1). As done for covering problem, initially the whole region is covered with regular hexagons whose each side is R . Then, with each of the vertices as a center, circles of radius R are drawn.

The following properties of the vertices in Figure 5.1 should be noted:

Property-1: Each vertex v is joined to three other vertices.

Property-2: The lines joining these three vertices to vertex v make an angle of 120° ($2\pi/3$ radians) with each other.

Property-3: Each vertex is at a distance of R from each of its neighboring vertices.

To compute the number of BNs, N_{BN} required in this case, it should be observed that ideally a BN is located at each vertex and each vertex of a hexagon belongs to two other hexagons. Thus, for large networks, total number of BNs can be approximated as

$$N_{BN} \approx \frac{2 * A}{3\sqrt{3}R^2/2}$$

5.3 Efficient Coordination Protocol

ECP adaptively elects Backbone Nodes (BNs) from all nodes in the network. BNs stay awake continuously and perform multi-hop packet routing within the

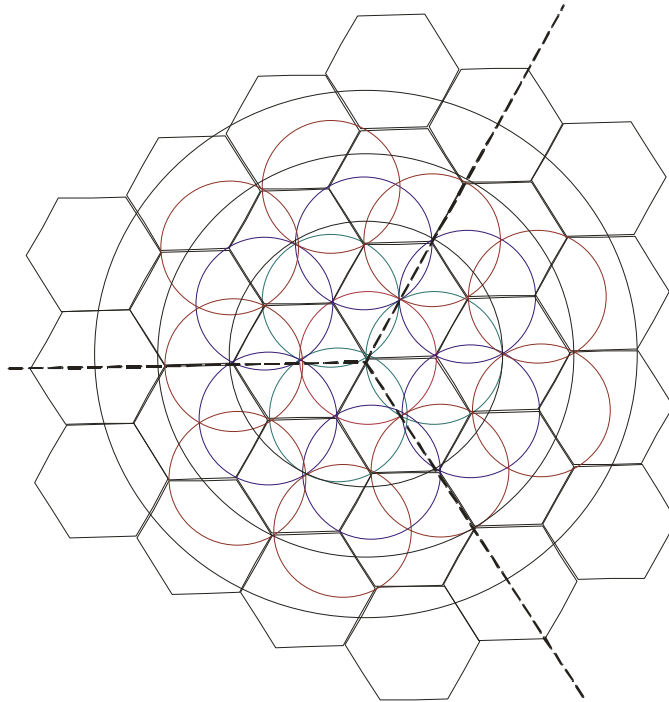


Figure 5.1. Covering a plane with circles in an efficient way

node network, while other nodes remain in power saving mode and periodically check if they should wake up and become a BN.

Our protocol achieves following objectives: First, it ensures that enough BNs are elected so that every node is the radio range of at least one BN. Second, it rotates the BNs to ensure that all nodes share the task of providing global connectivity roughly equally. Third, it attempts to minimize the number of nodes elected as BNs, thereby decreasing network energy consumption and thus increasing network lifetime, but without suffering a significant loss of capacity or an increase in latency. Finally, by dynamically having more number of nodes alive when needed, it keeps the delay low while still achieving high throughput.

We assume that each node knows its location which itself is a requirement for various routing protocols, sensing, target tracking and other applications. Various techniques like GPS [88], Time Difference of Arrival [89], Angle of Arrival [90] and Received Signal Strength Indicator [91] have been proposed to enable a node to discern its relative location. We also assume that the nodes are loosely synchronized.

Our protocol runs above the link and MAC layers and interacts with the routing protocol. ECP leverages a feature of modern power-saving MAC layers, in which if a node has been asleep a while, packets destined for it are buffered at the forwarding BN. When the node awakens, it can retrieve these packets from the buffering BN. ECP also requires a modification to the route look up process at each node - at any times, only those entries in a node's routing table that correspond to currently active BNs can be used as valid next hops (unless the next hop is the destination itself). We initially present the coordination protocol and later in the section present the enhancement that enables load adaptation.

5.4 The Protocol

In ECP, a node switches its state from time to time between BN and RN. Here we describe how a node decides it should be a BN.

Periodically, base station initiates the backbone reconfiguration procedure. The solution is based on the extended covering problem. The objective is to select nodes in the network that would form the best approximate for a hexagonal lattice structure to cover the whole area. The algorithm is as follows:

The Initiator constructs a $BN_{formation}$ message with two location fields L1 and L2 in the header. Whenever a node transmits a broadcast message, it sets L1 to the location of the node from which it received the message and sets L2 to its own location.

The protocol is as follows:

The Initiator S sets both L1 and L2 to its location (S_X, S_Y) and transmits the message.

1. A node M, upon receiving a $BN_{formation}$ message, first determines if the message can be discarded. A message can be discarded under any of the following conditions:
 - If the node has transmitted the message earlier.
 - If there is a Backbone Node (BN) closer to it by a distance less than Th , where Th is a threshold and the is further discussed later in this section.

2. If the message isn't discarded, M determines if it received the message directly from the initiator S. Else, if M hasn't received the message directly from the source S, but from some other node K, then using properties 1, 2 and 3 mentioned in section 3 and with the nearest strategic location. The message transmission is delayed by $d = l/R$.
3. After delay d , M again determines if it has received the same message again and if the message can be discarded (for the same reasons mentioned above). Thus, delaying enables a node to decide if it is the nearest node to the strategic location.
4. If the message cannot be discarded, M advertises itself as a Backbone Node, updates L1 to location of the node from which it received the message and L2 to its own location and transmits the message.

ECP resolves contention by delaying BN announcements with a back-off delay. Each node chooses a delay value, delays the $BN_{announcement}$ message that announces the node becoming a BN.

5.4.1 Selection of Th

The purpose of having Threshold is to prevent two nodes that are very close to each other to become backbone nodes. The key factors affecting Th are number of transmissions and reachability.

- **Number of transmissions**

As Th increases, the size of the backbone network decreases. This is because, at high Th values, the minimum distance between any two transmitting nodes is more. This in turn implies that additional area covered is higher and hence number of transmissions needed for covering the entire network is lesser.

- **Reachability**

It is the percentage of nodes that are either backbone nodes or neighbor of a backbone node. Higher the size of the backbone network, higher would be reachability. So, for higher reachability, lower Th is preferred.

As shown in 3.9, we think that the value of $0.35 \cdot R$ is the best choice. Higher values result in decrease in reachability. Values lower than this resulted in larger backbone networks, but only incremental improvement in terms of reachability.

5.4.2 Energy Balancing through Rotation

In this section, we present the details on how we rotate the role of backbone node. For this, we observe that, just by shifting the location of the initial vertex, the whole hexagonal lattice shifts/rotates accordingly.

Periodically the base station initiates a new *backbone_formation* message. It also, randomly selects one of its neighbors to initiate the backbone formation. The *chosen* neighbor, chooses a random orientation of the hexagonal lattice, and initiates the backbone formation protocol.

This simple procedure ensures distribution of energy to a great extent as verified by our simulation results. The interval between two backbone formations has to be chosen so as to obtain a good trade off between overhead and degree of uniformity required. If the interval is too small, the overhead might become significant. If the interval is too large, BNs might loose too much of energy. Typically, the interval could vary between few minutes to several hours depending on the network.

5.4.3 Load Adaptive Backbone Formation (ECP-A)

In a heterogeneous wireless network, nodes with different capabilities would co-exist. Thus it would be desirable to elect nodes with higher residual energy levels as BNs to prolong network lifetime as well as individual node life time and balance energy among the nodes. Also, the load might not be evenly distributed across the entire network. In such cases, load on some BNs might be higher than other nodes, causing excessive delay and packet drops. It is desired the load on the BNs is evenly distributed. In such scenarios, a BN can estimate the average packet transmission rate in its neighborhood and thus reduce/increase its transmission range to have a desired load. In this section, we propose ECP-Adaptive(ECP-A) that adapts to the network conditions.

However, if a node uses a transmission range that is lesser than the transmission range of the backbone node that it needs to communicate with, it may not be able to establish *direct* communication with the BN. Hence, once the backbone is established and BNs have been elected, each node uses the same transmission range as its cluster head.

Each node could estimate the *approximate* network load in its neighborhood based on the observed channel idle time. Thus, each node could periodically com-

pute the approximate load in its neighborhood and compute optimal transmission range. We use the analytical model developed in Section 2.2 for adaptation and computing the optimal transmission range. Then, each node would delay the transmission of broadcast message based on the additional area it would cover. For instance, consider two nodes separated by distance \bar{d} and with transmission ranges R and r respectively. Then the additional area covered can be computed as follows:

The area of intersection is given as

$$A_{int} = r^2 \cos^{-1} \left(\frac{\bar{d}^2 + r^2 - R^2}{2\bar{d}r} \right) + R^2 \cos^{-1} \left(\frac{\bar{d}^2 + R^2 - r^2}{2\bar{d}R} \right) - \frac{1}{2} \sqrt{(r + R - \bar{d})(-r + R + \bar{d})(r - R + \bar{d})(r + R + \bar{d})} \quad (5.1)$$

Thus, the additional area covered by a node with transmission range R and located at a distance $d = 2R\bar{d}$ from the strategic location can be obtained as

$$A_{additional} = \pi R^2 - A_{int} \quad (5.2)$$

Thus, the delay function at each node is computed as

$$delay_k = \frac{c}{A_{additional}(k)} \quad (5.3)$$

Thus a node with least delay would elect itself as a backbone node and broadcasts a $BN_{announcement}$. Whenever a node broadcasts a $BN_{announcement}$ message, it includes its transmission range in the message so that all nodes that decide to communicate with it use the same transmission range.

- **Energy balancing**

Heterogeneous Wireless Networks are envisioned to comprise of nodes with different capabilities leading to different energy levels of nodes. Even in Homogenous Wireless Networks, where all nodes have same energy levels during the bootstrapping stage, because of different roles/tasks each node would be performing, node energy levels vary from one another.

To simultaneously prolong the network lifetime as well as each nodes lifetime, it is required that nodes with higher energy levels are elected as backbone nodes more frequently than nodes with lower energy levels. We propose to acheive this by setting the delay d as follows:

$$delay_k = c \left(\frac{1}{A_{additional}} + \frac{Avg_energy_i}{Energy_i} \right) \quad (5.4)$$

where,

$A_{additional}$ is as explained in the previous section.

d_t is the distance to the nearest vertex.

$Energy_i$ is the energy level of node i .

Avg_energy_i is the average energy level of the neighbors of node i .

The intuition behind this is as follows: The lower the energy level of a node, the lesser it should participate in broadcasting. Thus, by having an energy component in the delay, a node with low energy will delay retransmitting a broadcast for a longer duration than a node with higher energy levels. Thus, the probability of a node with lower energy levels becoming a BN is lesser. We note that, though this mechanism requires energy levels of the neighbors, the information need not be accurate as we observe that the changes in the energy levels is not drastic.

5.5 Analysis of ECP

In this section, we present the analysis of ECP. We first present some observations on the structure of the backbone network, use these observations to obtain the average delay at a BN. Later, we also present the time taken for backbone formation.

5.5.1 Backbone Structure

Let B be the base station located at the center of the network that initiates the backbone formation. Let ξ_i be the number of BNs at a hop distance of i from the base station. Then,

$$\xi_i = 3 * i; i = 1, 2, 3, \dots \quad (5.5)$$

It should be observed that rotating figure 5.1 about the base station by 120° results in the same figure. Consider just one 120° sector in figure 5.1. A BN on the edge of two sectors is considered to belong to the left sector. The number of nodes at a hop distance of i from the base station in a sector is i . Thus, the average branching factor at level i is given as $(i+1)/i$. From now on we treat all BNs that are equidistant from the base station to be at the *same level*.

Consider all the BNs to form a tree rooted at the base station. The tree is formed such that a BN at level i has at least one child at level $i+1$. Also a BN can have at most *two* children. For this, it can be seen that the structure is symmetrical about 120° . Consider one of the 120° sectors as shown in figure 5.1. Since at each level, there are at most three more BNs than in the previous level, it follows that within a sector, at any level at most one more BN than the previous level can be present (because of symmetry). Thus, if all BNs at level i have at least one child at level $i+1$, then only one of BNs at level i can have two children.

Now, we proceed to calculate the number of levels (i.e., maximum hop distance) present in the tree in a regular network. This is equivalent to the maximum number of hops to a BN from the base station. For this, note that the number of BNs at each level follows the following series: 1, 3, 6, 9, 12 ...

Let μ be the number of levels. Then, total number of BNs at all levels can be expressed as

$$\begin{aligned} N_{BN} &= 1 + 3 \left(\frac{\mu(\mu + 1)}{2} \right) \\ \Rightarrow \mu^2 + \mu - \frac{2}{3}(N_{BN} - 1) &= 0 \end{aligned} \quad (5.6)$$

Solving for μ by assuming $N_{BN} \gg 1$, we have

$$\mu = \left\lceil 0.816\sqrt{N_{BN}} - 0.5 \right\rceil \quad (5.7)$$

Thus, one can compute . Consider a BN_i at level $\epsilon < .$ Then, the average number of BNs in the subtree rooted at BN_i can be expressed as:

$$\begin{aligned} \Phi_{BN_i} &= \sum_{j=\epsilon}^{\mu} \text{Number of BNs at level } j \\ &= \sum_{j=\epsilon}^{\mu} \prod_{k=\epsilon}^{\mu-j} \sigma_k \\ &= \sum_{j=\epsilon}^{\mu} \frac{j}{\epsilon} \\ \Rightarrow \Phi_{BN_i} &= \frac{(\mu + \epsilon)(\mu - \epsilon + 1)}{2\epsilon} \end{aligned} \quad (5.8)$$

where, σ_k is the average branching factor at level k and is given as $\frac{k+1}{k}$.

5.5.2 Arrival Rate

Consider the following simple scenario: A node wants to deliver a packet to the base station. We are given a set of nodes, $S = \{s_1, s_2, \dots, s_n\}$, in a two dimensional area A . Nodes are randomly placed in the region according to a Poisson process with density ρ nodes per unit area. This model is adequate in situations in which nodes are randomly deployed and is also appropriate for a first evaluation of the performance of our scheme. Each node senses its environment and thus is exposed to an input function γ corresponding to its sensed input traffic. Without loss of generality, we assume each node has a unit transmission range and network area is much larger than a nodes transmission region i.e., $A \gg 1$.

It should be noted that each node would be generating packets at an average rate of γ that are being forwarded by the backbone network to the base station. As there are N nodes in the network and N_{BN} backbone nodes, we assume that on an average each backbone node receives packets from (N/N_{BN}) nodes. Also, each BN would be receiving packets from BNs that are its children except in the case when the BN belongs to level μ .

Now we derive the expression for the average number of nodes forwarding sensed traffic to BN_i at level i . For this, note that BN_i has Φ_{BN_i} BNs in its subtree. Thus, the average arrival rate of packets at BN_i can be expressed as:

$$\begin{aligned} \gamma_{BN_i} &= \gamma \left(\frac{N}{N_{BN}} \right) \cdot \Phi_{BN_i} \\ \Rightarrow \gamma_{BN_i} &= \gamma \left(\frac{N}{N_{BN}} \right) \cdot \left(\frac{(\mu + \varepsilon)(\mu - \varepsilon + 1)}{2\varepsilon} \right) \end{aligned} \quad (5.9)$$

5.5.3 Average Number of Hops

Now, we would like find the average number of hops h_k which are necessary in order to reach a given node k at a distance d_k . Let, \bar{d}_k be the distance from the source to the nearest strategic location to node k . Now, if \bar{h}_k is the number of hops to this strategic location then

$$\bar{h}_k \leq h_k \leq \bar{h}_k + 1 \quad (5.10)$$

Thus, once \bar{h}_k is computed, it is straightforward to derive bounds on h_k . For computing \bar{h}_k , it should be observed every odd hop in the ideal scenario results

in a progress of $\sqrt{3}/2$, while every even hop results in a progress of 1. In a practical scenario, the average progress with each odd hop is $(\sqrt{3}/2) * E[\zeta]$ and with each even hop is $E[\zeta]$. Thus, the average progress made per hop toward a given destination can be approximated to

$$\bar{d} = \left(1 + \frac{\sqrt{3}}{2}\right) * \frac{E[\zeta]}{2} \quad (5.11)$$

Thus, the number of hops to a strategic location can be expressed as

$$\bar{h}_k = \frac{\bar{d}_k}{\bar{d}} \quad (5.12)$$

Thus, a packet sent by node k has to traverse the BNs at levels 1, 2... \bar{h}_k to reach the base station. We now proceed to calculate the average delay at each BN at different levels.

5.5.4 End-to-End Delay

The average service time and queueing delays can be computed using equations 2.14 and 2.12. The end-to-end delay for a packet from node k can be derived by summing up the packet delays at BNs at levels 1, 2, ..., \bar{h}_k and thus can be expressed as

$$D_k = \sum_{j=1}^{\bar{h}_k} (W_{BN_j} + \bar{T}_{BN_j}) \quad (5.13)$$

5.5.5 Time Taken for Backbone Formation

Let ϵ be the distance of the nearest node to the strategic location selected. The probability distribution of ϵ can be calculated by finding the area of intersection of two circles with centers (S and S') at distance unity and with radii 1 and ϵ (see figure 5.2). More specifically the probability that the distance to the strategic location is at least ϵ is the probability that the area of intersection does not contain any nodes. If A_ϵ is the area, then we have

$$P[x \geq \epsilon] = e^{-\rho A_\epsilon} \quad (5.14)$$

The area under consideration is

$$A_\epsilon = 2 \int_0^\epsilon x \cos^{-1} \left(\frac{x}{2} \right) dx = 2 \left[\left(\frac{x^2}{2} - 1 \right) \cos^{-1} \left(\frac{x}{2} \right) - \frac{x}{2} \sqrt{4 - x^2} \right]_0^\epsilon$$

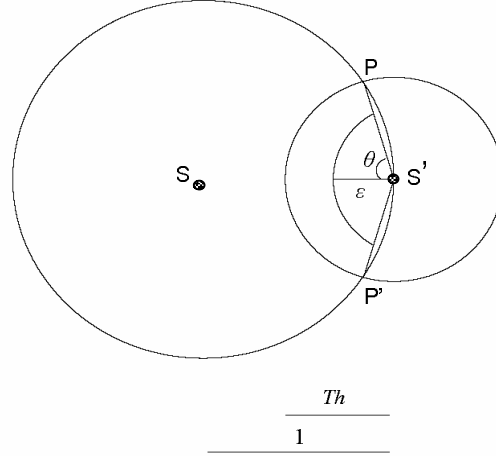


Figure 5.2. Circle intersection for the analysis

$$i.e., \quad A_\varepsilon = \pi + 2 \left(\frac{\varepsilon^2}{2} - 1 \right) \cos^{-1} \left(\frac{\varepsilon}{2} \right) - \frac{\varepsilon}{2} \sqrt{4 - \varepsilon^2} \quad (5.15)$$

We approximate the area of intersection to a sector $S'PP'$ as shown in figure 5.2. Now the area under consideration is

$$A_\varepsilon = \theta \varepsilon^2 \quad (5.16)$$

where, $\theta = \frac{1}{2} \cos^{-1} \left(\frac{Th}{2} \right)$

The distance from the strategic location is x with probability distribution

$$P[\zeta \geq \lambda] = \begin{cases} e^{-\rho A_\delta}, Th \leq \lambda \leq 1 \\ 0, \lambda < Th \\ 1, \lambda > 1 \end{cases} \quad (5.17)$$

Let

$$f_\zeta(\lambda) = f_\zeta^c(\lambda) + P[\zeta = Th] \delta(\lambda)$$

be the pdf of the advancement where $f_\zeta(\lambda)$ is the derivative of $P[\zeta \geq \lambda]$ in $\lambda \in (Th, 1)$. The average advancement is then found as

$$E[\zeta] = \int_{Th}^1 \lambda f_\zeta(\lambda) d\lambda = \int_{Th}^1 \lambda f_\zeta^c(\lambda) d\lambda \quad (5.18)$$

Notice that ζ depends on the density of the network. As density increases, ζ approaches unity. From Equation 5.7, we have the number of levels in the backbone tree in an ideal case. In practical scenario, the number of levels can be calculated

by approximating the armlength of the hexagon with ζ and can be expressed as

$$\bar{\mu} = \frac{\mu}{\zeta} = \left\lceil \frac{0.816\sqrt{N_{BN}} - 0.5}{\zeta} \right\rceil \quad (5.19)$$

Also, note that each node waits for duration of d before self-selecting as a BN. Duration d is proportional to the node's distance from the strategic location. Thus, average delay at each level can be expressed as

$$E[d] = c * (1 - E[\zeta]) \quad (5.20)$$

where, c is the proportionality constant. Assuming that transmission delay is negligible when compared to d , the total time taken for the formation of the backbone network can be expressed as

$$T_{BN_formation} = \bar{\mu}E[d] \quad (5.21)$$

Thus, for a given network, time taken for backbone formation scales as $O(\text{network diameter})$ and also reduces as node density increases.

5.6 Performance Evaluation

We use *ns-2* simulator to evaluate the performance of our protocol. We first present the results pertaining to the efficiency of our protocol in terms of the size of the backbone network and its scalability. In section 5.6, we validate our analytical model for calculating delay and in section 5.6, we present the efficiency in balancing the energy consumption.

- **Energy Efficiency**

Wireless networks of different physical areas with different number of nodes were simulated. To be more specific, square regions of size varying from $3 * 3$ to $10 * 10$ have been simulated, where the transmission range of each node is considered as one unit. The nodes were uniformly distributed all over the region with the density varying from 6.25 nodes per unit area to 25 nodes per unit area.

Initially, we study the size of the backbone network and study the performance of ECP with respect to the delay and energy savings. Then we present the performance in terms of balancing the energy.

Figure 5.3 shows the size of backbone network for different network sizes for different densities. The values obtained are an average over hundred Backbone

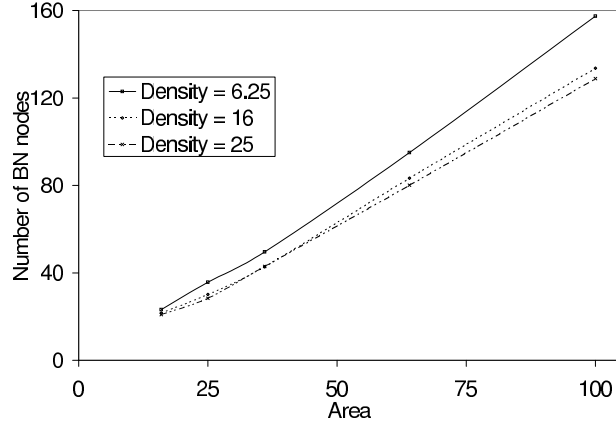


Figure 5.3. Number of Backbone Nodes for different network sizes and densities

network formations. The graph shows that BN size scales with density and is fairly constant for a given network area. The energy savings with ECP as compared to other protocols is shown in figure 5.4. At lower densities, both ECP and MBN-TSA consume similar energy amounts than SPAN. In case of GPSR [92], all nodes are awake for the entire duration of simulation. As density increases, the ECP does the best in minimizing energy consumption. This is because ECP is successful in choosing small backbone networks than others. The size of backbone networks for different protocols for varying densities is presented in figure 5.5. It can be seen that the size of backbone network in case of ECP decreases as the number of nodes (density) increases, while it increases for other protocols. In fact, as observed from the figure, ECP approaches the ideal case as density increases.

- **Delay Analysis**

For the specific purpose of validating the analytical model, we considered the following scenario. We consider an $4R * 4R$ network with the base station at the center. We positioned 11 nodes at the strategic locations to act as backbone nodes. We then place 50 nodes uniformly in the network. All nodes transmit to the base station according to the same CBR source rate with fixed packet sizes of 512 bytes. Each run corresponds to 30 minutes of data traffic. Regarding the physical layer, we use Direct-Sequence Spread Spectrum (DSSS) with a raw bit rate of 2Mbps. Table 5.1 summarizes the parameters used for our simulations.

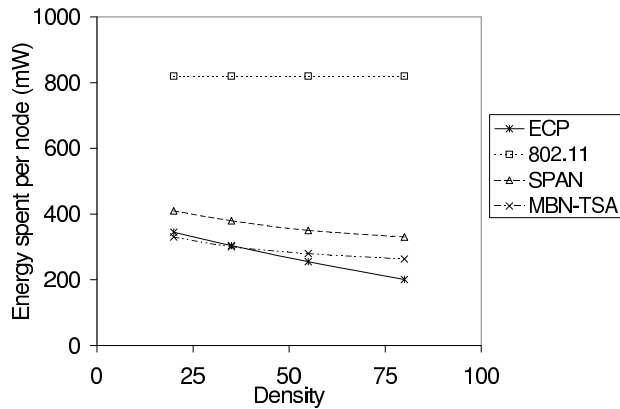


Figure 5.4. Energy consumption for different protocols

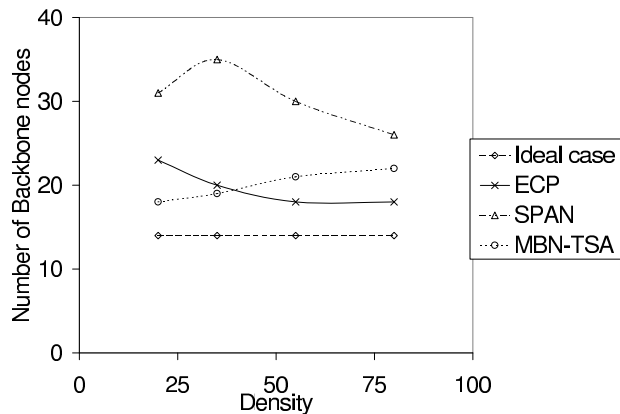


Figure 5.5. Comparison of backbone size networks for varying node densities

Figure 5.6 shows the numerical results for the average service time for both simulations and analytical models. As seen, our analytical model performs quite well in providing an upper bound on the average delay. Regarding the increasing discrepancy observed as the rate grows, we note two main reasons. First, in the analytical model [23], a packet can backoff infinitely in time, whereas in simulations (as in the standard) retry counters help the MAC determine when it is no longer worth it to continue attempting to transmit a packet. Therefore, only packets that were not discarded had their service time considered in the statistics. Secondly, to simplify the analysis, we took a conservative approach in calculating the number of active nodes in the neighborhood of a node.

TABLE 5.1. Physical layer parameters

Simulation parameters of 802.11	
W^*	64
MAC Header	34 bytes
ACK	38 bytes
CTS	38 bytes
RTS	44 bytes
Slot Time	20 μ sec
SIFS	10 μ sec
DIFS	50 μ sec
ACK Timeout	212 μ sec
CTS Timeout	348 μ sec
Raw Bit Rate	2 mbps
Packet Size	512 bytes

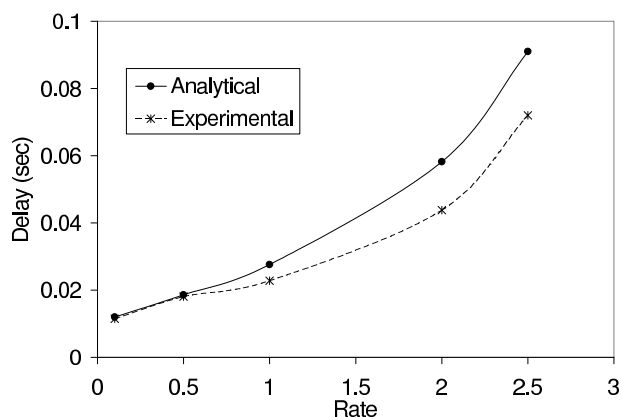


Figure 5.6. Average end-to-end delay, analytical and experimental

A comparison with GPSR (with all nodes awake) is presented in figure 5.7. It can be seen that, the delay with ECP is greater than delay with GPSR. This can be attributed to the fact that each packet might traverse longer hops with ECP when compared to the global shortest path in case of GPSR.

- **Energy Balancing**

The distribution of the number of times each node is elected as a BN for two different scenarios is shown in figure 9. Scenario-1 (figures 5.8(a) and 5.8(b)) is for a 8*8 network with 1024 nodes. The values are over a run of 500 BN formations. The average number of times a node is elected as BN is 39.38. It should be noted that the number of nodes within the range $(0.5 * \text{avg}, 1.5 * \text{avg})$ is 92.77%. Scenario-2

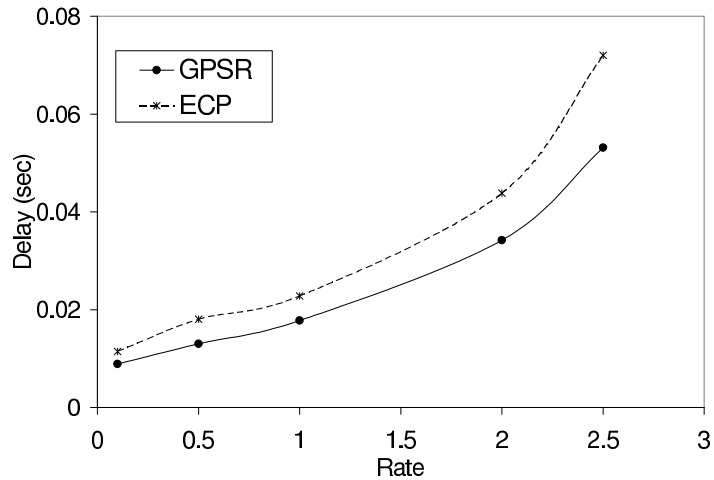


Figure 5.7. Average end-to-end delay, analytical and experimental

(figures 5.8(c) and 5.8(d)) is for a 10×10 network with 1600 nodes. The values are over a run of 500 BN formations. The average number of times a node is elected as BN is 41.25. It should be noted that the number of nodes within the range $(0.5 \cdot \text{avg}, 1.5 \cdot \text{avg})$ is 93%.

The energy distribution, as expected, has the same distribution as the number of times a node is elected as backbone node and hence not presented here due to space constraints. From the distribution, it should be observed that randomly rotating the role of backbone nodes significantly balances the energy among all the nodes. Also, we observed that most of the nodes that are elected fewer times as BNs were present at the network edges.

- **Adapting to the Network Conditions**

The focus of this section is to study how effectively ECP can adapt to the prevailing network conditions. Initially we present the size of the backbone for varying loads for networks with different densities. In each simulation run 20 nodes were randomly selected to transmit packets at a given rate. Data has been collected from 10 simulation runs, each simulation run being for 100 seconds. Backbone nodes are elected every 10 seconds. A network of 8×8 size is considered.

The average size of the backbone is presented in Figure 5.9. Initially, when load is very low, nodes are able to use high transmission ranges, thus resulting in smaller backbone. As load increases, the nodes adapt to a smaller transmission range thus

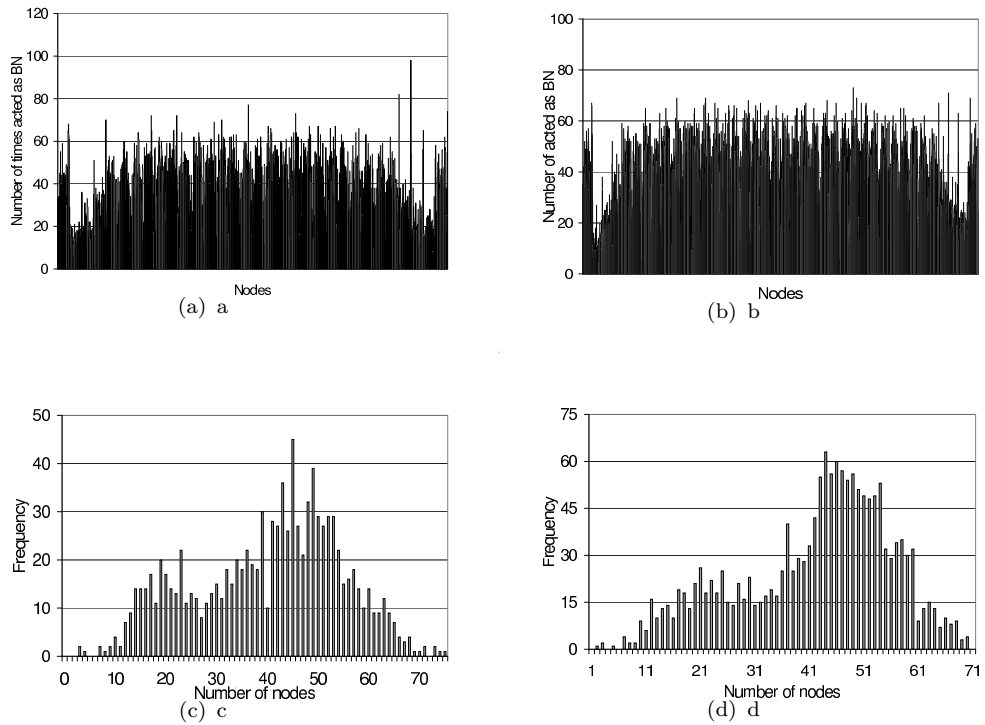


Figure 5.8. Distribution and histogram of the number of times each node is elected as BN (a), (b) 8*8 network (c), (d) 10*10 network

resulting an increase in the backbone size. Also, at higher densities, as observed earlier, nodes are closer to the strategic locations and hence the backbone size is smaller.

Next we consider the delay and energy consumption. The simulations are over a network area of $8 * 8$ with a density of 5 nodes per unit area. For comparison, we simulated ECP with the transmission range of all the nodes being unity. The average end-to-end latency for each packet is presented in 5.10(a). We observe that at low loads, with ECP-A there is improvement in delay by more than 50% when compared to ECP without adaptation. The reason is that at low loads, the transmission range of the backbone nodes is high and number of hops is smaller. As load increases, even though the transmission range is sometimes lesser than 1 resulting in more hops, the delay is lower with ECP-A. We attribute this to the lower queuing delays and lower contention delays in case of adaptation.

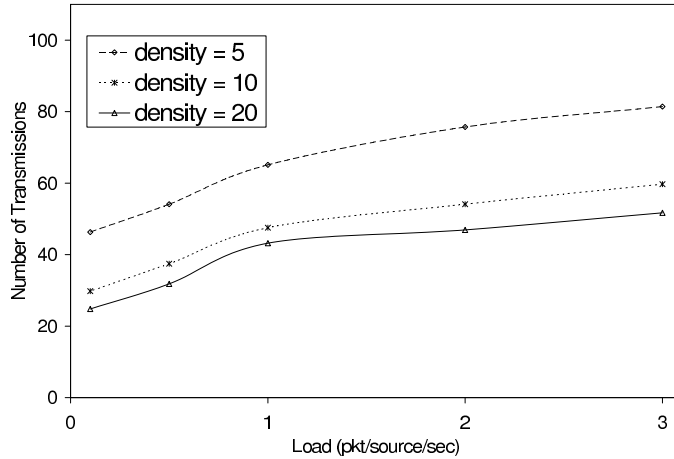


Figure 5.9. Average size of the backbone

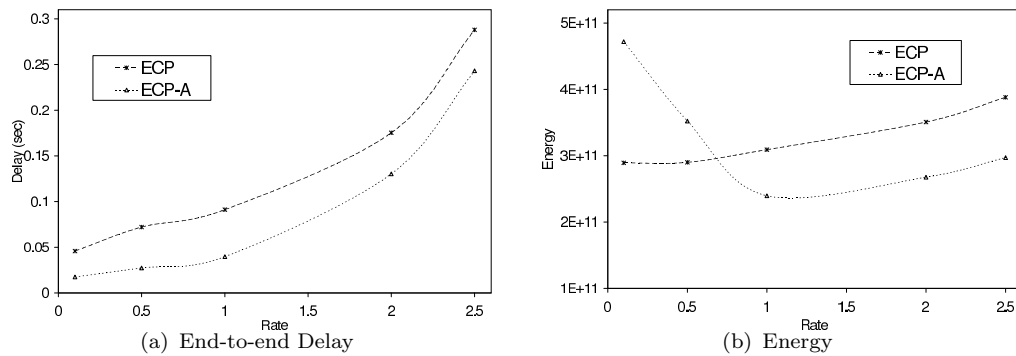


Figure 5.10. Adaptation of ECP to network conditions

Figure 5.10(b) presents the average energy consumption per backbone cycle for both ECP and ECP-A. We note that each backbone cycle is for a period of 10 seconds after which backbone nodes are re-elected. At lower loads, ECP-A results in a much higher energy consumption because of higher transmission ranges. However, we emphasize that this could be easily avoided by limiting the transmission range to a certain value, say 1.5. But, at moderate and high rates, the consumption is lesser. This is mainly because of reduction in range. We also observe that the energy consumption decreases initially and then increases in spite of decrease in transmission range. The reason is that the increased number of transmissions (packets) over compensate for the decrease in the transmission range.

The simulations are over a network area of $500 * 500$ with 100 and 250 nodes. We also introduce background traffic where nodes transmit data packets at a rate depending on *the term Average Load, L_{avg}* i.e., each node selects a data transmission rate randomly from the interval $(0, L_A)$ and transmits packets at this rate

throughout the simulation. Data has been collected from 10 simulation runs, each simulation run being for 100 seconds and consisting of 50 broadcasts.

5.7 Summary

This chapter presented ECP, an algorithm of constructing backbone in ad hoc wireless network for energy conservation. ECP employs a different procedure from other backbone construction algorithms. We use a geometric approach and extend the modified Covering problem. Our experiments with ECP show a superior capability of conserving energy in comparison existing protocols. ECP constructs backbones that are smaller, it results in energy savings that translate into extended network lifetimes, and at the same time ECP does not deteriorate network performance. We have validated these results through both analytical and simulation results. Also, through analytical modeling, ECP adapts to network conditions and improves both lifetime and performance.

Chapter 6

Adaptive Clustering Protocol for Wireless Networks

Someone has to be the leader, and no one else is doing it.

- G. G. Allin

Efficiently organizing nodes into clusters is an important application in wireless networks. Clustering divides the network into disjoint subsets, wherein a node from each subset is elected to represent that cluster. Many proposed protocols for both sensor networks and ad-hoc networks rely on the creation of clusters of nodes to establish a regular logical structure on top of which efficient functions can be performed. For example, clustering can be used to perform data aggregation to reduce communications energy overhead [1, 2]; or to facilitate queries [3]; to form an infrastructure for scalable routing [4, 5]; clustering also can be used for efficient network-wide broadcast [6]. Clustering also facilitates in resolving other aspects like MAC layer contention resolution [7], coverage, security [8, 9] and in-network processing. The efficiency of many higher level applications and network functions is pertinent on the regular and efficient structure attained in clustering.

We define the clustering problem as follows: At the end of the clustering algorithm, the nodes should be organized into disjoint sets (*clusters*). Each cluster consists of a *cluster-head* (cluster leader) and several cluster *followers*, all of which should be within one communication¹ radius of the cluster-head, thus causing the overall shape of the cluster to be roughly a circle of one communication radius, centered on the cluster-head. Each node belongs to exactly one cluster (i.e., every node chooses only one leader, even if there may be several leaders within range). Given these constraints, our goal is to select the smallest set of cluster heads such that all nodes in the network belong to a cluster. The problem is similar to the *minimum dominating set* problem in graph theory.

¹Some applications might require k -hop clusters, but for simplicity we only consider 1-hop clusters. Nevertheless, it is straightforward to extend our protocol for k -hop cluster formation.

We note that if each node is in *exactly* one cluster, then maximizing the average cluster sizes while maintaining full coverage is exactly equivalent to minimizing the number of cluster heads while maintaining full coverage. The purpose of minimizing the number of cluster heads is to provide an efficient cover of the network in order to minimize cluster overlap. This reduces the amount of channel contention between clusters, and also improves the efficiency of algorithms (such as routing and data aggregation) that execute at the level of the cluster-heads.

In this chapter, we present Adaptive Clustering Protocol (ACP), a simple but efficient clustering protocol. The key advantages of our protocol are: a) With ACP the number of clusters required scales with density of the network; i.e., the number of clusters required does not increase with the density; b) ACP has very low communication overhead while performance is comparable to other protocols; c) In ACP, a node does not need to know locations/addresses of all its neighbors and hence ACP does not impose any bandwidth overhead such as *hello* messages; d) Behavior of ACP in large networks has been presented and it is shown that ACP performs well even in very large networks. Because of the above-mentioned advantages, ACP is very well suited as an efficient clustering protocol for Heterogeneous Wireless Networks.

The rest of this chapter is organized as follows: Section 6.1 discusses related work, Section 6.2 presents our approach for clustering - ACP, Section 6.3 presents the simulation results of ACP.

6.1 Related Work

In this section, we review related work in clustering algorithms. Several clustering methods such as weighted clustering [93], hierarchical clustering [4] and emergent algorithms [6, 7] have been proposed to organize nodes as a cluster. Most algorithms elect leaders based on certain weights or iteratively optimize a cost function or use heuristic to generate minimum number of clusters. The Distributed Clustering Algorithm (DCA) [94] assumes quasi-stationary nodes with real-valued weights. The Weighted Clustering Algorithm [2] elects a node based on the number of neighbors, transmission power and so on. The Max-Min d -Clustering Algorithm [95] generates d -hop clusters with a run time of $O(d)$ rounds. This algorithm does not minimize the communicating complexity of sending information to the information center.

The hierarchal clustering scheme proposed by Banerjee et. al. [4] uses spanning tree-based approach to produce cluster with certain properties however energy efficiency is not addressed in this work. Perrig et. al. [8] have proposed an emergent algorithms that iteratively tries to achieve high packing efficiency, however negotiation among nodes to be cluster head and join cluster based on degree and proximity leads to high amount of communication overhead, thus wastage energy. Most protocols only discuss initial cluster formation only, re-configuration of clusters to balances the energy among nodes of the cluster are not addressed. These localized and emergent algorithms require high node densities to achieve good results.

LEACH [96] proposed by Heinzelman et al. uses two-layered architecture for data dissemination. In this scheme, nodes periodically elect themselves as cluster-heads with some probability and broadcast an invitation message for nearby nodes to join the cluster. The nodes that do not intend to be cluster-heads join the cluster based on the proximity of cluster-head, thus minimizing the communicating cost. Since the algorithm runs periodically, every node gets a chance to become cluster-head, this is ensured by selecting an appropriate probability value. This balances the energy across the nodes of cluster, thus increases the network lifetime. However since the events are not uniform across the wireless network, some parts of the network may be highly active while others may not be. Hence periodic re-configuration across the network is in-efficient. Reconfigurations of clusters should be localized to areas of higher activity. LEACH assumes that the nodes have transceivers with variable transmission power to optimize communication cost.

Unlike most of the existing approaches, GS^3 [97] is a geography-aware approach that enables network nodes to organize themselves into a cellular hexagonal structure. The time complexity of this algorithm is $\Theta(|Nodes|)$. In GS^3 , the hexagonal structure is fixed and thus GS^3 is not completely geography aware. For efficient performance GS^3 needs that number of $R_t - gaps$ (circular areas of radius R_t with out any node inside) is very low, so that the radius of each cell deviates by at most $2R_t/\sqrt{3}$. For instance, for a density of 20 nodes per R^2 , to ensure the probability of having a non-ideal cell to be less than 0.02, R_t is greater than 0.44R and for a density of 100, R_t is around 0.2R. Thus, for GS^3 to have a low deviation (lesser than 0.2R), the network density needs to be very high (greater than 100 nodes/ R^2). GS^3

also does not provide reconfiguration of the clusters so as to balance the energy consumption across all nodes.

6.2 Adaptive Clustering Protocol

In this Section, we present our algorithm for Cluster Establishment. The algorithm consists of two logical parts - the first deals with the formation of clusters and the second deals with dynamically reconfiguring the clusters to take into account the network dynamics.

Our protocol is based on hexagonal packing as shown in Figure 6.1 and referred as the Covering Problem [32]. The arm length of each hexagon is same as the range of the nodes, R . The reason for this is that, this ensures that all nodes in a cluster are within the transmission range of a node at the center of the hexagon. It should be noted that by defining the orientation and the position of any hexagon is fixed, we can define the whole hexagonal lattice structure.

In real conditions, though, it is rare to have nodes located exactly at the strategically selected locations. Thus, if nodes are not present at the optimal strategy locations, the coverage figure will get distorted; moreover, the distortion effect may propagate. Our goal is to extend the Covering Problem to meet this restriction. A simple solution is to select the nearest node to the strategic location that has received the message to retransmit.

We first present a high level overview of the protocol. In the simplest terms, the network is divided into a uniform hexagonal grid (Figure 6.1) and a node closest to the center of the hexagon is selected as the cluster head. Thus, higher the density of the network, closer is the approximation of the hexagonal grid, thus providing for scalability. A challenge with such a scheme would be that if no node is present close enough to the center of the hexagon, then some nodes might not be clustered (see Figure 6.2). Our approach is not only to find an approximation to the hexagonal grid, but also ensure a formation methodology such that all nodes are clustered.

Next, we present the reconfiguration. Reconfiguration of clusters is critical due to different reasons. Network wide reconfiguration periodically is needed for ensuring balancing of energy. Also, the load pattern might not be same over the entire network. Hence, by reconfiguration, clusters could be formed to better adapt to the prevailing network conditions and thus result in improved performance.

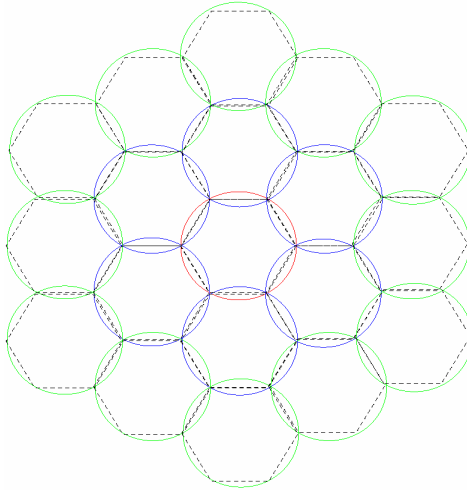


Figure 6.1. Hexagonal packing: Covering a plane with circles in an efficient way

It should also be observed that a node could receive a message more than once - from different directions and from different nodes, each node specifying different optimal strategy location (because of the distortion). This may cause two nodes very close to each other retransmit. We propose to avoid these transmissions by having a node keep track of its distance to the nearest cluster head and to have a node become a cluster head only when its distance to the nearest cluster head is greater than a threshold Th . In Section 6.3, we study the performance of ACP with different threshold values and show that a Th value of $1.2 * R$ is a good choice to ensure less number of cluster heads while keeping the number of unclustered nodes very low. R is the transmission range.

6.2.1 Hexagonal Clustering Protocol

In this section, we present Hexagonal Clustering Protocol (HCP), a simpler version of our protocol to put forth the basic idea of ACP. Unlike ACP, HCP does not adapt to the network conditions. HCP assumes all nodes have equal capabilities and thus each node is equally likely to be a cluster head based on its location. We assume that the base station has more energy and computational resources than all other nodes and hence, we place the constraint that the base station remains to be a *cluster head* all the time.

A node can be in three possible states: it can be *unclustered*, *clustered* or it may be a *cluster head*. In the beginning of the protocol, all nodes are *unclustered*. The

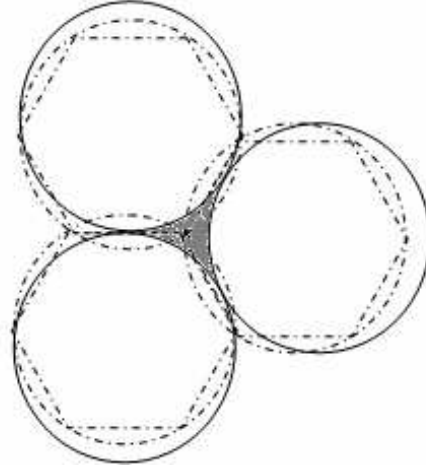


Figure 6.2. Hexagonal Packing and Practical scenarios: Choosing the nearest node to the center of the circles/hexagons might lead to gaps (shaded region). Dotted circles/hexagons represent the ideal locations and solid circles represent the coverage regions of nearest nodes to the centers.

formation of the clusters is initiated by an *Initiator*. A simple method is to have the base station select a node as an initiator whenever necessary.

Initially all nodes in the network are *unclustered*. The Initiator starts the clustering protocol. The Initiator node defines its (hexagonal) cluster by randomly selecting some orientation of a hexagon centered at the Initiator itself. The Initiator designates itself as the *cluster head* and broadcasts Cluster Head Announcement (CHA) claiming itself as the *cluster head* and includes the orientation of the hexagon in the message. The broadcast is limited to 2-hops.

If an *unclustered* node A receives a CHA directly from a *cluster head* C , it accepts C as its *cluster head* and changes its status to *clustered*. The node A also retransmits the CHA. An already *clustered* node simply ignores any CHA message it receives. An *unclustered* node X upon receiving a CHA message, but not directly from a *cluster head*, first computes the orientation and position of its (would be) cluster (hexagon). Then, X calculates its distance d to the center of its cluster (hexagon) and sets its timer to $t = f(d)$. (The choice of $f(d)$ is discussed later in the section.) If X receives a broadcast message from any other node belonging to its cluster before the timer expires, then it nullifies the timer and sets its status to clustered node and recognizes the node from which it received the message as the *cluster head*. If X does not receive any broadcast before the timer expires, it

```

Protocol execution at node  $i$  once it receives a  $CHA_k$  from node  $j$ ;  $k$  is the CH that formed  $CHA_k$ .
CASE 1:  $status(i) == Cluster\ Head$ 
    Discard CHA
CASE 2:  $status(i) == Clustered$ 
    IF  $D_{nearest-CH} < Th$ 
        Discard  $CHA_k$ 
    ELSE
        GOTO CASE 3
CASE 3:  $status(i) == Unclustered$  OR ( $status(i) == Clustered$  AND  $D_{nearest-CH} > Th$ )
    SET  $D_{nearest-CH} = D(P_i, P_k)$ 
    IF  $k == j$ 
        SET  $status(i) == clustered$ 
        SET  $ClusterHead = k$ 
        SET  $TTL = 1$ 
        Forward the packet
    Else
        Calculate distance  $D(P_i, L_s)$ ,  $L_s$  being distance to the nearest CH strategic location
        Initialize a countdown timer to  $d = D(P_i, L_s)$ 
        When the timer expires, check if it received a  $CHA_x$  such that  $D(P_x, L_s) < D(P_i, L_s)$ .
        If true
            Discard all CHAs
        else
            SET  $status(i) == Cluster\ Head$ 
            Construct  $CHA_i$  including next strategic locations.

```

Figure 6.3. HCP - Protocol description

considers itself as the *cluster head* and broadcasts a message. The choice of $f(d)$ depends on the density, time required to process/transmit/receive a message. The complete protocol description is presented in Figure 6.3.

6.2.2 Cluster Reconfiguration

We propose to periodically form completely new clusters along the entire network so as to balance energy among all the nodes in the network. Whenever network wide reconfiguration of the clusters has to be made, the current Initiator starts the process by selecting a new Initiator. The new initiator can be a randomly selected neighbor of the current initiator and the cluster orientation is also fixed randomly. The randomness is to ensure uniform balancing of load over the entire network.

Though the above methodology is very simple, we study the performance of this method in ensuring nodes are equally likely elected as cluster heads and show its effectiveness. To understand the intuition behind this, we observe two facts: (i) Constructing the hexagonal lattice from any vertex would result in the same lattice; (ii) Irrespective of the initiator's location, at least one of the neighbors of the base station has to be a cluster head (because of the implicit clustering

property). Thus, the initiator always being a neighbor of the base station does not affect the performance.

6.2.3 Adaptive Clustering Protocol

In a heterogeneous wireless network, nodes with different capabilities would co-exist. Thus it would be desirable to elect nodes with higher residual energy levels as cluster heads to prolong network lifetime as well as individual node life time and balance energy among the nodes. Also, the load might not be evenly distributed across the entire network. In such scenarios it is desirable to have clusters of varying sizes such that each cluster would have roughly equal loads. In this section, we propose Adaptive Clustering Protocol (ACP) that adapts to the network conditions.

6.2.4 Adaptive Clustering Protocol (ACP)

In several scenarios, traffic rates might be different in different regions of the network. Hence, it is desirable for the ideal power control scheme to support distributed coordination among nodes. ACP allows nodes to adapt it's transmission range to its neighborhood conditions. However, if a node uses a transmission range that is lesser than the transmission range of its cluster head, it may not be able to establish *direct* communication with its cluster head. Hence, once all clusters are established and cluster heads have been elected, each node uses the same transmission range as its cluster head.

Each node could estimate the *approximate* network load in its neighborhood based on the observed channel idle time. Thus, each node could periodically compute the approximate load in its neighborhood and compute optimal transmission range. We use the analytical model developed in Section 2.2 for adaptation and computing the optimal transmission range. Then, each node would delay the transmission of broadcast message based on the additional area it would cover. For instance, consider two nodes separated by distance \bar{d} and with transmission ranges R and r respectively. Then the additional area covered can be computed as follows:

The area of intersection is given as

$$A_{int} = r^2 \cos^{-1} \left(\frac{\bar{d}^2 + r^2 - R^2}{2\bar{d}r} \right) + R^2 \cos^{-1} \left(\frac{\bar{d}^2 + R^2 - r^2}{2\bar{d}R} \right) - \frac{1}{2} \sqrt{(r + R - \bar{d}) (-r + R + \bar{d}) (r - R + \bar{d}) (r + R + \bar{d})} \quad (6.1)$$

Thus, the additional area covered by a node with transmission range R and located at a distance $d = 2R\bar{d}$ from the strategic location can be obtained as

$$A_{additional} = \pi R^2 - A_{int} \quad (6.2)$$

Thus, the delay function at each node is computed as

$$delay_k = \frac{c}{A_{additional}(k)} \quad (6.3)$$

Thus a node with least delay would elect itself as a cluster head and broadcasts a CHA. Whenever a node broadcasts a CHA message, it includes its transmission range in the message so that all nodes that decide to be part of its cluster use the same transmission range.

- **Energy Balancing**

Heterogeneous Wireless Networks are envisioned to comprise of nodes with different capabilities leading to different energy levels of nodes. Even in Homogeneous Wireless Networks, where all nodes have same energy levels during the bootstrapping stage, because of different roles/tasks each node would be performing, node energy levels vary from one another.

To simultaneously prolong the network lifetime as well as each nodes lifetime, it is required that nodes with higher energy levels are elected as cluster heads more frequently than nodes with lower energy levels. We propose to achieve this by setting the delay d as follows:

$$delay_k = c \left(\frac{1}{A_{additional}} + \frac{Avg_energy_i}{Energy_i} \right) \quad (6.4)$$

where, $A_{additional}$ is as explained in the previous section.

d_t is the distance to the nearest vertex.

$Energy_i$ is the energy level of node i .

Avg_energy_i is the average energy level of the neighbors of node i .

The intuition behind this is as follows: The lower the energy level of a node, the lesser it should participate in broadcasting. Thus, by having an energy component in the delay, a node with low energy will delay retransmitting a broadcast for a longer duration than a node with higher energy levels. Thus, the probability of a node with lower energy levels becoming a cluster head is lesser. We note that,

though this mechanism requires energy levels of the neighbors, the information need not be accurate as we observe that the changes in the energy levels is not drastic.

Finally, we note that under uniform distribution of load over the entire network and when all nodes have similar energy levels, ACP and HCP are equivalent.

6.3 Performance Evaluation

In this section, we evaluate the performance of our clustering algorithm using simulations. First, we study performance in networks with out any data packets to study the efficiency in number of clusters. Then, we introduce data packets to study the effectiveness of ACP in adapting to the network conditions.

A wireless network of varying network sizes with varying node density and threshold values were simulated. The nodes were uniformly distributed all over the region with the density varying from 5 nodes per $R * R$ region to 20 nodes per $R * R$ region. Every simulation is repeated until the 95% confidence intervals of all average results are within $\pm 5\%$.

The simulations are aimed at studying the performance of ACP in networks of different sizes and densities. Initially, we simulated the ideal case where some node always exists at the strategically selected location. Then, we studied the effect of different threshold values on the performance of ACP. Then, we concentrated on the algorithm efficiency by studying the performance of ACP in networks of different sizes and densities. We compare our protocol with ACE [8].

6.3.1 Ideal Case Scenario

We define *ideal case scenario* as a scenario in which some node exists exactly at each of the strategically selected locations. The number of clusters required to cover circular and rectangular regions in the ideal case scenario are observed and are as presented in Table 6.1 and Table 6.2. The number of clusters required in the ideal case present a lower bound on the number of clusters required in any network.

6.3.2 Effect of Threshold Th

The purpose of this study is to evaluate the effect of different threshold values on the performance of clustering algorithm. The percentage of nodes that were unclustered and percentage of nodes that were selected as cluster-heads for

TABLE 6.1. Number of clusters in a circular network in an Ideal Case

Radius of Circular region	Number of transmissions
2R	7
3R	13
4R	19
5R	31
6R	43
7R	61
8R	79

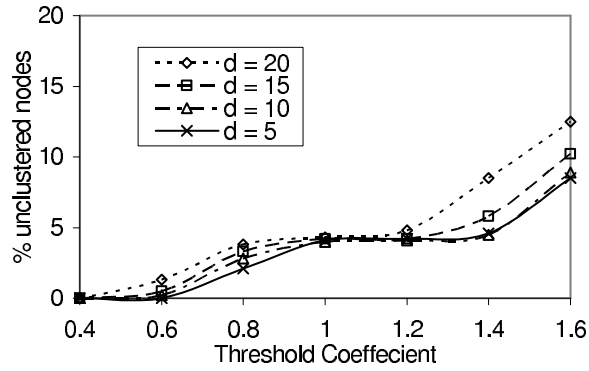
TABLE 6.2. Number of clusters in a rectangular area in an Ideal Case

Size of the rectangular region	Number of transmissions
3R*3R	5
4R*4R	9
5R*5R	14
6R*6R	16
7R*7R	22
8R*8R	30
10R*10R	42

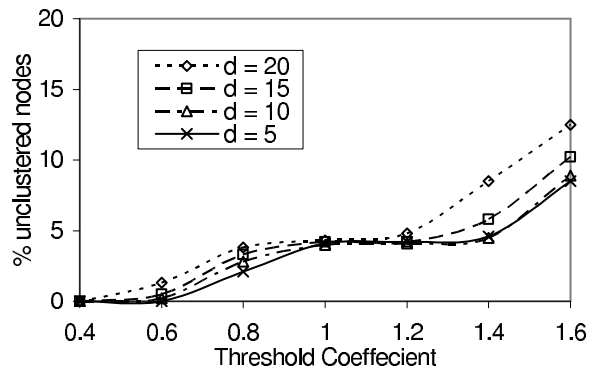
varying densities and threshold value were simulated. Figure 6.4(b) shows, that as the threshold coefficient is increased the percentage nodes becoming cluster-heads decreases. On the contrary increasing the threshold coefficient also increases the percentage of unclustered nodes (nodes that along the edges of the deployment area) as shown in Figure 6.4(b). Thus an appropriate value for threshold coefficient has to be determined for given density such that optimal clustering achieved with least percentage of the un-clustered nodes. It should be noted that at a Th value between 1.05 and 1.35, the percentage of unclustered remains almost a constant except for the case when density is 5 and in this range the percent of *unclustered nodes* is around 6% . When density is 5, the increase in percent of unclustered nodes is gradual till $Th = 1.35$. Thus, the optimal value of Th is between 1.05 and 1.35. For all further simulations, we use threshold value of $Th = 1.2$.

6.3.3 ACP Efficiency

The purpose of this study is to evaluate the performance of ACP in networks of different sizes and different densities. We include a "best-case" bound provided by the simulation results in *ideal case scenarios*. It is impossible for any algorithm to perform better than the performance in *ideal case scenario*. Thus, this bound



(a) Percent of cluster heads vs Th



(b) Percent of unclustered nodes vs Th

Figure 6.4. Effect of Th on the performance of ACP

provides a useful spectrum to gauge the performance of our protocol. For this study we varied the network size from $5 * 5$ to $12 * 12$. The transmission radius of each node is unity. We also varied the density of the network from 5-nodes to 20-nodes per unit square.

First, fixing the Th value and varying density of the nodes in the region, we simulated the number of clusters needed to cover a square/rectangular region completely. The coverage figure gets distorted a lot as in most of the cases no node exists at the strategic location. Figure 6.5 shows two such cases. Figure 6.5(a) corresponds to a network of size $10 * 10$ with node density of 10. Number of clusters here is 41. Figure 6.5(b) corresponds to a network of size $12 * 12$ with node density of 8 and the number of clusters is 59.

Next, we simulated the number of clusters formed to cover areas varying from $3 * 3$ to $12 * 12$ for various densities. As shown by Figure 6.6, the number of clusters formed is linearly proportional to the area and is almost independent of the density; this is depicted in the Figure 6.6 below. Hence our clustering scheme

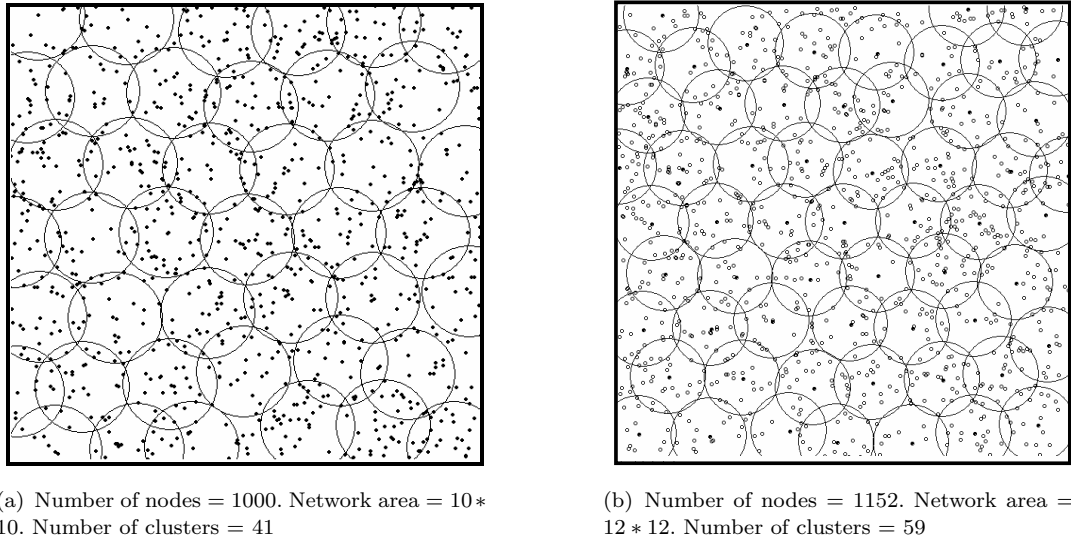


Figure 6.5. Distorted coverage figures in non-ideal cases

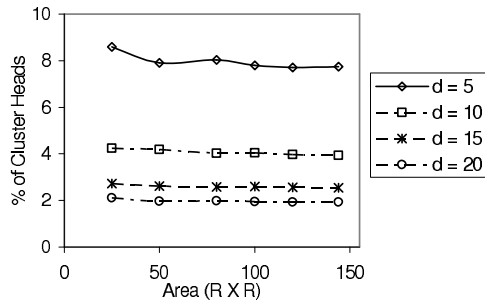


Figure 6.6. Percentage of cluster heads formed for various network areas varying from 3 * 3 to 12 * 12

is not hindered by dense networks. Figure 6.7 presents the performance as density varies. It should be observed that ACP is scalable with respect to density. In fact the performance slightly improves as density increases.

6.3.4 Distortion

Fixing the density in the region, we simulated the number of transmissions needed to cover a square/rectangular region completely. The coverage figure gets distorted considerably and, in most of the cases, no node exists at the strategic location. In order to quantify the distortion, we define Degree of Distortion as follows:

Degree of Distortion (DoD) is defined as the average distance between the nearest node that would retransmit the packet and the strategic location normalized to

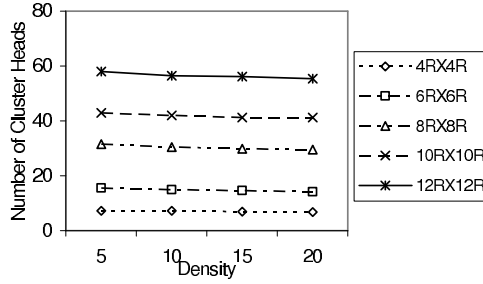


Figure 6.7. Number of clusters formed for various network areas varying from $3 * 3$ to $12 * 12$

the communication range of the nodes, i.e.,

$$DoD = \frac{1}{|S|} \sum_{\forall i \in S} \frac{d(L_i, L_P)}{R} \quad (6.5)$$

where S is the set of nodes that are selected as cluster heads and $|S|$ is the size of S .

$d(L_i, L_P)$ is the distance between the strategic point L_P and L_i location of CH i ($i \in S$) nearest to L_P . R is the transmission range.

Ideally, the degree of distortion should be same as the expected nearest neighbor distance. For a completely random distribution, the expected nearest neighbor distance is given by

$$d(ran) = 0.5 \sqrt{\frac{A}{N}} \quad (6.6)$$

We compare the observed values of the Degree of Distortion with the ideal values for various densities in Figure 6.8. We observe that experimental DoD values is higher than the ideal values. This can be explained from the observation that, in ACP, not all nodes surrounding the strategic location might receive the message at the same time.

6.3.5 Average Delay per Hop

Assuming an idealistic MAC in which there are no collisions, we observed the average delay a node has to wait at each hop before selecting itself as a cluster head is presented in Table 6.3. The maximum allowed delay is 50 ms. We observe that even at low densities, the delay is around 17ms, while at high densities the delay is very low and nearly negligible.

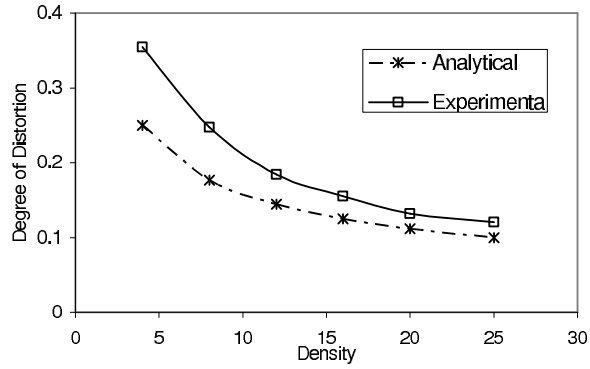


Figure 6.8. Degree of Distortion - experimental and analytical

TABLE 6.3. Delay observed for various densities

Density	Delay per hop (<i>msec</i>)
4	16.9
6.25	14.1
16	8.4
25	7.2
100	3.7

6.3.6 Performance Comparison

We compare our clustering approach against ACE [8], which uses localized emergent approach to clustering. The ACE and ACP was simulated for various areas and densities and the number of clusters required where compared. This is depicted in Figure 6.9(a) for a density of 10 and Figure 6.9(b) for a density of 20 respectively. In both cases the numbers of cluster of given area required by ACP is significantly lesser that of ACE and node degree based clustering algorithm.

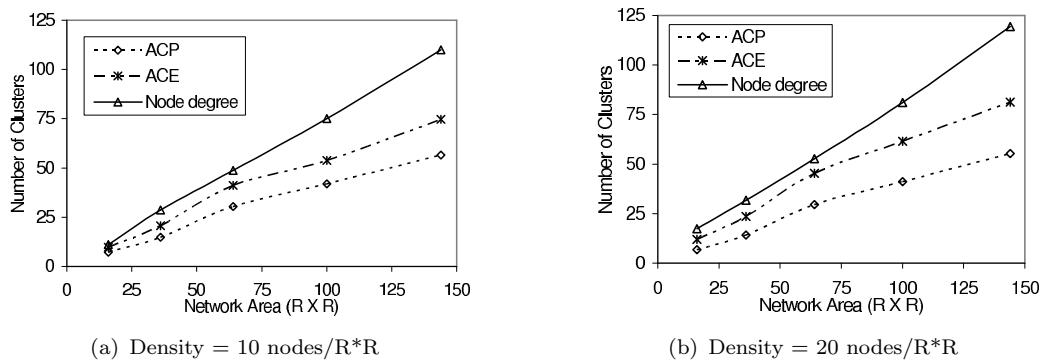


Figure 6.9. Performance comparison

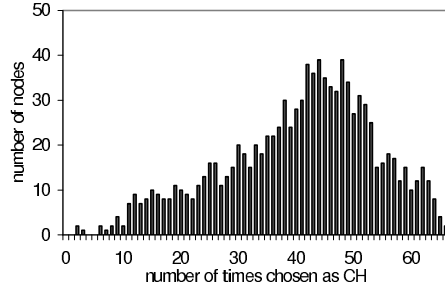


Figure 6.10. Distribution of the number of times each node is elected as a cluster head

6.3.7 Energy Balancing

Figure 6.10 shows the distribution of the number of times each node is elected as a cluster head. The Scenario is for a $10 * 10$ network with 1000 nodes. The values are over a run of 500 cluster formations. The average number of times a node is elected as BN is 41.04 corresponding to a standard deviation of 14.31. It should be noted that the number of nodes within the range (20, 60) is 894. We observe that the distribution is distorted to the left. The reason is that nodes closer to the edge of the network are selected fewer number of times.

6.3.8 Adaptation to Network Conditions

The focus of this section is to study how effectively ACP can adapt to the prevailing network conditions. The simulations are over a network area of $6 * 6$ with 180 nodes. We also introduce background traffic where nodes transmit data packets at a rate depending on *the term Average Load, L_{avg}* i.e., each node selects a data transmission rate randomly from the interval $(0, L_{avg})$ and transmits packets at this rate through out the simulation. Data has been collected from 10 simulation runs, each simulation run being for 100 seconds. Reconfiguration of clusters takes place every 20 seconds. The initial energy of each node is set to a value between 10 and 30 power units. The energy level is a function of node address, i.e., a node i has an initial energy of $10 + \frac{(30-10)*i}{N}$, where N is the total number of nodes in the network. Thus ideally, we expect the number of times a node is selected as a cluster head to be proportional to its address (energy level). For baseline comparison, we simulated the performance of HCP with all nodes assigned unit transmission range. In ACP, nodes select a transmission range that ensures that at least 80% of its neighbors receive broadcast messages transmitted by it for the given network conditions.

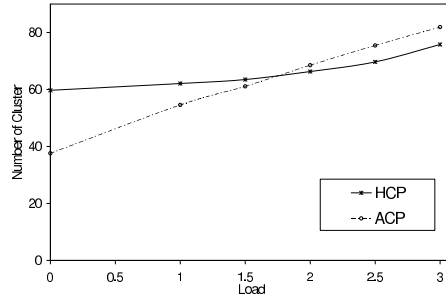


Figure 6.11. Number of clusters as a function of load

We focus on the following metrics:

- *Number of Clusters* - The average number of clusters is computed. Once the average number of clusters is computed, it is straight forward to compute the average cluster size.
- *Number of Clustered Nodes* - The average number of clustered nodes is an important metric. While several works considered this metric, the major difference lies on how it is measured. Previous works assume that each node receives all broadcast packets, which might not be practical. In our simulations, we consider packet losses due to collisions and interference which impact the number of clustered nodes.
- *Energy Distribution* - We study how ACP uses the node energy levels of the nodes to select cluster heads and decrease the disparities in the energy levels.

The number of clusters for different loads is presented in Figure 6.11. With ACP, when load is low, nodes choose high transmission ranges and hence the number of clusters is less. As load increases, nodes reduce transmission ranges to ensure reliability and hence number of clusters is higher. Even in case of HCP, the number of clusters increases with load, albeit for a different reason. At higher loads, the chances receives a broadcast message is low and hence the cluster formation is not very efficient leading to increase in number of clusters.

The percentage of clustered nodes for different loads is presented in Figure 6.12. As observed, with ACP, since nodes select transmission ranges that ensure the given reliability, the number of clustered nodes remains very high (above 90%) even at high loads. The reason the number of clustered nodes is significantly higher than

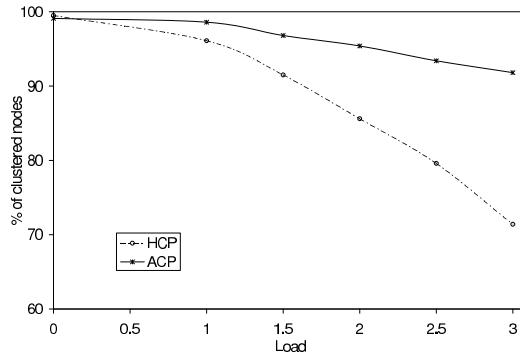


Figure 6.12. Percentage of clustered nodes as a function of load

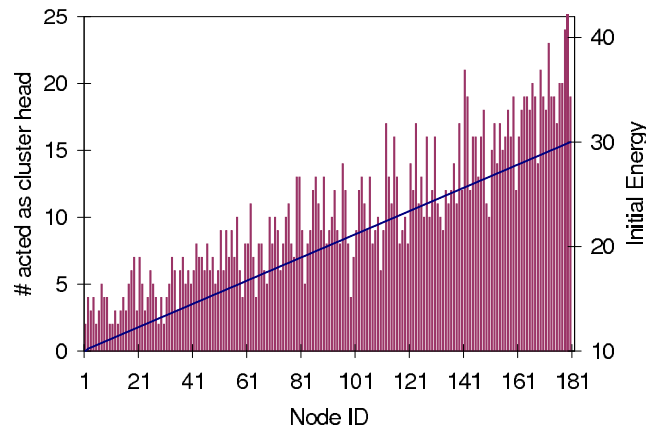


Figure 6.13. Distribution of the number of times each node is elected as a cluster head

the expected reliability of 80% is because several nodes receive the CHA message from more than one node. Thus, if a node receives a CHA from at least one node, then it can change its status to *clustered node*. With HCP, with out any adaptation, we observe that around 29% of nodes remain unclustered.

Finally, the distribution of the number of times each node is elected as a cluster head is presented in Figure 6.13. We observe that there does exist a relation between the address of a node (i.e., the energy level) and the number of times it is elected as a cluster head. While there is not complete dependency on the energy level, ACP does significantly balance the energy levels.

6.4 Summary

We present Adaptive Clustering Protocol (ACP), a novel protocol for clustering. ACP is an extension of the Covering Problem encountered in geometry. The protocol is performed in an asynchronous and distributed manner by each node in the network. The protocol does not require a node to have any neighborhood information and thus poses minimal storage overhead. ACP has a number of ad-

vantages over other approaches considered in the literature. The best features of ACP is that it scales with density and adapts to network conditions resulting in clusters with uniform load. ACP imposes lesser communication overhead and the efficiency of ACP remains very high even in large networks.

Chapter 7

A Hierarchical Anonymous Communication Protocol for Heterogeneous Wireless Networks

The things most people want to know about are usually none of their business.

- *George Bernard Shaw*

Wireless networks, applied to monitoring physical environments, have recently emerged as an important application resulting from the fusion of wireless communications and embedded computing technologies [98, 99, 100, 101]. Wireless sensor networks consist of hundred or thousands of sensor nodes, low power devices equipped with one or more sensors. Potential applications include monitoring remote or inhospitable locations, target tracking in battlefields, disaster relief networks, early fire detection in forests, and environmental monitoring.

With the growth and acceptance of the wireless networks, there has been increased interest in maintaining anonymity in the network. The mere fact that a node has sent some information to the base station can reveal extremely important information. For instance, consider a sensor network deployed for intruder detection in which a sensor keeps sensing for intruders. Thus, when an intruder, once in the network area, sees a transmission from a sensor close to his location, can rightly assume that the his presence is sensed and might pursue evasive actions immediately. In general, interception of messages containing the physical locations of wireless nodes allows an attacker to locate the nodes and destroy them. The significance of hiding location information from an attacker lies in the fact that the wireless nodes have small dimensions and their physical location cannot be trivially traced. Thus, it is important to hide node locations. In the case of static nodes, the location information does not age and must be protected throughout the lifetime of the network. Moreover, it should be noted that adversaries can correlate data flow patterns to event locations/active areas using traffic analysis. Therefore, there is a strong need to develop anonymity mechanisms which hide the location of nodes and obscure the correlation between event zones and data flow from snooping adversaries.

Privacy International [10] defines four categories of privacy: information privacy, bodily privacy, communication privacy, and territorial privacy. Location privacy is a particular case of information privacy and can be defined as the ability to prevent other parties from learning one's current and past locations [11]. Anonymity can be defined as the state of being not identifiable within a set of subjects called the anonymity set [12].

Conventional protocols [13, 14, 15] proposed to ensure user anonymity in the Internet are based on the communication model in which high traffic conditions and high processing power is assumed, which might not be true with respect to wireless networks.

We present Hierarchical Anonymous Communication Protocol (HACP), a novel protocol that prevents traffic analysis from revealing node information including its location. We use token ring approach for achieving anonymity of communication between cluster heads. Routes are chosen and frames are scheduled to traverse these routes. Each frame is assigned a token and a node can send a message through a frame only if the token is free.

The rest of the chapter is organized as follows: Section 7.1 deals with related work, section 7.2 discusses our design goals and network model, section 7.3 presents our protocol, section 7.4 discusses security and performance results of HACP.

7.1 Related Work

In this section, we discuss some existing research efforts related to wireless networks, secure routing, anonymity and location privacy.

The problem of routing in wireless networks has been initially studied in a non-adversarial setting, and recently the focus of research shifted to the design of secure routing protocols; researchers have already devised a number of proposals to secure both reactive (on-demand) and proactive routing protocols and identified a number of attacks [102, 103, 104, 9]. There are several recent research efforts exploring different aspects of wireless network security, for example key management [105, 106], secure multicast communication [107], authentication [108, 109, 110] and location privacy [111, 112, 113, 114].

Anonymous communication for wired networks is a well-studied aspect. A seminal work in the domain of anonymity was notably reported by Chaum in [115]. In

[15], Reiter and Rubin present Crowds, a scheme that enables anonymity of web transactions. The concept of a mix is introduced in [15]. A single processor in the network, called a mix, serves as a relay. Each processor P that wants to send a message m to a processor Q encrypts m using Q's public key to obtain m' . Then P encrypts the pair (m', q) using the public key of the mix. The mix decrypts the message and forward m' to q . This scheme has been extended where several mixes are used to cope with the possibility of compromising the single mix. Another approach is to interpose an additional party (an anonymizer) between the sender and receiver to hide sender's identity from the receiver.

The Mist routing project [116] addresses the problem of routing a message to the user while keeping its location private. Mist operates by making use of a set of mist routers organized in a hierarchical structure that provides location privacy. In [117], Smailagic et al. present two location sensing systems and compare them to the existing location sensing proposals. They further perform a user privacy study and show that users expect two unique behaviors from the system: an introvert model, where privacy is preferred, and an extrovert model where availability is preferred. Recently, Kong and Hong have proposed a protocol for anonymous communication in mobile ad hoc networks [118].

We consider a more general attacker model considered for the Internet [13, 14, 15] in which the attacker (that may not be part of the network) has access to the entire networks traffic information. The protocols described here are designed to be resilient to traffic analysis i.e., to make it difficult for observers to learn identifying information about the origin/destination of a connection. Also, we aim at hiding information about a node transmitting/receiving a message. Thus, the attacker would not be able to even figure out if a node is transmitting any data.

7.2 Design Goals and Network Model

7.2.1 Design Goals

We want to design a system that enables anonymous communication. Anonymity is the state of being not identifiable within a set of subjects called the anonymity set. Here, we define these terms more precisely in the context of hybrid ad hoc networks.

Anonymity is generally classified into source and destination anonymity. *Source anonymity* is defined as the property that a particular message is not linkable to any source, and vice-versa. A similar definition applies to *destination anonymity*. *Unlinkability* in this context means that the probability that a particular message was sent by a given source and/or received by the same destination is the same as imposed by the a priori knowledge. This means that the process of sending and/or receiving messages does not reveal any additional information about the identities of the source and/or destination that was not already known to the attacker prior to the message transmission.

7.2.2 Network Model

We consider clustered wireless networks because clustering allows for scalability of MAC and routing. Cluster heads also serve as fusion points for aggregation of data, so that the amount of data that is actually transmitted to the base station is reduced. Clustering nodes into groups, so that nodes communicate information only to cluster heads and then the cluster heads communicate the aggregated information to the processing center, may save energy. Many clustering algorithms in various contexts have been proposed [119, 2, 120]. These algorithms aim at generating the minimum number of clusters such that any node in any cluster is at most d hops away from the cluster head.

We use the communication graph $G(V_{CH}, E)$ to represent the network in terms of cluster heads. V_{CH} is the set of cluster heads and E is the set of communication edges (might be paths involving intermediate non-cluster heads) connecting the cluster heads. We assume that G is connected.

We initially fix a spanning tree in the graph. Next, using an Euler tour (that is a DFS tour) of the spanning tree in the graph, we define a ring. Also, the ring formation can use the underlying routing protocol to achieve energy efficiency and load balancing.

We base our protocol on symmetric key cryptographic techniques because they require lesser computational and energy consumption requirements. There exist a number of key pre-distribution schemes for wireless networks to set up secret keys among nodes [121, 105, 106]. We assume that each node shares a secret key with its cluster head. Also, each cluster head shares a symmetric key with its neighboring

cluster heads in the ring. We use $E(M, K_{ij})$ to represent encryption of message M with K_{ij} , the secret key shared by nodes i and j and $D(M, K_{ij})$ to represent decryption of message M with K_{ij} , the secret key shared by nodes i and j .

Tokens and Frames: At anytime there can be only one frame traversing through the ring. The nodes use a *token passing access mechanism* to access a frame passing through the network. A node wishing to send data should first receive permission. When it gets control of the token, it may transmit data in that frame. Each *frame* is of fixed length and contains the *status* of the token itself. A token can be either in *free status* or *occupied status*. The format of the frame is as follows:

$$\langle E((Token || E(Frame_{Header}, K_{sd}) || E(Frame_{Data}, K_{sd})), K_{si}) \rangle$$

where K_{si} is the secret key shared between the source node s and node i that is the upstream neighbor of sender s and K_{sd} is the secret key shared between the source node s and destination node d .

The format of the Token is as follows:

$$\langle Redundancy\ predicate || Status \rangle$$

Redundancy predicate is used for checking the validity of the frame. For the frame to be verified successfully by node i , upon decryption the *Redundancy predicate* must be fulfilled. *Status* specifies if the token is *occupied* or *free*. If a token is *free*, a node can send data through that frame; else it cannot.

The format of the *Frame Header* is as follows:

$$\langle Redundancy\ predicate || Source\ Address || Destination\ Address \rangle$$

Again *Redundancy predicate* is used for checking the validity of $E(FrameHeader, K_{sd})$.

The format of Frame Data is as follows:

$$\langle Datalength || Data || Padding \rangle$$

Data length specifies the length of the total *data* in the packet. This is crucial when the amount of data needed to be sent is not enough to fill the whole frame. In that case, data to be sent is padded with some random number to meet the constraint that the size of the frame is of fixed length.

7.3 Hierarchical Anonymous Communication Protocol (HACP)

HACP provides two different mechanisms to achieve anonymity - one is based on introducing dummy messages for anonymity with in a cluster and the other is based on ring based approach for anonymous communication with in cluster heads.

7.3.1 Anonymous Communication with in a Cluster

Inserting dummy traffic in a network is a technique that hides the traffic patterns inside the network, making traffic analysis more difficult [122]. The generation of dummy traffic increases the anonymity of the messages sent through the mix network.

A dummy message is a *fake message* created by a node. The final destination is its cluster head; the dummy message is discarded by the cluster head. Observers of the network and other nodes cannot distinguish the dummy from a real message.

In HACP, each node (including the cluster head) transmits messages at a Poisson rate r_t . Thus, on an average each node would send a message every $1/r_t$ seconds. Let r_t denote the sensing rate of each node. Thus, whenever there is sensed data to be sent, the node encrypts the data message with the secret key it shares with the cluster head and transmits it. Else the node sends dummy messages. Hence, the dummy messages are sent at a rate of $(r_t - r_s)$.

Whenever a cluster head has a message to be sent to one of its cluster nodes, the cluster head simply encrypts the message with the secret key it shares with that node and sends. Whenever a node senses a packet transmission, it receives the packet and decrypts it with its key and checks if it is a valid packet.

7.3.2 Anonymous Communication between Cluster Heads

Whenever a node i receives a frame, it decrypts the frame using the key shared with its downstream node in the ring and verifies the redundancy predicate. Once the *Redundancy predicate* is fulfilled, the following algorithm is executed.

1. If the node has no data to send, it just encrypts the resultant plain frame with the common key shared with its upstream node and retransmits the packet on to the ring.
2. If the *status* of the token is *free* and the node has some data to send to another node D , then i constructs the frame as follows:

- Node i constructs *FrameHeader* and *FrameData* as explained earlier using key shared with the Destination.
- Node i sets the *status* field in the token to *occupied*.
- Computes the following using its shared key with upstream node and transmits the packet on the ring

$$\langle E((Token || E(FrameHeader, K_{sd}) || E(FrameData, K_{sd})), K_{si}) \rangle$$

3. If the *status* of the token is set to *occupied*, the node checks if the data in the frame is destined to itself by decrypting $\langle E(FrameHeader, K_{sd}) \rangle$ with the shared key and checking if the *Redundancy predicate* is fulfilled.

- If the node is able to check the validity of the frame header, then it is addressed to node i , which makes a copy of it. It encrypts the whole frame with the key shared with its upstream node and transmits the frame on to the ring.
- Else, if the node i is not able to check the validity of the frame header, then it is not the destination and the node just encrypts the whole frame with the key shared with its upstream node and transmits the frame on to the ring.

Once the frame returns to the source, the source repeats the procedure as long as it has data to send. When it has no more data to send it sets the *status* field of the token to *free*, assigns the whole frame to some randomly generated data. Then it encrypts the whole frame with its shared key with upstream node and transmits the frame on the ring.

7.3.3 Multiple Rings

In a network consisting of n nodes, the ring size is n . Thus, a message needs to be transmitted along the whole ring and hence, each message is transmitted n times. To reduce the communication overhead (complexity), we divide the graph into sub-graphs and construct rings within each sub-graphs. An example partition is shown in 7.1. The dark circle indicates the base station to which all the nodes are communicating with.

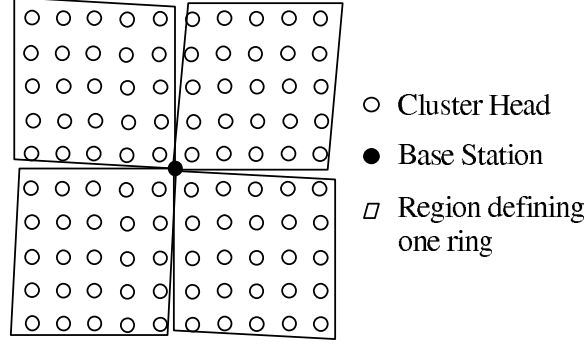


Figure 7.1. A partition of a network into multiple rings

Once we have the partition to sub-graphs, we have one ring in each sub-graph, which is formed by an Euler tour on the spanning tree of the sub-graphs. We call the nodes that are part of more than one ring as *Junction nodes*. There are at most δ_x nodes in each sub-graph, thus the time complexity is $O(\delta_x)$ within a sub-graph.

In order to enable communication with node outside a sub-graph, we assign each ring a unique identifier, RID. Also, each node knows the RID of the ring to which the destination belongs. We introduce a new header - $E(Frame_{RID}, K_{sJ})$ - in the frame in order to identify the destination's RID, where K_{sJ} is the common key shared by the source with the Junction node that is also part of a ring that has to be traversed to reach the destination.

The modified format of the frame as follows

$$\langle E((Token||E(Frame_{RID}, K_{sJ})||E(Frame_{Header}, K_{sd})||E(Frame_{Data}, K_{sd})), K_{si}) \rangle$$

The format of FrameRID is

$$\langle RIP||RID_D \rangle$$

RIP is the redundancy predicate that has to be fulfilled so as to indicate successful decryption. RID_D is the Ring Identifier of the destination's ring. The sender encrypts FrameRID with the key shared with the Junction node that is part of ring that is on the way to the destination's ring.

When a node in one ring has data to send to a node in another ring, then the frame need to be transferred from one ring to another until it reaches the ring of the destination. For this each Junction node maintains a forwarding routing table that specifies the ring a frame addressed to a particular destination ring has to be transferred to. A *Junction Node* upon successful decryption of $E(Frame_{RID}, K_{sJ})$

stores a copy of the frame and then retransmits the frame. The junction node based on the RID of the destination node, decides to which ring the frame has to be transferred. Then, it waits for a free token on the other ring it has to transmit the copied frame, encrypts the frame with the common key it shares with the next junction node on the way to the destination's ring and transmits the frame. The process continues till the frame reaches the destination's ring, where the Junction node that of RID_D that receives the frame just assigns some random string to $E(Frame_{RID}, K_{sJ})$ and transmits the frame on to the ring RID_D .

This mechanism prevents local traffic from traversing the whole network. Even if an adversary were able to compromise a Junction node, he would just be able to know the ring to which frame was destined to and no more. The attacker could not even figure out the originating ring of the frame. Thus, this mechanism does not reduce the anonymity provided by the protocol.

In some situations, only some nodes might have a need for anonymity in which case a ring has to be established only among those nodes. In such cases, the neighbors in a ring need not be physical neighbors in the network and these nodes can communicate using the shortest path available.

7.4 Performance of HACP

In this section we present the performance of HACP in terms of the overhead imposed and the anonymity provided. Initially, we describe the metrics we would be considering and present the performance of HACP in terms of these metrics.

7.4.1 Metrics

Anonymity can be measured with various metrics, among which the most common is based on anonymity set. In our system, if the attacker holds the list of registered network nodes, the maximum degree of anonymity that the system can provide is proportional to the size of the list; in this case, the list corresponds to the anonymity set of the network. We will assume that the network has a sufficiently large anonymity set, so that it thus provides a reasonable anonymity to the users. In our protocol, the size of the anonymity set is same as the size of the ring. Thus, bigger the ring is more is the anonymity provided.

We present a new metric *Data Exposure Index* in section 7.4.3. This metric effectively captures the probability with which an attacker can guess if a node is

sending data. We consider the communication overhead imposed by the protocol and also discuss the average delay encountered by a packet before it reaches the destination.

7.4.2 Communication Overhead

In HACP, whenever a node has data to send, it captures a free token and sends data in that frame. Else, it just forwards the idle frame. Thus, even if node has any data to send, at least one frame would be traversing the ring. We use the term *communication overhead* to represent the number of transmissions that correspond to idle frames. It should also be noted that the power consumption of a node can be derived from the average current drain [123] given by

$$I_{avg} = T_{on} * I_{on} + (1 - T_{on}) * I_{stby} \quad (7.1)$$

where

T_{on} is fraction of time receiver or transmitter is *on*

I_{on} is current drain from battery when receiver or transmitter is *on* and

I_{stby} is current drain from the battery when both transmitter and receiver are off.

Thus, higher the communication overhead higher is the T_{on} , which implies higher is the power consumption. Thus, communication overhead also acts as a direct measure of power consumption.

Consider a ring with N number of nodes out of which N_a nodes have data to send at a rate of R packets per unit time. Let us say, a frame can traverse the ring at a maximum of t times in one unit of time. The value of t depends on ring latency, which in turn depends on the transmission time of the frame (T_{tr}), ring traverse time delay (T_t) and processing delay at a node (T_{proc}). Here, we ignore the delay incurred at a node to process the frame before forwarding it.

$$t = \frac{1}{(N * T_{tr} + T_{proc} + T_t)} \quad (7.2)$$

If n tokens are present in the ring, then a maximum of $n * t$ frames can be transmitted across the ring. Thus, ideally, we would like to have the following condition satisfied, so that no idle frame is transmitted:

$$\frac{N_a}{N} * R = n * t \quad (7.3)$$

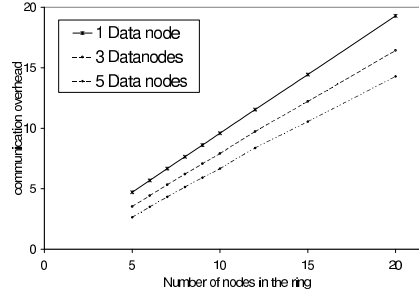


Figure 7.2. Communication Overhead Vs number nodes in a ring

Thus, the fraction of idle frames being transmitted over the ring is $1 - \frac{N_a * R}{N * n * t}$. Thus, communication overhead i.e., number of transmissions corresponding to idle frames, is given by

$$\begin{aligned}
 \text{Communication overhead} &= \text{number of idle frames} * \text{number of nodes in the ring} \\
 &= N - \frac{N_a * R}{n * t}
 \end{aligned} \tag{7.4}$$

The communication overhead in rings for varying sizes and for different number of tokens is presented in 7.2. The communication overhead increases almost linearly as number of nodes in the ring increases. This behavior is as expected because with more number of nodes in a ring more number of transmissions occur corresponding to each frame generated by any node.

7.4.3 Data Exposure Index

We introduce a new metric called *Data Exposure Index* (DEI) defined as follows:

$$\text{DEI} = \frac{\text{Number of data generating nodes on the Ring}}{\text{Total Number of Nodes on the Ring}} \tag{7.5}$$

The worst case scenario is when the DEI is equal to one. In this case all nodes on the ring generate data and the attacker's assumption that data is being sent by some node is valid. The goal of hiding the information a node is sending some data cannot be achieved in this case. On the other hand lower DEI is achieved by having few data source nodes on the ring. Less data sources or more the total nodes on the ring reduces the chances of the attacker to identify the data sources.

Figure 7.3 shows the trade off between communication overhead and exposure degree. When the total number of nodes on the ring increases, while having the

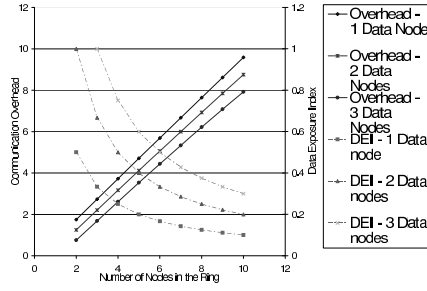


Figure 7.3. Trade off between Communication overhead and Data Exposure Index

data sources the same, it can be observed that the DEI (right y axis) decreases but the bandwidth/power overhead (left y axis) increases. The user can get different trade offs by changing the number of data sources on the ring. For instance, for high anonymity, rings with high number of nodes have to be used, but which results in high communication overhead. Also, to keep the DEI low, ring formation should be such that only few nodes are transmitting at a given point of time. It should be noted most of the related works aim at hiding the communication pattern (i.e., who is talking to who) and not hiding the information if a node is transmitting or not. For these works, the DEI would be one as the attacker would be able to figure out who is transmitting and who is receiving, though he is not able to find out who is receiving from whom.

7.4.4 Mean Waiting Time

The mean waiting $E[W]$ for a frame normalized to X is computed in [124] and is given as

$$\frac{E[W]}{X} = \frac{\rho(1 + 2a' + a'^2) + \left(1 + \frac{\rho}{M}(1 + a')\right)}{2 \left(1 - \left(1 + a'(1 + \frac{1}{M})\right) \rho\right)} \quad (7.6)$$

where,

M is number of nodes in the ring

X is frame transmission time

ρ is load of a station and defined as arrival rate at a station X transmission time.

This assumes exponential inter-arrival times

t' is the ring latency i.e., the propagation delay for a frame to traverse the ring

$$a' = t'/X$$

Figure 7.4 presents the waiting time in rings with different number of nodes. As it can be observed, the wait time increases very fast as the number of nodes in the

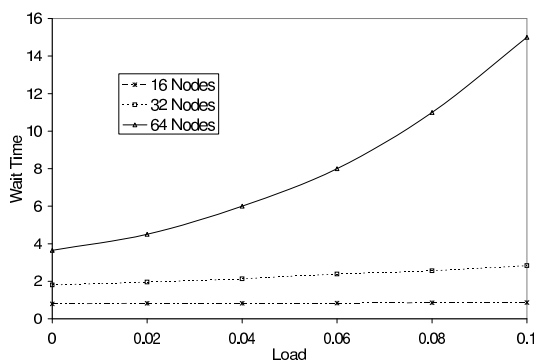


Figure 7.4. Wait time in rings of different sizes

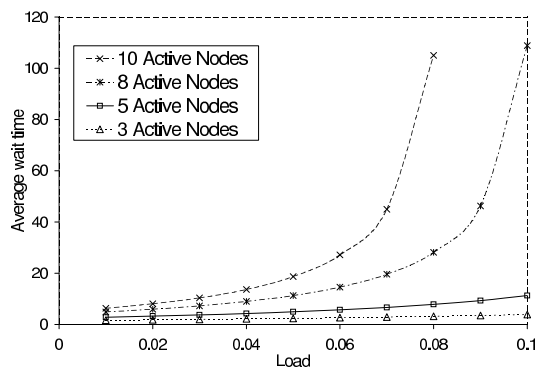


Figure 7.5. Wait time vs. number of active nodes in the ring. Total nodes = 32.

ring increases. Figure 7.5 shows the variation in wait time as the number of active nodes in the ring is varied. As expected, with increase in the number of nodes that have data to send, the wait time increases.

From figures 7.4 and 7.5, the trade off between number of nodes in the ring and anonymity degree is clear. For time sensitive data which require low latencies, rings with less number of nodes have to be formed which in turn results in lesser communication overhead and at the same time lesser anonymity.

7.5 Summary

The data-centric behavior of wireless networks leaves them vulnerable to traffic analysis and identification of event locations and active areas. Therefore, ensuring data anonymity is a crucial research area. We presented Hierarchical Anonymous Communication Protocol (HACP) to achieve anonymous communications in

a wireless network. We divide the network into rings and use the concept of tokens and rings to achieve anonymity.

We also present the tradeoffs between the overhead imposed and ring sizes. We show that higher anonymity comes at a cost - either higher communication/energy overhead or at higher latency. The choice of the parameters is left to the network administrator and depends on level of security needed and the type of traffic in the network.

Chapter 8

Lightweight Data Integrity Protocol for Wireless Networks

There is one safeguard known generally to the wise, which is an advantage and security to all...What is it? Distrust.

- *Demosthenes* (384-322 BC)

Heterogeneous wireless networks have recently emerged as a critically important disruptive technology resulting from the fusion of wireless communications and embedded computing technologies [125, 126, 98, 99, 100, 101]. Potential applications include monitoring remote or inhospitable locations, target tracking in battlefields, disaster relief networks, early fire detection in forests, and environmental monitoring.

Security is a crucial part of the architectures for these wireless networks. Wireless networks are vulnerable to a vast number of security threats [127, 128, 129] with variable application-specific attack mechanisms and variable impact on the network. Due to their nature and operational resource constraints wireless networks are vulnerable to various types of attacks. While designing the new network architecture for future wireless networks, the research community has a unique chance to integrate security and privacy since the beginning as a fundamental part of the architecture. As shown by the Internet example, security cannot be implemented properly as patches to an existing network architecture, rather security mechanisms must be developed as part of an integral security framework.

Wireless networks, in general, are more vulnerable to security attacks than wired networks, due to the broadcast nature of the transmission medium. Furthermore, wireless networks have an additional vulnerability because nodes are often placed in a hostile or dangerous environment where they are not physically protected. Note that security issues in ad-hoc networks are similar to those in sensor networks and have been well enumerated in the literature [130], but the defense mechanisms developed for ad-hoc networks are not directly applicable to sensor networks. For example, some ad-hoc network security mechanisms for authentication and secure

routing are based on public key cryptography [128, 131, 132, 133, 134] which is too expensive for sensor nodes. Similarly, security solutions for ad-hoc networks based on symmetric key cryptography have been proposed [135, 136, 137]. They are too expensive in terms of node state overhead and are designed to find and establish routes between any pair of nodes—a mode of communication not prevalent in sensor networks. The authors in [138, 139] consider the problem of minimizing the effect of misbehaving or selfish nodes through punishment, reporting, and holding grudges. The application of these techniques to sensor networks is promising, but these protocols are vulnerable to blackmailers.

There are several recent research efforts exploring different aspects of wireless network security, for example key management, secure multicast communication, authentication and anonymous routing [140]. Among the original sensor network security solutions, SPINS [141] presents two building block security mechanisms for use in sensor networks, SNEP and μ -TESLA. SNEP provides confidentiality and authentication between nodes and the sink, and μ -TESLA provides authenticated broadcast.

Ad hoc and Sensor networks are expected to consist of hundreds to thousand of nodes dispersed in hostile environments. It is clearly impractical to monitor and protect each individual node from physical or logical attack. An enemy can easily alter existing data or even inject spurious data in the wireless network by capturing or insert new malicious nodes into the network. A key technical challenge is to detect such activity by distinguishing fake/altered data from the correct one and identifying the malicious nodes. In data-centric wireless networks, data is typically aggregated for energy-efficiency [142]. Since wireless networks are highly unstructured, both routing and aggregation of data occurs in an ad-hoc manner depending on current resource distributions and current (localized) sensing activity. It is therefore extremely difficult to identify vulnerable nodes/network zones a priori. Therefore there is a need to develop a broad spectrum of dynamic defense mechanisms for detecting such malicious behavior.

We propose a new lightweight security protocol to provide data integrity for ad hoc and sensor networks. Data integrity is the assurance that the data received by the destination is the same as generated by the source. Data Integrity ensures that

data is unchanged from its source and has not been accidentally or maliciously altered. Integrity attacks modify content without the knowledge or permission of the owner. The key advantages of the protocol are: 1) The protocol is simple; 2) it needs very few bits in the header (as low as three bits). This results in negligible bandwidth overhead; 3) the protocol poses very less computational overhead (it needs to compute just a hash as compared to multiple complex operations required by any cryptographic implementation for verifying authenticity).

The rest of the chapter is organized as follows: Section 8.1 describes current network security trends specially for data integrity. Section 8.2 describes our protocol for providing integrity. In Section 8.3 we present the communication and computational overhead imposed by our protocol and analyze the performance.

8.1 Related Work

Data integrity is the assurance that the data received by the destination is the same as that generated by the source and has not been accidentally or maliciously altered en route. Integrity attacks modify content without the knowledge or permission of the owner.

The security community has paid vast attention to confidentiality issues, which are solved through encryption of data transmissions such as email or encrypting files in storage. While encryption has been possible for decades, this security technique lags in implementation in wireless networks due to complex key management and low processing and memory capabilities of these networks. Asymmetric cryptographic techniques might not be possible at all in sensor networks [141]. Symmetric cryptographic techniques though implementable, still consume lot of energy. The issue of denial of service attacks began to be solved through better intrusion detection, high-speed reaction mechanisms, redundancy, fault tolerance, better disaster planning and system reconstitution.

Integrity mechanisms have been part of the computer security professional's arsenal in many forms. The simplest method is called CRC or a Cyclic Redundancy Check. The contents of the file are XORed with another set of (random) data and the results create an integrity key. When the reverse CRC process is run, and if the integrity key does not match the original, the file has been corrupted in some form and cannot be trusted.

A stronger integrity method is called Message Authentication Code (MAC) [143], a cryptographic technique that is based on the Data Encryption Standard. Again, a key is generated when the file is 'sealed'. Upon decoding, the key must match if the files are to be trusted. MAC is based on a secret key shared between the communicating parties, i.e., source and destination. Key distribution in wireless networks is an on going research issue [106, 144, 9].

Though, cryptographic techniques can ensure complete security, they are very computational intensive and consume lot of energy. Moreover, in many scenarios, all security issues need not be addressed. For instance, consider a wireless network deployed for intrusion detection. Once a wireless node S detects an intruder, it sends an alert message to the base station. Encrypting the alert message need not essentially prevent someone from realizing the contents of the message itself. For another scenario, consider a sensor network deployed to detect fires by monitoring the temperature. Once the base station gets a packet from a sensor that has a temperature reading greater than some threshold, a warning might be issued. In these cases, it is more important that message integrity is ensured than message secrecy.

8.2 Data Integrity-Lightweight Network Layer Security

We present a lightweight algorithm to preserve the integrity of messages in a wireless network even in presence of compromised nodes. Our protocol prevents compromised nodes from changing the contents of a packet. Our mechanism can be used upon any other security protocol with slight modifications. Our mechanism can be modified to work even with as low as three bits. Even with just three bits in header, a compromised node could send only a few packets (less than 10 packets in 99.9% of cases) before being detected.

Our schemes add little or no overhead to the node's critical forwarding path. In fact, the only invariant that we can depend on is that a packet from the attacker must traverse all of the nodes between it and the victim.

8.2.1 Assumptions

We assume a wireless network that is logically represented as a set of clusters. Several protocols have been proposed to efficiently divide the network into clusters and elect cluster heads [2, 96, 120]. The cluster heads form a d -hop dominating

TABLE 8.1. Header space allocation for different fields for authentication purpose

ONAF	TNAF	Flag
4 bits	4 bits	1 bit

set. A node either becomes a cluster head, or is at most d hops from a cluster head. Cluster heads form a virtual backbone and may be used to route packets for nodes in their cluster. The value of d is a parameter of the network. We assume that multiple malicious nodes might be present, but the nodes do not collaborate.

8.2.2 The Protocol

We propose an effective mechanism to prevent compromised router nodes from modifying the contents of a packet. Our mechanism can work even with as low as three bits as we illustrate later in this section. With just three bits, a compromised router could send few packets (less than 10 packets in 99.9% of cases) before being detected. We first present a generalized version of our mechanism in which we assume the available header space to be $2t + 1$ bits. Later in the section, we examine the different choices of t .

We divide the $(2t + 1)$ -bit header space into three fields a t -bit One-hop Neighbor Authentication Field (ONAF), a t -bit Two-hop Neighbor Authentication field (TNAF) and a 1-bit flag field as shown in Table 8.1.

Our scheme is based on a lightweight strategy. We define for each cluster head x the set $N(x)$, which contains the nodes in G that are neighboring cluster heads of x (which does not include the x itself). That is:

$$N(x) = \{y : (x, y) \in E \text{ and } y \neq x\} \quad (8.1)$$

The security of our scheme is derived from a secret key $k(x)$ that is shared by all the cluster heads in $N(x)$, but not by x itself. This key is created in a setup phase and distributed securely to all the members of $N(x)$. Note, in addition, that $y \in N(x)$ if and only if $x \in N(y)$.

When a cluster head s wishes to send a packet P to be forwarded by a neighboring cluster head, x , it sets the above fields as follows:

$$ONAF = h[P, k(x)]$$

$$Flag = 0$$

where h is a cryptographic keyed-hash function that is collision resistant using a key. s and d are the source and destination addresses present in the packet P and the T is the marking in the packet used for traceback and set by the underlying traceback mechanism being used. When the cluster head x receives a packet P from one of its neighbors y , it verifies the authenticity of the packet as follows:

Node x computes $h[P, k(y)]$ and compares it with $TNAF$.

- If they are the same, then x does the following operations:

$$TNAF \leftarrow ONAF$$

$$ONAF \leftarrow h[P, k(z)]$$

$$Flag = 1$$

z being the cluster head to which the packet is being forwarded by x

Setting of $Flag$ is to indicate that it is not the originator of the packet.

- If the values are different, and if the $Flag$ is not set to 0 then immediately x can decide that y has been compromised.
- If $Flag$ is set to 0, then x definitely marks the packet.

The protocol follows a leap frog approach. Each cluster head verifies if the packet was modified by previous node by checking the hash value of the packet generated by the *up tree* node that is two hops far away from it. If the verification fails, the previous node has either originated the packet (which is indicated by the flag) or has modified the packet.

8.3 Analysis and Results

In this section, we analyze the overhead (bandwidth and computational) and the performance of our protocol.

8.3.1 Bandwidth Overhead

We present simulation results regarding the number of header bits the protocol needs. A node malicious y can successfully escape from being detected with a probability of $\frac{1}{2^t}$. When $t = 4$, this probability is $\frac{1}{16}$, and the node will be discovered with a probability of $\frac{15}{16}$. The probability of y passing this test for more than three packets is less than 0.00025. That is, in more than 99.97% of cases, y will be

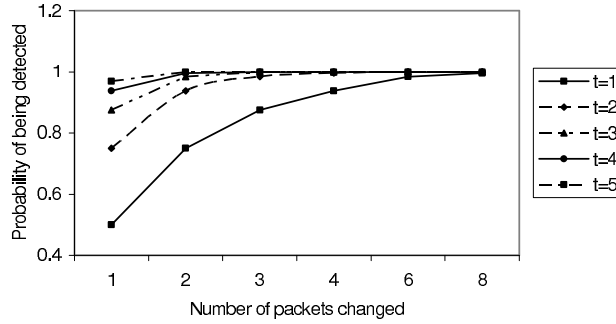


Figure 8.1. Probability that a node can change p

discovered even before it could modify three packets. To generalize, the probability that a node can change and send p packets without being detected is $(\frac{1}{2^t})^p$. Figure 8.1 illustrates this for different values of t and p . It should be noted that even with $t = 1$, 99.6% of times, a malicious node will be detected even before it can change 8 packets. In Figure 2 it is shown the required size of the header field (t) so as to detect a malicious node before modifying a given number of packets with a given probability. It could be noted that even with a modest total header space of 5 bits, a node would be detected even before it is able to modify four packets.

Once a cluster head discovers that one of its neighbors is compromised, it can report it to the base station for further action and also broadcast the entire network alerting all nodes about this node.

8.3.2 Computational Overhead

As explained earlier, our mechanism is quite effective even with just three bits. The hash function generates a keyed hash value on source and destination addresses and all other fields being marked by the underlying mechanism. Thus, the hash function operates over an input of 70 bits apart from the key itself and at each node at most two such hash values will be generated for each packet. Hence, computing the hash values hardly poses any processing or computational overhead on the node.

8.4 Summary

In this chapter, we presented a novel lightweight strategy to ensure data integrity. Our protocol is based on leap-frog strategy in which each cluster head verifies if its previous node has preserved the integrity of the packet using the secret key it shares with two hop *up tree* node. The key advantages of the protocol

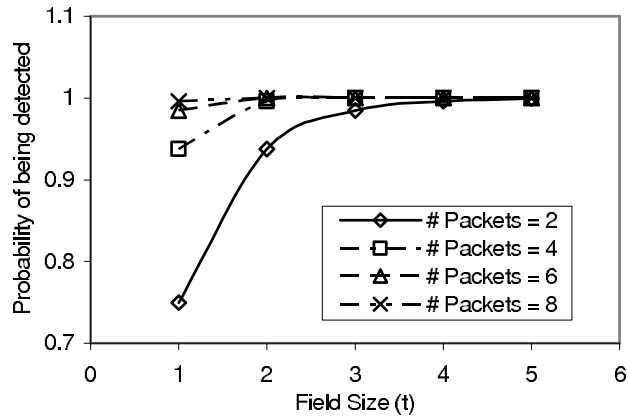


Figure 8.2. Probability of detecting a malicious node for a given number packets

include: the protocol is simple; it needs very few header bits, as low as three bits, thus resulting in negligible bandwidth overhead; the protocol poses very low computational overhead, it needs to compute just a hash as compared to multiple complex operations required by any cryptographic implementation for verifying authenticity. We also discussed the performance of the protocol.

Chapter 9

Conclusions

I not only use all the brains that I have, but all that I can borrow.
- *Woodrow Wilson* (1856 - 1924)

This dissertation proposes analytical models to study the impact of collisions and interference in heterogeneous wireless networks and simple scalable and lightweight protocols that use these models to adapt to network conditions thus increasing efficiency, decreasing energy consumption and prolonging network lifetime.

Several works have proposed protocols for addressing various issues of heterogeneous wireless networks, but few works have considered adapting the protocols to the prevailing network conditions and the trade offs between delay, reliability and energy consumption, simultaneously. The work described in this dissertation has developed analytical models to enable protocols to adapt to the network conditions and demonstrated the advantages of adapting by designing and evaluating protocol for broadcasting, routing, backbone formation and clustering.

Analytical characterization of the impact of collisions/interference on both broadcast and unicast messages is presented. The models can be used to study the trade off between performance and energy consumption and enables the network administrator to choose the best transmission ranges based on the network requirements.

The Optimized Flooding Protocol presented for efficient and reliable network-wide broadcasting is a geometric approach and thus inherits the nice features like scalability, immunity towards non-circular propagation and mobility, and no communication overhead. By adapting the transmission range (either locally or globally) to the network conditions, OFP can ensure the required reliability criteria. OFP can also be extended to three dimensional networks for equally efficient performance.

Adaptive Routing and Energy Management is based on simple techniques - random wakeup and forwarding set based routing, but clearly outperforms previous works both in terms of end-to-end latency and energy consumption. AREM ben-

efits from the analytical framework and balances the load across all nodes in the network thus further prolonging the network lifetime.

Efficient Coordination Protocol is also based on a geometric approach and incorporates adaptation techniques developed by the analytical model to form small backbones, each backbone node serving varied number of nodes so that all backbone nodes have similar loads.

Adaptive Clustering Protocol is a simple, lightweight protocol that minimizes the number of clusters formed. Again, the clusters are formed in such a way that the number of packets generated from each cluster is uniform. Thus, instead of number of nodes or size of the cluster, the load on the cluster is the criteria which not only increases the efficiency but also prolongs the network lifetime.

This dissertation also proposes two protocols to address anonymous communication and lightweight data integrity. The administrators have the option of selecting the parameters so as to achieve an optimal trade off between security and overhead based on network requirements.

To summarize, this dissertation presents works in the following directions: adaptability to different existing network scenarios and the ability to tune the protocols to attain specific performance based on network requirements.

References

- [1] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences*, 2000, p. 8020.
- [2] S. Bandyopadhyay and E. Coyle, “An energy efficient hierarchical clustering algorithm for wireless sensor networks,” in *Proceedings of IEEE INFOCOM*, March 30 - April 3, 2003.
- [3] M. Demirbas and H. Ferhatosmanoglu, “Peer-to-peer spatial queries in sensor networks,” in *P2P '03: Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, 2003, p. 32.
- [4] S. Banerjee and S. Khuller, “A clustering scheme for hierarchical control in multi-hop wireless networks,” in *INFOCOM*, 2001, pp. 1028–1037.
- [5] C. R. Lin and M. Gerla, “Adaptive clustering for mobile wireless networks,” *IEEE Journal of Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.
- [6] W. Chen, J. Hou, and L. Sha, “Dynamic clustering for acoustic target tracking in wireless sensor networks,” in *11th IEEE International Conference on Network Protocols*, 2003.
- [7] G. Jolly and M. Younis, “An energy-efficient, scalable and collision-free mac layer protocol for wireless sensor networks,” in *Wireless Communications and Mobile Computing*, vol. 5, no. 3, 2005, pp. 285–304.
- [8] H. Chan and A. Perrig, “ACE: An emergent algorithm for highly uniform cluster formation,” in *Proceedings of the First European Workshop on Sensor Networks (EWSN)*, Jan. 2004.
- [9] S. Zhu, S. Setia, and S. Jajodia, “Leap: Efficient security mechanisms for large-scale distributed sensor networks,” in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, October 2003.
- [10] “Privacy international,” www.privacyinternational.org.
- [11] M. Gruteser and D. Grunwald, “Anonymous usage of location-based services through spatial and temporal cloaking.” in *MobiSys*, 2003.
- [12] A. Pfitzmann and M. Hansen, “Anonymity, Unobservability, and Pseudonymity: A Consolidated Proposal for Terminology,” in *Proceedings*

of *International Workshop on Design Issues in Anonymity and Unobservability*, July 2000.

- [13] G. D., R. M., and S. P., “Onion routing for anonymous and private Internet connections,” in *Communications of the ACM*, vol. 2, no. 42, February 1999, pp. 39–41.
- [14] A. Beimel and S. Dolev, “Buses for Anonymous Message Delivery,” *Journal of Cryptology*, vol. 16, no. 1, pp. 25–39, 2003.
- [15] M. K. Reiter and A. D. Rubin, “Anonymous Web transactions with crowds,” *Communications of the ACM*, vol. 42, no. 2, pp. 32–48, 1999.
- [16] H. Takagi and L. Kleinrock, “Optimal transmission ranges for randomly distributed packet radio terminals,” *IEEE Trans. on Communications*, vol. 32, no. 3, pp. 246–257, March 1984.
- [17] T.-C. Hou and V. O. K. Li, “Transmission range control in multihop packet radio networks,” *IEEE Trans. on Communications*, vol. 34, no. 1, pp. 38–44, Jan 1986.
- [18] R. Ramanathan and R. Hain, “Topology control of multihop wireless networks using transmit power adjustment.” in *INFOCOM*, 2000, pp. 404–413.
- [19] P. Gupta and P. Kumar, “Critical power for asymptotic connectivity,” *Proceedings of Conference on Decision and Control, Tampa, USA, 1998.*, 1998.
- [20] R. Ramanathan and R. Hain, “Topology control of multihop wireless networks using transmit power adjustment,” in *Proceedings of IEEE INFOCOM*, 2000, pp. 404–413.
- [21] S. Gabriel, R. Melhem, and D. Mossé, “A unified interference/collision analysis for power-aware adhoc networks,” in *Proceedings of IEEE INFOCOM*, Mar. 2004.
- [22] X. Li, T. D. Nguyen, and R. P. Martin, “An analytic model predicting the optimal range for maximizing 1-hop broadcast coverage in dense wireless networks,” in *ADHOC-NOW*, 2004, pp. 172–182.
- [23] G. V. Z. D. Myers and V. R. Syrotiuk, “An Adaptive Generalized Transmission Protocol for Mobile Ad Hoc Networks,” *Mobile Networking and Applications (MONET)*, vol. 7, no. 6, pp. 493–502, 2002.
- [24] G. Bianchi, “Performance analysis of the ieee 802.11 distributed coordination function,” *IEEE Journal on Selected Areas of Communications*, vol. 3, no. 18, pp. 535–547, March 2000.

- [25] M. M. Carvalho and J. J. Garcia-Luna-Aceves, "Delay analysis of ieee 802.11 in single-hop networks," in *ICNP '03: Proceedings of the 11th IEEE International Conference on Network Protocols*, 2003.
- [26] E. Poon and B. Li, "Smartnode: Achieving 802.11 mac interoperability in power-efficient ad hoc networks with dynamic range adjustments." in *ICDCS*, 2003, pp. 650–657.
- [27] X. Li, T. D. Nguyen, and R. P. Martin, "Using adaptive range control to maximize 1-hop broadcast coverage in dense wireless networks," in *Proceedings of IEEE Conference on Sensor and Ad Hoc Communication Networks (SECON)*, October 2004.
- [28] J. Cartigny, D. Simplot, and I. Stojmenović, "Localized minimum-energy broadcasting in ad-hoc networks," in *Proceedings of IEEE INFOCOM'2003*, 2003.
- [29] M. Agarwal, L. Gao, J. H. Cho, and J. Wu, "Energy efficient broadcast in wireless ad hoc networks with hitch-hiking." *MONET*, vol. 10, no. 6, pp. 897–910, 2005.
- [30] X.-Y. Li and P.-J. Wan, "Constructing minimum energy mobile wireless networks," in *ACM Symposium on Mobile Ad Hoc Networking and Computing MobiHoc*, 2001.
- [31] V. Rodoplu and T. Meng, "Minimum energy mobile wireless networks," in *Proceedings of the IEEE International Conference on Communications(ICC)*, vol. 3, 1998.
- [32] R. Kershner, "The number of circles covering a set," *Amer. J. Math*, no. 61, 1939.
- [33] S. Y. Ni and et. al., "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of ACM MOBICOM*, August 1999, pp. 151–162.
- [34] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," in *Proceedings of European Symposium on Algorithms (ESA)*, 1996.
- [35] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, "A lower bound for radio broadcast," *J. Comput. Syst. Sci.*, vol. 43, pp. 290–298, October 1991.
- [36] Gaber and Y. Mansour, "Broadcast in radio networks," in *Proceedings of 6th Annu. ACM-SIAM Symp. Discrete Algorithms*, January 1995, pp. 577–585.
- [37] W. Peng and X. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proceedings of MOBIHOC*, 2000.

- [38] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc net-works," in *Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, 2000.
- [39] Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," in *Technical Report 3898, INRIA - Rapport de recherche*, 2000.
- [40] W. Peng and X. Lu, "AHBP: An efficient broadcast protocol for mobile ad hoc net-works," *Journal of Science and Technology*, 2002.
- [41] J. Sucec and I. Marsic, "An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks," in *CAIP Technical Report 248 - Rutgers University*, September 2000.
- [42] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of the third ACM international symposium on Mobile ad hoc networking & computing*, June 2002.
- [43] M. Sun and T. Lai, "Location aided broadcast in wireless ad hoc network systems," in *IEEE WCNC 2002*, March 2002, pp. 597–602.
- [44] J. Wu and H. Li, "On calculating connected dominating sets for efficient routing in ad hoc wireless networks," in *Proceedings of the International Workshop on Discrete Algorithms and methods for Mobile Computing and Communication (DIAL-M)*, 1999, pp. 7–14.
- [45] J. Wu and F. Dai, "Broadcasting in ad hoc networks based on self-pruning," in *In Proceedings of IEEE INFOCOM 2003*, September 2001, pp. 7–14.
- [46] Haas and L. Halpern, "Gossip based ad hoc routing," in *Proceedings of IEEE INFOCOM*, June 2002.
- [47] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor network," in *Proceedings of 5th ACM/IEEE Mobicom Conference (MobiCom '99)*, August 1999, pp. 174–185.
- [48] S. Tilak, A. Murphy, and W. Heinzelman, "Non-uniform information dissemination for sensor networks," in *Proceedings of the 11th International Conference on Network Protocols ICNP'03*, November 2003.
- [49] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and ro-bust communication paradigm for sensor networks," in *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking MobiCOM '00*, August 2000.

- [50] S. Dolev, T. Herman, L. Lahiani, and Ben-Gurion, “Polygonal broadcast for sensor networks,” in *2nd IEEE Upstate New York Workshop on Sensor Networks*.
- [51] W. Lou and J. Wu, “Double-covered broadcast (DCB): A simple reliable broadcast algorithm in manets,” in *Proceedings of IEEE INFOCOM 2004*, 2004.
- [52] P. Rogers and N. Abu-Ghazaleh, “Towards Reliable Network Wide Broadcast in Mobile Ad Hoc Networks,” *ArXiv Computer Science e-prints*, Dec. 2004.
- [53] P. Rogers and N. Abu-Ghazaleh, “Selective additional rebroadcast: An approach for robustness control for network wide broadcast.”
- [54] M. Mohsin, D. Cavin, Y. Sasson, R. Prakash, , and A. Schiper, “Reliable broadcast in wireless mobile ad hoc networks,” in *Proceedings of the Thirty-Ninth Hawaii International Conference on System Sciences (HICSS’06)*, Jan. 2006.
- [55] ns 2, “Network simulator.”
- [56] T. Camp, J. Boleng, and V. Davies, “A Survey of Mobility Models for Ad Hoc Network Research,” *Wireless Communication and Mobile Computing (WCMC)*, vol. 2, no. 5, pp. 483–502, 2002.
- [57] J. Conway and N. Sloane, “Sphere packings, lattices and groups.”
- [58] J. K. F.C. Frank, “Acta cryst.” 11, 184 (1958).
- [59] ———, “Acta cryst.” 12, 483 (1959).
- [60] H. Steinhaus, “Mathematical snapshots,” pp. 185–190, 1999.
- [61] P. Kunkel, “Whistler alley - the octahedron,” in <http://whistleralley.com/poly-hedra/octahedron.htm>.
- [62] “Wireless lan medium access control (MAC) and physical layer (PHY) specifications,” *IEEE Standard 802.11*, June 1999.
- [63] R. Zheng, J. Hou, and L. Sha, “Asynchronous wakeup for ad hoc networks,” *The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 03)*, January 2003.
- [64] C. S. Raghavendra and S. Singh, “PAMAS-power aware multi-access protocol with signaling for ad hoc networ,” *Computer Communication Reviews*, 1998.
- [65] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, “Optimizing sensor networks in the energy-latency-density design space,” *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, January-March 2002.

- [66] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in *Proceedings of IEEE Infocom*, June 2002, pp. 1567–1576.
- [67] T. van Dam and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks,” *ACM Sensys*, November 2002.
- [68] M. Zorzi and R. Rao, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance.” *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, 2003.
- [69] —, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Energy and latency performance,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 349–365, 2003.
- [70] J. Monks, V. Bharghavan, and W. mei W. Hwu, “A power controlled multiple access protocol for wireless packet networks,” in *Proceedings of IEEE INFOCOM*, 2001, pp. 219–228.
- [71] D. Qiao, S. Choi, A. Jain, and K. G. Shin, “MiSer: an optimal low-energy transmission strategy for ieee 802.11a/h.” in *MOBICOM*, 2003, pp. 161–175.
- [72] N. Bambos, S. C. Chen, and G. J. Pottie, “Radio link admission algorithms for wireless networks with power control and active link quality protection,” in *Proceedings of IEEE INFOCOM*, 1995, pp. 97–104.
- [73] S. Ulukus and R. Yates, “Stochastic power control for cellular radio systems,” *IEEE Transactions on Communications*, pp. 784–798, 1998.
- [74] Q. Fang, J. Gao, and L. J. Guibas, “An energy efficient MAC protocol for wireless LANs,” *Proceedings of IEEE Infocom 2004*, June 2004.
- [75] “Crossbow MPR/MIB mote hardware users manual,” www.xbow.com/Support/manuals.htm.
- [76] A. Durrezi, V. Parachuri, S. Iyengar, and R. Kannan, “Optimized Broadcast Protocol for Sensor Networks,” *IEEE Transaction on Computing*, vol. 54, no. 8, pp. 1013–1024, August 2005.
- [77] B. Das, R. Sivakumar, and V. Bharghvan, “Routing in ad- hoc networks using a minimum connected dominating sets,” in *Proceedings of IEEE International Conference on Communications*, 1997.
- [78] I. Stojmenovic, M. Seddigh, and J. Zunic, “Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks,” in *Proceedings of IEEE Hawaii International Conference on System Sciences*, jan 2001.

- [79] P. Wan, K. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proceedings of IEEE INFOCOM*, 2002.
- [80] P. Sinha, R. Sivakumar, and V. Bharghavan, "Enhancing ad hoc routing with dynamic virtual infrastructures," in *Proceedings of IEEE INFOCOM*, 2001, pp. 1763–1772.
- [81] J. Wu and W. Lou, "Forward-node-set-based broadcast in clustered mobile ad hoc networks." *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 155–173, 2003.
- [82] B. Gao, Y. Yang, and H. Ma, "An effective distributed approximation algorithm for constructing minimum connected dominating set in wireless ad hoc networks," *Fourth International Conference on Computer and Information Technology*, pp. 658–663, 2004.
- [83] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "SPAN: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *The Seventh ACM MOBICOM Rome, Italy*, July 2001.
- [84] Y. Xu, J. S. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Mobile Computing and Networking*, 2001, pp. 70–84.
- [85] R. Kravets and P. Krishnan, "Application-driven power management for mobile communication," *Wireless Networks*, vol. 6, no. 4, pp. 263–277, 2000.
- [86] H. Liu and R. Gupta, "Selective backbone construction for topology control," in *Proceedings of the First IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, October 2004.
- [87] H. Ju, I. Rubin, K. Ni, and C. Wu, "A distributed mobile backbone formation algorithm for wireless ad hoc networks," *Proceedings of BroadNets*, vol. 00, pp. 661–670, 2004.
- [88] G. Dommety and R. Jain, "Potential networking applications of global positioning systems (GPS)," in *Technical Report, TR-24, The Ohio State University*, April 1996.
- [89] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic Fine-Grained Localization in Ad Hoc Networks of Sensors," in *ACM MOBICOM'01, Rome, Italy*, July 16-21 2001.
- [90] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AoA," in *IEEE INFOCOM'03*, April 1 - 3 2003.

- [91] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," in *IEEE INFOCOM'00*, March 26 - 30 2000.
- [92] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Mobile Computing and Networking*, 2000, pp. 243–254.
- [93] M. Chatterjee, S. Das, and D. Turgut, "Wca: A weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, 2002.
- [94] S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAAN)*, 1999, p. 310.
- [95] A. D. Amis, R. Prakash, D. Huynh, and T. Vuong, "Max-min d -Cluster Formation in Wireless Ad Hoc Networks," in *INFOCOM*, 2000, pp. 32–41.
- [96] M. Handy, M. Haase, and D. Timmermann, "Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection," in *IEEE International Conference on Mobile and Wireless Communications Networks*, Stockholm, 2002.
- [97] H. Zhang and A. Arora, "Gs3: scalable self-configuration and self-healing in wireless networks," in *PODC '02: Proceedings of the twenty-first annual symposium on Principles of distributed computing*, 2002, pp. 58–67.
- [98] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. F. Knight, R. Nagpal, E. Rauch, G. J. Sussman, and R. Weiss, "Embedding the Internet: amorphous computing," *Communications of ACM*, vol. 43, no. 5, pp. 74–74, May 2000.
- [99] G. Borriello and R. Want, "Embedding the Internet: embedded computation meets the World Wide Web," *Communications of ACM*, vol. 43, no. 5, pp. 59–59, may 2000.
- [100] G. J. Pottie and W. J. Kaiser, "Embedding the Internet: wireless integrated network sensors," *Communications of ACM*, vol. 43, no. 5, pp. 51–51, may 2000.
- [101] G. S. Sukhatme and M. J. Mataric, "Embedding the Internet: embedding robots into the Internet," *Communications of ACM*.
- [102] P. K. Dutta, "Security Considerations in Wireless Sensor Networks," *Sensors Expo*.
- [103] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," in *First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003, pp. 113–127.

- [104] S. Slijepcevic, M. Potkonjak, V. Tsiatsis, S. Zimbeck, and M. B. Srivastava, "On Communication Security in Wireless Ad-Hoc Sensor Networks," in *WETICE '02: Proceedings of the 11th IEEE International Workshops on Enabling Technologies*, 2002, pp. 139–144.
- [105] H.Chan, A.Perrig, and D.Song, "Random key pre-distribution schemes for sensor networks," in *Proceedings of IEEE Symposium on Security and Privacy*, 2003.
- [106] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, October 2003.
- [107] R. di Pietro, L. V. Mancini, Y. W. Law, S. Etalle, and P. Havinga, "A directed diffusion-based secure multicast scheme for wireless sensor networks," in *First International Workshop on Wireless Security and Privacy (WiSpr'03)*, 2003.
- [108] M. Bohge and W. Trappe, "An authentication framework for hierarchical ad hoc sensor networks," in *WiSe '03: Proceedings of the 2003 ACM workshop on Wireless security*, 2003, pp. 79–87.
- [109] D. Balfanz, D. Smetters, P. Stewart, and H. Wong, "Talking to strangers: Authentication in adhoc wireless networks," Feb. 2002, In Symposium on Network and Distributed Systems Security (NDSS '02), San Diego, California.
- [110] D. Liu and P. Ning, "Multilevel *µtesla*: Broadcast authentication for distributed sensor networks," *Transactions on Embedded Computing Systems*, vol. 3, no. 4, pp. 800–836, 2004.
- [111] S. Capkun, J. P. Hubaux, and M. Jakobsson, "Secure and privacy-preserving communication," in *Hybrid Ad Hoc Networks, EPFL-IC Technical report no. IC/2004/10*, January 2004.
- [112] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 46–55, April 2003.
- [113] D. G. Marco Gruteser, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the First International Conference on Mobile Systems, Applications, and Services*, May 2003.
- [114] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald, "Privacy-aware location sensor networks," in *USENIX 9th Workshop on Hot Topics in Operating Systems (HOTOS IX) 2003*, 2003, pp. 163–167.
- [115] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications ACM*, vol. 24, no. 2, pp. 84–90, 1981.

- [116] J. Al-Muhtadi, R. Campbell, A. Kapadia, M. D. Mickunas, and S. Yi, "Routing through the Mist: Privacy Preserving Communications in Ubiquitous Computing Environments," in *Proceedings of International Conference of Distributed Computing Systems (ICDCS 2002)*, July 2002.
- [117] A. Smailagic, "Location sensing and privacy in a context aware computing environment," in *Proceedings of Pervasive Computing*, 2001.
- [118] J. Kong and X. Hong, "ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, 2003.
- [119] M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in clustering-based sensor networks," in *Proceedings the 10th IEEE/ACM Sym. on Modeling, Analysis and Simulations of Computer and Telecom. Systems*, October 2002.
- [120] A. Durresi and V. Paruchuri, "Adaptive clustering for sensor networks," in *Accepted in 2005 IEEE Aerospace Conference*, Big Sky, Montana, March 6-15 2005.
- [121] R. Kalidindi, V.Parachuri, R.Kannan, A. Durresi, and S.Iyengar, "Subquorum based key vector assignment: A key pre-distribution scheme for wireless sensor networks," in *Proceedings of International Conference On Wireless Networking, (ICWN04)*, July 2004, pp. 440–446.
- [122] C. Díaz and B. Preneel, "Reasoning about the anonymity provided by pool mixes that generate dummy traffic." in *Information Hiding*, 2004, pp. 309–325.
- [123] E. H. Callaway, *Wireless Sensor Networks – Architectures and Protocols*. Boca Raton, Florida: Auerbach, 2003.
- [124] A. Leon-Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*. McGraw Hill, 2000.
- [125] I. F. Akyldiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.
- [126] D. Estrin and R. Govindan, "Next century challenges: Scalable coordination in sensor networks," *Proceedings of ACM/IEEE Mobicom'99*, pp. 263–270, August 1999.
- [127] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, 2002.

- [128] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.
- [129] "NAI lab," www.nai.com/nai_labs/asp_set/crypto/crypt_senseit.asp.
- [130] F. Stajano and R. J. Anderson, "The resurrecting duckling: Security issues for ad-hoc wireless networks," in *Proceedings of the 7th International Workshop on Security Protocols*, 2000, pp. 172–194.
- [131] J.-P. Hubaux, L. Butty, and S. Capkun, "The quest for security in mobile ad hoc networks," in *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, 2001, pp. 146–155.
- [132] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," in *International Conference on Network Protocols (ICNP)*, 2001, pp. 251–260.
- [133] M. G. Zapata, "Secure ad hoc on-demand distance vector routing," *SIG-MOBILE Mobile Computure Communications Review*, vol. 6, no. 3, pp. 106–107, 2002.
- [134] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing ad hoc wireless networks," *ISCC*, vol. 00, p. 567, 2002.
- [135] Y.-C. Hu, D. B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," *WMCSA*, vol. 00, p. 3, 2002.
- [136] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti, "Secure pebblenets," in *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, 2001, pp. 156–163.
- [137] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," in *Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, 2002.
- [138] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Mobile Computing and Networking*, 2000, pp. 255–265.
- [139] S. Buchegger and J. L. Boudec, "Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks," in *10th Euromicro Workshop on Parallel, Distributed and Network-based Processing (PDP), January 2002*, pp. 403–410.
- [140] J. Kong, X. Hong, and M. Gerla, "An anonymous on demand routing protocol with untraceable routes for mobile ad-hoc networks," in *UCLA Computer Science Department TechnicalReport 030020*.

- [141] A.Perrig, R.Szewczyk, V. Wen, D.Culler, and J. Tygar, "Spins: Security protocols for sensor networks," in *Proceedings of 7th International Conference on Mobile Computing and Networks*, July 2001.
- [142] B. Krishnamachari, D. Estrin, and S. Wicker, "Modeling Data-Centric Routing in Wireless Sensor Networks," in *Wireless Sensor Network Applications WSNA*, June 2002.
- [143] H. Krawczyk, M. Bellare, and R. Canetti, "RFC 2104: HMAC: Keyed-hashing for message authentication," Feb. 1997.
- [144] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," in *Proceedings of the 10th ACM conference on Computer and communication security*, October 27-30 2003.

Vita

Vamsi Krishna Paruchuri was born on May 15, 1980, in Renigunta, India. He received his Bachelor of Science in Electronics and Communications Engineering degree from Sri Venkateswara University, Tirupati, India, in 2001. He received his Master of Science in Electrical Engineering degree from the Ohio State University, Columbus, Ohio, in 2003. He attended the doctoral program at the Department of Computer Science at the Louisiana State University, Baton Rouge, Louisiana, from August 2003 to August 2006. His research has focused on heterogeneous wireless networks and network security, including design, analysis, implementation, and experimentation of protocols.