## ABSTRACT

BHATT, BHUSHAN. Design and Implementation of a Distributed Scheduling Algorithm using Period Inflation for Sensor Networks. (Under the direction of Dr. Rudra Dutta).

Wireless Sensor Networks (WSNs) are fast emerging as a new and ubiquitous networking arena which will enables many new applications and pervades many old ones. One of the motivations for the development of WSNs is their ability to be deployed in any environment in a comparatively ad-hoc manner. The most important challenge faced by WSNs is battery-limited lifetime of the network. Physically replacing batteries is infeasible in most real-life deployments of WSNs. It has been demonstrated both theoretically and practically that intelligent operation of WSN nodes can improve network lifetime. For example, turning off wireless transceivers at WSN nodes, minimizing idle listening, can increase battery lifetimes by large factors, especially in many passive data sensing applications where the sense-receive-transmit cycle of the sensors is periodic.

In particular, we focus on some previous work in which an adaptive scheduling algorithm was proposed for this purpose, under unpredictable but small clock drift (so called quasi-periodic traffic). While this approach can adapt effectively to unknown transmission periods and unknown changes in transmission periods, the fundamental problem remains: a few nodes close to the base station deplete their batteries sooner than the rest resulting in early network death. Further, this phenomenon reduces the effectiveness of the method even more when (a) the periods of the various nodes are very disparate, and (b) when nodes artificially reduce their periods to maintain end-to-end delay bounds. In this thesis, we advance a new technique called "period inflation", by which the nodes of a WSN can cooperatively create a schedule in which nodes close to the base station have higher periods. We investigate the performance of the inflated and non-inflated cases for scenarios where all nodes have similar periods as well as when some nodes have very disparate periods, and also under bounded delay conditions. Numerical results show that the new technique of period inflation performs better, as expected.

# Design and Implementation of a Distributed Scheduling Algorithm using Period Inflation for Sensor Networks

by

**Bhushan Bhatt**

A thesis submitted to the Graduate Faculty of
North Carolina State University
in fulfillment of the
requirements for the Degree of
Master of Science

**Electrical and Computer Engineering**

Raleigh, North Carolina

2006

**Approved By:**

Dr. Mihail Sichitiu

Dr. Do Young Eun

Dr. Rudra Dutta
Chair of Advisory Committee

To my family . . .

## Biography

Bhushan Bhatt was born on December 15, 1982 at Ahmedabad, India. He graduated with a Bachelor of Engineering degree in Electronics and Communication Engineering from Dharamsinh Desai Institute of Technology, India, in May 2004. He then joined the Master of Science program in Computer Networking at North Carolina State University(NCSU), USA.

## Acknowledgements

I would like to express my gratitude to my advisor, Dr. Rudra Dutta, for his guidance, patience and the countless hours he spent with me to help bring this thesis to the current form. He has been a constant source of inspiration and insight, and this thesis would not have been possible without his support, enthusiasm and encouragement. He was always a source of motivation for me during the work on my entire thesis. I am grateful to my committee members, Dr. Do Young Eun and Dr. Mihail Sichitiu, for devoting time and providing valuable input.

I would like to thank my parents, Mr. Upendra Bhatt and Mrs. Kusum Bhatt, and my sisters, Samira Bhatt and Dharti Thacker, for the love, support and encouragement they have given me throughout my life. They have always been a source of inspiration for the all the endeavors in my life.

I would also like to thank my friends, Vineet Sahijwani, Pritesh Patwa, Pranav Parikh, Joerg Winterhoff and Teju Patel for being caring, understanding and supportive and constantly motivated through my education. My life as a graduate student, would not have been the same without these friends. Also, I will like to thank my childhood friends, Jay Mehta, Prashant Pandya, Sharnil Dalal, and Manish Shewaramni for all the fun times that we shared.

# Contents

# List of Figures

# Chapter 1

# Introduction

Ad- hoc wireless networks are a comparatively new paradigm in multi-hop wireless networking that is increasingly becoming popular. The credit for growth of ad-hoc network goes to its self organizing and self configuring properties. Ad-hoc network can be placed anywhere without the need of an existing infrastructure. Wireless sensor network fall under the special class of ad-hoc networks which are mainly used for data collection purposes. Depending on the modules they are equipped with, they sense some characteristics of the environment and connect with each other wirelessly to forward data to the monitoring station. Sensor nodes generally are deployed in such a manner that the topology may not be predictable beforehand. Thus they need to organize themselves in an ad hoc manner, set up the routing and start the communication process. Generally a sensor network would have a monitoring station in the vicinity of one or more nodes of the network which stores and processes the collected data. The nodes in the vicinity of the monitoring station are the ones which forward the data from the nodes in the entire network and hence drain their battery the most. To sum up, the life of the entire network is dependent on the lifetime of these nodes [10]. Hence power saving is a very important aspect of wireless sensor nodes.

The power saving approach that we focus on turns off the on board transceiver off when the node is not in the process of forwarding any data packets. There is a major risk in employing this strategy as there is no signaling between the nodes as when they might be forwarding data hence a sleeping node might miss some data packets. This is a very important point to be addressed especially in networks that sense and forward delay critical data. It is important to have an on/off schedule coordinated with a good routing algorithm so a node is not sleeping when it has to forward data [12]. Hence it becomes crucial that nodes have sufficient information as to when the next packed should be expected to be received.

The idea behind our algorithm is that the node can switch its transceiver off to save battery and know in advance when to wake up to receive the next packet from an upstream neighbor. A similar algorithm has been explored before in [16]. In our work, we extend the models already proposed to develop a relationship between the theoretical results proposed with the actual values we obtain from implementing the algorithm in our real network. We propose a new technique called "period inflation", by which the nodes closer to the base station have higher periods and thus an increased benefit from sleeping. We also give a comparison between performance of the network by varying values of different parameters like sleep and delay.

The rest of the thesis is documented as follows. In Chapter 2 we discuss the background and different model proposed in the literature typically focusing on the area of power conservation and routing for sensor networks. In Chapter 3 we discuss and summarize the prior work and discuss in detail the approach we adopted to address the problem. In Chapter 4 we describe the modifications we made in the existing algorithms by implementing deviations in generation periods and period inflation at specific depths. In Chapter 5 we present the numerical results we obtained by making the modifications. In Chapter 6 we discuss briefly about sensor motes and present our algorithm and results we obtained from our tests. We conclude our thesis in Chapter 7 and discuss direction for possible future work.

# Chapter 2

# Context

From the period during 1970's packet oriented radio networks have evolved to a great extend [2] [1]. Till this date, most of the wireless network fall under either of the two categories i.e. Infrastructure networks like the Wireless LANs normally used in homes and offices. These networks have preinstalled network components to facilitate their operation. The other category is the wireless Ad-Hoc network which does not have any existing infrastructure at disposal for use. This type of network is used when it is not possible to set up the required infrastructure for use. Some quick examples would be a combat zone, natural emergencies, or places where it is not feasible to set up an infrastructure due to environmental conditions or cost of operation.

Wireless sensor networks fall under a special category of ad-hoc networks. Unlike ad-hoc networks, which might support different data or voice application, the purpose of deploying wireless sensor is to take periodic measurement by sensing application. One another important difference is that sensor networks are equipped with a limited source of battery which in most cases is not replenishable. This results in a dynamic topology changes as some nodes may drain their battery over a period of time. Hence efficient routing and power saving algorithms become very important areas to focus while deploying a sensor network. In the first part of this chapter, we give an overview of Ad-hoc networking while in the latter part we discuss wireless sensor networks in detail. We present a brief overview from various literature surveys we undertook especially in the areas of routing and power saving.

## 2.1 Wireless Ad-Hoc Networks

### 2.1.1 Network Architecture and Properties

Nodes which form an ad-hoc network generally also perform the task of routing packets for other nodes in the network. This is the reason most communication that takes place in an ad-hoc network is multi-hop. Thus a node, besides transmitting its own data to the nodes closer to the base station also does forwarding of packets it receives from other upstream nodes to its downstream nodes. Thus the sensor network very aptly is called a network formed by self-configuring and self-organizing structure of numerous peer nodes. Almost all the nodes in a sensor network environment share similar properties and similar functionalities.

Ad hoc network can be mobile or stationary. Besides mobile networks where the topology cannot be predetermined, in most cases, even in stationary ad-hoc network the location and position of nodes is not determined beforehand. Hence it requires the nodes in the network to have a capability of organizing themselves in such a way that multihop communication is possible between different nodes and the monitoring station.

Given to the attribute of their small size, the nodes in a sensor network have limited resources on board to carry out certain tasks. They are supplied with very low onboard computational memory and a limited amount of battery power. Besides in an environment when more than one transmission and reception is taking place on a common channel, collision and packet loss is also an important factor to be considered. Thus these challenges make their operation more difficult than the operation of the nodes in a wired environment.

### 2.1.2 Challenges in Ad hoc Networking

Ad hoc networks are posed with several performance challenges. While some of the challenges are faced due to their size and available resources like power, processor speed and memory, some are posed to the environment in which they operate. These challenges are briefly described below.

*Power:* Wireless Ad hoc sensor nodes are battery powered and in most cases the area where they are deployed makes it very difficult or impossible to replace the battery or recharge the battery on the nodes. Hence it can be said that the lifetime of the entire network is dependent on the life of the battery.

There might be a number of nodes at strategic locations in the network which forward data from multiple upstream nodes. Battery consumption of these nodes may be much higher as compared to other nodes which forward data from fewer upstream nodes. Thus battery drainage of such a node might make a part or whole of the network unusable. Thus it becomes important to develop a routing algorithm for these types of networks to make optimum use of the battery and hence increase the lifetime of the network.

*Wireless Medium:* Sensor networks are typically broadcast networks. Hence all the nodes in range of a particular node can receive the data it transmits. In a network where hundreds of nodes are deployed and multiple transmissions are going on at the same time, collision and packet loss become an important issue to address to improve the network performance. While some the properties like predictable traffic pattern and low mobility are useful factors, properties like limited power source and low computational memory pose major challenges to design and deployment of sensor networks.

In this thesis work, we address the issue of power conservation. We have implemented a power saving algorithm on the wireless sensor nodes to help them save power and thus increase the lifetime of the network.

In the following section we present the power saving mechanisms that have been proposed and discuss them briefly.

### 2.1.3 Power Consumption

Wireless devices use a battery as their source of energy. As discussed in the sections above, limited battery power is the most fundamental constraint and hence power saving becomes a very important issue to be addressed. There has not been enough work done to address this problem as there has been done for issues like routing in wireless networks. The proposed approaches for power saving in wireless sensor networks can be broadly categorized in two perspectives. Both of the perspectives adopt a different approach in trying to address the same issue to save power in wireless networks.

### Node non Switch-off approach

The idea behind this approach is not to focus on any individual node but on the network as a whole to solve the problem. Most of the network level approach fall under this category and address issues mostly related to routing of packets in the network. Power aware routing techniques utilize one of the two proposed approaches: Minimizing the power consumption or maximizing the network lifetime. Both of these approaches have one common goal: to reduce power consumption in the network but still maintaining optimum network connectivity [4].

### Protocols maximizing battery life-time

In various networks, it may be the job of routing protocols to select the best route to forward packets in such a way that the overall power consumption of the entire network on a whole is minimized. Many different route selection approaches have been proposed to achieve this[18].

Minimum total transmission power routing (MTPR) Distance and the error or interference rate on a channel between the source and destination node are important aspects to drive the amount of power with which the packet should be transmitted. The most logical approach would be to find a path such that the sum of all the required transmission power is minimum. However this tends to create longer paths then necessary which may not be desired as longer paths are more unstable due to more number of points of failure. Hence an alternative approach is to not only consider the transmission power but also the power required at the receiving end. This will tend to push the selected path towards the shorter one. MTPR is not fair to all the nodes in the network i.e. it might over-utilize a single node which requires the least power to transmit thus burdening it.

Minimum Battery Cost Routing (MBCR): MBCR corrects the shortcomings of MTPR by including remaining battery life at each node on the path in its calculations of cost. This way, it avoids overusing a single node. However, it may still select paths in which nodes have little battery left when it is possible to select a path with a higher cost, but with more battery power remaining and thus attain a longer network lifetime.

Min-max Battery Cost Routing (MMBCR): In MBCR, the protocol will avoid a route with the minimum total battery capacity node in it. The individual nodes are not paid attention to. Thus it might be possible that we may decide avoid a route which is the lowest single battery capacity node.

The Conditional Min- Max Battery Cost Routing (CMMBCR) algorithm is proposed as an improvement to the performance of MMBCR. When all nodes in some possible routes between a source and a destination pair have sufficient remaining battery capacity (i.e. above a threshold), a route with minimum total transmission power is chosen. However if all routes have nodes with low battery capacity (below threshold), routes including nodes with the lowest battery capacity are avoided to extend the lifetime of these nodes.

COMPOW [11]protocol addresses three different issues and specifies methods to achieve them. These issues can be classified as to maximize the battery life, to increase the traffic carrying capacity and to reduce the contention at the MAC layer.

The COMPOW protocol propose that all the nodes in the network transmit at low power. With consideration of the low transmission power of different nodes of the network, it ensures the bi-directionality of communication between the nodes. Transmission of packets at low power by nodes ensures they conserve power which is an important resource for wireless nodes. Transmitting at low power directly results into lesser neighbors which in turn ensure that each node has a smaller number as neighbor hence less contention at MAC layer. Nodes that may have been unable to communicate among themselves may now communicate because there is less interference from surrounding nodes.

Hence COMPOW tries to find the minimum common power level at which the different nodes comprising of the network still remains connected. COMPOW maintains multiple routing table, one for each transmit power level available. This table can be formed by sending hello packets at the different power levels. Once the table has been formed, the optimal power level is the smallest power level whose routing table has the same number of entries as in the routing table if the node were operating without the COMPOW protocol.

*Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection* (LEACH DCS)[6]: This paper focuses on reducing the power consumption of wireless microsensor networks by stochastic clusterhead selection algorithm by a deterministic component. Depending on the network configuration, an increase of network lifetime by about 30 percent can be accomplished by implementing this protocol. This protocol replaces cluster heads after a specific amount of time to prevent overloading on a certain node in the area.

*A Low Computation Routing Algorithm for Sensor Networks* (RWPS) [13]: The recent interest in sensor networks has led to a number of routing schemes which make use of the limited resources available at sensor nodes more efficiently; the most important of all is the limited power source. The idea presented in this paper typically tries to find the minimum energy path to optimize the power consumption at a node. The approach addresses the typical application of wireless sensor network in which all the nodes forward data to a single monitoring station. The Remote Watching Power Saving (RWPS) communication system involves routing and MAC layers, with the purpose of maximizing the lifetime of the sensor network distributing semi-randomly the load in the network and minimizing the routing signaling traffic.

### Node Switch-off approach

Each node is addressed individually under this approach to deal with power conservation. It attempts to increase the lifetime of individual nodes by switching them off, usually operating in the MAC layer. Hence the goal of this approach is to save the power utilized in every node while still getting optimum performance.

*Pulse Protocol Routing and Power Saving* (PPSRP)[7]: The Pulse protocol utilizes a periodic flood initiated by a single node to provide both routing and synchronization to the network. This periodic pulse forms a pro-actively updated spanning tree rooted at the pulse source. Nodes communicate by forwarding packets through this tree. In addition, nodes are able to synchronize with the periodic pulse, allowing idle nodes to power off their radios a large percentage of the time when they are not required for packet forwarding. This results in substantial energy savings. A new mechanism called intermediate wake-up periods is introduced in this work in order to reduce the energy costs of low delay applications.

Pulse protocol thus helps save power consumed by a node by turning it off but it also has a drawback that it wakes up a node intermittently to send out packets and thus wastes power.

*Power Saving in Wireless Ad hoc Networks without Synchronization*[19] Power saving strategies generally attempt to maximize the time that nodes spend in a low power consumption sleep state. Such strategies often require the sender to notify the receiver about pending traffic using some form of traffic announcement. Although asynchronous traffic announcement mechanisms are particularly suitable for the ad hoc environment, they also provide relatively limited power savings. This paper proposes a mechanism that improves the efficiency of asynchronous traffic announcement mechanisms by reducing the proportion of time that nodes need to spend awake, while still maintaining good connectivity properties. The mechanism is based on allowing traffic announcements to be rebroadcast by neighboring nodes.

Power saving is an important and challenging issue in wireless networks, in particular for battery operated nodes such as mobile devices and sensor nodes. A popular approach to save energy is to periodically switch off a node or a few of its components for a certain time interval. In wireless ad hoc networks, switching off network nodes might not only have impact on the reachability of a single node but also on the connectivity of the whole network. Several approaches therefore propose to introduce synchronization mechanisms among the ad hoc network nodes. Nodes may wake up in a synchronized way, exchange data, and fall into sleep again after data exchange. Synchronization however is not easy to achieve and introduces also some overhead. Thus the paper proposes a mechanism that avoids synchronization and tries to take advantage of intermediate nodes that can relay traffic indication map messages between a sender and a receiver node.

## 2.2 Wireless Sensor Networks

Sensor networks comprise of small-size, low powered wireless devices deployed for specific application. Most of the sensor network nodes are equipped with sensing devices to sense parameters like temperature, pressure etc periodically. The nodes forward this data at a regular interval to its downstream neighbors to send the data to the sink. The location of the sensor nodes may not be known beforehand. Thus most of the research works in the field of sensor networks adopt the sensor node distribution as a Poisson process. There has been an increased interest in the field of sensor networks due to its inherent size and cost.

### 2.2.1 Network Architecture and Properties

Most of the nodes in a wireless sensor network equipped with sensing devices perform two tasks: sense data for its own application and forward the packets it receives from its neighbors to sink. The reason wireless networks are self-organizing and self-configuring is because the pattern in which they are deployed is not always predetermined. Thus as the nodes power up they are expected to form a network, know its neighbors and start routing packets. There might be only one monitoring station, the sink, which is the destination for routing all the packets in the network.

### 2.2.2 Challenges in Sensor Networks

Sensor network are usually deployed in a large number. Hence scalability becomes a very important issue. There is no unique ID for sensor nodes. They can be assigned beforehand or can be configured once they are deployed. Mobility is not a major consideration factor; because once the nodes are deployed they are usually not moved intentionally. However the topology still remains dynamic because nodes may die out and new nodes may be deployed to compensate for them. Nodes in the sensor network often are even more constrained then other form of ad hoc networks in memory, power and computational capabilities.

### 2.2.3 Routing

Routing in wireless sensor networks is a very important aspect. There has been a great amount of research done in this field in the recent times. The routing protocol running on the nodes of a network have two important things to consider; route packet as efficiently as possible to help the nodes to stay longer in low power mode while still maintaining full network connectivity and respect the application with delay bounds and forward packets for such application within the specified maximum delay bound.

*Agent Based Energy Efficient Routing*[8] In the paper, each data packet sent from a source node is carried by an autonomous mobile data agent, which can make its own routing decision based on its local information. Moreover, authors propose a novel data routing idea on energy efficient route choosing. Simulation results demonstrate that routing schemes based on the proposed idea can achieve a better performance of energy load balancing in the network, and a shorter time delay for data agents to travel from a source to a sink than other schemes discussed in the paper. Furthermore, by endowing the data agents with some intelligence, data aggregation can be performed locally and dynamically during their transmissions in sensor networks.

*Modelling Data-Centric Routing in Wireless Sensor Networks* Sensor networks differ from traditional networks in several ways: sensor networks have severe energy constraints, redundant low-rate data, and many-to-one flows. The end-to-end routing schemes that have been proposed in the literature for mobile ad-hoc networks are not appropriate under these settings. Data-centric technologies are needed that perform in-network aggregation of data to yield energy-efficient dissemination. In this paper a comparison between data-centric routing and traditional end-to-end routing schemes is shown. The impact of source-destination placement and communication network density on the energy costs, delay, and robustness of data aggregation is examined. Data-centric routing offers significant performance gains across a wide range of operational scenarios. authors make an underlying assumption that most of the nodes sense very similar data and hence there is always data redundancy. They propose a method in which the nodes enroute the packet to the sink verify the content of packet. Thus this requires much computational capacity for all the nodes in a network which is always not true. Also looking at the content of the packets is not always allowed in all the application because of data integrity reasons.

*Hierarchical Power aware Routing in Sensor Networks* [15] Wireless sensor nodes can create ad hoc networks and be used as distributed sensors to monitor large geographical areas, as communication enables for field operations, or as grids of computation. These applications require great care in the utilization of power due to limited onboard source of battery. The power level is provided by batteries and thus it is finite. Every message sent and every computation performed drains the battery. The paper adopts an approach of calling the network dead when the first node dies. Hence the lifetime metric is very important for ad-hoc networks where messages have to be delivered at high rates. An approximation algorithm for power-aware message routing that optimizes the lifetime of the network is proposed by the authors. The algorithm combines the benefits of selecting the path with the minimum power consumption and the path that maximizes the minimal residual power in the nodes of the network.

*Locating and Bypassing Routing Holes in Sensor Networks*[14] This works keeping in mind the limited power source for sensor networks, proposes a scheme for routing packets through the network. The authors in the work show how flooding is not an optimum solution for routing in sensor networks due to its high usage of resources. Also routing based on geographic forwarding has an inherent drawback. Some of the packets are forwarded to a node whose one hope neighbors are all far away from the destination than the node itself. Thus there arises a condition in which a packet is stuck and never makes to its destination. The sensor network topology is considered in this paper and the area where there is less number of nodes, routing around such situation is taken into consideration.

The assumption made in this work is that the network topology is known prior to deployment of the sensor nodes. In most of the sensor networks and applications this is not always the case. Hence the idea presented in this paper is only applicable to the networks whose topology is predetermined.

*Power Saving Mobility Protocols*[5]: In this paper the authors focus on sensor ID assignment scheme and ID creation algorithm, and mobility protocol for the management of the sensor or relay nodes movement. The relay nodes described in this literature are the nodes which forward data from its own sensing application as well as its upstream neighbor's data. Service gateway nodes are gateway nodes that provides service to the user outside WSN and manages the inside sensor or relay nodes. Relay nodes relay data between service gateway nodes and sensor nodes or between another relay nodes and sensor nodes. The authors consider a case in which both relay nodes and sensor nodes may be a mobile device. Hence, a new mobility protocol to manage the sensor or relay nodes movement is proposed. This literature work also focuses on issues such as Power Management considering the fact that Wireless Sensor Nodes are equipped with a very small amount of available power source.

### 2.2.4 Power Saving Techniques in Sensor Networks

As discussed in 2.3.3, it is important to address the issue of power consumption in wireless networks. Sensor networks have its nodes deployed at places or conditions where it is not possible to recharge or replace the battery. Some of the work done in this area deals at the MAC layer proposing the nodes to go into low power mode when not in use[9][20][12] while some address this issue by developing routing techniques [8][15] that help conserve limited amount of power available in sensor nodes.

*Energy-Latency Tradeoffs for Data Gathering in Wireless Sensor Networks* [20]: The paper focuses on a real-time scenario where the data gathering must be performed within a specified latency constraint. The idea behind the shutdown of nodes adopted in this paper is every nodes turns off by itself when it is not receiving or transmitting data. Another major assumption this paper makes is that all the nodes send out packets ( whether generated by its own application or forwarding packets that are received from its upstream nodes) at the highest speed and more priority and than turn off the transmitter and receiver.

The idea presented in this work differs from our proposition that each node builds its own forwarding table and is not dependent on the performance of other nodes i.e. each node acts independently in forwarding data.

*Latency of Wireless Sensor Networks with Uncoordinated Power Saving Mechanisms* [12]: This research work considers a wireless sensor network, where nodes switch between on and off mode, to save energy. The basic assumptions are that the on/off schedules are completely uncoordinated and that the sensors are distributed according to a Poisson process. Moreover, the durations of active and sleeping periods are such that the number of active nodes at any particular time is low hence a resilient routing mechanism is important. Thus for delay critical application it is required to have bounds on the latency, which is the delay elapsed between the time at which an incoming event is sensed by some node of the network and the time at which this information is retrieved by the data collecting sink. It is shown that the messages sent by a sensing node reaches the sink with a fixed asymptotic speed, which does not depend on the random location of the nodes, but only on the network parameters like on/off periods and network density.

The only drawback of implementing this method is nodes broadcasts the data as soon as it receives from its upstream neighbor, it broadcasts the packet. This approach thus makes sure that the packet reaches the sink but uses unnecessary redundant transmissions from multiple nodes thus consuming more power for transmission of a single packet than normally required.

*Delay-bounded Adaptive Power Saving for Ad hoc and Sensor Networks* [9]: This research paper focuses on the importance of the power saving in sensor networks while keeping it connected at most times. This duration, commonly known as the network lifetime, is often limited by a set of nodes that deplete their energy faster than others and cause parts of the network to be disconnected. The paper exploits the idea of switching off the nodes when they are dormant. In this dormant network condition, nodes put their radio in a power-saving mode. While putting nodes in power saving mode conserves their energy it can cause additional delay in transmission.

The disparity in the energy levels of the nodes can result from the network topology as well as differences in the hardware or the quality of the wireless links. The adaptive power-saving scheme increases the lifetime of a network up to 30 percent when variation in energy profiles of the nodes is solely due to the network topology. It can result in even higher performance gain when considering other factors that differentiate the energy levels of the nodes.

*Dynamic Power Management in Wireless Sensor Networks* [17]: This research paper proposes an OS-directed power management to improve the energy efficiency of sensor nodes. DPM is an effective tool in reducing system power consumption without significantly degrading performance. The basic idea is to shut down devices when not needed and wake them up when necessary. The work in this paper focuses more on hardware aspect in implementing power conservation. It discusses adaptation in the operating voltage and frequency besides sleep and wake up to achieve optimum power saving. The only problem this approach faces is predetermining the power levels necessary for certain applications which consume high power. Thus failing to provide such applications with required power levels make it not usable for all kinds of wireless sensor applications.

*Adaptive Ad-hoc Self-Organizing Scheduling for Quasi-Periodic Sensor Network Lifetime* [16]: Wireless Sensor Networks have been very popular for various applications in the recent times because of its size and cost. But these benefits come with a major drawback that they are equipped with a limited amount of battery. Authors of this paper identify power saving as important issue and propose node swithoff technique when the node is not being used. The algorithm in the paper allows the nodes in the network to learn about the behavior of its neighbors and adapt accordingly. Thus this helps the nodes in the network to save power but still not compromising with packet loss rate.

Hence power saving has been identified as a very important feature to prolong the life of any sensor network in various literature survey we discussed. The idea of controlling the power consumption by Operating System has been proposed in [17], while [1], [2][7] focuses on turning off the node components like transmitter and receiver when in idle state to save energy. [12]Discusses the effect of turning the nodes on/off to save power. It also focuses on having a delay bound for packets to reach the sink for delay critical application. But a different approach that models the traffic as a stream of data and shows that traffic shaping is required because of quasiperiodicity is discussed in [16]. It is shown how each node in the network shapes the traffic it sends to its downstream neighbor. By doing so, the nodes can learn the behavior of its upstream nodes and thus can form an optimum sleep schedule so that it does not miss any transmission from its upstream node. In the following section we present a short summary of the work done in [16] based on which we have extended our work.

# Chapter 3

# Problem Description

[12] identifies power consumption as very important aspect for sensor networks to extend its lifetime. But at the same time it stresses on having a certain routing algorithm and parameters to define delay bounds for the network nodes so that delay critical applications can send their data to the monitoring station within a specified time frame. A very similar problem is addressed but a slightly different approach is adopted in [16]. The idea is to make nodes in the network aware of the traffic as a flow from the upstream neighbor to downstream neighbor. Most of the sensor network algorithms are independent as in [16], which means there is not centralized component or monitoring station to overlook the working of the entire network. The only monitoring station present is one which collects all the application data for processing.

The algorithms in most of the previous work focus on three basic functions of the sensor node: generating sensor data, receiving transmissions from upstream neighbors and transmitting to downstream neighbors. Every time when the node generates a data packet, instead of transmitting it, it stores the data in a buffer and waits for a timer to fire indicating the time for transmission. The sole reason for buffering data is to have traffic shaping in the network.

## 3.1 Traffic Shaping

Sensor networks are always expected to have a large number of nodes. Most of the nodes in the sensor networks sense an application data and forward it to a monitoring station. Hence the node closest to the monitoring station carries data from its multiple upstream nodes. Thus we define the flow of data from upstream nodes to downstream node as a stream. This helps us in identifying transmission as flows instead of dealing with them as individual packets. Doing so reduces the overhead on a node as it has just to keep track of all the flows it handles rather than keeping track of every packet it receives from different nodes. Hence it implies that a node has to keep track of only a limited number of streams and thus can reduce the overhead in terms of buffer it might need to store information of every upstream node based on each packet it receives from them. The nodes that observe data packets from definite sources remember the sources they receive packets from.

## 3.2 Adaptive Algorithms

All the adaptive algorithms proposed have three different phases. In the first phase of the algorithm, the node determines its own period and its forwarding period. It also receives data from its upstream nodes which it needs to forward to the sink. The node records the inter-arrival times of the packets from its neighbors which is used to shape the traffic. The node does not sleep till the end of this first phase.

In the second phase the node actively starts shaping its data based on the traffic from the upstream neighbors. This phase is meant to allow the node to adapt its shape and the node will be stable at the end of this phase. Even in the second phase, the node does not sleep as its upstream neighbors themselves might not be stable.

In the third phase the node is allowed to sleep. By the end of the second phase, the node has a good estimate of the time when its upstream neighbors sends data to it. After it receives transmission from its upstream neighbor, it can calculate its sleep time based on the time when it expects another transmission from other upstream neighbor.

### 3.2.1 Non-Adaptive Approach

Each node knows the behavior of its upstream neighbor and it believes it to be constant. The node stores the value of the Algorithm 1 at the end of phase three. It does not modify it once it has stored it. Which means the node does not observe any change in pattern of traffic from its upstream neighbors neither does it adapt to it.

Non-adaptive approach would result in increase in loss if the behavior of its nodes is not constant. Consider a case when an upstream neighbor speeds up its transmissions. In this case, the node might be sleeping and it may miss a packet. The algorithm is to stay awake until it receives the next packet thus the sleep decreases with the increase in the packet loss. Other case to be considered is when an upstream neighbor slows down transmission. In this scenario the node wakes up ahead of time and so less transmissions will be lost but the sleep would also reduce.

### 3.2.2 Use of Sequence Numbers

The use of sequence numbers can actively help the node to track any missed packet. Each node transmits packets with sequence numbers and they are independent of the sequence numbers of another node. Thus a node, when receiving packet from its upstream node, notes the sequence number of the packet and knows what the expected sequence number of the next packet should be. In case the sequence number of the next packet received from a particular node is out of order i.e. greater than what it was expecting, the node knows that it had missed a transmission from one of its upstream neighbor.

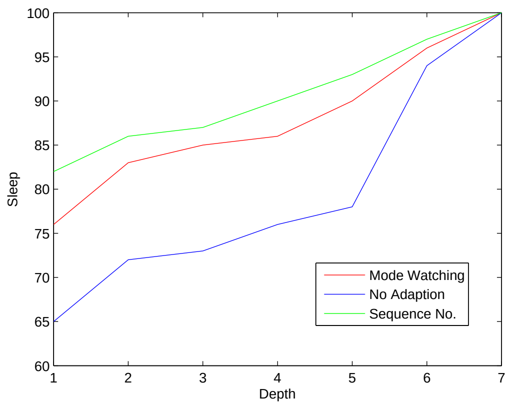
When a node misses a transmission, the inter-arrival time between two consecutive packets is very large. This is divided equally to get the successive inter-arrival times. When an upstream neighbor slows down, the node wakes up ahead of time and gets less sleep. But it will very soon adapt to this change as it is noting the inter-arrival times. The same will hold true for the case in which an upstream speeds up transmission. Initially the node would miss a packet but it will soon adapt to this change with the help of inter-arrival times it records.

### 3.2.3 Adaptation without Sequence Numbers

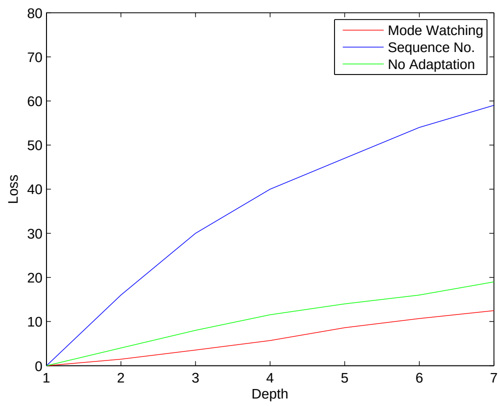
Sometimes it might not be possible to make use of sequence numbers due to the nature of application or limitation of the sensor nodes. The algorithm should adopt to the changes quickly even though there are no sequence numbers used. The mode of the histogram is not distorted by sleep but it changes when the behavior of the upstream node changes. Thus if there is lot of change in the mode of the histogram, the node knows that the behavior of its upstream neighbor has changed. The node in this situation switches back to the phase 1 of the algorithm and attempts to learn the new behavior of its neighbors.

Consider the case when an upstream neighbor slows down its transmission rate. In this condition the mode and mean of inter-arrival time distribution and mode increases. This helps the downstream neighbor to detect the shift in the behavior of its upstream neighbor. Even in the case when the upstream neighbor speeds up, the mode changes and thus helps the downstream neighbor to detect it. In either case, the node will switch to phase one of the algorithm and learn the new behavior of its neighbor and construct the histogram based on it.

### 3.2.4 Comparison of Adaptive Algorithms




(a) Sleep Comparison




(b) Loss Comparison

Figure 3.1: Comparison of sleep and loss for Non Adaptive, Sequence no. and Mode Watching cases

We show the comparison of the performance of the adaptive algorithms discussed.

To further investigate the claims, the performance of the algorithms is shown under conditions when the node speeds up its transmission or it slows down.

Fig 3.1(a)and 3.1(b) shows the values for non-adaptive, sequence number and mode watching algorithm. It can be seen that the non-adaptive case has the worst performance in terms of sleep and loses more data than other two algorithms. The adaptation by mode watching performs better than non-adaptive case. It sacrifices some sleep in the process of adapting to the changed behavior of its neighbors. Adaptation using sequence number works the best of all the algorithms. It quickly adapts to the change in the behavior of the neighbors and does not sacrifice its sleep in the process of adapting.

## 3.3 Our Contribution

We identify the importance of sleep for the nodes in the network to prolong the network lifetime. For this, it is important to have an algorithm which would help the nodes near the sink to conserve power. We therefore propose modifications in the algorithms discussed in 3.2 which will allow the inner tier nodes to sleep more without compromising on loss or delay in the network. To this end we introduce a technique called *Period Inflation* which makes possible for the inner tier nodes to transmit less frequently and help them sleep more.

Further we implemented the algorithm in an environment with wireless motes. We implemented the existing algorithm using sequence numbers and later tested the network with period inflation algorithm to maximize sleep for the nodes in the network.

# Chapter 4

# Modifications in the Existing Adaptive Algorithms

The algorithm in [16] introduced a sleep scheduling concept to save the battery power in the sensor nodes. There is a trade-off between sleep, loss and delay parameters shown in Fig 3.1. We have identified that varying different factors in the algorithm can help increase the values of sleep but maintaining loss and delay values. We inflated the generation and forwarding periods to enhance the sleep factor and keeping the loss to minimum.

*Period Inflation* helps the nodes to increase their sleep fraction by modifying their periods. There are two different approaches we adopt to implement this enhancement in the previous work. We inflate the periods of the nodes which belong to the inner tiers in the network in the proximity of the sink thereby enabling them to increase their sleep time. As the nodes in the inner tiers sleep more, they help in increasing the overall network lifetime. Our work addresses both the scenarios with delay bounds and non specified delay bounds.

When the inflation in employed at all the nodes in the network from the outermost tier to the innermost tier, each node is able to inflate its period depending on the location of the node in reference to the sink. The nodes situated closer to the sink get to inflate their periods more than the nodes in the outermost tier. Thus with all the modifications with inflation we expect the overall network lifetime to increase as compared to the network which does not implement period inflation. We discuss the period inflation and deviated generation period algorithms in the following sections.

## 4.1 Increased Deviation in Generation Periods

Most nodes in a wireless sensor network are responsible for carrying two different tasks: generating its own sensor data and forwarding the data it receives from its upstream neighbors to the monitoring station. Generation times in different networks may be different but the generation period i.e. time between two consecutive applications sensing in most case is identical for all the nodes as they are a part of one network and sensing data for the similar application.

But if we consider a situation in which some nodes in a network though forwarding data to the same sink as other nodes are performing different task. Thus in the prior work this issue has not been addressed. The deviation parameter *(Pdev)* discussed in prior work had a limited deviation from the default value. Pdev in prior work had a value of 10 - 30 percent above the default generation periods. The algorithm under such cases had very negligible effect on the performance of the network. We increased the value of Pdev by 100-200 percent of the generation period for randomly distributed nodes.

The goal of this modification was to check if the simulation algorithm adapts to such drastic variation in values that had previously not been dealt with. The overall sleep and loss is not expected to vary largely because the nodes when adapt to the behavior of its upstream nodes, define their sleep periods and it takes only two cycles to adapt to the behavior of the upstream nodes. We expected the delay, a packet experiences in reaching the sink, to increase in both cases with unbounded delay and bounded conditions.

The expected increase in the case of unbounded delay is self-explanatory. When the nodes with different delay bounds are distributed randomly, in any data stream from the outermost tier to the innermost node, there is a possibility of having a node whose generation time is more than 500ms, hence the delay increases. Even though in any stream, a single node whose generation period is the least drives the entire flow of data, it is likely to have nodes in the outer tiers who have a larger generation period and thus add to the delay in the network.

But even for the case of the bounded delay there would be a very slight increase in the delay as compared to the bounded delay case without any deviations in the generation periods. The explanation for the increase is the same as that for the unbounded case. Presence of nodes in the outer tiers with higher generation periods adds to the overall delay incurred on a packet to reach the sink.

## 4.2 Period Inflation at Nodes in inner tiers

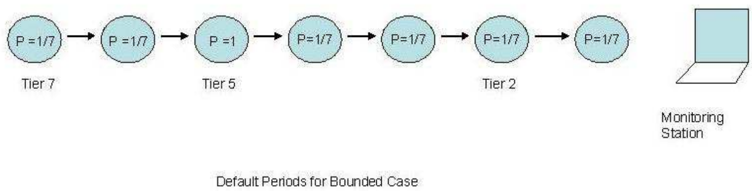



Figure 4.1: Bounded Delay Default Case

The correlation between loss and sleep of any node in the network can be shown in terms of benefit. This dependency can be expressed in terms of Benefit (B), Sleep (S) and loss (L) as [16]:

$$B = (1 - L)/(1 - S) \tag{4.1}$$

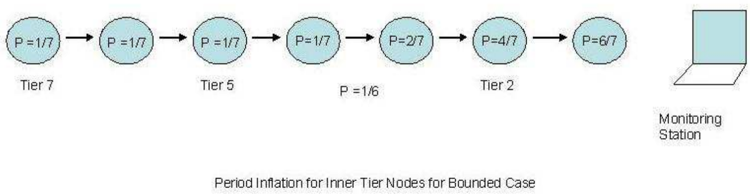



Figure 4.2: Bounded Delay Case with Inflation $i_f = 2$

As shown in [10] the outermost nodes have a high loss and delay rate while the nodes which lie in the innermost tier has the least value of sleep as they forward most of the data. The lifetime of the network is determined by the lifetime of the innermost nodes. Hence when the innermost nodes run out of power, the rest of the nodes in the network get disconnected from the monitoring station and thus the network becomes useless. Thus our focus was to increase the network lifetime by increasing the value of B. To achieve this we increased the available sleep for nodes in different tiers such that the loss does not increase but the overall network lifetime does.

As we already mentioned, nodes in inner tiers are the ones which forward data from multiple upstream neighbors and get less sleep as compared to outer tier nodes. Hence *Period Inflation* forces the nodes to hold the packets they generate or the ones they receive from their upstream neighbors for the amount of time specified by the inflation factor. It is possible to vary the value of inflation factor for nodes belonging to different tiers in the network. By forcing the nodes belonging to tier 1,2 and 3, we can help them sleep more and forward data less frequently.

**Calculate Transmission Period for Default Case**

**Algorithm 1** Calculate Transmission Period for Default Case

$\{nd_{max}$ : delay bound for the node$\}$
$\{n_p t$ : transmission period of the node$\}$
$\{n.d$ : depth of the node$\}$
$\{np_g$ : generation period of node n$\}$
$\{i_f$ : inflation factor$\}$
$\{$U : set of all upstream neighbors$\}$
$d_{max} \leftarrow \min(\mathrm{U}.d_{max})$
**if** self.$d_{max} < d_{max}$ **then**
  $d_{max} \leftarrow$ self.$d_{max}$
**end if**
$n_p t \leftarrow \min(\mathrm{U}.n_p t)$
self.$n_p t \leftarrow d_{max}.n_p t$

Thus we modified the algorithm such that, it allows the nodes in tier 1, 2 and 3 to adjust its forwarding period to twice that of the outer tier nodes. Thus letting the nodes to wake up as previously but forward only once in every two cycles. These scenarios can be expressed diagramatically as show in Fig. 4.1 and Fig. 4.2. The default bounded case as seen from the figure divides its generation time by the number of hops its away from the sink. But when period inflation is applied to the inner nodes, they calculate the packet transmission time according to Algorithm 2 and Fig. 4.2. In the unbounded case, this increase in period would add up to the delay in the network as the inner tier nodes hold packets from upstream neighbors for 2 cycles. 3 nodes in the inner tier buffering packets to be transmitted thus would add up to the delay for the unbounded case. For the packets which have specified delay bounds, the outer tiers will forward data 1/7 period of every cycle for their generation period. But the inner tier nodes buffer this data and transmit it only once in every two cycles. This would add up to the delay even with the bounded delay case and help increase the sleep fraction for nodes in tier 1, 2 and 3.

**Calculate Transmission Period for Period Inflation**

**Algorithm 2** Calculate Transmission Period for Period Inflation

$\{nd_{max}$ : delay bound for the node$\}$
$\{n_pt$ : transmission period of the node$\}$
$\{n.d$ : depth of the node$\}$
$\{np_g$ : generation period of node n$\}$
$\{i_f$ : inflation factor$\}$
$\{$U : set of all upstream neighbors$\}$
$d_{max} \leftarrow \min(\text{U}.d_{max})$
**if** self.$d_{max} < d_{max}$ **then**
  $d_{max} \leftarrow$ self.$d_{max}$
**end if**
$n_pt \leftarrow \min(\text{U}.n_pt)$
self.$n_pt \leftarrow d_{max}.n_pt * i_f$

The delay is expected to increase in the unbounded delay case as the inner nodes hold the packet for twice the period of time they hold in default case. But again the delay in the bounded delay case increases and allows the inner nodes to sleep more. Hence the sleep can be increased while still respecting the delay bounds of the application.

## 4.3 Inflating Period for all Tier Nodes

As discussed in 4.2, inflating periods for nodes in the inner tiers help them sleep more hence help conserve their battery power which can result in overall increase in the network lifetime. We decided to check the performance of the network under the condition where all the nodes in the entire network inflate their period whether it is bounded case or unbounded.

The sleep fraction for each node is decided by the number of upstream neighbors it has. Thus to inflate the period for all the nodes in the network, we decided to take the depth and indegree parameters into consideration. The generation periods of each node in the network is inflated by the location of the node in the network. Hence nodes belonging to the inner tiers in the network in this case will get more sleep as they have more upstream neighbors as compared to the nodes in the outer tiers.

$$n_d = \text{2d+1/ 2d-1} \tag{4.2}$$

Changing the value of inflation for every node in a network should result in a delayed transmission of the packet in unbounded case to the sink. The delay added by each node would be the amount of inflation it is allowed to add on each packet it generates. Hence the overall delay a packet faces in reaching the sink for unbounded delay case would be finally increase to almost 10 times the delay in the default case without the inflation.

**Inflation at all tier nodes**

**Algorithm 3** inflation at all tier nodes

{$n_pt$ : nodes transmission period}
{U : set of all upstream neighbors}
{$n_pg$:nodes generation period}
{$i_f$ : inflation factor}
$n_pt$ = min (U.$n_pg$)∗i$_f$

In the bounded delay case when we investigate the scenario when all the nodes inflate their periods as discussed, the delay in the nodes is expected to increase by about 2 times the delay faced by the node in reaching the sink than under the default run. By inflating the period at each node in the network, the overall sleep in the network is expected to increase because they forward data less frequently than they used to under the non period inflation condition.

# Chapter 5

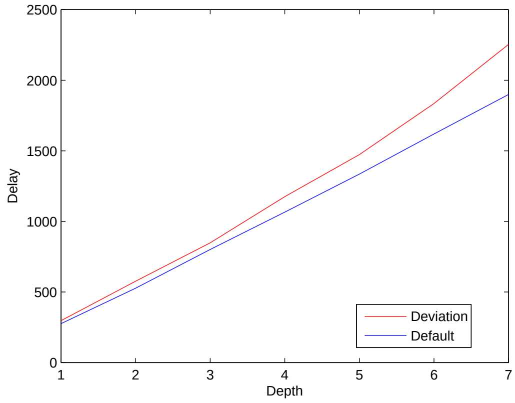
# Results for Period Inflation

The simulation setup we used has several topologies with uniform density. There are 200 nodes in the network topologies we used with maximum depth of 7. The following sections show the performance of the network with large variations in deviation periods and inflated periods for nodes at various depths in the network.

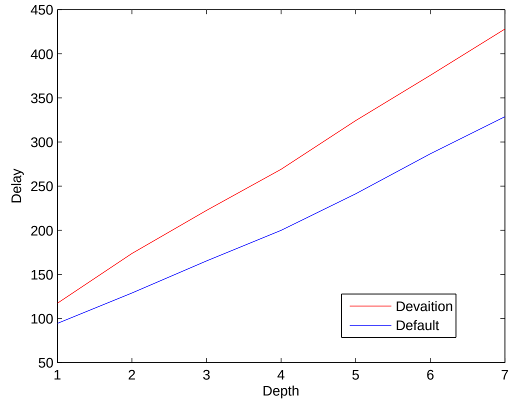
## 5.1 Performance with increased deviation in generation periods

The generation period for all the nodes in the network is 500ms + x.Pdev where Pdev is 50ms and x = (0,1,2,3). As discussed in 4.1, we use the value of Pdev as 250ms. Thus this results in generation periods for different nodes as 500ms, 750ms, 1000ms and 1250ms. We generate a random number and according to the value of the number, the nodes are assigned different generation times. We used numerous different topologies hence we say that the the different generation times are equally distributed among the nodes. Thus one fourth of the nodes have generation period of 500, one fourth have it as 750, one fourth have a period of 1000 and the rest generate data at 1250ms.

Fig 5.1 shows the comparison of delay curves. There is an increase in the overall delay in the network because the nodes generate and transmit data less frequently as com-

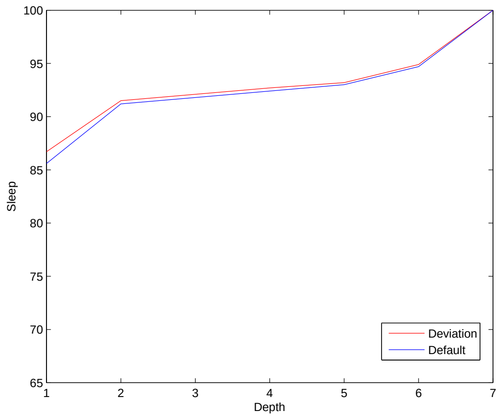



(a) Pdev = 250. Unbounded Delay

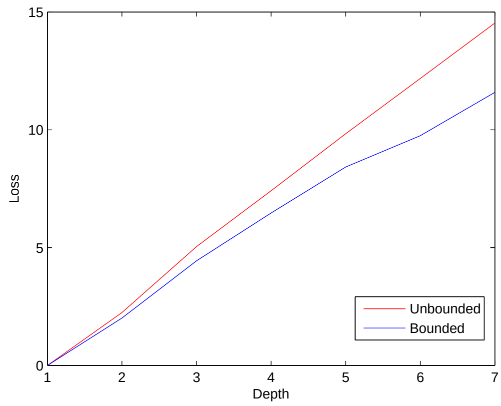



(b) Pdev = 250. Bounded Delay

Figure 5.1: Delay comparison for Default case and Case with Deviation in Generation Periods




(a) Pdev = 250. Unbounded Sleep



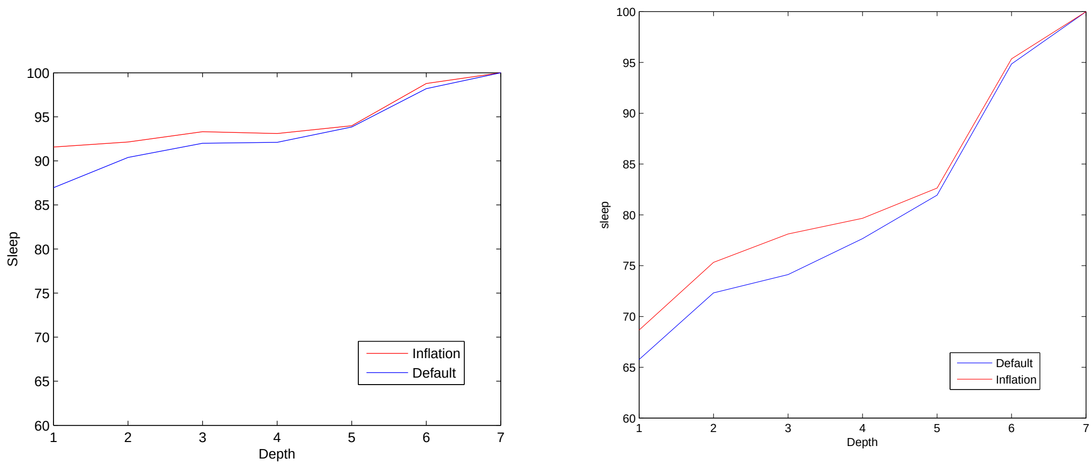

(b) Pdev = 250. Loss

Figure 5.2: Sleep and Loss for Increased Deviation Periods

pared to the default generation period case. There is hardly any noticeable difference in the packet loss because though the generation periods are varied, the algorithm still adapts to changes and misses minimal packets.

Fig 5.2 shows the comparison in performance of the default algorithm and our algorithm in terms of sleep and loss fraction. As expected, the value of sleep increases marginally because some nodes now forward data less frequently as compared to the default case. There can be an argument that though some nodes generate data once in every 1250ms, they may have upstream neighbors with period of 500ms thus forcing the node to adjust its period to 500ms. But the sleep increases becasue when the generation periods are distributed according to Poisson distribution, there will be some nodes in the outer tiers with period of 1250ms or 1000ms.

## 5.2 Period Inflation at inner tier nodes

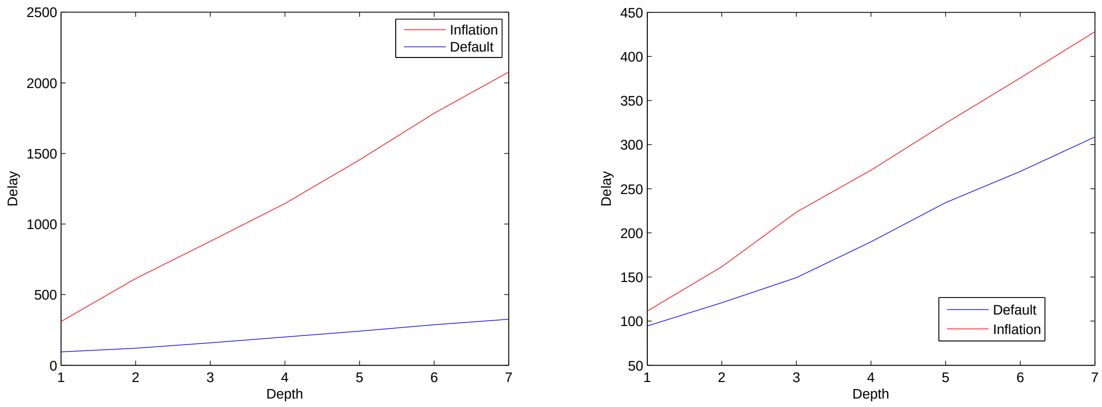



(a) $i_f = 2$. Unbounded Sleep

(b) $i_f = 2$. Bounded Sleep

Figure 5.3: Sleep Comparison for Default case and Inflation for $i_f$ =2

We inflated the period at the nodes in tier 1, 2 and 3 by an inflation factor of 2. Nodes in this tier wake up to receive packets from its upstream neighbors once every cycle, but the forward data to downstram neighbors once in every two cycles. This allows




(a) $i_f = 2$. Unbounded Delay

(b) $i_f = 2$. Bounded Delay

Figure 5.4: Delay Comparison for Default case and Inflation with $i_f$ =2

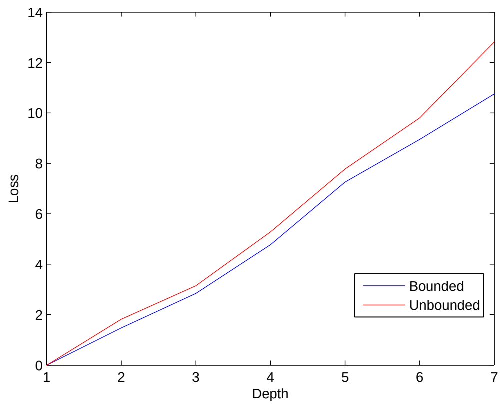
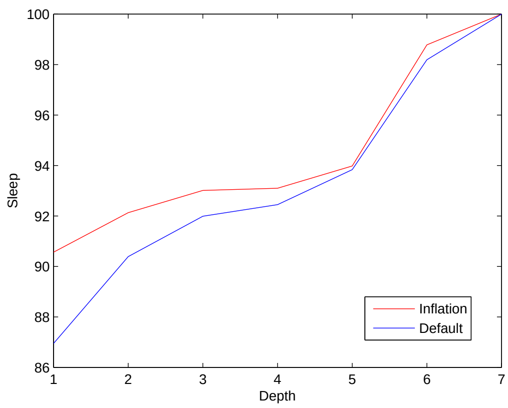



Figure 5.5: Loss Comparison for Default and for Inflation with $i_f$ =2

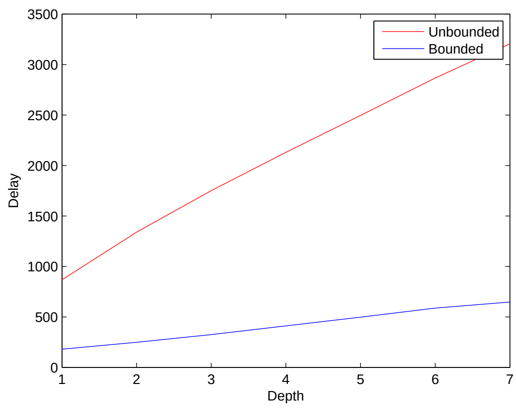
this nodes to sleep more and conserver power as discussed in 4.2 which in turn should help increase the network lifetime. Fig 5.3 shows the increase in sleep obtained at inner tiers by implementing inflation. As compared to network with no inflation, the sleep value for the inner tier nodes increases for both bounded and unbounded cases.

The delay a packet incurs in reaching the sink, without delay bounds, increase greatly due to inflation. For non delay sensitive application increase in delay is not an issue as far as network lifetime increases. The delay for the applications that have a specific delay bound also increases but this increase is marginal. Hence there is a tradeoff between reducing the delay and maximizing sleep fraction. Thus the nature of the application will be deciding factor for the use of algorithm. Fig 5.5 shows that loss in the network is almost unaffected due to inflation and thus adaptive algorithm holds true even with inflation.

## 5.3 Effect of period inflation for all the nodes




(a) $i_f = n_d$. Unbounded Sleep




(b) $i_f = n_d$. Bounded and Unbounded Delay with Inflation

Figure 5.6: Sleep and Delay Comparison for $i_f = n_d$

According to equation 4.2, we inflated the periods of all the nodes in the network considering the location of the nodes. Hence before forwarding a transmission to its downstream neighbor, each node multiplies its generation period by the inflation factor

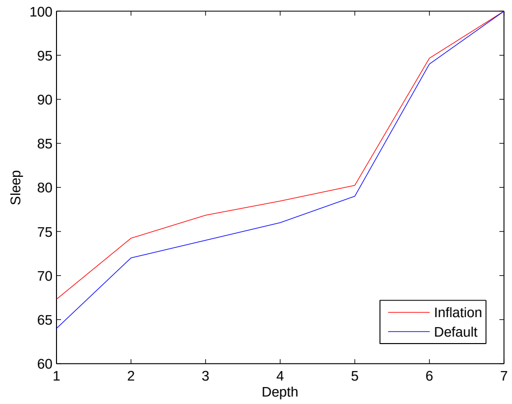



Figure 5.7: Sleep Comparison for Default and Inflation Cases with Delay Bound

depending on its location in the network. Nodes in each tier calclute their inflation factor from equation 4.2. The expected result of this change was the increase in the delay for a packet to reach the sink. Inflating the periods of all the nodes would result in increase in fraction of sleep by a small factor. Fig. 5.6(a) shows the increase in the sleep fraction as predicted in 4.3. Sleep increases in all the nodes in the network but it increases especially at the nodes in the inner tiers. To make this possible, we had to inflate the periods in the outer tiers which resulted in increase in sleep in the outer tier nodes too. Fig. 5.6(b) shows the increment in delay for the bounded and unbounded cases. Fig. 5.7 shows the increase in the sleep for the network with delay bounds.

Various results from the modifications show that increasing the fraction of sleep for different nodes in the network helps increase the network lifetime. This increase is very noticeable at the inner tier nodes and the difference reduces as we go from inner tiers to outer tiers. The delay in different scenarios increase as predicted. The loss parameter of the entire network does not vary while varying sleep and delay. This helps us propose that modifying the algorithm to increase the benefit in terms of sleep does not affect the overall loss in the system as the adaptation algorithm works well even with period inflation.

# Chapter 6

# Implementation on Motes

In the previous chapters we discussed the work already done by developing the adaptive algorithms for the sensor network. We later identified some issues not addressed in the prior work and tried to focus on them by utilizing the algorithm used in [16]. We showed theoretically how modifying few parameters can help the inner tier nodes to increase their sleep fraction and in turn increase the network lifetime. In this chapter we discuss the work we did by implementing one of the adaptive algorithms, using sequence numbers on a test bed with motes. Later in the chapter, we discuss the implementation of period inflation in our mote network. In the last section we present the various numerical results we obtained from implementation of the algorithm in our network.

## 6.1 Adaptation Using Sequence Numbers

As discussed in 3.1.3, making use of sequence numbers is a way to keep track of lost packets. A downstream neighbor when receives a packet from its upstream neighbor notes the sequence number of the packet it last received. So it knows what would be the expected sequence number for the next packet. In case the sequence number of the packet it receives is greater than expected, it knows how many packets it missed while it was asleep.

We extend the work done priorly and the cases discussed in chapter 3 by implementing an algorithm on the test bed with motes. We focused on the algorithm where nodes in the network make use of sequence numbers when forwarding their data. Hence the goal of the algorithm would be for a downstream neighbor to quickly realize if it misses a packet from its upstream neighbor in case it was not able to turn its trans-receiver on.

The approach adopted in [16] was to note down the inter-arrival time of the packet in case of a missed transmission. It builds a histogram and thus nodes know the sleep and wakeup time from the values they retrieve from the histogram. If a downstream misses a packet, it does not sleep for the next cycle until it receives the next packet from its upstream neighbor. Our approach is slightly different in case of missed packets. The node which misses a packet notes down the inter-arrival time but we do not focus on building a histogram. Rather after obtaining a value of the inter-arrival time between the packets, it can form a new sleep schedule for the next packet it expects to receive.

## 6.2 Routing in Sensor Nodes

Sensor networks typically have hundreds of nodes generating data and forwarding it to a sink. Except the tier 1 nodes, all the rest of the nodes are not in the range of the sink and thus there is a need to have an ad-hoc routing between the nodes. Hence all the nodes forward the generated data to its downstream neighbors at regular intervals. Thus a multi-hop routing protocol is required for the communication to work.

### 6.2.1 Routing using Surge

| Source | Origin | Sequence No. | Hop Count | Data |
|---|---|---|---|---|
| 2 bytes | 2 bytes | 4 bytes | 2 bytes | 4 bytes |

Message Format

Figure 6.1: Packet Format

Wireless sensor nodes use a platform called TinyOS. Surge is a multi-hop protocol which can be used using TinyOS on wireless nodes. Instead of developing a routing algorithm we use Surge for implementation of the adaptive algorithm. The packet format of Surge is show in Fig 5.1

The packet format used for our messages is shown in Fig 6.1. The origin address is the address of the node which generated the reading. The source address is the address of the node which is forwarding the data to its next hop downstream neighbor. In a multi-hop network with more than 3 tiers, each node which receives the data replaces the source address with its own address before forwarding to its downstream neighbor. The sequence number field contains the sequence number of that particular packet. Each node upon successful transmission of the packet increments the value used for sequence numbers. As discussed earlier, using sequence number helps us keep a record of missed transmissions. The data field contains the random data generated each time before transmitting out a packet. It also contains the updated value of the sleep time with every transmission.

## 6.3 Power saving

We address the power saving, which we have focused as a key issue in wireless sensor network by turning the on board transmitter and receiver present on the nodes to off when they are not involved in any activity. As discussed in 5.1 each node actively notes down the time and sequence number of the packet it receives from the various upstream neighbors it has. When the node is not expecting any transmissions from its neighbors for a threshold period (thp) of 50ms, it turns off the transmitter and receiver. When the radio on the mote is turned off, it consumes less than 1 percent of power it consumes otherwise. Hence by efficient design of the algorithm we can increase the lifetime of our network manifold.

## 6.4 Adaptation using Sequence Numbers

The network we modeled for implementation makes use of sequence numbers to identify if a node missed transmission from its upstream neighbors. The nodes in our network are not synchronized by a global clock for the network; hence we do not address the delay issues related to transmissions in our network.

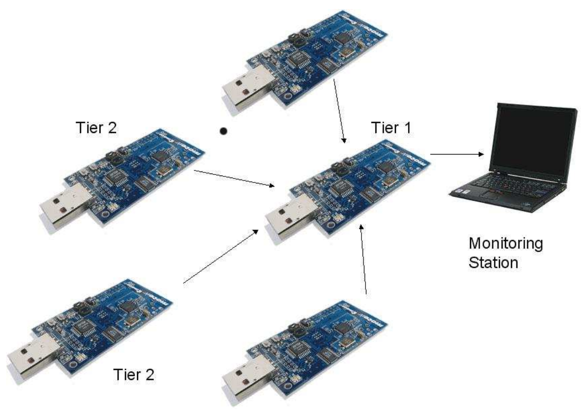


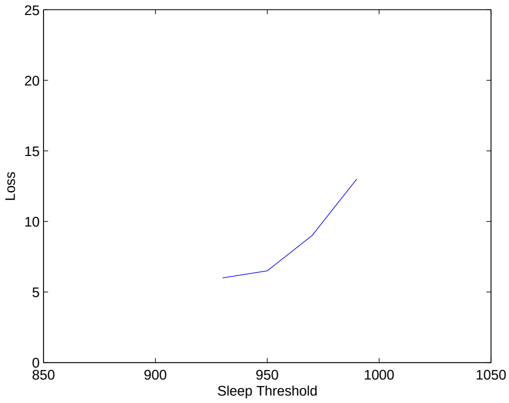

Figure 6.2: Network Setup

### 6.4.1 Network Setup

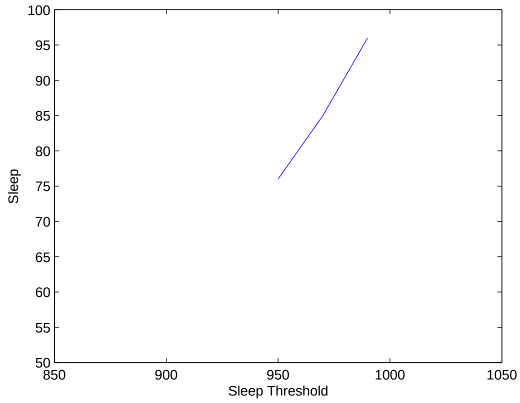
Our network consists of 5 motes. 4 motes form tier 2 in the network which generates packets at a regular interval of time. Random data is generated for each transmission at an interval of 1 second. These motes forward the data to the tier 1 mote. This mote is connected to a monitoring device i.e. computer to log the data it receives from its upstream neighbors. Fig. 6.2 shows the network setup we used.

In the initial 2 cycles, the tier 1 mote observes the traffic pattern form its neighbors. It does not turn off its transceiver during this period. It records the source address and sequence numbers of the packets we receives from its upstream neighbors. It calculates the inter-arrival time between the packets and determines its sleep schedule. After 2 cycles, the mote sets it clock to the time of next expected reception from any of its upstream neighbors. This node uses a threshold period of 20ms to schedule its sleep i.e. if the next expected reception is less than 20ms, than the node does not turn off its transceiver.

Let us consider a case when a packet is received out of sequence from one of the upstream neighbor, the mote does not sleep until it receives the next packet from the same neighbor. It calculates the inter-arrival time between the packets it received and makes the change in its wakeup schedule for that particular upstream node. We expect the motes to adapt to this algorithm quickly and incur minimum loss of packets.




(a) Packet loss for $s_{th}$ = 20ms




(b) Sleep for $s_{th}$ = 20ms

Figure 6.3: Loss and Sleep for $s_{th}$ = 20ms

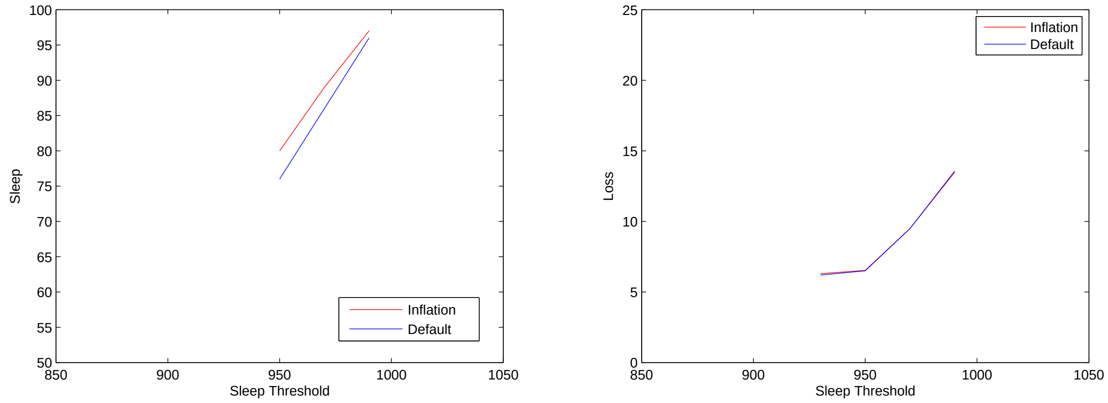
### 6.4.2 Validation of sleep model with sequence numbers

To validate the results obtained from the algorithm we implemented, we changed the threshold for the wake-up time. In default case, the node wakes up 30ms ahead of the next expected reception. We modified this reception threshold from 30ms to 50ms allowing the node to sleep more. The expected result would be increase in throughput and decrease in sleep. With the threshold reduced from 30ms to 10ms, the result shows reduced throughput indicating more packet loss but increased sleep. Fig. 6.3(a) and 6.3(b) shows this comparison.

### 6.4.3 Period Inflation

The topology for a network consisting of 5 motes is shown in Fig. 6.2. We implemented period inflation technique on the tier 1 mote. Similar to the algorithm discussed in 4.2, we inflated the period of tier 1 node to twice the regular period. The goal behind this implementation was to make the tier 1 node transmit data it receives from the tier 2 nodes less frequently hence enabling it to have more sleep. This would help us increase the lifetime in out network as tier 1 mote forwards its data as well data it receives from its 4 upstream nodes.

Fig. 6.4 shows that due to period inflation, the sleep in tier 1 node increases as




(a) Sleep Comparison for Default and Inflation Cases (b) Loss Comparison for Default and Inflation Cases

Figure 6.4: Loss and Sleep for $s_{th} = 20$ms with $i_f = 2$

compared to its sleep fraction shown in Fig 6.3(a). As the mote forwards data once in every two cycles, it is expected that the delay the packets from tier 2 and the packet mote 1 itself generated would face increased delay. Due to non-synchronized clocks the motes work with, we do not take the delay into account.

We show the comparison for loss factor between the defualt case and case with inflation. It can be seen from the figure that there is very slight variation in the loss case after we ran the topology several times. Hence we say that period inflation algorithm adapts to the mote environment as it did to our simulated network.

# Chapter 7

# Conclusion

We show that by increasing the generating periods of various nodes increases the delay factor for a packet to reach the sink. But we implemented our algorithm such that the adaptive algorithm holds true and causes minimum effect on sleep and loss.

Implementing period inflation, the nodes in inner tiers are able to sleep more which directly results in increased lifetime of the entire network. The delay increases due to inflation but the increase is minimal for application with bounded delay. Increase in delay for non-delay sensitive application helps us increase the lifetime of the network greatly.

We also implemented the inflation period technique we propose on our mote network to show the validity of our modification in real networks. The results obtained shows increase in sleep for inner tier node improving the total lifetime of the network.

# Bibliography

[1] Genetic algorithm to solve optimum TDMA transmission schedule in broadcast packet radio networks. In *IEEE Transactions on Communications*, pages 765–777, May 2004.

[2] A.H Abdelmonem and T.N Saadawi. Performance analysis of spread spectrum packet radio network with channel load sensing. In *IEEE Journal on Selected Areas on Communication*, pages 161–166, Jan 1989.

[3] Bhaskar Krishnamachari, Deborah Estrin, and Stephen Wicker. Modelling Data-Centric Routing in Wireless Sensor Networks. In *IEEE INFOCOM*, 2004.

[4] I. Chlamtac, M. Conti, and J. J. Liu. Mobile ad hoc networks: Imperatives and Challenges. In *Ad Hoc Networks*.

[5] Dong-Hyun Chae, Kyu-Ho Han, Kyung-Soo Lim, Kyeung-Hak Seo, and Kwang-Ho Won. Power Saving Mobility Protocol for Sensor Network. In *Second IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems pages=*.

[6] M. J. Handy, M. Haase, and D. Timmermann. Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection. In *IEEE-Moblie Wireless Communication Networks*, 2002,.

[7] Kirk Chang and J Wang. The Pulse Protocol: Sensor Network Routing and Power Saving. In *IEEE Military Communications Conference*, 2004.

[8] Long Gan, Jiming Liu, and Xiaolong Jin. Agent-Based, Energy Efficient Routing in Sensor Networks. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 472–479, 2004.

[9] Maryam Owrang, Diba Mirza, and Curt Schurgers. Delay-bounded Adaptive Power Saving for Ad hoc and Sensor Networks. In *Vehicular Technology Conference*, September 2005.

[10] Mihail Sichitiu and Rudra Dutta. Benefits of Multiple Battery Levels for the Lifetime of Large Wireless Sensor Networks. In *Proceedings of Networking 2005*, pages 1440–1444, May 2005.

[11] S. Narayanaswamy, V. Kawadia, R. Sreenivas, and P. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol. In *Proc. of European Wireless 2002. Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, pages 156–162, February 2002.

[12] Olivier Dousse, Petteri Mannersalo, and Patrik Thiran. Latency of Wireless Sensor Networks with Uncoordinated Power Saving Mechanisms. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, 2004.

[13] Pierpaolo Bergamo, Daniela Maniezzo, Mario Gerla, and Gianluca Mazzini. A Low Computation Routing Algorithm for Sensor Networks. In *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, Sep 2004.

[14] Qing Fang, Jie Gao, and Leonidas J. Guibas. Locating and Bypassing Routing Holes in Sensor Networks. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, March 2004.

[15] Qun Li, Javed Aslam, and Daniela Rus. Hierarchical Power-aware Routing in Sensor Networks. In *International Conference on Internet Computing*, 2001.

[16] Sharat Visweswara, Rudra Dutta, and Mihail Sichitiu. Adaptive Ad-hoc Self-Organizing Scheduling for Quasi-Periodic Sensor Network Lifetime. In *Elsevier Computer Communications Journal*.

[17] A. Sinha and A. Chandrakasan. Dynamic Power management in Wireless Sensor Networks. In *IEEE Design and Test of Computers*, pages 62–74, March-April 2001.

[18] C.-K. Toh. Maximum battery life routing to support ubiquuitious mobile computing in wireless ad hoc networks. In *IEEE Communication Magazine*, pages 138–147, June 2001.

[19] Torsten Braun and Laura Marie Feeney. Power Saving in Wireless Ad hoc Networks Without Synchronization.

[20] Yang Yu, Bhaskar Krishnamachari, and Viktor K. Energy-Latency Tradeoffs for Data Gathering in Wireless Sensor Networks. In *IEEE Infocom*, 2004.