DATA RELIABILITY CONTROL IN WIRELESS SENSOR NETWORKS FOR DATA STREAMING APPLICATIONS

by

L. D. Tuan

Bachelor of Computer Engineering (Honours)

University of Tasmania, 2003

A thesis submitted in fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science and Engineering

University of New South Wales

2009

This thesis entitled:
DATA RELIABILITY CONTROL IN WIRELESS SENSOR NETWORKS FOR DATA STREAMING APPLICATIONS
written by L. D. Tuan
has been approved for the Department of Computer Science and Engineering

Supervisor: Prof. Sanjay Jha

Signature _____ Date _____

The final copy of this thesis has been examined by the signatory, and I find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Declaration

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Signature _____ Date _____

Abstract

This thesis contributes toward the design of a reliable and energy-efficient transport system for Wireless Sensor Networks. Wireless Sensor Networks have emerged as a vital new area in networking research. In many Wireless Sensor Network systems, a common task of sensor nodes is to sense the environment and send the sensed data to a sink node. Thus, the effectiveness of a Wireless Sensor Network depends on how reliably the sensor nodes can deliver their sensed data to the sink. However, the sensor nodes are susceptible to loss for various reasons when there are dynamics in wireless transmission medium, environmental interference, battery depletion, or accidentally damage, etc. Therefore, assuring reliable data delivery between the sensor nodes and the sink in Wireless Sensor Networks is a challenging task.

The primary contributions of this thesis include four parts. First, we design, implement, and evaluate a cross-layer communication protocol for reliable data transfer for data streaming applications in Wireless Sensor Networks. We employ reliable algorithms in each layer of the communication stack. At the MAC layer, a CSMA MAC protocol with an explicit hop-by-hop Acknowledgment loss recovery is employed. To ensure the end-to-end reliability, the maximum number of retransmissions are estimated and used at each sensor node. At the transport layer, an end-to-end Negative Acknowledgment with an aggregated positive Acknowledgment mechanism is used. By inspecting the sequence numbers on the packets, the sink can detect which packets were lost. In addition, to increase the robustness of the system, a watchdog process is implemented at both base station and sensor nodes, which enable them to power cycle

when an unexpected fault occurs. We present extensive evaluations, including theoretical analysis, simulations, and experiments in the field based on Fleck-3 platform[1] and the TinyOS operating system. The designed network system has been working in the field for over a year. The results show that our system is a promising solution to a sustainable irrigation system.

Second, we present the design of a policy-based **S**ensor **R**eliability **M**anagement framework for Wireless Sensor Networks called SRM. SRM is based on hierarchical management architecture and on the policy-based network management paradigm. SRM allows the network administrators to interact with the Wireless Sensor Network via the management policies. SRM also provides a self-control capability to the network. This thesis restricts SRM to reliability management, but the same framework is also applicable for other management services by providing the management policies. Our experimental results show that SRM can offer sufficient reliability to the application users while reducing energy consumption by more than 50% compared to other approaches.

Third, we propose an **E**nergy-efficient and **R**eliable **T**ransport **P**rotocol called ERTP, which is designed for data streaming applications in Wireless Sensor Networks. ERTP is an adaptive transport protocol based on statistical reliability that ensures the number of data packets delivered to the sink exceeds the defined threshold while reducing the energy consumption. Using a statistical reliability metric when designing a reliable transport protocol guarantees the delivery of adequate information to the users, and reduces energy consumption when compared to the absolute reliability. ERTP uses hop-by-hop Implicit Acknowledgment with a dynamically updated retransmission timeout for packet loss recovery. In multihop wireless networks, the transmitter can overhear a forwarding transmission and interpret it as an Implicit Acknowledgment. By combining the statistical reliability and the hop-by-hop Implicit Acknowledgment loss recovery, ERTP can offer sufficient reliability to the application users with minimal

[1] Fleck-3 is a wireless sensor network hardware platform which has been developed by CSIRO

energy expense. Our extensive simulations and experimental evaluations show that ERTP can reduce energy consumption by more than 45% when compared to the state-of-the-art protocol. Consequently, sensor nodes are more energy-efficient and the lifespan of the unattended Wireless Sensor Network is increased.

In Wireless Sensor Networks, sensor node failures can create network partitions or coverage loss which can not be solved by providing reliability at higher layers of the protocol stack. In the final part of this thesis, we investigate the problem of maintaining the network connectivity and coverage when the sensor nodes are failed. We consider a hybrid Wireless Sensor Network where a subset of the nodes has the ability to move at a high energy expense. When a node has low remaining energy (dying node) but it is a critical node which constitutes the network such as a cluster head, it will seek a replacement. If a redundant node is located in the transmission range of the dying node and can fulfill the network connectivity and coverage requirement, it can be used for substitution. Otherwise, a protocol should be in place to relocate the redundant sensor node for replacement. We propose a distributed protocol for **Mo**bile **Se**nsor **R**elocation problem called **Moser**. Moser works in three phases. In the first phase, the dying node determines if network partition occurs, finds an available mobile node, and asks for replacement by using flooding algorithm. The dying node also decides the movement schedule of the available mobile node based on certain criteria. The second phase of the Moser protocol involves the actual movement of the mobile nodes to approach the location of the dying node. Finally, when the mobile node has reached the transmission of the dying node, it communicates to the dying nodes and moves to a desired location, where the network connectivity and coverage to the neighbors of the dying nodes are preserved.

Dedication

To my parents

and Tina Chau Nguyen

Acknowledgements

I am deeply grateful to my supervisor, Prof. Sanjay Jha, whose thoughtful guidance, insightful vision, and continuing support have leaded me to grow immensely over the last four years. One of his best attributes as a supervisor was the perfect balance he achieved between granting me the freedom to pursue research problems that interested me and always being involved to guide the overall direction of my research that kept me on the right track. I am proud to be one of his students, and I am thankful for the opportunity to learn from Sanjay. Thank you!

Dr. Wen Hu has taught me more than I can describe in these short lines. Throughout my PhD, Wen has been a friend, an insightful co-supervisor, and a guide in a field of research that was new and uncertain. The path to my PhD would have been a lot less rewarding without Wen. I am thankful to Wen for his valuable comments, his tremendous ongoing help and support, and for many fun and motivating discussions. His ability to ask the challenging questions that were truly fundamental to the problem at hand forced me to think deeper about the ideas and come up with more general or more elegant solutions.

I am also indebted to all the members of Network Research Lab (NRL) at the University of New South Wales. Long discussions with my friends Sarfraz Nawar, Nadeem Ahmed, Jerry Quan Jun Chen, Yuvraj Kris Rana have significantly improved my work. I am particularly thankful to Dr. Nandan Paramesh, whose research on network management, provides part of the foundation for thesis. I have immensely enjoyed being part of one of the most collaborative and productive groups that I have seen.

I am also grateful to all the members in the CSIRO ICT Centre, particularly Dr. Peter Corke, who invited me to the lab. I want to thank Peter for this opportunity and for his support, which had such an immense positive impact on my PhD. The lab has provided the most comfortable working environment and all the facilities that I need for my experiments. I am also grateful to Dr. Zvi Rosberg, whose mathematical insight is sharp and accurate. My work on the Implicit Acknowledgment has greatly benefited from his suggestions and his kind encouragement. I also want to thank Dr. Ren Ping Liu for his useful discussions, which have inspired many new research directions in the development of a transport protocol for Wireless Sensor Networks.

My research would not have been possible without the financial support from the University of New South Wales and research grants from CSIRO. I am indebted to these institutions for their support.

Finally, I would like to thank my parents for their endless love, support and encouragement. Although our life paths may have been tortuous, your care and love have been immutable over my life. I am indebted to my dear Tina Chau Nguyen for all her support, love and joy. I simply could not have reached this stage without her. Thank you with all my heart!

Contents

Chapter

List of Tables

Table

List of Figures

Figure

Acronyms

WSN Wireless Sensor Network.

eACK Explicit Acknowledgment.

iACK Implicit Acknowledgment.

NACK Negative Acknowledgment.

HBH Hop-by-Hop.

E2E End-to-End.

RTO Retransmission Timeout.

ADSL Asymmetrical Digital Subscriber Line.

SN Sequence Number.

SRM Sensor Reliability Management.

ERTP Energy-efficient and Reliable Transport Protocol.

RSSI Received Signal Strength Indicator.

Moser Mobile Sensor Relocation Protocol.

Chapter 1

Introduction

The recent advances in micro-electro-mechanical system, wireless communication technology, and digital electronics have enabled the development of low-cost, low-power, small size distributed devices, a reality. Such small devices are called sensor nodes. Sensor nodes, which comprise sensing, data processing, and wireless communicating components, are capable of local processing and transmitting information wirelessly to base stations thus establishing a Wireless Sensor Network (WSN). WSNs offer significant advances over traditional networks and can be applied in many applications because of their flexibility, cost-effectiveness, and ease of deployment. Detecting environmental hazards, monitoring remote terrain, monitoring building structures and tracking and surveillance of borders are among many common sensor network applications. Other applications include managing complex physical systems such as airplane wings and ecosystems.

However, this new area in WSNs presents many new technical challenges for the research communities as well as many untapped opportunities for a diverse set of industries and early adopters, from environmental sensing to ubiquitous computing. In this thesis, we address some important networking problems associated with this nascent field that could limit this broad vision.

1.1 Chapter Organization

This chapter is organized as follows. In Section 1.2, we provide a brief overview of WSNs including the details of currently available sensor node hardware, WSN challenges, and WSN applications. In Section 1.3, we present the motivation of our work. In Section 1.4, we present the problem that this thesis aims to solve and the contributions of this thesis. Finally, Section 1.5 outlines the structure of the remaining thesis.

1.2 Overview of Wireless Sensor Networks

1.2.1 Overview

The notion of sensor networks has been around for over two decades, but recently the coming together of sensing and wireless communications has revolutionized the field and enabled significant advances. Early sensor networks emerged in the 1980s such as radar networks used in air traffic control systems and the national power grid. However, these early incarnations were wired networks. WSNs today comprise tiny and distributed sensor nodes that are capable of sensing, collecting, and transmitting information wirelessly to base stations for archiving, processing, or further analysis. Each sensor node has an embedded processing capability, onboard storage and communication capability, and has multiple onboard sensors interfacing with the physical environment such as temperature sensors, humidity sensors, etc.

Figure 1.1 shows examples of three common commercial sensor nodes from different vendors: Mica-2 from Crossbow Technology [1], Tmote Sky from Sentilla [2], and Robomote from the University of Southern California [3]. Typically, a sensor node has the following components:

- Microcontroller and Memory: The microcontroller controls functionality of other components in the sensor node. Memory is included for storage of data. Typically, flash memories are used due to their low cost, energy-efficiency, and large

(a) Mica-2 (b) Tmote Sky (c) Robo Mote

Figure 1.1: Sensor Platform.

storage capacity.

- Sensors and Analog-to-Digital Converter: The sensors sense data from the environment and send it to the microcontroller. If they are analog sensors, the signals are sent to an analog-to-digital converter which converts them to a digital format and then sends them back to the microcontroller.

- Transceiver: This unit transmits and receives radio signals or optical signals. Radio Frequency RF-based communication is the most common that is used in most of the wireless sensor platforms.

- Power Source: The main consumers of power in the sensor node are the sensors themselves, communications, and data processing. Batteries and solar cells are the main sources of power for sensor nodes.

CSIRO [4] has developed a WSN platform called Fleck-3 [5], a platform for real-world outdoor sensor networks for use in environmental monitoring, agriculture, etc. The Fleck-3 is the first device that incorporates a real-time clock allowing the CPU to attain the deepest sleep state while reducing the time-keeping overhead on the microcontroller. In addition, the Fleck-3 also provides more reliable radio front-end and a large transmission range of up to 1 km [6].

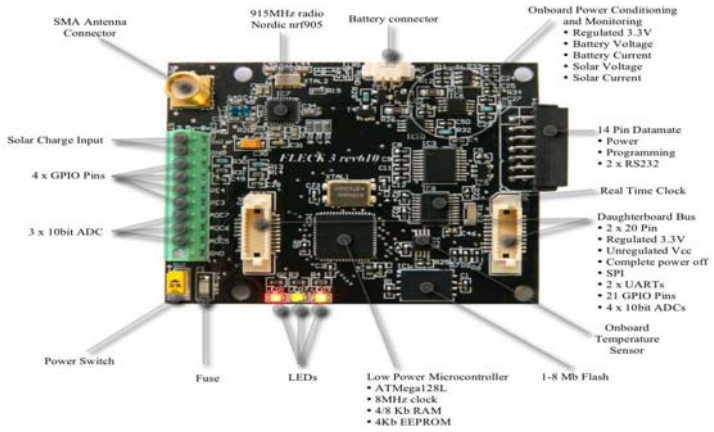


Figure 1.2: Fleck-3 platform.

This thesis has used Fleck-3 primarily for experimental evaluation. As shown in Figure 1.2, the Fleck-3 is based on the Atmel Atmega128 micro-controller running at 8 MHz. The Fleck-3 uses the packet-based Nordic NRF905 transceiver for communication. The NRF905 radio works in the 915MHz Industrial, Scientific and Medical (ISM) band and supports a bit rate of 100 kbps with Manchester encoding, providing a data-link operating at 50 kbps. The NRF905 radio uses GFSK modulation and provides a large transmission range of up to 1 km using standard unity-gain quarter-wavelength antennas.

1.2.2 Challenges

WSNs represent a new class of distributed systems that operate under a new set of constraints. Unlike traditional networks, where the focus is on maximizing channel throughput or minimizing node deployment, prolonging the network lifetime and building a robust data collecting system are the primary concern for WSN research communities. The challenges of WSNs are briefly summarized as follows:

- Limited Access: WSNs are usually deployed in a remote environment that is inaccessible or expensive to access, making the sensor nodes largely untethered

and unattended. Inaccessibility, as well as a large number of sensor nodes, implies that they must operate without human attendance. Thus, it is required that WSNs must possess self-organizing capabilities.

- Resource Constraints: Sensor nodes are small-scale devices and are very limited in both the amount of energy they can store or harvest from the environment and resources (CPU performance, memory, and wireless communication bandwidth). In most WSN deployments, there is only a finite source of energy because sensor nodes are often battery-powered. Moreover, the unattended nature of sensor nodes and the hazardous sensing environments preclude battery replacement as a feasible solution, while many sensor network applications demand that the network must operate for long periods of time. To reduce energy consumption, most of the components, including the CPU and radio, are likely to be turned off for most of the time and thus, the number of messages exchanged in the network should be minimized. Therefore, it is challenging to utilize energy resources in the most efficient way.

- Network Dynamics: Node mobility, node failures, and environmental interference can cause high degree of dynamics in WSNs. Because sensor nodes are closely tied to the changing physical world, they experience extreme dynamics. Environmental factors can dramatically influence propagation characteristics of a low-power radio frequency and can create high dynamics even in stationary configurations. Sensor nodes also experience extreme variation in network connectivity and are subject to potentially harsh environmental conditions including node failures, frequent network topology changes, or even network partitions. To cope with resource limitations in the presence of such dynamics, WSNs should adapt to changes in the environment.

Figure 1.3: The sensor node deployed at one of the pump sites at the Burdekin Site. (Top) Sensor node details. (Bottom) (B) Sensor node housing. (C) Bore containing water level sensor. (D) Pump. (E) Flow meter. (F) EC sensor. (G) Tank.

1.2.3 Applications

WSNs have a broad application domain including environmental monitoring, real-time target tracking, structural monitoring, health care, etc. Some significant WSN deployments are outlined below.

- Environmental Monitoring [7–10]: Examples of WSN applications to environmental monitoring are water quality monitoring, air quality monitoring, flood detection, fire detection, etc. The recent WSN deployment [7] at Burdekin, Queensland for water quality (water flows, water level, and salinity) monitoring that will be described in Chapter 3 is one such example. The application streams sensed data periodically from many scattered sensors to the sink node which in turn relays them via IP network for processing offline. Figure 1.3 shows a sensor node deployed at one of the pump sites. The flow meter and the electrical conductivity sensor (to measure salinity) are mounted in the pipe connecting the pump to the reservoir tank. The pressure sensor is mounted in an observation bore next to the pump.

- Real-time Target Tracking [11–13]: WSNs can be used for battlefield surveillance and position tracking of the enemy. Critical terrains, approach routes, paths and straits can be covered with sensor networks for surveillance. For example, Bokareva et al. [13] designed a surveillance system which can detect and classify multiple targets (e.g., vehicles and troop movements) using off-the-shelf wireless sensor nodes, capable of sensing acoustic and magnetic signals generated by different target objects. Such a system is capable of performing detection and tracking of targets as well as sending real time enemy mobility information to a command center.

- Structural Monitoring [14]: WSNs can be used to monitor vibration or to assess integrity of structures such as buildings, bridges, aero-space structures and off-shore oil rigs. For example, Xu et al. [14] discuss Wisden, a WSN data acquisition used for structural health monitoring, which was deployed on a realistic large structure. Wisden has tens of wireless sensor nodes, placed at various locations on a large structure. Each node measures structural vibrations by using a vibration card specifically designed for vibration sensing and sends the data to a base station. Such a system promises enormous benefits such as ease and flexibility of deployment in addition to low maintenance and deployment costs.

- Health Care [15, 16]: WSNs can be used for the long-term monitoring of a patient's health. Patients can benefit from WSN monitoring by using it as part of a diagnostic procedure, for achieving optimal maintenance of a chronic condition, or during recovery from an acute event or surgical procedure. An example of the WSN health care system is a Wearable Wireless Body/Personal Area Network (WWBAN) [16], which is designed for health monitoring. WWBAN is comprised of a number of physiological sensors and location sensors that monitor vital signs and environmental information (temperature, humidity, and light). WWBAN allows long-term, unobtrusive, ambulatory health monitor-

ing for the users with instantaneous feedback about their current health status and real-time updates of their medical records. Such a system can be used for computer-supervised rehabilitation in various conditions, and even for early detection of medical conditions.

1.3 Motivation

In most WSN applications, the common task of sensor nodes is to sense the environment and send the sensed data to a sink node. Thus, data reliability is one of the most important requirements in WSNs. The effectiveness of a WSN depends on how reliably the network can deliver the sensed data from sensor sources to a sink. In order to offer sufficient information required by the applications, it is important to ensure reliable data delivery between sensor nodes and base stations at a desired level.

However, because WSNs are subject to the limitations and constraints of the real world and interact closely with the physical environment in which they reside, ensuring data reliability is a challenging task. Data transfer in WSNs is more susceptible to loss than over wired networks such as Internet. This is because in wired networks data loss occurs primarily due to congestion, whereas there are many reasons for data loss in WSNs such as node failures, environmental interference, nodes joining/leaving a network, power depletion, etc. In addition to this, energy conservation needs to be taken into account because it is a critical resource in WSNs.

Moreover, sensor nodes in WSNs are prone to failure. These failures may occur upon the deployment or over time after the deployment because the node operation may drain the power, or external factors may physically damage part of the nodes such as fire or extreme heat, malicious activity, accidentally damage, extended use, etc. In many deployments, it is usually difficult to access the deployed nodes for replacement. If sensor node failures cause the network to be disconnected or lack other desired properties, data reliability is significantly affected.

For these reasons, ensuring reliable data delivery between sensor nodes and the base station and energy conservation are a crucial and challenging task. This is in contrast to traditional networks, in which energy is usually not a major concern and reliable networking has been well studied and applied with great success in support of everyday applications.

Definition 1. *The data reliability α of a sensor node ($0 < \alpha < 1$), is described by the probability of a data packet being delivered from the sensor node to the sink.*

Definition 2. *Data reliability management is the task that ensures data reliability of each sensor node in the network.*

1.4 Thesis Contributions

In this thesis, we investigate some of the key solutions for supporting data reliability in WSNs. The contributions of this thesis are:

1. A cross-layer design for reliable data transfer in WSN data streaming applications.
2. A policy-based management framework for data reliability in WSNs.
3. An energy-efficient and reliable transport protocol for WSNs.
4. An energy-efficient sensor relocation protocol for maintaining network connectivity and coverage.

The contributions of this thesis are comprised of four parts. First, **we design, implement, and evaluate the cross-layer communication protocol for reliable data transport** for WSN data streaming applications. We employ reliable algorithms in each layer of the communication stack. At the MAC layer, a CSMA MAC protocol with a Hop-by-Hop Explicit Acknowledgment (HBH eACK) loss recovery is employed. To ensure the End-to-End (E2E) reliability, the maximum number of retransmissions

is estimated and used at each sensor node. At the network layer, an E2E Negative Acknowledgment (NACK) with an aggregated positive Acknowledgment mechanism is used. By inspecting the sequence numbers on the packets, the base station can detect which packets were lost. The base station sends a NACK to a source after receiving a preset number of packets from the source. In addition, to increase the robustness of the system, a watchdog process is implemented at both the base station and sensor nodes, which enable them to power cycle when a unexpected fault occurs, e.g., sensor nodes hang up, the radio does not work properly, sensors are malfunctioning, etc. The designed sensor network system has been working in the deployed field for over a year. The collected results show that our sensor network system is a promising solution to allow gathering of water quality data to establish a sustainable irrigation system.

Second, we argue that a reliability management framework is necessary for controlling data reliability in WSNs. We study the technical challenges for reliability control in WSNs and present the design of **a policy-based reliability management framework for WSNs called SRM**. SRM is based on a hierarchical management architecture and on the policy-based network management paradigm formulated by [17]. SRM allows the network administrators to interact with the WSN via the management policies. SRM also provides a self-control capability to the network. This thesis restricts SRM to reliability management, but the same framework is also applicable for other management services by providing the management policies. SRM consists of four core modules: a user policy specification module, an evaluation module, a decision making module, and an action module. The cooperation among these modules enables the management framework to be efficient and adaptive to the network dynamics. Our experimental results show that SRM can offer sufficient reliability to the applications while significantly reducing energy consumption when compared to other approaches.

Third, we present **an Energy-efficient and Reliable Transport Protocol (ERTP)**, which is tailored for WSN data streaming applications. ERTP is an adaptive

transport protocol based on a statistical reliability that ensures the number of data packets delivered to the sink exceeds the defined threshold, while reducing the energy consumption. Using a statistical reliability metric when designing a reliable transport protocol guarantees the delivery of adequate information to the users, and reduces energy consumption when compared to the absolute reliability. ERTP is comprised of two components: a HBH reliability component and a HBH Retransmission TimeOut (RTO) component. The first component ensures the required E2E reliability by dynamically controlling the maximum number of retransmissions for each data packet in each intermediate node based on the channel quality. ERTP uses HBH Implicit Acknowledgment (iACK) for loss recovery. The HBH iACK mechanism operates by the transmitter overhearing the packet being forwarded by the receiver to its next hop and considers this as an iACK. The transmitter will retransmit the packet if it has not received the iACK after a time-out interval. Determining how long the node should wait for an iACK is non-trivial and depends on the time it takes a packet to be forwarded by the downstream node. A premature RTO value for HBH iACK may increase sensor energy-consumption because transmitters will send duplicate packets. On the other hand, a large RTO value tends to increase transmission latency and thus reduces network goodputs. In order to achieve energy efficiency, ERTP dynamically adjusts the RTO value at each node by observing the channel quality. By combining the statistical reliability and the HBH iACK loss recovery, ERTP can offer sufficient reliability to the application users with the minimal energy expense. Our extensive simulations and experimental evaluations show that ERTP can reduce energy consumption by more than 45% when compared to the state-of-the-art protocol. Consequently, sensor nodes are more energy-efficient and the lifespan of the unattended WSN is increased.

In WSNs, sensor node failures can create network partitions or coverage loss which can not be solved by providing reliability at higher layers of the protocol stack. In the final part of this thesis, we investigate the problem of maintaining the network

connectivity and coverage when the sensor nodes fail. Although this work handles a very important aspect of reliability, it is peripheral to the reliability protocol work described in rest of this thesis and hence has been added as an appendix. We consider a hybrid sensor network where a subset of the nodes has the ability to move at a high energy expense. When a node has low remaining energy (dying node) and it is a critical node which constitutes the network, i.e. a cluster head, it will seek a replacement. If a redundant node is located in the transmission range of the dying node and can fulfill the connectivity and coverage requirement, it can be used for substitution. Otherwise, a protocol should be in place to relocate the redundant nodes to the location of the dying node for replacement. **We propose a distributed protocol for MObile Sensor Relocation problem called Moser**. Moser works in three phases. In the first phase, the dying node determines if network partition occurs, finds an available mobile node, and asks for replacement by using flooding algorithm. The dying node also decides the movement schedule of the available redundant node based on certain criteria. The second phase of the Moser protocol involves the actual movement of the mobile redundant node to approach the transmission range of the dying node. Finally, when the mobile redundant node has reached the transmission range of the dying node, it communicates to the dying node and moves to a desired location, where the network connectivity and coverage to the neighbors of the dying nodes are preserved.

1.5 Thesis Organization

The rest of this thesis is organized as follows:

- **Chapter 2** provides a brief overview of the related work on transport protocols, network management, and maintaining network connectivity and coverage in WSNs.
- **Chapter 3** presents our reliable cross layer design solution for data transfer in WSN data streaming applications.

- **Chapter 4** presents the details of the SRM management framework.
- **Chapter 5** presents the design of the proposed ERTP in detail.
- **Chapter 6** summarizes the work presented in this thesis and discusses some future directions that can be pursued to improve data reliability for WSNs.

Chapter 2

Literature Review

In this chapter, we provide a brief survey on the transport protocols, network management, and maintaining connectivity for WSNs.

2.1 Transport Protocols for WSNs

Applying transport protocols from traditional networks such as TCP [18], for example, to achieve reliability in WSNs is hampered by several mismatches since sensor networks have different constraints from the traditional wired networks. First, energy constraints are paramount in sensor networks since in most of the cases sensor nodes can not be recharged. In addition to the energy constraints, low wireless bandwidth makes in-network processing both feasible and desirable. Moreover, the primary reason for packet loss in traditional wired nets is network congestion while in WSNs, packets may be lost for various reasons such as environmental inference, contention, congestion, etc.

There is no single protocol covering all the applications, but different solutions have been developed for a single application or small sets of application. We categorize the transport protocols for three different types of applications:

- Data Streaming Applications: Applications require periodic data reporting from sensor nodes to a sink. In these applications, sensor nodes continuously generate

and send the sensed data packet to a sink. Often, the reliability requirements for data streaming applications is determined by the quantity of data packets delivered to the sink rather than the reliability of each data packet. Therefore, these applications can tolerate a certain degree of packet losses. In addition to reliability requirement, energy is also a primary concern, since sensor nodes are usually battery-powered. Examples of transport protocols designed for this type of applications include Surge Reliable [19], RMST [20], Wisden [14], and RCRT [21].

- Bulk Data Applications: Applications require reliable delivery of block data such as disseminating new codes or new queries into the network. In these applications, a source has a large block of data and needs to deliver to a destination with a very high reliability requirement, i.e. 100%. For example, during code transferring from a sink to a sensor node, if a packet is lost, the code can not be reconstructed at the node. Thus, an absolute reliability is required in this type of application. In addition to the high reliability requirement, these applications prefer to minimize the transfer time, i.e., achieves high network throughput. Examples of transport protocols designed for this type of application include PSFQ [22], RBC [23], and Flush [24].

- Event Data Applications: Applications require event detection such as target tracking, fire detection, etc. These applications do not require absolute reliability of all data packets, but require successful event detection. Examples of transport protocols designed for this type of applications are ESRT [25], PORT [26], and STCP [27].

Next, we will briefly discuss several transport protocols for each type of applications.

2.1.1 Transport Protocols for Data Streaming Applications

2.1.1.1 Surge: a Reliable Multihop Routing in WSNs

Woo et al. proposed **Surge Reliable** [19], a reliable multi-hop routing for WSNs. Although Surge Reliable is a routing protocol, it can be thought of as implementing a simple transport protocol for WSN data streaming applications. Surge Reliable uses link quality as its routing metric. Surge Reliable dynamically forms a reliable spanning tree that covers every node in the network, using link connectivity estimation and neighbourhood table management techniques. In Surge Reliable protocol, each node periodically measures the link qualities between itself and its neighbours, and selects "the best" neighbour as its parent to forward the data to the base station. The two important components of Surge Reliable protocol are: the link estimation and neighborhood table management components.

- Link Estimation: The objective of link estimation is to estimate the channel quality. Surge Reliable uses a Window Mean with Exponentially Weighted Moving Average (WMEWMA) for link estimation. WMEWMA(t, α) computes an average success rate over a time period and smoothens the average with EWMA. The tuning parameters are t and α, where t is the time window represented in number of message opportunities and α controls the history of the estimator. In Surge Reliable, the minimum message rate and the periodic timer event are assumed to be known so the estimator can infer losses prior to next packet reception.

- Neighborhood Table Management: A node performs neighbour discovery by monitoring information about nodes from which it receives packets. Link estimation is used to determine which nodes should be selected as the neighbours. The neighborhood management has two policies: the insertion policy, and the eviction and reinforcement policy. The insertion policy determines upon hear-

ing from a node whether to insert it into the neighborhood table or not. Surge Reliable uses an adaptive down sampling scheme, which sets the probability of insertion as the ratio of the neighbor table size to the number of distinct neighbours. The eviction and reinforcement policy keeps a frequency count for each entry in the table. On insertion, a node is reinforced by incrementing its count. A new node will be inserted in the table if there is an entry with a count of zero; otherwise, the count of all entries is decremented by one and the new candidate is dropped.

Surge Reliable is the current state-of-the-art protocol for data streaming in WSNs. However, it does not guarantee E2E reliability. Moreover, it is not energy-efficient when the link quality is good, since it introduces a significant number of acknowledgments (ACK).

2.1.1.2 RMST: Reliable Multi-Segment Transport

Akan et al. proposed **RMST** [20], a Reliable Multi-Segment Transport. RMST is a reliable transport layer protocol which is built on the top of Directed Diffusion [28] routing protocol. Reliability in RMST refers to the eventual delivery of all fragments related to a unique RMST entity to the subscribing base station. A unique RMST entity is a data set which comprises one or more fragments coming from the same source. An unfragmented data entity has an application specific attribute (RMSTNo) that can be differentiated from other fragments during the data transfer. Each fragment has a sequential fragment ID (FragNo) and the total number of fragments for a unique RMST entity is known (MaxFrag).

RMST uses HBH NACK, which requests only the lost packets for retransmission. When a node finds a missing packet, a NACK is sent to the next node on the reinforced path toward the source. If the lost packets are found in the local cache, the retransmissions will occur. Otherwise, the NACK message is forwarded to the next node on the

reinforced path toward the source.

RMST has two working modes: the non-caching mode, and the caching mode. In the non-caching mode, only the base station and the source node are involved in loss detection and recovery. By inspecting the sequence number stamped in the received packet, the base station is able to detect missing data packets and sends the requests for retransmissions to the source node, where the missing packets are retransmitted. In the caching mode, each intermediate node on a path from the source node to the base station caches the data packets it received. The timer handler inspects the holes in the sequence number and sends a NACK for the holes that have aged for too long. To conserve on control traffic, multiple hole numbers are aggregated into a single NACK.

The drawback of RMST is that it is tightly bound to Directed Diffusion routing protocol [28] in which packet losses are recovered HBH using caches in the nodes along the path to the sink. Furthermore, RMST is not scalable because it requires each intermediate nodes to cache all packets received from each upstream source. Memory limitation on resource-constrained sensor nodes requires intelligent caching strategies to be considered.

2.1.1.3 Wisden: A WSN for Structural Monitoring

Xu et al. presented **Wisden** [14], a WSN system for structural-response data acquisition. Wisden is designed for structural health monitoring, which continuously collects structural response data from a multi-hop network of sensor nodes for detection and localization of damages in the structures. Wisden provides reliable data transport by using a hybrid recovery scheme that recovers packet losses both HBH and E2E. Wisden aggressively uses overhearing and piggybacking techniques in order to detect and repair packet losses.

In the HBH NACK-based reliability scheme, each source stores the generated vibration data in its EEPROM, and then transmits the data to its parents. Parents keep track of the sequence number of packets that they have received. A gap in the

sequence number of sent packets indicates packet loss. Each node maintains a list of missing packets. When a loss is detected, the "missing packets" list is piggybacked in outgoing transmissions, and children infer loss by snooping the channel. Nodes keep a small cache of recently transmitted packets, from which they can repair losses reported by their parents.

In addition to the HBH NACK-based scheme, Wisden also uses an E2E NACK-based scheme for loss recovery. The approach used in the E2E recovery scheme is very similar to the HBH scheme. It leverages the fact that the base station has significantly more memory and can keep track of all missing packets. When a node detects a packet loss but does not have a cached copy of that packet in its queue, it adds the recovery request to its missing packets list. This request is propagated downward to the next hop toward the source (using the same mechanisms described for HBH recovery) until it reaches the node that has the lost packet. Since the source maintains generated packets in its EEPROM, it can repair the missing packet.

Wisden is similar to the scheme proposed in RMST discussed eariler in Section 2.1.1.2. Both require each intermediate nodes to cache all the packets received from each upstream source. For resource-constrained sensor nodes, the Wisden scheme is not scalable when the number of sensor nodes in the network increases.

2.1.1.4 RCRT: Rate Controlled Reliable Transport for WSNs

Paek et al. proposed **RCRT** [21], a centralized rate-controlled reliable transport protocol for WSNs. RCRT aims to ensure reliable delivery of data from a collection of sensor nodes to a base station, while avoiding congestion collapse. RCRT is a centralized protocol, in which the traffic management functionality resides at the sink.

RCRT contains four components: E2E retransmission, congestion detection, rate adaptation, and rate allocation. RCRT uses E2E NACK for loss recovery. Each sensor node stores a copy of every data packet that it sent out to the sink. The sink keeps track of sequence numbers of packets that it received for each flow. A gap in the

sequence number of received packets indicates packet loss. When losses are detected, the E2E retransmission component inserts sequence numbers of the lost packets into a list. Entries in this list of missing packets are sent as NACK messages by the sink to each source. The congestion detection component observes the behaviour of packet loss across every flow in the network, and decides if the network is congested. In RCRT, if the E2E losses are repaired slower than the estimated time to recover loss (a congestion indicator), network is congested. The time to recover loss is set as a multiple Round Trip Time (RTTs). Once it determines that the network is congested, the rate adaptation component estimates the total sustainable traffic, called R(t), in the network. Then, the rate allocation component decreases the flow rates $r_i(t)$ for flow i to achieve R(t), while conforming to defined policy. Conversely, when the network is not congested, the rate adaptation component additively increases the overall rate R(t), and the rate allocation component determines corresponding maximum rate $r_i(t)$ for each flow i.

RCRT focuses on achieving 100% reliability and high throughput via congestion control without consideration of energy-efficiency. It is not designed for the applications where the energy efficiency is highly concern, i.e. environmental monitoring applications.

2.1.2 Transport Protocols for Burst Data Applications

The burst data applications require reliable delivery of block data such as disseminating new codes or new queries into the network. In these applications, a source has a large block of data and needs to deliver to a destination with a very high reliability requirement, i.e. 100%. In addition to the high reliability requirement, these applications prefer to minimize the transferring time, i.e. achieve high network throughput. The following subsections review a number of transport protocols in literature designed for this class of applications.

2.1.2.1 PSFQ: Pump Slowly, Fetch Quickly

Wan et al. proposed **PSFQ** (Pump Slowly Fetch Quickly) [22], a transport protocol designed for reprogramming WSNs. PSFQ aims to distribute data from a sink to sensor nodes by pacing data at a relatively slow speed, but allowing sensor nodes that have data loss to quickly fetch any missing segments from their neighbours. PSFQ uses a HBH NACK scheme and requires that each node caches the data packets it receives.

PSFQ contains three components: the pump operation, the fetch operation, and the report operation. In the pump operation, the source transmits data packets one by one to its neighbours every period of time. Each packet contains a sequence number that allows sensor nodes to identify the packet loss. When a sensor node receives a packet, i.e. p_i, the first time, it will store the packet p_i in its local data cache and broadcast the packet to the next-hop with a random delay. The sensor node will drop the packet p_i if it is already in the data cache. Also, forwarding is suppressed when the sensor node finds that four or more of its neighbors have already forwarded the same packet, since expected additional coverage achieved by the forwarding the packet tends to be small. Because the time transfer between the different segments is comparably large, the pumping operation is considered as slow pumping.

The sensor node will go into fetch mode once a sequence number gap in a data fragment is detected, i.e. packet loss. The fetch operation corresponds to a NACK or a retransmission request and is triggered by observing the sequence number. When a sensor node receives a packet out-of-sequence, i.e. p_{i+1} is received where p_i has not been received, it will issue a request (NACK message) for retransmission of all the lost packets with sequence numbers lower than p_{i+1} from its neighbors. If the neighbors do not have the missing segments, they forward the NACK packet further upstream until it eventually reaches the node having the missing packets. As soon as the node receives the request, it switches to the pumping mode and starts forwarding the missing packets. The NACK packets are broadcasted and any upstream neighbor having some

of the missing segments is invited to respond. To reduce the collisions among these packets, the nodes use random delays before replying.

Finally, the report operation is initiated by the source to check data delivery status information to users. To reduce the number of report messages, each node can append its own feedback information to the original report message sent by the most distant target node as it propagates toward the source that initially requested the report.

2.1.2.2 RBC: Reliable Bursty Convergecast in WSNs

Zhang et al. proposed **RBC** [23], a Reliable Bursty Convergecast in WSNs. RBC is designed for transferring a large burst of packets from sensor sources to a sink. To improve channel utilization, RBC uses a window-less block acknowledgment scheme that enables continuous packet forwarding in the presence of packet and acknowledgment loss. The sender S organizes its packet queue as a number of linked lists called virtual queues, denoted as $\{Q_0, Q_1, ..., Q_{M+1}\}$ (assuming that there are $M+2$ linked lists). The virtual queues are ranked such that a virtual queue Q_k ranks higher than Q_j if $k < j$. Each virtual queue buffers packets waiting to be sent or to be acknowledged, and Q_{M+1} collects a list of free queue buffers. The virtual queues are maintained as follows:

- When a new packet arrives at S to be sent, S puts the packet into the head buffer of Q_{M+1}.
- Packets stored in a virtual queue Q_k $(k > 0)$ will not be sent unless Q_{k-1} is empty. Packets in the same virtual queue are sent in FIFO order.
- After a packet in a virtual queue Q_k $(k \geq 0)$ is sent, it is moved to the tail of Q_{k+1}. However, if the packet has been retransmitted M times, it is moved to the tail of Q_{M+1}.
- When a packet is acknowledged to have been received, the buffer holding the packet is released and moved to the tail of Q_{M+1}.

To ameliorate retransmission-incurred channel contention, RBC introduces differentiated contention control, which ranks nodes by their queuing conditions and the number of times that the enqueued packets have been transmitted.

By maintaining the virtual queues, RBC provides window-less block acknowledgment. Moreover, this scheme allows new packets to be sent out without waiting for the previously sent packets to be acknowledged. As a result, the network throughput is improved.

2.1.2.3 Flush: a Reliable Bulk Transport Protocol for Multihop Wireless Networks

Kim et al. proposed **Flush** [24], a reliable transport protocol for WSNs designed for transferring bulk data across a multihop path from a source to a sink. Flush uses a sink-initiated control protocol to coordinate transfers, with E2E selective NACK and retransmissions to provide reliability.

To initiate a data transfer, the sink sends a request to a source in the network. After a request is made, Flush moves through four phases: topology query, data transfer, acknowledgment, and integrity check. The topology query phase probes the depth of a target node to tune the Round Trip Time (RTT) and compute a timeout at the receiver. The sink uses an estimate of the RTT to decide when to send a request for packet loss. In the data transfer phase, the source sends packets to the sink. On long paths, Flush pipelines packets over multiple hops. To minimize the transfer time, Flush proposed a distributed rate control algorithm, which dynamically estimates the sending rate that maximizes the pipeline utilization. The algorithm follows two rules: 1) A node should only transmit when its successor is free from interference. 2) A node's sending rate cannot exceed the sending rate of its successor. The rules allow sensor nodes to find and send packets at the maximum rate that will avoid intra-path interference and allow spatial reuse of the channel.

The sink also needs to keep track of packets it received. In the acknowledgment

phase, the sink sends the sequence numbers of the lost packets back to the data source. Similar to RCRT, Flush uses E2E NACK for loss recovery. Each NACK message can hold up to 3 sequence numbers. When the source receives a NACK packet, the source retransmits the missing data packets. This process repeats until the sink has received all the requested lost packets. The sink then verifies the integrity of the data. Integrity is checked at the level of both packets and data objects. If the integrity check fails, the sink discards the data and sends a fresh request.

PSFQ, RBC, and Flush are designed for bulk data transfer. These protocols aim to achieve 100 % reliability and high throughput. They are not designed for data streaming applications in which energy efficiency is highly concern but not throughput.

2.1.3 Transport Protocols for Event Data Applications

The event data applications require event detection such as target tracking, fire detection, etc. These applications do not require absolute reliability of all data packets, but require successful event detection. The following subsections review a number of transport protocols in literature for this class of applications.

2.1.3.1 ESRT: Event-to-Sink Reliable Transport

In [25], Akan et al. proposed **ESRT**, an Event to Sink Reliable Transport Protocol for E2E reliability based on the notion of event-to-sink reliability. ESRT achieves the reliable detection of an event and congestion avoidance by controlling the transmission rate of each source at the sink.

Assume that the sink must determine the transmission rate on the event features every τ time units. Here, τ represents the duration of a decision interval and is fixed by the application. At the end of each decision interval, the sink makes an informed decision based on reports received from sensor nodes during that interval. Let us denote r_i as the number of data packets received by the base station in the decision interval

i, and R as the number of data packets required for detection and extraction of event features in a decision interval. r_i can be computed by stamping source data packets with an event ID and incrementing the received packet count at the sink. If $r_i > R$, then the event features can be reliably detected. Otherwise, an appropriate action needs to be taken to achieve the desired reliability R. The main idea of ESRT is to configure the reporting rate, f, of source nodes so as to achieve the required event detection reliability, R, at the sink with minimum resource utilization. Based on the simulation studies, the authors observed that the achieved reliability shows a linear increase with reporting rate f until a certain $f = f_{max}$, beyond which the reliability drops. This is because of network congestion. Based on the simulation results, ESRT defines five network states:

- (NC,LR) : $f < f_{max}$ and $\eta < 1 - \epsilon$ (No Congestion, Low Reliability)
- (NC,HR) : $f \leq f_{max}$ and $\eta > 1 - \epsilon$ (No Congestion, High Reliability)
- (C,HR) : $f > f_{max}$ and $\eta > 1$ (Congestion, High Reliability)
- (C,HR) : $f > f_{max}$ and $\eta \leq 1$ (Congestion, Low Reliability)
- (OOR) : $f < f_{max}$ and $\eta > 1 - \epsilon \leq \eta \leq 1 + \epsilon$ (Optimal Operating Region)

The primary motive of ESRT is to achieve and maintain network operation in state OOR by dynamically adjusting the reporting frequency f. For example, if the network is in (NC, LR) state, the base station instructs the sensor nodes to increase the reporting rate f. If the network is in (C, HR) state, the base station instructs the sensor nodes to decrease the reporting rate f.

Although ESRT does not require packet retransmissions, it is not as energy-efficient as HBH loss recovery schemes since the rate decision is controlled centrally. Moreover, ESRT assumes that the sink can communicate with all sources directly, which may not be a reasonable assumption in practical WSN deployments.

2.1.3.2 PORT: A Price-Oriented Reliable Transport Protocol in WSNs

Zhou et al. proposed **PORT** [26], a Price-Oriented Reliable Transport protocol. PORT aims to provide fidelity of interested events while minimizing energy consumption.

PORT employs node price to measure the communication cost from a node to the sink. Node price is defined as the total number of transmissions attempts across the network from a source to a sink for achieving successful packet delivery. To ensure the fidelity of the collected events, PORT estimates the optimal reporting rate for each source based on the current contribution of the packets and the node price at each source. To improve the data reliability from a sensor source to a sink, each node in the network dynamically allocates its outgoing traffic based on the neighboring nodes' feedback of their node prices and the link loss rates between the neighbors. This approach can alleviate network congestion. PORT also employs a source reporting rate control mechanism which controls the source reporting rates based on the node prices of the source. For example, the source node with a high node price might slow down its transmission rate whereas the source node with a low node price may increase its transmission rate, if it still ensures that the sink can obtain enough information. Hence, the in-network congestion-avoidance mechanism and the E2E reporting-rate adjustment mechanism can provide fidelity of interested events while minimizing energy consumption.

2.1.3.3 STCP: Sensor Transmission Control Protocol in WSNs

Yogesh et al. proposed **STCP** [27], a Sensor Transmission Control Protocol for WSNs. STCP controls variable reliability, congestion detection and avoidance, and supports multiple flows in the network.

The sensor nodes need to establish an association with the base station via a session initiation packet before transmitting the packets. The session initiation packet

informs the base station the number of flows originating from the node, the type of data flow, the transmission rate, and the required reliability. When the base station receives the session initiation packet, it stores all the information, initiate the timers and other parameters for each flow, and acknowledges this packet.

STCP supports two types of data flow traffics: continuous and event-driven flows. For the continuous flows, an E2E NACK scheme is used. Since the base station is aware of the transmission rate of sensor nodes, the estimated arrival time for a packet traveling from a sensor node to the base station can be estimated. The base station will send a NACK if it does not receive the packet within the estimated time and sensor nodes retransmit packets upon receiving the NACK. For the event-driven flows, an explicit E2E ACK scheme is used. Each sensor node buffers the transmitted packets until it receives the ACKs from the base station. The sensor nodes also maintain a buffer timer that fires periodically. When the timer fires, packets in the buffer are assumed to be lost and thus, are retransmitted. When an ACK is received, the corresponding packet is deleted from the buffer.

In STCP, sensor nodes specify the required reliability for each flow in the session initiation packet. For the continuous flows, the reliability is measured as the fraction of packets successfully received. When the base station does not receive a packet within the expected time interval but the current reliability satisfies the required reliability, it may not send a NACK. For event-driven flows, the base station calculates reliability as a ratio of packets received to the highest sequence numbered packet received. Before transmitting a packet, the sensor nodes calculate the effective reliability assuming that the packet will not reach the base station. If the result is satisfactory, the node does not buffer the packet, thus saving memory space.

STCP adopts the method of explicit congestion notification. Each STCP data packet has a congestion notification bit in its header. Every sensor node maintains two thresholds in its buffer: t_{lower} and t_{higher}. When the buffer reaches the threshold

t_{lower}, the congestion bit is set with a certain probability. When the buffer reaches the threshold t_{higher}, the node will set the congestion notification bit in every packet it forwards. On receiving this packet, the base station informs the source of the congested path by setting the congestion bit in the acknowledgment packet. The source that receives the congestion notification will either route successive packets along a different path or slow down the transmission rate.

Similar to ESRT, STCP is not energy-efficient since the rate decision is controlled centrally. Moreover, STCP requires clock synchronization for all the sensor nodes in the network. To the best of our knowledge, STCP as well as PORT have not been evaluated in a real-testbed.

2.1.3.4 CODA: Congestion Detection and Avoidance

Wan et al. proposed **CODA** [29], an energy efficient congestion control scheme for WSNs. CODA detects congestion by periodically sampling the channel load and comparing the fraction of time that the channel is busy to the theoretical channel utilization. The system responds to congestion with a combination of HBH flow control and closed-loop regulation. There are two mechanisms for congestion control in CODA: open-loop HBH backpressure and closed-loop multi-source regulation mechanisms.

In the open-loop HBH backpressure mechanism, a node uses its local queue length to indicate the congestion level. If the queue length exceeds a pre-defined threshold, then there is congestion. Once congestion is detected, the receiver will broadcast a suppression message to its neighbours and at the same time make local adjustments to prevent propagating the congestion downstream. A node broadcasts backpressure messages as long as it detects congestion. Backpressure signals are propagated upstream toward the source. Nodes that receive backpressure signals will reduce their sending rates or drop packets based on the local congestion policy. When an upstream node (toward the source) receives a backpressure message, it decides whether or not to further propagate the backpressure upstream, based on its own local network conditions. For

example, depending on the local congestion policy a node may simply start to drop its incoming data packets upon receiving a backpressure message, preventing its queue from building up, rather than propagating the backpressure signal further upstream because of an overflowing queue.

In the closed-loop multi-source regulation mechanism, the sink will detect and control congestion. When the sink consistently receives a less than desired reporting rate, it can infer that packets are being dropped along the path, most probably due to congestion. Let r and (S_{max}) denote the source event rate and the maximum theoretical throughput (S_{max}) of the channel. When the source rate exceeds the channel capacity ($r > \eta S_{max}$) where η is a constant, a source is more likely to contribute to congestion and therefore closed-loop control is triggered. At this point, acknowledgements (ACKs) are used by the sources to determine their sending rates. A source triggers sink regulation when it detects ($r > \eta S_{max}$) by setting a regulate bit in the event packets it forwards toward the sink. Reception of packets with the regulate bit set forces the sink to send ACKs to regulate all sources associated with a particular data event. ACKs could be sent in an application specific manner. For example, the sink could send the ACK only along paths it wants to reinforce in the case of a directed diffusion [6] application. The reception of ACKs at sources would serve as a self-clocking mechanism allowing the sources to maintain the current event rate (r). When congestion is detected, the source will reduce the sending rate (r) according to some rate decrease function (e.g., multiplicative decrease).

However, the main drawback of CODA as well as ESRT is that they only take the loss due to congestion into account whereas there are additional reasons for data loss in WSNs such as environmental interference. Moreover, it does not consider data recovery methods in order to achieve reliability.

2.2 Network Management in WSNs

Network management is the process of managing, monitoring, and controlling the performance of a network [30]. WSNs have fundamentally different architecture than normal wired data networks due to their unique characteristics which have been discussed in Section 1.2.2. Based upon the information collection and communication strategy, there are three types of network management architectures: centralized, distributed, and hierarchical.

2.2.1 Centralized Management System

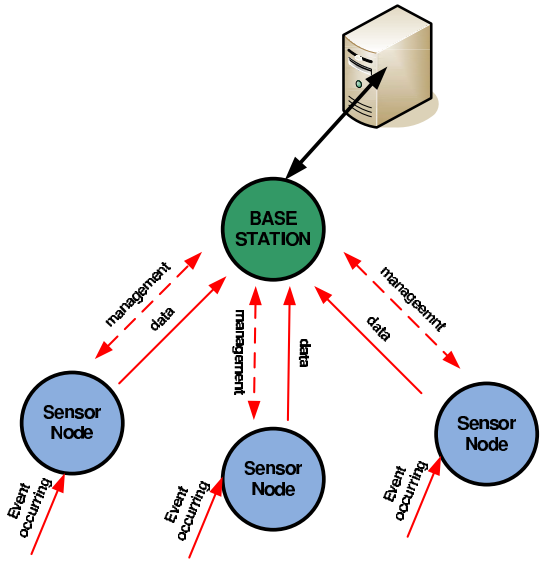


Figure 2.1: Centralized Architecture.

Figure 2.1 depicts a centralized management system. In a centralized system, there is a single manager station, i.e., base station, which controls the operations of the entire network. Centralized management system creates a trade-off between energy consumption, precision of control updates, and the size of the network which can be managed. In most of the system, the base station is connected to the computer, which has unlimited resources and has a comprehensive view of the performance of the net-

works. Thus, the base station is able to perform management tasks in a more efficient way than would be possible with a decentralized approach.

However, in centralized management systems, management information flows upwards from the sensor nodes to the base station, and the commands issued by the base station are sent down to the sensor nodes, causing a high energy consumption bottleneck. Moreover, centralized management systems have a single point of failure at the bottleneck. If a network partition occurs due to node failures, the portion that is disconnected from the base station is left without any management functionality.

Next, we discuss several centralized management systems for WSNs including MOTE-VIEW [31], SNMS [32], and SYMPATHY [33].

2.2.1.1 MoteView: a Sensor Network Monitoring and Management Tool

In [31], Turon proposed **MoteView**, a sensor network monitoring and management tool. MoteView is designed to be an interface between a user and a deployed network of wireless sensor nodes. The functionalities of MoteView include historical and real-time charting, topology map, and sensor-value gradient visualization, etc.

MoteView consists of four layers: the data access abstraction layer, the node abstraction layer, the conversion abstraction layer, and the visualization abstraction layer. Each of the four layers has a plug-in capability to provide modular extensions. The data access abstraction layer allows clients to access sensor network data. The node abstraction layer is responsible for storing the sensor node's meta data information such as name, set of sensors, configuration, and calibration coefficients. This layer also allows the users to configure the mote parameter settings, including radio frequency, power selection, sample rate, and custom calibration. The conversion abstraction layer converts and calibrates the raw sensor readings to final readings in the engineering unit. It also allows adding extension modules into a library of conversion for handling new unit types. Finally, the visualization layer provides a graphical display of sensor data in various representations: spreadsheet, chart with time, and network topology map. It

also allows the users to browse historical data and creates animated movies of the data.

The drawback of MOTEVIEW is that it does not allow networks to self-configure and requires an end-user to participate in the network management process.

2.2.1.2 SNMS: Application-cooperative Management for WSNs

Toll and Culler [32] proposed **SNMS**, a Sensor Network Management System which is designed for monitoring the health of WSNs. SNMS provides two core services: a query system to enable user-initiated acquisition of network health and performance data, and a logging system to enable recording and retrieving of system-generated events. The query system allows the users to collect and monitor the network parameters such as a node's available power level, temperature and humidity. The logging system enables the users to log the network parameters into a database.

The SNMS has two components: a collection component and a dissemination component. The collection component is a collection tree construction protocol which retrieves the network health from sensor nodes and transports it to the base station. The tree construction protocol constructs a tree only in response to a message sent from the root. Thus, a sensor node does not need to maintain a neighbour table or any explicit initiation of tree construction. Moreover, to maintain a high-quality tree, each node continually updates the parent selection as new messages arrive. To avoid contention while flooding the construction message, each node randomly staggers the retransmission time. The second component is a dissemination protocol called Drip [32], which provides an interface and stack for transport level reliable dissemination of messages. When a component such as a user or sensor node wants to make a query, it needs to select and subscribe to a particular dissemination channel. The Drip protocol then transports received messages on that channel to the registered component and returns a reply. As Drip is application independent, it is robust to network failures and is able to provide management functions even when the application fails.

2.2.1.3 SYMPATHY: Sympathy for the Sensor Network Debugger

In [33], Ramanathan et al. proposed **Sympathy**, a debugging tool for detecting and debugging failures in WSNs. Sympathy is designed for data streaming applications, which periodically gather data at a sink from many sensor sources. Sympathy primarily recognizes the network failures by the interactions among sensor nodes such as no route, no neighbor, etc. To detect these type of failures, Sympathy gathers and analyzes management metrics which represent the states of the network such as nodes' next hops and neighbors. Based on these metrics, Sympathy finds out which nodes and components have not delivered sufficient data to the sink and infers the causes of these failures.

Sympathy detects and localizes failures using information from both sensor nodes and from the sink. The sensor nodes are responsible for collecting and monitoring network metrics, detecting environmental events, and providing requested data to the Sympathy-sink. The failure detection and localization is primarily done by the sink. The fault detection process has four stages. The sink first uses flooding to request the sensor nodes to report their event data and their management metrics such as sampled data and packet loss. After receiving the metrics, the sink analyzes them and detects if a failure has occurred. When a failure is detected, the sink probes the root cause of the failure by analyzing the available metrics and the sink may also initiate execution of additional tests if they are required to determine the cause. After the sink verifies the hypothesis of the root cause, it will inform the client. For example, if the next-hop-metric changes frequently, this event may indicate bad route configurations or network instability. Therefore, by analyzing detected events and management metrics, Sympathy is able to detect failures and localize their causes.

2.2.2 Decentralized Management System

In the decentralized management system (Figure 2.2), the management tasks are assigned to every sensor node in the network. Each node collects management information, analyses it, and performs management tasks itself without any central supervision. The decentralized network management system usually achieves higher reliability and lower communication overhead than the centralized network management system, however, it is more complex and difficult to manage. Distributed management algorithms may require significant resources and thus, may be too computationally expensive for the constrained sensor nodes. Moreover, the overall system performance may not be as good as the centralized approach since it does not have a complete view of the network.

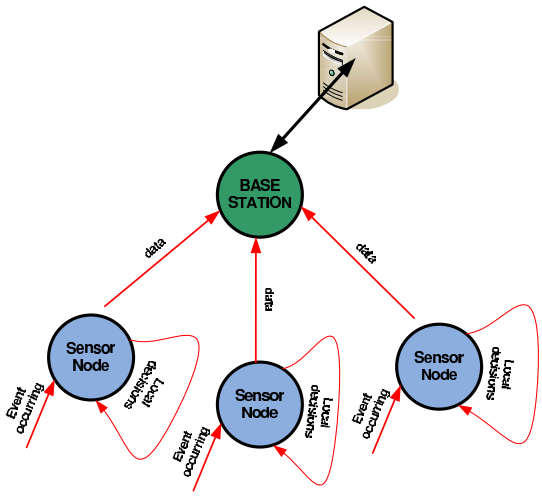


Figure 2.2: Decentralized Architecture.

Next, we discuss different decentralized management systems for WSNs including Role Assignment [34], two-phase self-monitoring system [35], and SORA [36].

2.2.2.1 Algorithms for Generic Role Assignment in WSNs

Frant et al. [34] proposed a generic **role assignment** framework for managing the roles of sensor nodes. In this framework, each sensor node is assigned with a specific role based on its properties such as remaining battery, number of neighbours, etc. A role is an identifier which consists of a list of rules. Rules are Boolean expressions that may contain predicates over the local properties of a sensor node and predicates over the properties of well-defined sets of nodes in the neighbourhood of a sensor node. The following is an example of a coverage role specification.

```
1 ON :: {
2   temp-sensor == true &&
3   battery >= threshold &&
4   count(2 hops) {
5   role == ON &&
6   dist(super.pos, pos) <= sensing-range
7   } <= 1 }
8 OFF :: else
```

Lines 1-7 of the rule specify the conditions required for a node to have ON status. The node is ON only if it has a temperature sensor and enough battery power. As the third condition, it is required that at most one other ON node should exist within 2-hop range from this node, which is specified by the count operator in the line 4.

The generic role assignment provides a programming abstraction that reduces the complexity of programming sensor networks at the system level. The developer can specify parts of the system behaviour using a high-level configuration language, rather than implementing low-level protocols and node functions. The assignment of these roles depends on a variety of parameters. For example, the cluster heads should be a powerful device because they act as a router for many slaves. All sensor nodes in the network have a copy of the same role specification. This reflects the understanding that

all sensor nodes are in the same initial software state.

Frant et al. [34] also proposed a distributed role assignment algorithm. The objective of this algorithm is to assign roles to sensor nodes, taking into account role specifications and sensor node properties. The algorithm is based on a fixed-point iteration [37], where each node would repeatedly fetch the current values of all relevant remote properties in order to evaluate the role predicates, eventually deciding on a role for itself. These evaluation cycles would have to be properly sequentialized among neighbouring nodes in order to ensure consistent role assignments. Assuming that there is a fixed-point configuration, each node would end up with a role that does not change in subsequent evaluation cycles. The advantage of role assignment is that with generic role assignment, the configurations can be easily generated and changed.

2.2.2.2 A Distributed Monitoring Mechanism for WSNs

Hsin et al. [35] proposed the **two phase self-monitoring system** for WSNs. The two phase self-monitoring system is designed for detecting malfunctioning nodes in the network by employing two types of fault detection: implicit and explicit. The explicit fault detection monitors the readings of sensor nodes. For example, if a temperature reading exceeds a pre-defined threshold, the sensor node will notify the base station. The implicit fault detection, on the other hand, refers to the detection of node communication failures due to energy depletion or environmental factors. The implicit fault detection is performed as follows. Each sensor node monitors the status of its neighbours by periodically sending a hello message to each. If a sensor node does not receive the hello message from a neighbour within a pre-specified period of time, it will assume that the neighbour is dead. Since neighbours monitor each other, the monitoring effect can be propagated throughout the network. As a result, the control centre only needs to monitor a potentially very small subset of sensor nodes.

To reduce false alarm probability, the two phase self-monitoring system maintains a two-phase timer. A sensor node uses the first phase to wait for updates from a

neighbour and uses the second phase to consult and coordinate with other neighbours in order to reach a more accurate decision. The use of two timers will assist the node in deciding whether the neighbour has not received the hello message due to environmental interference or whether the neighbour is actually dead.

Since it is a distributed monitoring system, the scheme offers low communication overhead. However, this scheme fails when there is a network partition. Also, time synchronization between neighbors may be difficult due to clock drifts, resulting in inaccurate fault detection.

2.2.2.3 SORA: Self-Organizing Resource Allocation

Mainland et al. proposed **SORA** [36], a Self-Organizing Resource Allocation for achieving efficient resource allocation in sensor networks. SORA is an approach for determining efficient node resource allocations in WSN by using a market-based approach. SORA defines a virtual market in which nodes sell goods (such as data sampling, data relaying, data listening, and data aggregation) in response to global price information that is established by the end-user.

In the SORA model, each sensor node acts as an agent that attempts to maximize its profit for taking a series of actions, subject to energy constraints. Each action consumes some amount of energy and produces one or more goods that have an associated price. Nodes receive payments by producing goods that contribute value to the network's overall operation. The actions in SORA are sampling a sensor, aggregating multiple sensor readings, or broadcasting a radio message. Prices are determined by the client of the sensor network, which can be considered as an external agent that receives data produced by the network and sets prices to induce network behaviour. Given a set of actions, goods produced by those actions, prices for each good, and energy costs for each action, each agent tries to maximize the profit by employing a greedy action selection algorithm such as the Exponentially Weighted Moving Average (EWMA) method. For example, a node may be paid for transmitting a sensor reading that indicates the

proximity of a target vehicle, but not be paid if the vehicle is nowhere nearby. Reacting to this payment feedback is the essential means of adaptivity in SORA.

The main advantage of SORA is that it offers low overhead for sensor resource management. Nodes self-schedule their local actions in response to the feedback in the form of payments. However, prices are determined and set by an external coordinator agent to induce a desired network's global behaviour. The system may not perform well when the network is dynamic since the pricing scheme action selection algorithms in SORA are simple and do not take into consideration the practical parameters such as the sequence of past actions or the state of neighbouring nodes, routing failures, etc.

2.2.3 Hierarchical Management System

Hierarchical management system is a hybrid between the centralized and decentralized management systems, in which the management tasks are distributed across sensor nodes in the network to ease the burden on a central manager. In hierarchical management system, sensor nodes are organized in interconnected clusters or sub-networks. As shown in Figure 2.3, the network is divided into a set of clusters. Each cluster has one cluster head, which is responsible for managing the sensor nodes within its cluster. The cluster head aggregates the management information received from sensor nodes within its cluster and passes it to the base station, and also disseminates management functions received from the base station to its sub-network. Furthermore, the cluster head can work cooperatively with other cluster heads to achieve an overall management goal, i.e., forming groups of nodes.

Many energy-efficient cluster algorithms have been proposed in literature and can be used here such as LEACH [38], HEED [39], etc. LEACH is one of the most popular clustering algorithms for WSNs. LEACH forms clusters by using a distributed algorithm, where each sensor node determines its cluster by choosing the cluster head that can be reached using the least communication energy. HEED considers a total cost of energy and communication when selecting cluster heads. HEED selects the sensor

nodes that have a high residual energy as the cluster heads. Both LEACH and HEAD could be used to maintain the cluster topology.

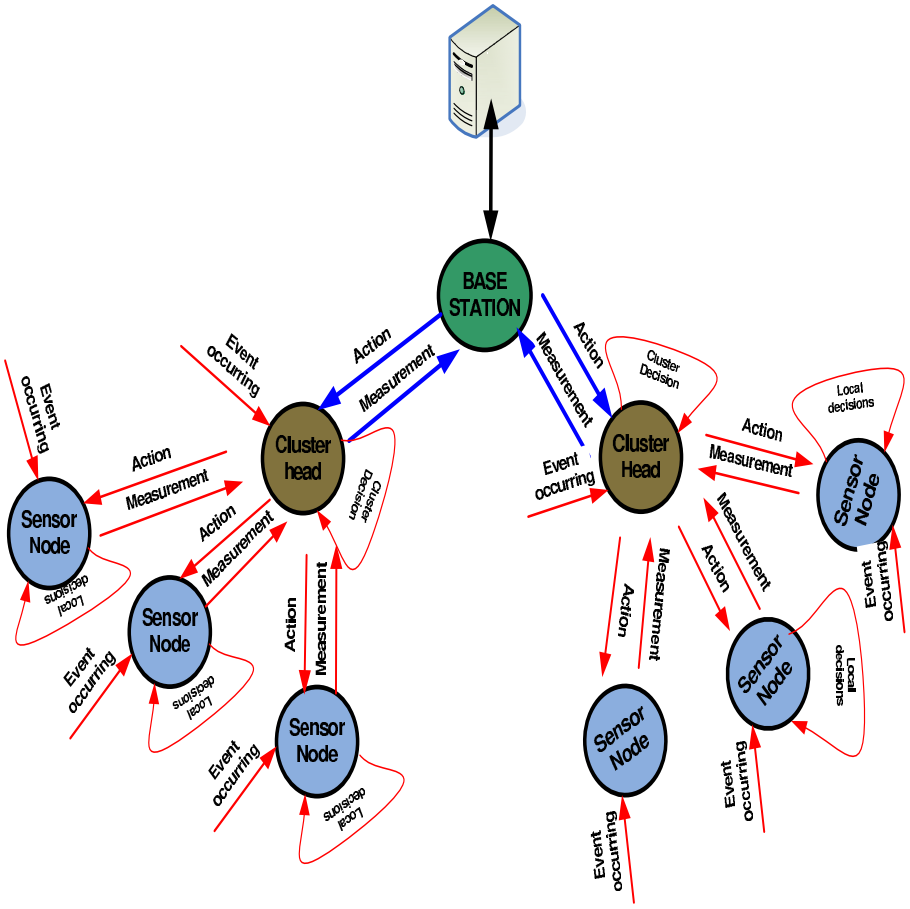


Figure 2.3: Hierarchical Architecture.

Next, we discuss different hierarchical management systems for WSNs including MANNA [30], SENOS [40], STREAM [41], and RRP [42].

2.2.3.1 MANNA: A Management Architecture for WSNs

Ruiz et al. [30] proposed **MANNA**, a policy-based Management Architecture in WSNs. Traditional network management consists of two planes: management functional areas and management levels. MANNA considers three planes: management functional areas, management levels, and WSN functionalities. From the abstractions of the three-

dimensional planes, MANNA builds a list of management functions and services that can be used for network management.

In the MANNA architecture, a WSN model represents an aspect of the network, and serves as a reference to the management functions. The WSN models periodically retrieve management information to monitor the state of the network. For example, to run a coverage area maintenance service, WSN models such as energy maps and topology maps are used to obtain management information for the service such as remaining battery status, network connectivity, etc. This information will be used to decide the appropriate management functions to be performed. Some examples of WSN models are given below:

- Sensing coverage area map: describes the actual sensing coverage map of the sensor elements.
- Communication coverage area map: describes the present communication coverage map from the range of transceivers.
- Network topology: represents the actual topology map and the reachability of the network. It may he used to obtain information about the necessity of adding new nodes.
- Residual energy: represents the remaining energy in a node or network. This information may also he available considering a region or time interval.

Based on the information obtained from WSN models, management services and functions are executed according to management policies. A management function represents the lowest level of management architecture. A management service consists of a set of functions. The policies are used to specify the conditions under which the management functions and services are executed. Examples of the management functions are environmental monitoring function, monitored area definition function, coverage area supervision function, node deployment definition function, etc.

The authors also suggested the use of an agent-based framework in which sensor nodes are organized in clusters and agents are located in the cluster-heads. Agents collect management information and transport this information back to the base station. Although no implementation details have been discussed in MANNA, it provides a conceptual view for designing a management system.

2.2.3.2 SENOS: Sensor Network Management Protocol for State-driven Execution Environment

In [40], Kim et al. proposed **SenOS**, a finite state machine based operating system for WSNs. In the SenOS, a valid input triggers a state transition and output generation, which moves the machine from the current state to the next state. Such a state transition takes place instantaneously and an output function associated with the state transition is invoked. Using this execution mechanism, a finite state machine sequences a series of actions or handles input events differently depending on the state of the machine.

The SenOS assumes that there are redundant sensor nodes in each cluster that are available to participate in network operation. In order to extend the cluster lifetime, SenOS employs dynamic power management (DPM) [43], which dynamically turns on and off sensor nodes when necessary. DPM provides a policy for determining state transitions based on observed events to reduce energy consumption. SensOS expresses state transitions produced by DPM in a finite state machine model and executes the power management of networked sensor nodes based on this model. The state-driven SenOS execution model is extensible to other sensor management protocols.

The main drawback of SenOS is that it requires network management tasks to be implemented on SenOS platform. Thus, it may not be easy to port it to other platforms such as Telos [44] or Fleck-3 [5].

2.2.3.3 STREAM: Sensor Topology Retrieval at Multiple Resolutions

Deb et al. [41] proposed **STREAM**, a distributed algorithm for sensor topology retrieval at multiple resolutions. STREAM retrieves the network state for multiple resolutions at different communication costs. Retrieved network topology ranges from the backbone to the complete network graph. STREAM uses snooping to identify the existence of other nodes in its communication range on the selected communication channel. By selecting a subset of nodes and merging their neighbourhood lists, an approximate topology can be constructed. The resolution of the topology depends on the cardinality and structure of the chosen set of nodes. For example, to construct a minimal backbone tree of the network, STREAM only needs to merge the neighbourhood lists of the minimal dominating set of the network graph.

STREAM is a colouring algorithm and consists of two stages. First, a monitoring node initiates a topology discovery request to all the nodes in the network using flooding. The request contains two parameters called virtual range and resolution factor. These parameters are used to select a minimal set of nodes, which is called the Minimal Virtual Dominating Set (MVDS), for retrieving topology at a desired resolution. During this stage, the nodes in the network are coloured red or black. Red nodes do not forward information, and the MVDS is the set of nodes coloured black. Further, at the end of the first phase, a black node tree rooted at the monitoring node is set up. In the second phase, the black nodes reply back to the request with a subset of their neighbourhood list, determined by the resolution factor. Each black node aggregates the data received from its child black nodes and sends it to its parent in the tree.

As STREAM selects a subset of nodes to reply to the topology discovery query, the number of these nodes determines the resolution of the retrieved topology. Thus, the overhead incurred is proportional to the resolution retrieved topology. Therefore, STREAM provides a trade-off between topological details and resource expended.

2.2.3.4 RRP: Managing Sensor Networks with Supply Chain Strategy

Liu et al. [42] proposed **RRP**, a Region based Routing Protocol based on the notion of a supply chain concept. RRP is designed for managing data gathering applications such as habitat monitoring and battlefield surveillance. In the business world, a supply chain is the series of links and shared processes existing between suppliers and customers, which involve all activities from the acquisition of raw materials to the delivery of finished goods to end consumers [42]. The objective of supply chain management (SCM) is to optimize all activities throughout the supply chain such as manufacturers, distributors, and retail outlets so that products and services are supplied in an optimal way.

RRP utilizes the knowledge of SCM to improve the performance of the sensor network. The sensor network is partitioned into several functional regions based on the supply chain management methodology. Different routing schemes for different regions and their inter-cooperation are applied in order to provide better performance in terms of reliability and energy usage.

RRP employs a hierarchical management system consisting of three areas: the manufacturing area, the transportation area, and the warehouse and service area. It manages the acquisition of raw data from the manufacturing area to the delivery of processed data to the warehouse and service area. Sensor nodes are heterogeneous and have different roles. In the manufacturing area, sensor nodes are aware of their missions. A sensor node may be either a source node that generates raw data, or an aggregation node which is responsible for filtering raw data. The aggregation node selects a transportation method and the proper transportation zones for forwarding the data to the transportation area. In the transportation area, sensor nodes undertake the task of relaying data to sink nodes. RRP uses a zone-flooding scheme to reduce the cost of topology maintenance and route discovery, which is a combination of geometric routing and flooding techniques. In zone-flooding scheme, when a node receives a packet

carrying parameters that identify a flooding zone, it needs to determine whether it is in the indicated zone or not. If the node is in the flooding zone, it will rebroadcast the packet. Otherwise, it will simply ignore the packet because it is not in the specified flooding zone for that packet. In the warehouse and service areas, sensor nodes are responsible for managing or reducing information implosion at base stations. Instead of zone flooding as the underlying routing protocol, a modified SPIN [45] protocol is used. SPIN allows nodes in a neighborhood to communicate with each other and use meta-data negotiation (ADV-REQ-DATA) to eliminate the transmission of redundant data.

The main advantages of RRP are that zone flooding ensures low message overheads, and adjusting the size of flooding zone ensures high reliability. However, RRP requires GPS-attached nodes in order to implement the zone-flooding protocol. Moreover, it requires a human manager to place sensor nodes in the field strategically at the initial network setup in order to support RRP hierarchical network management, which may not be suitable to many applications.

2.3 Maintaining Network Connectivity for WSNs

Network connectivity is a crucial requirement for most WSN applications. Maintaining network connectivity for a WSN is not a trivial task when there are environmental interferences, nodes joining/leaving the network, power depletion, etc. If the network is disconnected, the sensor nodes are no longer capable of delivering useful information to the end-users.

There are many relevant works on maintaining network connectivity in WSNs [46] [47] [48] [49] [50]. We classify these works into two groups: maintaining network connectivity for static WSNs, which comprises only static sensor nodes; and maintaining network connectivity for mobile WSNs, which comprises both static sensor nodes and mobile sensor nodes that have the ability to move. One example of a mobile node is

the Robomote [3]. These sensors are smaller than 0.000047 m^3 and cost less than 150 dollars.

Next, we will briefly discuss the relevant work for maintaining network connectivity in both static and mobile WSNs.

2.3.1 Maintaining Network Connectivity for Static WSNs

In static WSNs, the common solution for maintaining connectivity is to deploy redundant sensor nodes. When sensor nodes fail or the network is disconnected, the redundant nodes can be used for repairing connectivity [46] [47].

However, deploying redundant nodes for maintaining network connectivity is an expensive solution because a large number of backup nodes must be deployed together with the actual required sensor nodes. Moreover, in many cases it is difficult to ensure that redundant nodes are available for replacement, especially for a network in which the sensor nodes are randomly deployed. Next, we briefly discuss several related works in this area.

2.3.1.1 Span: An Energy-efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks

Chen et al. proposed **Span** [46], a power saving technique for multi-hop ad hoc wireless networks that reduces energy consumption while maintaining connectivity of the network. Span is a distributed, randomized algorithm where nodes make local decisions on whether to sleep or to stay awake based on the estimation of how many neighbors will benefit from it being awake and its remaining energy. A node will decide to be a coordinator (active node) if it discovers that two of its neighbors cannot communicate to each other directly or via one or two coordinators. Other nodes remain in a power-saving mode and periodically check if they should become a coordinator. To reduce the number of redundant coordinators, each node uses a random back-off delay to decide if it should become a coordinator. This delay is a function of the number of neighbours

and the amount of remaining energy. The random delay ensures a network connectivity and capacity, and provides significant energy saving.

2.3.1.2 ASCENT: Adaptive Self-Configuring Sensor Networks Topologies

Cerpa et al. proposed **ASCENT** [47], an adaptive self-configuration topology mechanism for WSNs. Similar to [46], the objective of ASCENT is to adaptively maintain a set of active nodes which stay awake and participate in a routing forwarding backbone. Other nodes in the network remain in a power-saving mode and periodically check if they should be active. In ASCENT, each node assesses network connectivity and decides its participation based on the measured operating region.

ASCENT algorithm works as follows. Sensor nodes are in one of four states: sleep, passive, test, and active. Each node initializes with a test state, maintains a timer T_t, and sends announcement messages to its neighbours. When T_t expires, the node determines if it should go to the passive state or active state. When the number of active neighbours is above the neighbour threshold NT (the degree of connectivity is high) or when the average data loss rate DL is higher than the average loss T_0, the node decides to move to the passive state. If these conditions do not occur, the node will go to the active state. In the passive state, the node sets up a timer T_p and sends a passive node announcement message to its neighbours. This information is used by the active neighbour nodes to estimate the density of nodes in their neighbourhood. The idea behind the passive state is to gather information regarding the state of the network without causing interference with the other nodes. Energy is still consumed in the passive state since the radio is still on (idle listening). When T_p expires, the node determines if it should go to the sleep state or return to the test state. If the number of neighbours is below NT (the degree of connectivity is low) and either the average data loss rate DL is higher than the loss threshold LT or DL is below the loss threshold but the node received a help message from an active neighbour, it makes a transition to the

test state. Otherwise, the node will move to the sleep state. A node that enters the sleep state turns the radio off, sets a timer T_s and goes to sleep. When T_s expires, the node moves into passive state. Finally, a node in active state continues forwarding data and routing packets until it runs out of energy. If the data loss rate is greater than LT, the active node sends help messages. The state machine takes both energy efficiency and packet loss into account. Therefore, it can adaptively maintain the number of nodes that need to be active in the network while reducing significant packet loss.

2.3.1.3 Improving Connectivity of Wireless Ad Hoc Networks

Li et al. [51] studied the problem called **Connectivity Improvement** of deploying additional wireless nodes to improve the connectivity of a wireless adhoc network. Specifically, given a disconnected wireless network, the connectivity improvement problem investigates how to deploy the minimal number of additional nodes to connect all network components. They proved the NP-completeness of the connectivity improvement problem and proposed a heuristic algorithm, called Connectivity Improvement using Delaunay Triangulation (CIDT). The CIDT constructs a Delaunay Triangulation in the disconnected network, and places new nodes in the selected triangle. CIDT selects triangles in the Delaunay Triangulation, one by one, with respect to certain criteria, and a connector is inserted into the selected triangle each time. The process repeats until the augmented network is connected. The results show that the CIDT algorithm can improve the network connectivity with a reasonable running time.

2.3.2 Maintaining Network Connectivity for a Mobile WSN

Using mobility for maintaining connectivity has been discussed in [48] [49] [50] [52] [53] [54]. When there are node failures, mobile nodes can be relocated to replace the failed nodes [48] [49] [50]. Mobile nodes can also relocate themselves from a densely deployed area to a sparse area for improving network connectivity [52] [53]. Another solution has been proposed in [54] in which mobile nodes are used as data carriers and

forward data between disconnected components of the network to the base station.

Next, we will briefly discuss the relevant work for maintaining network connectivity in both static and mobile WSNs.

2.3.2.1 Co-Fi: Coverage Fidelity maintenance algorithm

Ganeriwal et al. [48] proposed an algorithm called **Co-Fi**, a COverage FIdelity maintenance algorithm for WSNs. Co-Fi is a distributed algorithm which relocates sensor nodes to replace dying (low-energy) nodes for maintaining coverage and network connectivity.

Co-Fi has four phases: an initialization phase, a panic request phase, a panic reply phase, and a decision phase. In the initialization phase, each node estimates the sensing coverage of its neighbours by calculating its neighbour coverage region. In the panic request phase, a dying node notifies its status to its neighbours and sends a request for updating the new network topology. If the dying node does not have any exclusively monitored area to other nodes, its failure does not cause loss of the coverage of the network. In this case, the dying node just broadcasts a message, notifying the coverage neighbours of its death so that its neighbours can recalculate their coverage regions. However, if the dying node has some exclusively monitored area to other nodes, it will broadcast a panic request message and trigger the update of the network topology. In the panic reply and decision phase, if a neighbouring node, w, gets the panic request message of the dying node, v, node w will need to make a decision if it should move to v. If node w covers some exclusively monitored area, it only helps node v if its relocation does not lose its own coverage. On the other hand, if node w does not cover any exclusively monitored area, it will notify the dying node about its availability. During the decision phase, the dying node chooses the best candidate with the highest remaining energy for replacement.

The drawback of this work is that it requires all the sensor nodes in the network to be mobile, which may be expensive in many practical applications.

2.3.2.2 Sensor Relocation in Mobile Sensor Networks

Wang et al. [49] proposed a solution for **mobile sensor node relocation problem** to maintain the network connectivity. The sensor relocation has two phases: finding redundant sensor nodes and relocating them to a target location. A Grid-Quorum based solution is used in which the target field is divided into grids. Each grid has one grid head, which is responsible for collecting the information of its members and determining the existence of redundant sensors based on their locations. The grid head also monitors its group members and initiates a relocation process when nodes fail.

In the first phase, a dying node sends a request to seek a replacement. On the other hand, a redundant node also sends an advertisement to notify its availability. Instead of flooding the network with advertisements and requests, the advertisement and the request are only being sent to the nodes in the same row or the same column. Due to the intersection property of grid-quorums, they eventually intersect at a grid point. Thus, this scheme can reduce message complexity and response time significantly.

In the second phase, a cascaded movement algorithm is used. The main idea of the algorithm is to find cascading (intermediate) nodes, and ask them for relocation with the objective of balancing the response time and the energy consumption. For example, instead of asking a redundant sensor s_3 to move directly to the destination of s_0, s_1 and s_2 are selected as cascading nodes. Thus, to replace s_0, s_3 moves to replace s_2, s_2 moves to replace s_1, and s_1 moves to the destination of s_0. The cascaded movement solution can significantly reduce the relocation time for replacement.

The drawback of this work is that it only is designed for a grid network. However, in many applications where nodes are randomly deployed, this solution may not be applicable.

2.3.2.3 Dynamic Coverage Maintenance Algorithms for Sensor Networks with Limited Mobility

Sekhar et al. [50] proposed a **Dynamic Coverage Maintenance** (DCM) scheme for maintaining the network connectivity and coverage. Unlike the previous solution in [49], only the neighbors of the dying node will participate in the relocation process. Four DCM schemes were proposed: Maximum Energy Based (MEB), MinMax Distance (MMD), Minimum D/E (MDE), and Minimum Distance Lazy (MDL).

In the MEB scheme, only the neighbours that have high remaining energy will participate in the relocation process. A threshold of energy is defined and the neighbouring nodes that have remaining energy lower than that are not considered for movement. The MMD scheme tries to minimize the migration distance of mobile nodes. For each neighbour of the dying node, the maximum distance that it needs to move is calculated. The neighbour which has to move with the minimum of these maximum distances is chosen for migration. The MDE scheme combines the objectives of the MEB and MMD heuristics, which makes a decision based on the ratio of the maximum distance they can move to their available energy (D/E), and choosing the node with the least of these ratios. The MDL scheme moves the closest neighbour so that the uncovered area is likely to become covered.

Although this work only requires the neighbours to participate in movement, similar to [48], this work is only applicable to a network with all mobile nodes.

2.3.2.4 An Incremental Self-Deployment Algorithm for Mobile Sensor Networks

Howard et al. [53] proposed an **incremental self-deployment algorithm** for mobile sensor networks, in which the sensor nodes are deployed one at a time. Each node uses the data gathered from previously deployed nodes to determine its optimal deployment location.

The deployment algorithm has four phases: initialization, selection, assignment and execution. In the initialization phase, sensor nodes are in one of the three states: waiting, in which the sensor nodes are waiting to be deployed; active, in which the sensor nodes in the process of deploying; and deployed, in which the sensor nodes have already been deployed. Initially, there is a single node in the deployed state, which provides a starting point for the network. Other nodes are initialized with the waiting state. The selection phase determines the next deployment location. In the selection phase, sensor data from the deployed nodes is combined to form a common map of the environment. This map is analysed to select the deployment location, or goal, for the next node. The goal is to maximize network coverage under the constraint that nodes maintain line-of-sight with each other. The assignment phase attempts to assign the selected goal to a waiting node. In the assignment phase, the selected location is assigned to the first waiting node and the node changes from a waiting state to a active state. Finally, in the execution state, the active nodes are deployed sequentially to their goal locations. The state of each node is changed from active to deploy upon arrival at its goal.

The drawback of this approach is that it may incur high deployment time and has strong assumptions about the initial placement to guarantee the communication between the deployed and undeployed sensor nodes.

2.3.2.5 Movement-Assisted Sensor Deployment

Wang et al. [52] proposed an **incremental movement-assisted protocol** for improving the coverage in WSNs. The sensor deployment protocol uses a potential-field-based approach to move sensor nodes by considering them as virtual particles, subject to virtual forces.

The protocol runs iteratively until it terminates or reaches a pre-defined maximum round. In each round, sensor nodes construct their local Voronoi polygon. Each sensor node calculates the bisectors of its neighbours and itself based on the location information. These bisectors and the boundary of the target field form different poly-

gons and the smallest polygon encircling the sensor node is the Voronoi polygon of this sensor node. After the Voronoi polygons are constructed, if a coverage hole exists, a sensor movement is scheduled. They proposed three movement-assisted sensor deployment schemes to relocate sensor nodes: VEC (VECtor-based), which primarily pushes sensors away from a densely covered area, VOR (VORonoi-based), which pulls sensors to the sparsely covered area, and Minimax, which moves sensors to their local center area. VEC use the attributes of electro-magnetic particles: when two electro-magnetic particles are too close to each other, an expelling force pushes them apart. In VEC, the virtual force will push the sensors away from each other if coverage hole exists in either of their Voronoi polygons. VOR, on the other hand, is a pull-based algorithm which pulls sensors to their local maximum coverage holes. In VOR, if a sensor detects the existence of coverage holes, it will move toward its farthest Voronoi vertex. Similar to VOR, Minimax fixes coverage holes by moving closer to the farthest Voronoi vertex. Minimax chooses the target location as the point inside the Voronoi polygon whose distance to the farthest Voronoi vertex is minimized. These three protocols can provide high coverage within a short deploying time and limited movement.

The drawback of this approach is that it does not consider the uniformity of network coverage and does not guarantee oscillation avoidance if the threshold parameters are not set properly.

2.3.2.6 Deployment and Connectivity Repair of a Sensor Net with a Flying Robot

Corke et al. [55] proposed a **deployment algorithm with assistance from an autonomous helicopter**. The sensor nodes form a network on the ground and compute their connectivity map. If the network is disconnected, a localized algorithm determines the waypoints for the helicopter to drop additional nodes for maintaining connectivity.

The deployment algorithm has three phases. In the first phase, an initial au-

tonomous network deployment is executed. In the second phase, the entire network measures its connectivity topology. Two methods are used to measure network connectivity: a pingbased connectivity measure and a tokenpassing based measure. For the pingbased measure, a sensor that has been specially modified to add physical user interface controls is used to control and configure the sensor side of the ping connectivity. The token based connectivity algorithm is a distributed algorithm which computes the connectivity for the deployed network. Each node ends up with one token that denotes the group to which it belongs. These tokens are collected by the helicopter during a sweep of the field. If more than one token is collected, the network is not connected and new sensor deployments are needed. The locations of the collected tokens can be used to determine the repair regions. If this topology does not match the desired topology, a third phase is employed in which the waypoints for the helicopter are computed at which additional sensors are deployed. The last two phases can be run at any point in time to detect the potential failure of sensor nodes and to ensure sustained connectivity.

2.3.2.7 A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks

Zhao et al. [54] proposed a **Message Ferrying** (MF) scheme for data delivery in a sparse mobile ad hoc network. MF is a mobility-assisted scheme which utilizes a set of special mobile nodes called message ferries to provide communication service for nodes in the area. In the MF scheme, message ferries are responsible for carrying messages among nodes, while regular nodes are without such responsibility. Ferries move around the deployed area following pre-defined routes, collect messages from other nodes, and deliver messages to their destinations or other ferries. On the other hand, the regular nodes make proactive movement to meet up with the ferries. With the knowledge of ferry routes, regular nodes can adapt their trajectories to meet the ferries and transmit or receive messages. The main idea behind the MF approach is to introduce non-randomness in the movement of nodes and exploit such non-randomness to help the

data delivery task. By using message ferries as relays, nodes can communicate with distant nodes that are out of communication range.

However, this solution has a strong assumption of prior knowledge of connectivity patterns, which may not be feasible in many cases.

Chapter 3

A Cross-layer Design for Reliable Data Transport in WSN Data Streaming Applications

Figure 3.1: The Airdmillan Road Vicinity (approximately 2km × 3km, inside the yellow line) in the lower region Burdekin of Queensland, Australia, is an area of concern for salt-water intrusion into the aquifer.

Steadily rising salinity levels have been noticed in a number of production bores near the coast in the Lower Burdekin region. The Airdmillan Road area (approximately 2kmx3km, see Figure 3.1), which is centrally located within the Burdekin irrigation area, is an area of particular concern. There is concern that the ground water resource in this

area may be degrading, but the extent and cause of the problem are not well understood. Consequently, the management options available and the efficacy of particular options are also not well understood. One recommendation of a previous study [56] is that all the extraction bores in the monitoring area be metered (including date stamping), as it is unclear how much water is being extracted from the aquifer, and it is suspected that there may be some interplay between aquifer stress and the timing of the extraction.

In this chapter, we describe our design for reliable data transport for WSN data streaming applications, i.e. a WSN water monitoring application. Our goal was to design and deploy a WSN which could operate unattended, is capable of reliably reporting on the amount of water being pumped from each bore, and can measure the impacts of water extraction on water quality including water salinity, the underground water table level, the flow meter, and the flow ticks.

We employed reliable algorithms in each layer of the communication stack. At the MAC layer, a CSMA MAC protocol with an HBH eACK loss recovery is employed. To ensure the E2E reliability, the maximum number of retransmissions are calculated at each sensor node. An E2E NACK with an aggregated positive Acknowledgment mechanism is used in the transport layer. By inspecting the sequence numbers on the packets, the base station can detect which packets were lost. The base station sends a NACK message to a source after receiving a preset number of packets from it. In addition, other robustness requirements such managing node reboots are considered in the protocol design. The designed sensor network system has been working in the field for over a year. Our comprehensive evaluations, which include theoretical analysis, simulations, and experiments in the field, show that the reliable data transmission protocol in our sensor system is a promising solution to allow gathering of sufficient data to establish a sustainable irrigation system.

3.1 Chapter Contributions

The primary contribution of this chapter is that we designed, implemented, deployed, and evaluated a reliable data communication protocol for outdoor sensor networks across multiple protocol layers. We evaluated the reliable data communication protocol with theoretical analysis, simulations, and eventually field deployment. To make our system survive the hostile tropical environment, we have designed a tailored package for our sensor system. To increase the robustness of the system, we have implemented watchdog logic at both the remote gateway and at the sensor nodes. Our evaluation results show that the reliable data communication protocol can increase significantly E2E data delivery ratios.

This work could not be completed without the hardware supports from Dr. Peter Fitch and Dr. John Whitham and funding supports from CSIRO [4], Water Resource Observation Networks [57] and North Burdekin Water Board [58].

3.2 Chapter Organization

This chapter is organized as follows. In Section 3.3, we describe the requirements of the system. Section 3.4 describes the architecture of the sensor network system employed in our deployment. In Section 3.5, we describe the design of the reliable cross-layer communication protocol at each layer in detail. Section 3.6 presents our extensive evaluation in both simulation and in the deployment field. In Section 3.7, we discuss the lessons that we learned from the deployment. Section 3.8 briefly outlines the related work. Finally, Section 3.9 summarizes the work presented in this chapter.

3.3 System Requirements

The application requirements of our water quality monitoring sensor network include:

Sensors	Min	Max	Resolution	Unit
Electrical Conductivity (EC)	0	100,000	10	μ S/cm
Water level	0	3,000	5	cm
Flow rate	0	100	0.5	litre/s
Flow volume	0	200	1	ticks

Table 3.1: Sensor Resolution Requirements.

- Sensor Specification Requirements: Our system consists of four types of sensors: Electrical Conductivity (EC), water level, water flow, and water volume at each irrigation bore. As shown in Table 3.1, the water volume sensor provides digital pulses to the node. Each pulse (a tick) represents 1 litre of water passing through the irrigation pipe. The EC sensor must be able to measure up to 100,000 $\mu S/cm$, and provide a measurement resolution of 10 $\mu S/cm$ [1] . The water level sensor must be able to measure a water depth range of up to 30 m, and the flow rate sensor must be able to measure a flow rate up to 100 l/s. Details of ranges and resolutions are given in Table 3.1. The sample rate of the analog sensors is one sample per minute. The sensors must be robust enough to operate in a harsh tropical environment with high humidity, high temperature, iron deposits, and acidic cleaning liquid. Further, the diameter of the observation bores, where we deploy the pressure sensor to measure the level of the water table, is around 75 mm, which limits the size of the water level pressure sensor.

- System Maintenance/Service: Because the sensor system will be operating in a remote area (about 2,000 km from our lab), the sensor system must be capable of operating independently for long periods of time, i.e. months. Therefore, the system must be robust to environmental dynamics, software failures, power supply outages, etc.

[1] Siemen (S) is the inverse of resistance (Ohm)

- Sensor Platform and Package: Because the sensor network is sparse (5 nodes in an area of about 2km × 3km (see Figure 3.1), the radio range of the nodes must be large enough to form a connected network, i.e. more than 1km. This requirement necessitated that we deploy additional nodes to improve network connectivity. In order to make the system work in this harsh environment, the sensor housings must be waterproof and be able to tolerate high humidity.

- Network Delivery Ratio: Our target is for a 75% E2E packet delivery ratio. This is a challenging task because the access to the deployed field is limited (more than 2000km from our lab) and the field environment is very dynamic.

In next section, we will introduce the architecture of our sensor system, which is tailored to meet these challenging requirements.

3.4 System Architecture

In this section, we briefly describe the architecture of the Burdekin sensor network system utilised in our deployment. As shown in Figure 3.2, the system consists of the field gateway, the site gateway, the remote server, and the sensor nodes. The system was designed to reliably acquire data from the sensor network in the field and store it in a database server in the office.

The field gateway was designed for long-term, remote, and unattended operation. The gateway is a single board ARM-based computer running Linux. It also runs the 'C' version of the TinyOS [59] serial-forwarder program which acts as a gateway between Internet-clients and the sensor network. It is located in a shed near a pumping site on the farm. An Asymmetrical Digital Subscriber Line (ADSL) connection is available in the shed. It communicates with the remote server located in our lab at Brisbane (about 2,000 km away from the deployment field) using an ADSL modem/router connected via Ethernet. The ADSL modem/router has a static IP address and provides Network

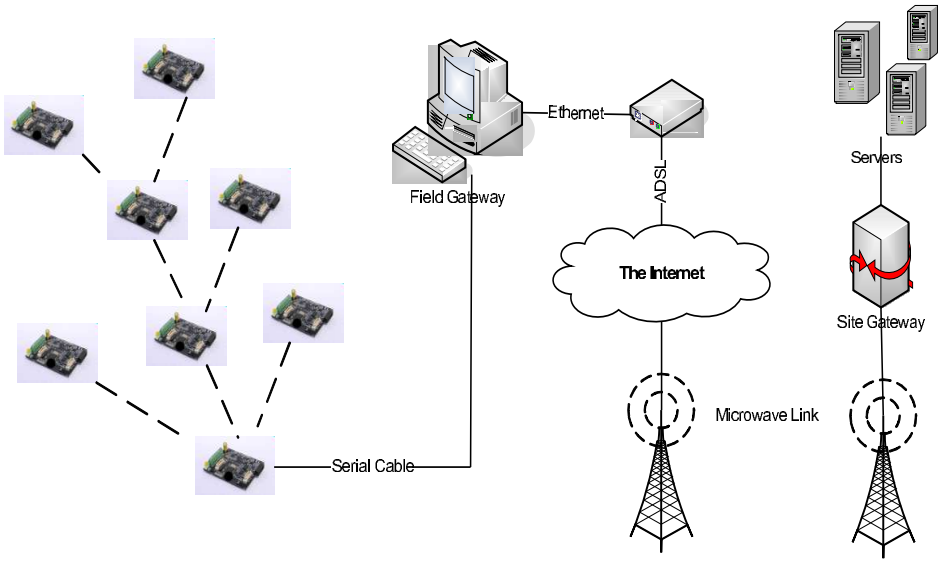


Figure 3.2: Sensor Network System Architecture.

Address Translation (NAT) firewall. It also runs a DHCP server which provides the gateway computer with an IP address. To ensure the continuous operation of the gateway, a watchdog process is implemented that monitors Internet connectivity and can power cycle the attached ADSL modem/router as well as the attached Fleck node when a fault occurs, i.e. the modem is not working, or the Fleck node is down.

The site gateway runs an interface program that collects all the messages coming from the sensor network, stores them in a MySQL database at the remote server, and sends ACK/NACK messages to the sensor nodes for packet loss recovery.

The remote server is a server class computer located in our lab at Brisbane. Our lab is connected to the Internet using a high-speed microwave link.

Finally, the sensor nodes are based on the Fleck platform [5], which has been described in Chapter 1. The sensor node has a long transmission range of up to 1 km using standard unity-gain quarter-wavelength antennas. This is critical for a sparse sensor network deployed in a large area such as the Burdekin. Figure 3.3 shows the sensors being used in the deployment. The Electrical Conductivity sensor is a Toroidal

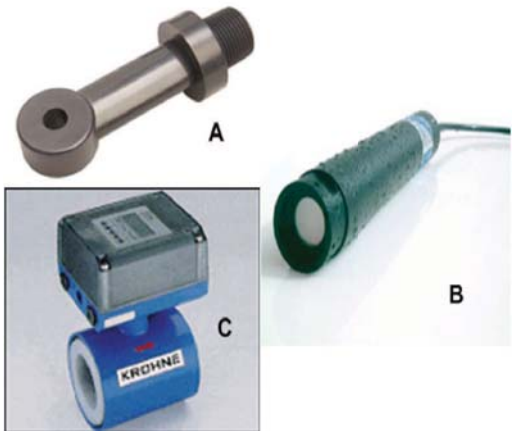


Figure 3.3: Pictures of the water quality sensors. A. Sensorex TCS1000 salinity sensor; B. Tyco PS100 pressure (water level) sensors; C. Krohne electromagnetic flow meter sensor.

Conductivity Sensor TCS1000 made by Sensorex. The depth of the water table is measured by a PS100 pressure sensor made by Tyco. The electro-magnetic flow meters are made by Krohne and provide both volume and flow rate. Figure 3.4 shows the deployed sensor node at one of the pump sites. The flow meter and the EC sensor are mounted in the pipe connecting the pump to the reservoir tank. The pressure sensor is mounted in an observation bore next to the pump. As well as the watchdog process in the gateway which power cycles the modem, we have embedded watchdog timers in each of the sensor nodes to ensure their robustness.

3.5 Communication Components

We used TinyOS [59] as the operating system for the Fleck-3. Taking into account the system requirements introduced in section 3.3, we employed reliable algorithms in each layer of the communication stack (see Figure 3.5). The cross-layer design ensures the required network delivery ratios while minimizing the energy consumption for communication activities in the network. At the MAC layer, a CSMA MAC protocol with a Hop-by-Hop Explicit Acknowledgment (HBH eACK) loss recovery is employed.

Figure 3.4: The sensor node deployed at one of the pump sites in the Burdekin. (A) Sensor node internal view. (B) Sensor node housing. (C) Bore containing water level sensor. (D) Pump. (E) Flow meter. (F) EC sensor. (G) Reservoir Tank.

To ensure the End-to-End (E2E) reliability, the maximum number of retransmissions is estimated and used at each sensor node. At the network layer, an E2E Negative Acknowledgment (NACK) with an aggregated positive Acknowledgment mechanism is used. In addition, to increase the robustness of the system, a watchdog process is implemented at both the base station and sensor nodes, which enable them to power cycle when a unexpected fault occurs, e.g., sensor nodes hang up, the radio does not work properly, sensors are malfunctioning, etc.

3.5.1 Application Layer

The application layer is simple and is only responsible for collecting the data that must be sent to the server. At the sensor nodes, it periodically queries the sensor readings and passes the values to the transport layer. At the site gateway, a java-based program was implemented to log the sensor readings to the MySQL database at the server (see Figure 3.2). To increase the robustness of the application layer, the java program periodically checks the connection to the field gateway as well as to the MySQL server and will restart the connection if a fault is detected.

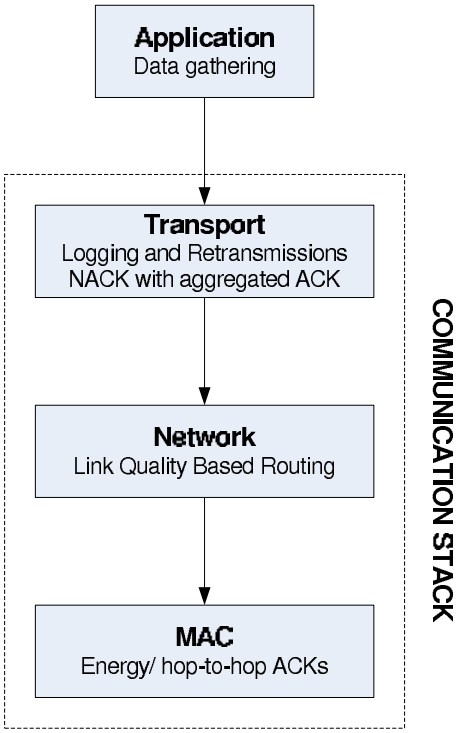


Figure 3.5: The architecture of reliable network protocols.

3.5.2 Transport Layer

This layer has a simple interface which composed of two different commands for sending data. Each of them creates a different kind of packet:

- Data packets: they contain the sensor readings which need to be sent to the sink.
- Control packets: they contain the network control information, i.e. NACKs, ACKs, or battery voltages, which need to be sent to a particular node or to all nodes in the network.

An E2E NACK with aggregated positive Acknowledgement (ACK) mechanism was used in the transport layer. By inspecting the sequence numbers (SN) on the packets, the site gateway can detect which packets were lost. The site gateway sends a NACK to a source after receiving a preset number of packets from it, i.e. 10 packets

(see Section 3.7.2 for the analysis). The NACK message contains the list of SNs of lost packets that the site gateway requests the source to retransmit. The sources use a circular queue to buffer their packets before sending them out. The source assumes that the packet has been delivered successfully to the sink if the source does not receive any NACKs. The source removes a packet from its buffer when it receives an ACK of the packet from the sink. To ensure the reliable delivery of the first packet to the site gateway, we require an ACK scheme for the first packet transmission from each source. We would like to have a queue as large as we can. In our system, the queue can hold up to 80 packets, which is equivalent to about 3.2kB (recall that the memory is 4kB).

The transport protocol also takes the node reboot issue into account by periodically monitoring the SNs. For example, assume that the SN of the last packet which the site gateway received from node i is n_1. If the site gateway receives a packet from node i with the SN n_2, where n_2 is less than n_1, the site gateway infers that node i has rebooted. In this case, the site gateway will clear the current list of the lost packets of node i to stop sending NACKs.

3.5.3 Network Layer

We chose a well known sensor network routing protocol, Surge [19], in the network layer. Surge is a reliable multi-hop routing protocol for any source to sink communication that uses link quality as its routing metric (see Section 2.1.1.1). Surge dynamically forms a reliable spanning tree that covers every node in the network, using link connectivity estimation and neighbourhood table management techniques. In the Surge protocol, each node periodically measures the link qualities between itself and its neighbours, and selects "the best" neighbour as its parent to forward the data to the base station. The performance of Surge has been shown to be superior [19] to other routing protocols including shortest-path, destination sequence distance vector routing (DSDV) [60] and Ad-hoc on-demand distance vector routing (AODV) [61], in unreliable wireless environments.

At the sink, the routing layer simply forwards data packets to the field gateway via the serial port, while control packets are passed to the MAC layer and sent to the sensor nodes. At the sensor nodes, the network layer passes both kinds of packets to the MAC layer.

3.5.4 MAC layer

The wireless links in sensor networks are typically unreliable since sensor nodes generally use low transmission power. We implemented a Carrier Sense Multiple Access (CSMA) style Medium Access Control (MAC) with a HBH eACK. That is, a sender will wait for an ACK from the receiver after sending out a packet. If it does not receive the ACK within a pre-defined time interval called MAC Layer Timeout, it will retransmit that packet. The process is repeated until either the sender successfully receives the ACK or the number of retransmissions exceeds a pre-defined threshold.

- ***MAC Layer Timeout***

  It is important to have a good hop-to-hop MAC retransmission timeout value (t). If t is too small, immature retransmissions may occur. Conversely, if t is too large, the transmission latency will increase, and network throughput will reduce.

  The packet transmission time of the NRF905 radio is calculated by [62]:

$$t = t_{startup} + t_{preamble} + \frac{N_{address} + N_{payload} + N_{CRC}}{BR} \tag{3.1}$$

  where:

  - BR is the bitrate , BR = 50kbps

  - $t_{startup}$ = 650 μs

  - $t_{preamble}$ = 200 μs

  - $N_{address}$ = 4 bytes

  - $N_{payload}$ = 32 bytes

- $N_{CRC} = 2$ bytes

Thus, the expected MAC layer timeout is $t > 6.93$ ms. Ideally, t should be as small as possible to increase network throughput. We performed empirical tests on different timeout values t ($t = 7, 8, 9, 10, 11, 12$ ms) and observed a significant number of immature timeouts when $t < 10ms$. We found that $t >= 10$ms gave better performance and thus, $t = 10$ ms was used in our communication protocol.

- ***The Expected Number of MAC Layer Retransmissions***

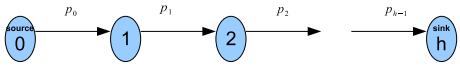


Figure 3.6: A Single Flow with Transmission Failure Probabilities.

Consider a single routing path of $h + 1$ sensor nodes arranged linearly from $0, 1, 2, ..., h$, where the source is Node 0 and the sink is Node h as depicted in Figure 3.6. Source 0 sends packets to the sink h through Nodes $1, ..., h - 1$. For each node i, we denote p_i as the upstream (from sources to the sink) link loss rate between Node i and Node $i + 1$ ($0 \leq p_i \leq 1$).

If Node i transmits a packet N_i times, the obtained E2E reliability r_o after N_i transmissions is

$$r_o = \prod_{i=0}^{h-1} \left(1 - p_i^{N_i}\right) \tag{3.2}$$

To achieve the E2E reliability at level r_d, the reliability (r_o) must be equal or greater than r_d. So, the reliability at each hop must be

$$\left(1 - p_i^{N_i}\right) \geq (r_d)^{1/h} \tag{3.3}$$

Therefore, the expected number of transmission retries N_i at Node i is

$$N_i = \frac{log(1 - r_d^{1/h})}{log(p_i)} \tag{3.4}$$

In our field deployment, the farthest node from the gateway is about 6.3km. With the maximum transmission range of about 1.5km, the farthest node is $h_{max} = 5$ hops away from the gateway. We would like to design a communication protocol that tolerates a link loss rate (p_i) up to 0.5 to cover high link dynamics, i.e., unstable link quality. With $r_d = 0.75$, we can calculate the maximum number of transmission retries is $N_{max} = 5$ by the Equation (3.3). We would like to have some redundant link-loss recovery capability in our protocol, and chose the number of retransmission retries $N = N_{max} + 1 = 6$.

3.6 Evaluation

In this section, we describe the evaluation of our system in both simulations and field experiments.

3.6.1 Simulation Results

We evaluate the performance of the cross-layer communication protocol introduced in Section 3.5 by discrete-event (ns-2 [63]) simulations. The purpose of the simulations is to evaluate analytical results (Section 3.5.4), and to evaluate the scalability and robustness of the protocol in different network conditions. Note that we do not focus the energy consumption, because most of the deployed sensing nodes are AC powered in our system (see Section 3.6.2.2).

We considered a network of 100 nodes (our ultimate goal) uniform-randomly distributed in a rectangular region 4000m x 4000m (see Figure 3.7). The communication transmission range of the nodes is 1000m. We selected the node at the top-left most corner as the sink and every other node generated a 40 byte data packet every minute. The hop-distance from the sink to sensor sources is between 1 hop and 6 hops. Each sensor source maintains a buffer of 80 packets. Our simulation setup is similar to the planned deployment. To study the impact of network dynamics on the reliability, we

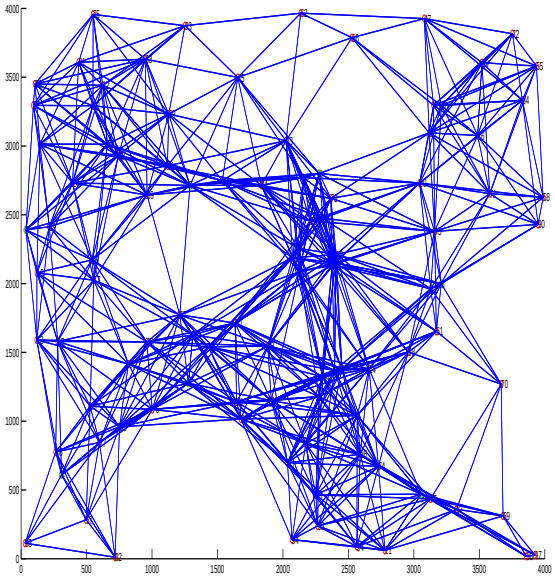


Figure 3.7: Simulation Network Topology.

simulated node failures as follows. We repeatedly turned on and off a fixed fraction of nodes for 30 minutes(0%, 10%, 20% node failures, respectively), which are randomly selected from the sensor field. We measured reliability in two cases: with (our communication protocol) and without E2E NACK transport protocol. The total simulation time is 12 hours[2] .

Figures 3.8-3.10 show the average obtained E2E reliability versus the route length along with 95% confidence intervals for different link loss rates (p = 5%, 20%, and 45%, respectively). The results suggest that the 6-MAC-layer retransmissions (obtained from the analysis in Section 3.5.4) can compensate the link loss adequately when no nodes have failed (E2E reliability is close to the 100%). However, when the network is more dynamic (with 10% or 20% node failures present in the network), the E2E transport protocol can improve delivery ratios significantly, and is able to recover up to 30% packet losses. These results indicate that our communication protocol will perform significantly

[2] 12 hours is chosen arbitrarily to obtain steady network state results.

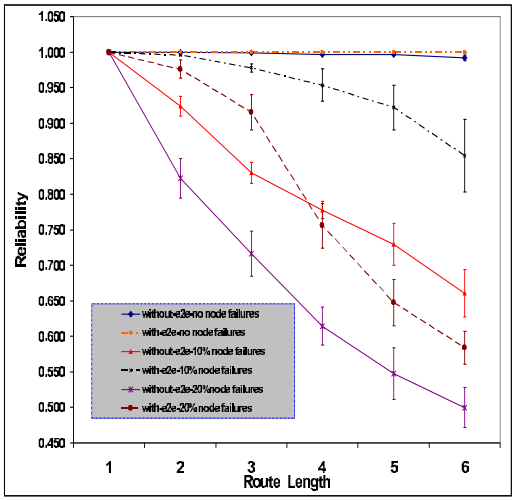


Figure 3.8: Simulated Delivery Ratio - Loss rate = 5%.

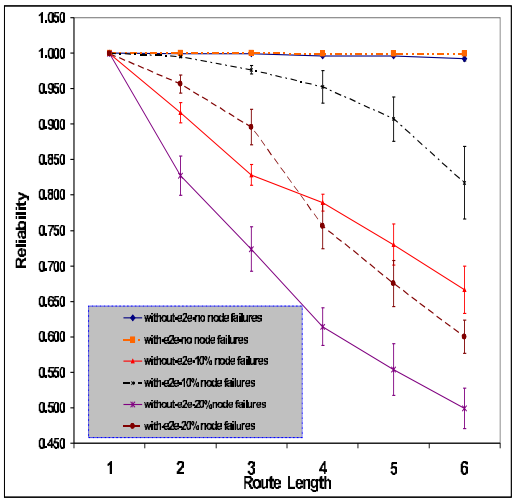


Figure 3.9: Simulated Delivery Ratio - Loss rate = 20%.

better than the communication protocol without E2E NACK in the field, where more network dynamics were expected.

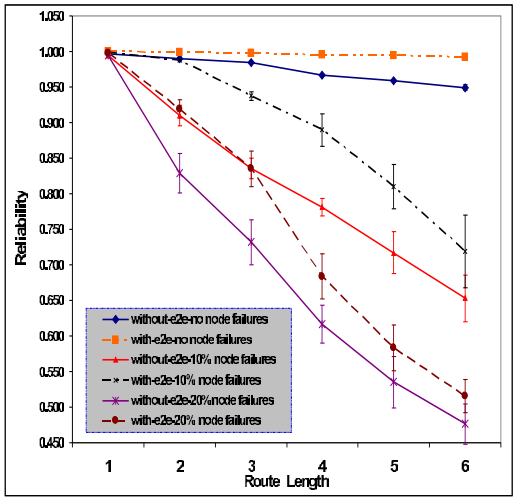


Figure 3.10: Simulated Delivery Ratio - Loss rate = 45%.

3.6.2 Field Results

3.6.2.1 System Deployment

At the end of Feb 2007, we deployed the sensor system with eight nodes (see Figure 3.11) during the southern hemisphere tropical dry season.

3.6.2.2 Energy Consumption

Most of the sensing nodes are AC powered since this is available at the pumping sites to operate the pump. One site has a diesel pump and we use a large solar panel and car battery to operate the node and the sensors. Relay nodes are standalone and self-contained with small solar cells, see Figure 3.13. Our solar power system has worked well in practice.

3.6.2.3 Dynamic Network Topologies

After the deployment, we observed a highly dynamic environment caused by the combination of many environmental parameters such as distance, antenna height, temperature, humidity and terrain.

Figure 3.11 shows the most common network topology of our deployment, with a mean transmission range of around 855 meters. The arrows in the figure represent the direction of data flow. With the link quality aware routing protocol (Surge) introduced in Section 3.5.3, the network stays in this topology more than 70% of time.

Figure 3.11: The Most Common Network Topology. The Mean Transmission Range is 855m.

Other than Node 11, all of the nodes choose the geographically closest node as their parent node. The distance between Node 11 and Node 2 is about 600 m, and we observed a link between them when we conducted transmission range tests in December 2006 (when the sugar cane growing in the surrounding field was only 0.5 m tall). Since deployment, we have not observed any link between Nodes 11 and 2 when the sugar cane was more than 4 m tall (the height of the antennas is just over 5 m). Because Node 12 is located in an open area, we observe consistently good link quality between Node 12 and Node 0. The link between Node 11 and Node 0 has intermittent connection only, and we plan to deploy an intermediate node between Node 11 and Node 0 to achieve a more reliable radio link. The new link may also act as a router between Node 2 and Node 0.

Figure 3.12 shows an extreme network topology of our deployment, with a mean transmission range of around 1,135 m. In this scenario, most of the nodes (1, 2, 11, and 13) choose alternative longer routes to parent nodes. Being closer to Node 0 and located at an open spot makes the link quality between Node 12 and 0 consistently good. Node 1 and Node 11 chose Node 12 as parent instead of transferring to Node 0 directly on a few occasions.

Figure 3.12: An Uncommon Network Topology. The Mean Transmission Range is 1,135m.

Figure 3.13: A typical relay node. The node is self-powered and simply needs to be attached to the outside of a convenient building.

3.6.2.4 System Delivery Ratio

For each sensor source i, we evaluate the following metrics:

- Delivery Ratio D: The ratio of the number of packets (N^r) successfully received at the server, to the number of packets sent (N^s) from a sensor node.

$$D = \frac{N^r}{N^s} \tag{3.5}$$

- Recovery Ratio R: The ratio of the number of retransmitted packets (N^t) successfully received at the server to the number of packets sent N^s from a sensor node.

$$R = \frac{N^t}{N^s} \tag{3.6}$$

The first metric measures the reliability of our entire system (from the source to the server through gateway) and the second metric measures the effectiveness of the transport layer.

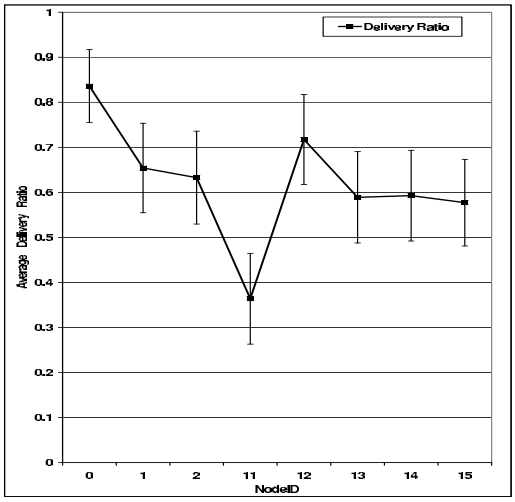


Figure 3.14: Weekly Average Delivery Ratio over the Period from 23/04/07 to 23/11/07.

Figure 3.14 and Figure 3.15 show the weekly average delivery and average recovery ratios per node along with 95% confidence intervals over the entire period between

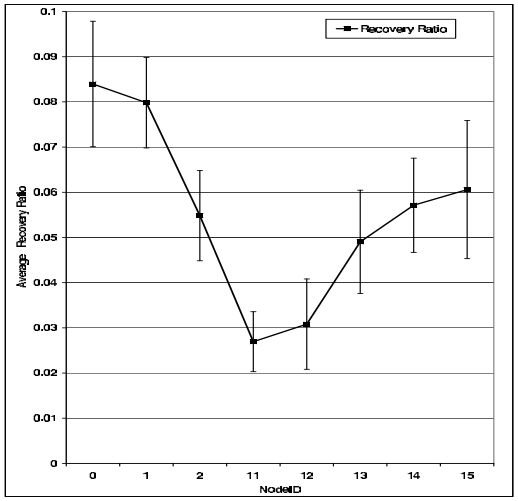


Figure 3.15: Weekly Average Recovery Ratio per node over the Period from 23/04/07 to 23/11/07.

23/04/07 and 23/11/07 (about 7 months collected data). In general, the average delivery ratio per node is around 62% and the average recovery ratio per node is around 5%.

First, the results show that the transport protocol can improve the delivery ratio up to 8.6%, e.g., Node 0 and Node 1. We observe that the packet losses also happened in the gateway (Node 0 - 84.4% delivery ratio) because the Internet connection between the gateway and server was down occasionally. The delivery ratio of Node 11, on the other hand, is significantly lower (36.38%) than the other nodes because of the intermittent communication problem between Node 11 and Node 0 (Section 3.6.2.3).

Second, we observed that the minimum recovery ratio is not significant (less than 2%) for the following reasons. If the communication link is stable, i.e. the link between 0 and 12, most data packets were routed successfully by the Surge protocol and only a few transport layer retransmissions happened. If the communication link is unstable, we observed that the Surge protocol does not route downstream (from sink to nodes) well (in fact, by purely broadcasting and without HBH recovery). Consequently, the source node, i.e. Node 11, receives few NACK packets, and therefore, does not attempt retransmissions. While the first case shows that routing protocols perform relatively

well for upstream (from nodes to sink) traffic, the second case shows some issues needed to be solved for the downstream traffic. In fact, we observed that the link connection between sensor nodes and gateway is down frequently during the night time. We will further investigate this behaviour in Section 3.7.2.

3.6.3 Sensor Measurements

The network has generated a lot of data since it began operation. Here we pick a small number of interesting examples that we have observed.

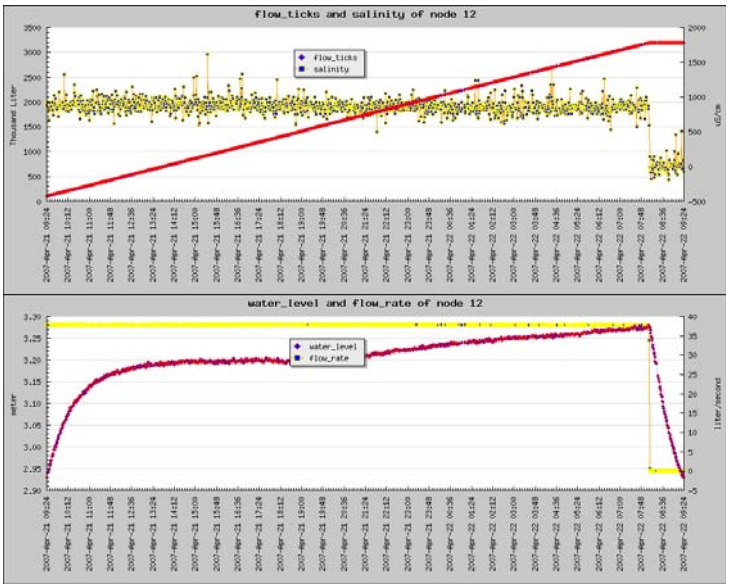


Figure 3.16: Sensor Measurements of Node 12 between 21/04/07 and 22/04/07.

Figure 3.16 shows EC, flow ticks, water level[3] and flow rate sensor measurements of Node 12 over 24 hours (21/04/07 to 22/04/07). It shows that the pump was turned on between 21/04/07 9:24am to 22/04/07 7:28am with a constant flow rate of 37.5 litre/second, nearly 3 Ml. The water level decreased gradually from 2.95 m below the ground to around 3.25 m below the ground (more than 30 cm in less than 24 hours). After the pump was turned off, the level of water table gradually rose back to

[3] Note that water level is expressed as distance below the ground surface, and increases as the water table gets lower.

2.95 m below the ground. Figure 3.16 shows that the EC level was constantly at around 1,000 μS/cm level.

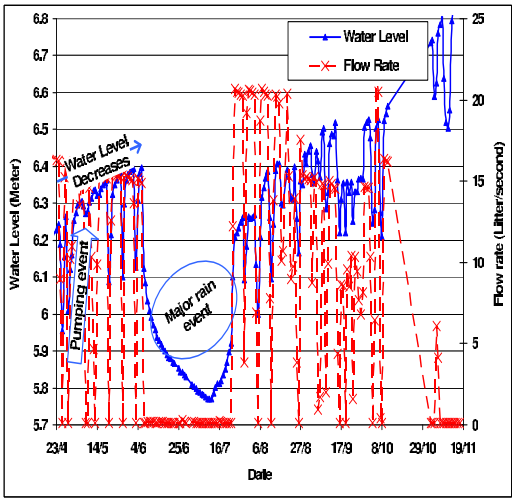


Figure 3.17: Flow Rate and Water Level of Node 14 between 23/04/07 and 23/11/07.

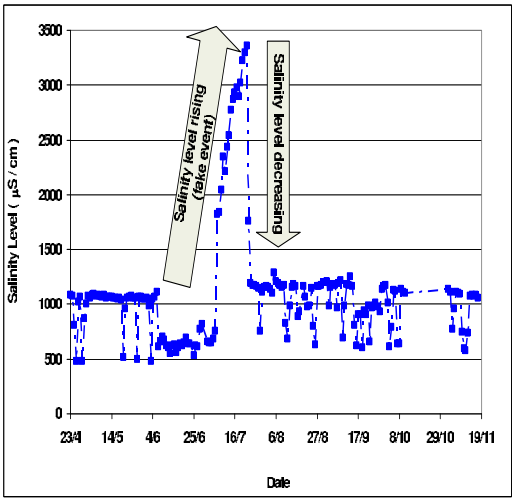


Figure 3.18: EC Measurements of Node 14 between 23/04/07 and 23/11/07.

Figure 3.17 shows the underground water level near Node 14 over the 7 months from 23/04/07 to 23/11/07. We observed that the farmer turned on the pump to irrigate sugar cane frequently before June, 2007 when little rainfall was recorded [64].

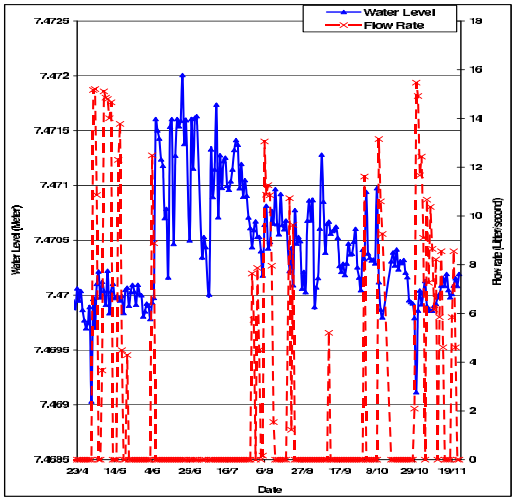


Figure 3.19: Flow Rate and Water Level of Node 15 between 23/04/07 and 23/11/07.

Consequently, the water level decreased from 6.2 m below the ground in the middle of April to 6.4 m below the ground in early June, 2007. The major rainfall in June, 2007 [64] resulted in the water level rising by 0.3 m and no pumping event was recorded during this period. We observed that water level peaked on 14 July. The farmers turned the pump on again on 24 July to irrigate the sugar cane.

To evaluate the performance of EC sensors in the field, we conducted an experiment by loading high-EC water into the pumping pipe during the rainfall period when the pump was not turned on. Our system successfully detected this "fake" event (see Figure 3.18). The observed water EC level was significantly higher between 7 July and 23 July. The EC level then dropped to the normal level on 24 July when fresh underground water was pumped through the pipe.

Figure 3.19 shows the underground water level for node 15. We observed that the water level of Node 15 is just slightly fluctuated because it is located next to the river (see Figure 3.11). Other than Node 15, we observed that the measurements collected from other nodes show similar phenomena to Node 14. This suggests that the collected sensor reading is consistent, and useful for long term salinity and water table study.

3.7 Deployment Lessons and Discussions

In this section, we discuss the lessons that we learned from the Burdekin remote water quality monitoring network deployment.

3.7.1 Wireless Radio Transmissions

Figure 3.20 shows the daily yield for the network. We plot, against time, the minimum, median and maximum yield across all nodes. Note that there are two significant outages in this dataset, around 24/6 and 1/07. The best nodes in the network sometimes have a yield of 100%. The worst nodes in the network often have a yield of zero.

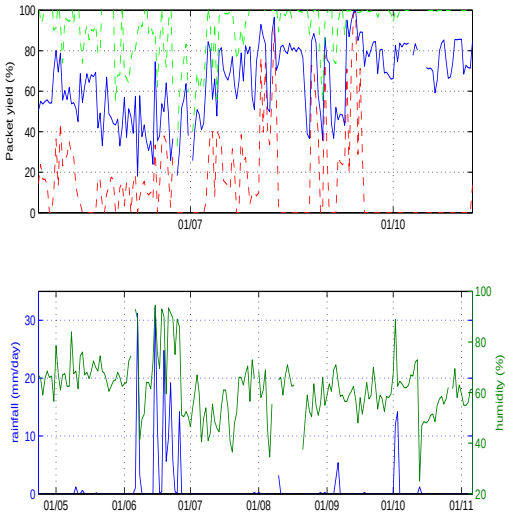


Figure 3.20: (Top) Daily yield (min, max and median) of the network nodes over the period 23/04/07 to 23/11/07. (Bottom) Corresponding rainfall and humidity.

There are some distinct low-points in the median yield value in June, 2007 that coincide with periods of very high rainfall and humidity, as measured at a nearby meteorological station. We have insufficient data (and rainfall) to determine whether this effect is statistically significant. In Figure 3.21, we see the temporal pattern of communications from Node 1 and Node 15 (Node 1 is the network topology parent of Node

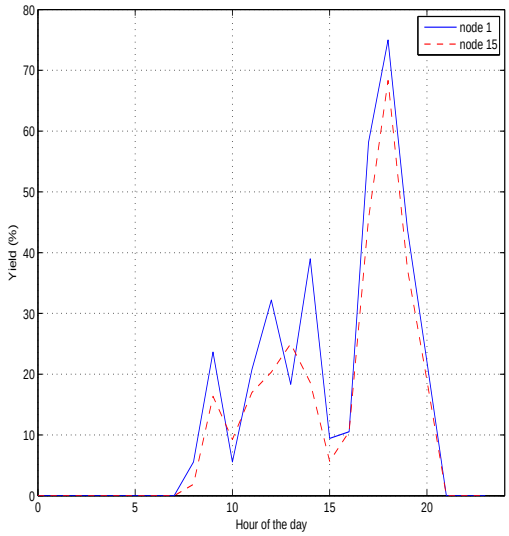


Figure 3.21: Daily temporal variation of yield for two nodes, computed over the week 20-26/5/2007.

15). We can see clearly that from 9pm to 7am there is no radio communications from both nodes. During the day the performance of both nodes track closely. This particular week corresponds to when the cane was fully grown, and more recent data after the cane has been cut shows a different pattern without the nightly communications loss. A similar pattern was noticed in early 2008. The data in Figure 3.20 and Figure 3.21 indicate that there are complexities in radio propagation which we do not yet fully understand or have a remedy for.

The wireless transmission model and range are important parameters for both network protocol and network deployment design. The research community has well observed that the "disc" transmission model is not applicable to most wireless transmission scenarios. Our experience shows that Surge can operate well in dynamic (asymmetric links and changes of link quality/connectivity) environments. However, we failed to find any network deployment methodologies that can model the environment well, and produce high connectivity networks. In particular, the methodology should take the deployment parameters, such as terrain, humidity, and height of the antennas, into

account when calculating the performance of radio links.

3.7.2 The Impact of Unreliable Downstream Link Transmissions

Routing protocols such as Surge [19] assume that all the network traffic in a WSN is toward one or a few gateways (sinks). Consequently, nodes store upstream (toward sinks) paths in their routing table only and use broadcast/flooding for downstream (toward sensors) traffic. Therefore, while the upstream traffic can be delivered efficiently, it is very inefficient to deliver downstream traffic, e.g., ACKs and NACKs. We observed the receipt of NACKs with substantial delay in our deployment, in particular to the nodes located deeply in the routing tree (e.g., Node 2 and Node 13 in Figure 3.11).

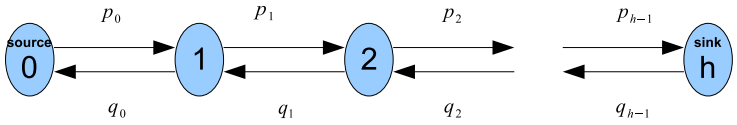


Figure 3.22: A Single Flow with Transmission Failure Probabilities.

Let us consider the impact of unreliable down link transmissions formally with the simplified model of linear topology in Figure 3.22. Let

$$a = \prod_{i=0}^{h-1}(1 - q_i^N) \tag{3.7}$$

and

$$b = \prod_{i=0}^{h-1}(1 - p_i^N) \tag{3.8}$$

where,

- p_i and q_i are the upstream and downstream link loss rate between Node i and Node $i+1$ $(0 \leq p_i, q_i \leq 1)$.
- $N = 6$ is the number of MAC-layer transmission retries derived from Section 3.5.4.
- $\boldsymbol{a}$ is the probability that the source receives NACK packets successfully.

- ***b*** is the probability that the sink receives retransmission packets successfully.

A lost packet can be recovered successfully if the source receives NACK packets from the sink successfully, and the packets are retransmitted to the sink successfully. Thus, the *loss recovery capability* c is

$$c = \left(1 - (1-a)^M\right) * b \tag{3.9}$$

where M is the number of times the sink requests retransmission from the source. In our case, the transmission delay, which is in terms of 10 milliseconds, is significantly smaller than the sending rate, which is in terms of 10 seconds (recall that the sampling rate is 1 minute per sample). The buffer size is 80 packets, and the sink sends a single NACK to the source every 10 packets received. Thus, for each lost packet detected at the sink, it may be requested for retransmission up to $M = 8$ number of times.

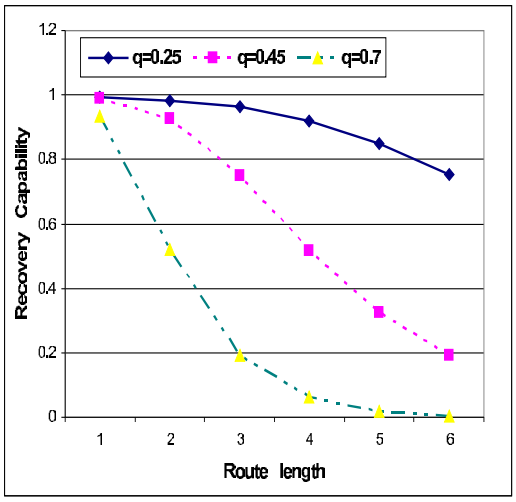


Figure 3.23: Recovery Capability for the loss rate p = 45%.

Figure 3.23 shows the recovery capability versus the route length for different downstream loss rates when the upstream loss rate is $p = 0.45$. The results suggest that the downstream reliability has a significant impact on the loss recovery ratio. When q is high, i.e. $q = 0.7$, the recovery ratio decreases exponentially as the path

length increases. Since the Surge protocol does not support downstream routing, we use flooding for downstream traffic in our case. The loss recovery capability of the node that has the most unreliable radio link, e.g., Node 11, is the worst (see Figure 3.14).

Nodes in routing protocols, such as Directed Diffusion [28], store bi-directional paths in their routing table. However, Directed Diffusion is not scalable with the number of traffic flows because intermediate nodes have to store the state of each flow that passes through them. The research community needs to address the downstream traffic problem in a scalable manner to improve the performance of reliable transmission protocols in the transport layer.

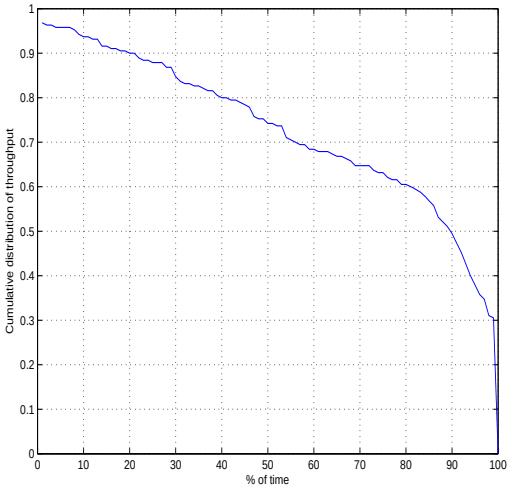


Figure 3.24: Cumulative Distribution of Internet Backlink Throughput over the Period 23/4/07 to 23/11/07.

3.7.3 Gateway and Internet Backlink

Our original plan was to link the sensor network directly to the office located about 4 km from the study area using several relay nodes with high-gain antennas. However, a site-survey in December 2006 identified a water tower located in the path loaded with Global System for Mobile communications (GSM) antennas that made it impossible to achieve this due to radio interference. Our interim solution was to use a

GPRS gateway. Our experience with General Packet Radio Service (GPRS) modems from three different vendors shows that they tend to hang after short periods of time (2-4 days) and can only be recovered by cycling power. Further, our Internet Service Providers (ISP) do not provide public Internet Protocol (IP) addresses to the GPRS devices, which made remote troubleshooting more difficult.

As described in Section 3.4 we switched to a wired ADSL service and our sensor network system has been operating independently since that time (April 11th, 2007).

However we have observed over the period quite poor internet connectivity, and Figure 3.24 shows a cumulative distribution computed over daily intervals. For 10% of the days we get better than 95% connectivity, more than 50% of the time we get less than 75% connectivity. This is worse performance than the sensor network itself. It is a combination of outage factors that includes the telco, our lab which uses a long-haul microwave link that has had major outages over the period, as well as scheduled weekend and after hours maintenance outages at our lab that have affected server rooms and core switch hardware.

3.7.4 Watchdog Timers

As well as the watchdog process in the gateway which power cycles the modem, we have embedded watchdog timers into each of the sensor nodes. This has proven to be very helpful for such a remote deployment (it takes more than 5 hours to travel from our lab to the Burdekin).

For example, two nodes frequently hung and were reset by the watchdog — we could see the reset behaviour from unexpected changes in their message sequence numbers. We also noticed that the resets correlated to changes in flow and speculated it was electrical interference from the pump motor switching on or off. We asked our local support person to disconnect the flow meter's pulse output from the node and the problem was rectified. We speculate that the interference led to a large burst of interrupts being generated by the digital input pin.

Figure 3.25: The Average Delivery Rate over the Period between April, 2008 and July, 2008

3.7.5 A Major Outage

We lost contact with our network on November 24 2007. Conflicting work pressures and the Christmas holiday season precluded a visit until early March, 2008. We found that at least two nodes had visible signs of lightning damage, through the AC power supply. Some others had nodes that were operational (LEDs blinking and serial debug output) but with very weak radio signal and we speculate this was also due to lightning damage. All faulty hardware was replaced, lightning arrestors fitted to all antennas, and surge arrestors fitted to all AC power supplies. To improve the robustness of the network in harsh environment, e.g., rain or the high humidity period before dawn, we also deployed an addtional node between Node 0 and Node 1. The network has been operational continuously since the repair trip.

Figure 3.25 shows the average delivery ratios for the period between April, 2008 and July, 2008. The results show that delivery ratios, after an additional node was deployed, are significantly improved, approximately 83%, which is 21% more than it was previously (62%).

3.8 Related Work

In the context of transport protocols, many tranport protocols have been proposed in literature such as Wisden [14], ESRT [25], RMST [20], etc. To the best of our knowledge, Wisden [14] (see Section 2.1.1.3) is closest in spirit to our work in that it uses the hybrid error recovery scheme that recovers packet losses both HBH and E2E. However, Wisden is designed for structural monitoring applications, where the packet latency is an important parameter. We design the reliable communication system for data streaming applications. Thus, latency is not a pressing concern, but the statistical reliability is. Moreover, Wisden uses HBH NACK for loss recovery, which is scalable since the sensor nodes are memory-constrained. Unlike Wisden, we achieve the HBH reliability through using the HBH eACK approach and hence, it is scalable when the network size or network densities increase (see Section 3.6.1). Akan et al. proposed ESRT (Event to Sink Reliable Transport) for statistical reliability in [25] (see Section 2.1.3.1), where each sensor node has the same sending rate. This protocol achieves the reliable detection of an event by controlling the sending rate of the sources for congestion avoidance. Since there is no packet retransmission in ESRT, it does not ensure the temporal relationship of the collected data when packets are lost. Moreover, ESRT assumes that the sink can communicate with all sources directly, which is not valid in many practical WSN deployments like ours. Stann et al. proposed RMST in [20] (see Section 2.1.1.2), a reliable transport protocols in sensor networks. RMST uses HBH NACK for loss recovery. However, RMST is tightly bound to Directed Diffusion routing protocol [28] in which packet losses are recovered HBH using caches in the nodes along the path to the sink. Similar to Wisden, RMST is not scalable because it requires each intermediate node to cache all packets received from each upstream source. To the best of our knowledge, the performance of the ESRT and RMST were evaluated by simulations only. In addition to simulations, we evaluated the performance of our communication protocol in a real outdoor environment, which allows us to study the

impacts of the highly dynamic radio transmissions on the data transmission reliability, and the benefits of E2E recovery.

In the context of sensor network deployment, many sensor network applications have been proposed for applications such as habitat monitoring [8, 9], health [15], education [65], structure monitoring [14], automatic animal vocalization recognitions [66], precision agriculture [67–70] and the military [11, 12] in the past few years. While these deployments can provide unprecedented fine-grained environmental data for scientific research, to the best of our knowledge, few sensor networks have deployed for long-term outdoor industrial applications. Further, limited success has been achieved by previous outdoor industrial application of sensornet deployments [67]. Our Burdekin sensor network deployment aims to provide a feasible solution for a critical problem (water salination) to an industrial partner, e.g., North Burdekin Water Board, by deploying a robust system, which can operate independently for a long time, in a harsh remote outdoor environment.

3.9 Chapter Summary

In this chapter, we have presented our entire cross-layer design system for a water quality sensor network deployment in a remote tropical area of north eastern Australia. Our goal was to collect real-time water quality measurements together with the amount of water being pumped out of the area, and investigate the impacts of current irrigation practice to the environment-in particular, underground water salination. This is a challenging task featuring wide geographic network coverage, highly dynamic radio transmissions, high E2E packet delivery rate requirements, and a hostile system deployment environment.

We have designed, implemented, and deployed a sensor network system, which has been collecting water quality measurements for over a year. The collected results show that sensor networks can provide a potential solution to deploy a sustainable irrigation

system, e.g., maximizing the amount of water pumped out from an area with minimum impact on water quality.

We have designed a reliable data system that features a communication component across multiple protocol layers to increase the E2E data delivery ratio. Our comprehensive evaluation, which include theoretical analysis, simulation, and experiments in the field, shows that the network is robust to the dynamics and provides a reasonably data delivery ratio.

Our experience shows that the environment at Burdekin is highly unstable. We observed that the link qualities are considerably lower at night time than in the morning time. Because of this, it is desirable to have multiple communication protocols which can run in different network conditions. In the next chapter, we present a sensor reliability management framework, which allows the sensor network to handle a range of possible parameter values or even handle a set of reliable communication protocols depending on node topology, network connectivity, and node status, to control the data reliability.

Chapter 4

Reliability Management Framework for Wireless Sensor Networks

Data reliability of a sensor node is described by the probability of a data packet being delivered from the sensor node to the sink. Data reliability management is the task that ensures the delivery of data from a sensor node to the base station. Ensuring data reliability across many sensor nodes in a network is a challenging task because data transfer in WSNs is susceptible to loss when there are node failures, environmental interferences, nodes joining or leaving the network, power depletion, etc. For example, in the Burdekin water quality monitoring application described in Chapter 3, it is observed that the link qualities are considerably lower at the night time than in the morning time. In this case, it is desirable to have multiple protocols which can run for different network conditions. Second, because of the scale of sensor networks, typically with tens, or even hundreds of nodes, coordinating the communication across these many nodes is complex.

On the other hand, in many applications, sensor nodes are battery-powered. The unattended nature of sensor nodes and hazardous sensing environments preclude battery replacement as a feasible solution while many sensor network applications demand that the network must operate for a long period of time. Minimizing energy consumption while ensuring the data reliability in such dynamic conditions is a complicated task. Thus, it is nearly impossible for a single protocol to be appropriate all the time, even

within a single sensor network application [71]. When the reliability of the network degrades, the network is no longer capable of delivering useful information to the users. Therefore, data reliability management, which is capable of handling a range of possible parameter values or even handling a set of reliable communication protocols depending on node topology, network connectivity, and node status, to control the data reliability for each sensor node, is crucial.

In this chapter, we propose and implement a **S**ensor **R**eliability **M**anagement framework called **SRM** for WSNs. SRM is based on a hierarchical management architecture and policy-based network management paradigm formulated by IETF [17]. To demonstrate SRM, we present two examples for data reliability management.

4.1 Chapter Contributions

The primary contributions of this chapter are:

- We proposed a sensor reliability management framework for WSNs called SRM. SRM is based on hierarchical management architecture and policy-based network management paradigm formulated by [17]. SRM allows the network administrators to interact with the network by defining the management policies. SRM also provides a self-control capability to the network. SRM consists of four modules: a user policy specification module, an evaluation module, a decision making module, and an action module. The cooperation among these modules provides adequate information to the users while reducing energy consumption. This thesis restricts SRM to reliability management, but the same framework is also applicable for other management services by providing the management policies.

- We implemented and evaluated SRM in a real-testbed. Our experimental results show that SRM can ensure the reliability requirement and reduces significantly energy consumption.

4.2 Chapter Organization

In Section 4.3, we investigate the requirements for a management system, specifically a reliability management system in WSNs. In Section 4.4, we describe the SRM in detail. Section 4.5 describes the implementation details of SRM framework. In Section 4.6, we provide two examples to demonstrate the benefits of using SRM for data reliability management. Section 4.8 summarizes the work presented in this chapter.

4.3 Requirements for Reliability Management Framework for WSNs

In this section, we discuss the desirable requirements sought in a management system for WSNs.

- Energy Efficiency: Although a network management system can improve network performance, it also introduces additional control traffic to regulate the operations of the network. The energy limitations of resource-constrained sensor nodes demand minimal network management traffic. This constraint greatly impacts on the choice of the mechanisms or protocols used for the management tasks.

- Robustness and Fault Tolerance: In WSNs, network topology often changes because of node failures, environmental interferences, node mobility, and nodes joining/leaving the network. The management system should be robust to the network dynamics by reconfiguring when necessary.

- Adaptivity: Given the dynamic nature of WSNs, an adaptive mechanism is required which enables the network to react to changes in network conditions. For example, the management system, upon being alerted of low energy of one or more sensor nodes, should reconfigure the network and allow graceful degra-

dation of performance. In order to accomplish this, the management system should monitor the capabilities and the performance of the sensor nodes involved, and uses this information as one of the criteria to decide appropriate actions for different types of sensor nodes.

- Response Time: Some applications such as target tracking require data to be received in a timely manner. To meet an application's latency requirement, the management tasks may need to be performed in a timely manner.

- Scalability: A generic WSN is envisioned as consisting of hundreds of sensor nodes. To support large-scale sensor networks, the management system should take scalability issues into account. The performance of the management system should not degrade when the number of nodes or the network densities increase.

- Programmability: Due to the dynamics of the system, programming and reconfiguring the network is necessary. It implies that the management system should support the programming paradigms as well as the reconfiguration methods. Given the limited energy on sensor nodes, network reprogramming is a challenging task.

4.4 SRM: a *S*ensor *R*eliability *M*anagement Framework for WSNs

In this section, we present the SRM framework and its major modules in detail. Section 4.4.1 discusses the management architecture used in SRM. The following subsections discuss each module of SRM. Finally, we present an algorithm for controlling data reliability.

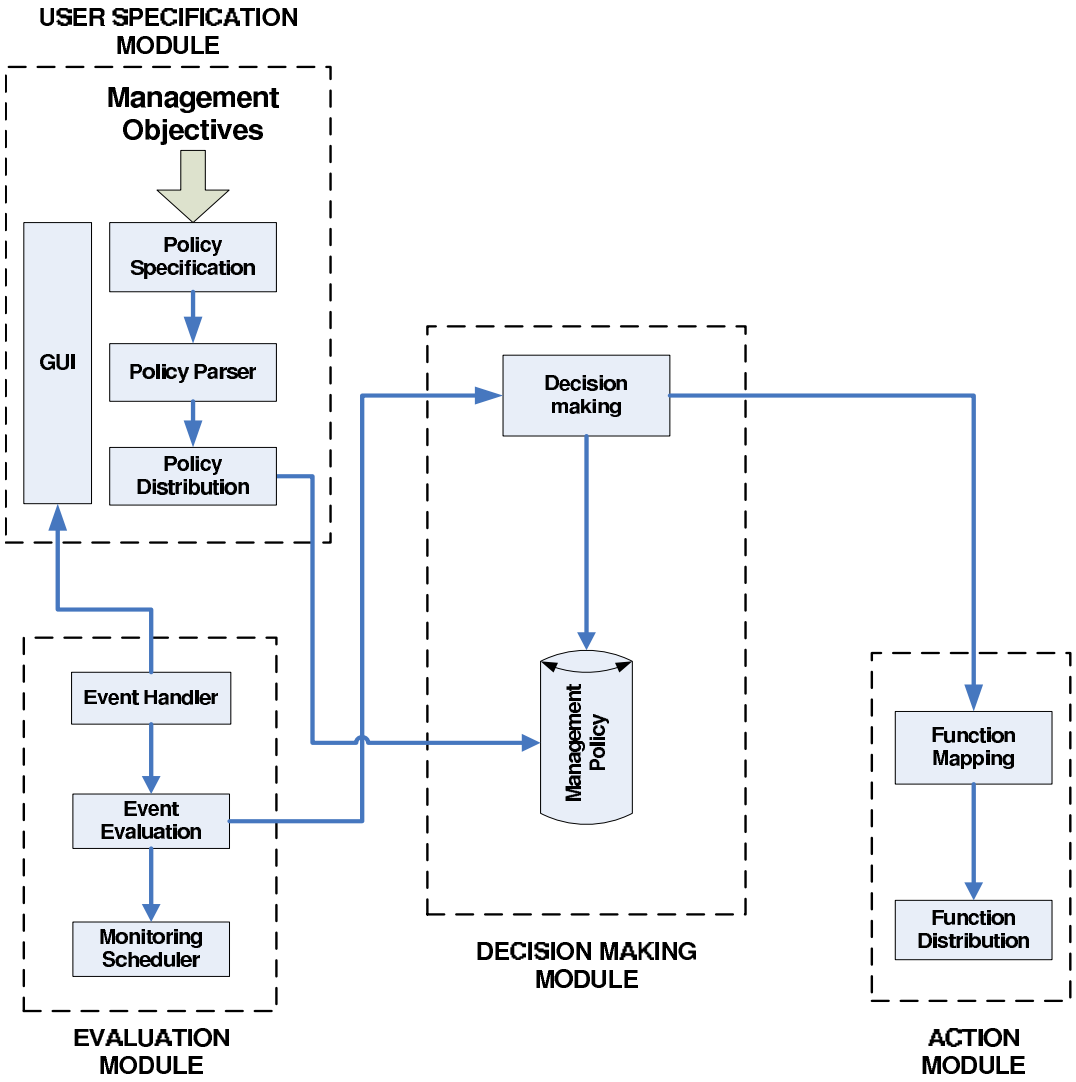
4.4.1 Management Architecture

To address the scalability issue, SRM uses a hierarchical management architecture. The management tasks are distributed to all sensor nodes including the cluster heads and the base station. Management policies are hierarchically distributed over the network. Management policies are divided into three levels: **node level**, which consists of a set of light-weight management rules that require less resources for estimation and can be performed locally; **cluster level**, which consists of a set of medium-weight management policies that control the reliability of the cluster; and **base station level**, which consists of a set of heavy-weight management policies that control the reliability of the entire network.

Next, we discuss the modules which constitute the SRM framework and their interdependencies. As shown in the Figure 4.4.1, a management architecture is adopted for the reliability management framework, consisting of four modules: an evaluation module, a user policy specification module, a decision making module, and an action module. The interaction among these modules enables the management framework to efficiently adapt to the network dynamics. The following subsections discuss the individual role played by these modules in the SRM framework.

4.4.2 Evaluation Module

The evaluation module is responsible for collecting the management information required to estimate the reliability of the network. The evaluation module consists of three components: an event handler, an event evaluation, and a monitoring scheduler. The event handler component is a packetizer, which captures the management packets, translates them to a known data structure, and parses them to the event evaluation component. The event evaluation component evaluates the reliability of the network by comparing the actual reliability to the required reliability and make a conclusion on the health of the network. To reduce energy consumption, the monitoring scheduler



The SRM Framework, consisting of four modules: an evaluation module, a user policy specification module, a decision making module, and an action module.

component is responsible for controlling how frequently the management information is collected. Typically, management information is exchanged periodically. However, when there are not many interesting events, the monitoring scheduler may request to reduce the collection frequency for energy-saving. The monitoring scheduler component has two modes: a passive mode, where the system collects information every pre-defined period of time; and a reactive mode, where the system collects information if an interesting event occurs.

Rule ID	Condition	Action	Scope
1	IF ENERGY <= E_TH AND DELIVERY_RATIO > DR_TH THEN	DECREASE_TRANSMISSION_RATE	LOCAL

Table 4.1: An Example of a Reliability Rule.

4.4.3 User Policy Specification

A policy is a set of rules governing decisions that will be implemented to achieve the objectives [17]. In SRM, we use the Policy Framework Definition Language (PFDL) [72] to express management rules. The PFDL simply expresses lists of `IF<condition> THEN <action> SCOPE <scope>` type of rules, where `<condition>` is a disjunctive normal form of condition expressions, `<action>` is a list of single action statements, and `<scope>` is the sensor node where the policy is enforced. Each rule comprises one or more terms joined by logical operators (e.g. `AND, OR`). A term comprises one management variable, a binary operator from the set $\{<, <=, =, >=, >\}$, and a reference value, which must be a real number or from a list of pre-defined constants. There are three scopes defined in SRM: `LOCAL`, for a single sensor node; `CLUSTER`, for a cluster; and `NETWORK`, for the entire network. A list of the rules will form a policy. If the evaluation of the condition expression request succeeds, the action list will be performed. In the SRM framework, a management policy is described by an XML schema document [73].

Table 4.1 shows an example of a rule in the reliability policy. In this example, E_TH, DR_TH denote the pre-defined thresholds of the remaining energy level of a sensor node and the desired reliability. The rule states that when the remaining energy of a sensor node is lower than E_TH it should reduce its transmission rate if the delivery ratio is satisfactory, i.e. greater than the threshold DR_TH.

Figure 4.2 shows the corresponding XML schema. The condition is specified by `<CONDITION> </CONDITION>` tag. It consists of management variable, defined by `<VARIABLE> </VARIABLE>` , e.g., `ENERGY`; the logical operators `<OPERATOR> </OPERATOR>`, e.g., `&lt;=`; and the reference value, defined by `<REFERENCE> </REFERENCE>`, e.g., `E_TH`. The action is specified by `<ACTION> </ACTION>` and is followed by a list of functions

```
<?xml version=‘‘1.0’’ encoding=‘‘UTF-8’’?>
<Rule>
 <CONDITION>
 <VARIABLE> ENERGY </VARIABLE>
 <OPERATOR> &lt;= </OPERATOR>
 <REFERENCE> E_TH </REFERENCE>
 <VARIABLE> DELIVERY_RATIO </VARIABLE>
 <OPERATOR> &gt </OPERATOR>
 <REFERENCE> DR_TH </REFERENCE>
 </CONDITION>
 <ACTION>
 <FUNCTION> DECREASE_TRANSMISSION_RATE </FUNCTION>
 </ACTION>
 <SCOPE SCOPE=‘‘LOCAL’’ />
</Rule>
```

Table 4.2: The XML Schema for the defined Reliability Rule.

which need to execute. The scope is specified by `<SCOPE>` and is followed by its value.

The user policy specification module has four components: a policy specification, a policy parser, a policy distribution, and Graphical User Interfaces (GUIs). The policy specification component allows users to define the management policies via a GUI or a web-page. The policy parser component is responsible for validating and translating the management policies into a data structure, which is understandable by the computer operating system. The policy distribution component distributes the management policies over the network. In SRM framework, we define GUIs as one of the components in the user policy specification module. By capturing and analyzing the packets received in the event handler component, the GUIs provide the visualization of the current state of the network including data reliability ratio, routing topology, link quality, etc.

4.4.4 Decision Making Module

The decision making module is the central layer of the framework. It has two components: a decision making component and a management policy component.

- Decision Making Component: Based on the estimated reliability from the evaluation module and the policies defined in management policy component, the

decision making component determines appropriate actions to be executed and passes the request to the action module.

- Management Policy Component: The management policy component is similar to the Management Information Base (MIB) in traditional network management [74], which stores the management policies defined by the network administrator. In SRM framework, two policies are defined: a data reliability policy and a congestion policy. Other policies can be easily integrated into the framework. The data reliability policy contains a set of rules which ensures the end-to-end delivery ratios. The congestion policy contains a set of rules which detect and control the network congestion. As the energy is a critical resource in WSNs, both policies take energy-efficiency into account.

4.4.5 Action Module

The action module is responsible for performing the action assigned by the decision making module. The core of the action module is a management function, which can be a single command such as an alarm operation; an algorithm such as adjusting retransmission algorithm; or a protocol such as a transport protocol. An action may be executed by a set of functions on the sensor node in response to a call from the decision making module.

The action module has two components: a function mapping, and a function scheduling. The function mapping component maintains a function list for each action, which maps the action to a set of functions being executed. The function scheduling component is responsible for distributing the assigned functions into the sensor nodes, which need to execute the functions.

After executing the action, the action module returns the state of execution to the decision making module. SRM defines three states: success, when the action is performed successfully; fail, when a failure occurs during the execution of the function;

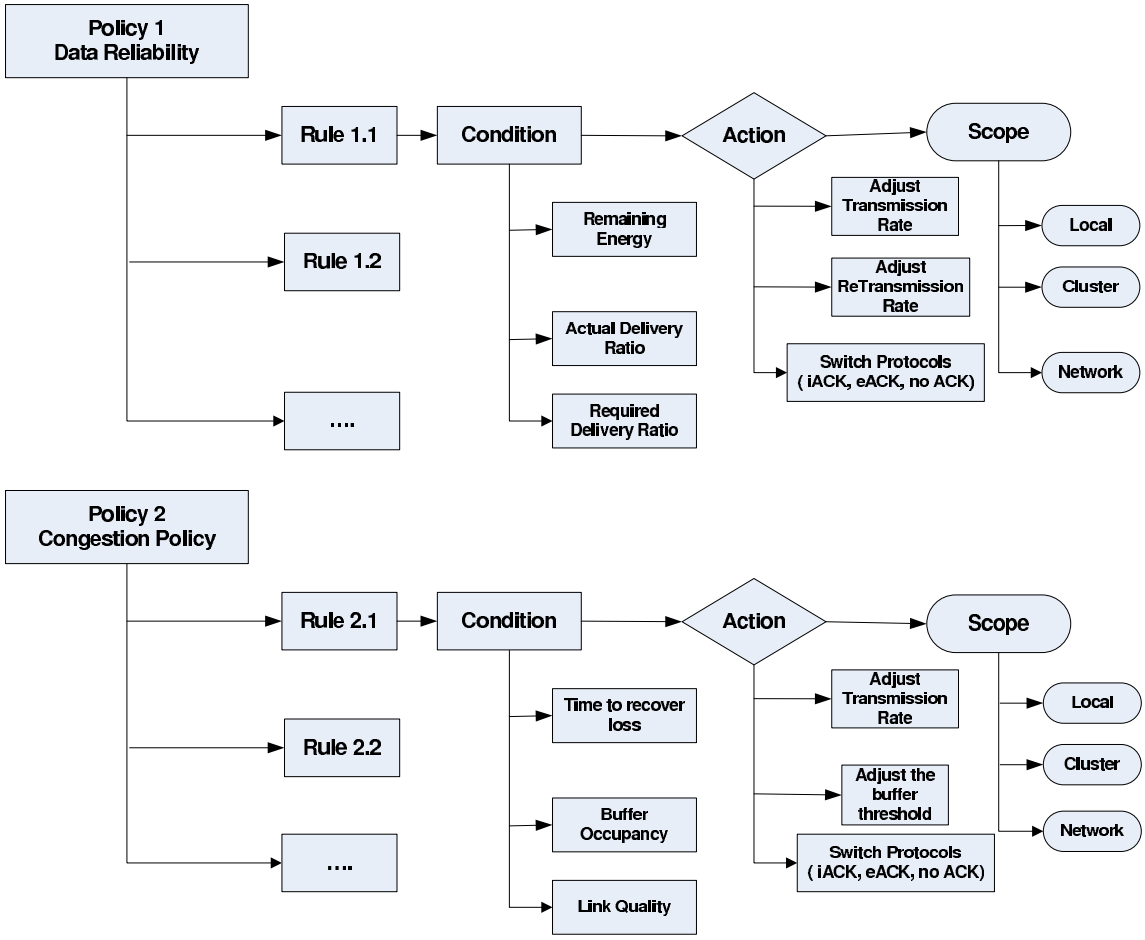


Figure 4.1: The Management Policies in the SRM Framework.

and executing, when the action is currently being executed.

Figure 4.1 describes management policies and actions used in SRM. In the data reliability policy, there are three defined actions: `ADJUST TRANSMISSION RATE`, `ADJUST RETRANSMISSIONS RATE`, and `SWITCH PROTOCOL`. `ADJUST TRANSMISSION RATE` is responsible for increasing/decreasing the transmission rate. Without introducing any extra protocol overhead, a sufficient quantity of data packets transmitted to the base station can ensure the E2E delivery ratio. For example, ESRT [25] can be used to implement this action. ESRT dynamically chooses a transmission rate that ensures event reliability for the application. Although `ADJUST TRANSMISSION RATE` action is simple and easy to implement, it is not energy-efficient when the link loss rates are high [75]. To reduce en-

ergy consumption, SRM also uses retransmission and acknowledgment schemes. `ADJUST RETRANSMISSIONS RATE` action is responsible for adjusting the number of retransmissions to balance the data reliability and energy consumption. The last action is `SWITCH PROTOCOL`, which allows the network to change the transport protocol on the fly. There are three protocols defined in SRM: noACK, eACK, and iACK. The choice of these transport protocols comes from the empirical studies in [71], which provides a baseline to compare energy consumption and data reliability between these three protocols. After a transmitter sends a packet, it will wait for an acknowledgment (ACK) from the receiver. If it does not receive the ACK before its pre-defined timer expires, the packet is retransmitted. The node will retransmit the packet until either the ACK is received or it has retransmitted the packet more than a pre-defined number of times called maximum number of retransmissions. If the ACK is explicitly sent from the receiver, the scheme is called eACK. In multi-hop wireless links, the transmitter can overhear forwarding transmissions from the receiver and interprets them as acknowledgment. This scheme is called iACK. There is a trade-off between energy consumption and data reliability in these three protocols. An insufficient maximum number of retransmissions may cause a packet to be lost as it travels to the base station, wasting energy and network resources, as well as degrading E2E reliability. Conversely, there will be energy-inefficiency when the maximum number of retransmissions is too high.

There are three actions defined in congestion management policy called: `ADJUST TRANSMISSION RATE`, `ADJUST BUFFER THRESHOLD`, and `SWITCH PROTOCOL`. The `ADJUST TRANSMISSION RATE` is responsible for increasing/decreasing the transmission rate for controlling network congestion. A high transmission rate may cause network congestion. Conversely, a low transmission rate reduces network throughput. The `ADJUST TRANSMISSION RATE` action tries to maintain the transmission rate at the highest possible rate which congestion does not occur. For example, RCRT [21] can be used to implement the `ADJUST TRANSMISSION RATE` action. RCRT dynamically adjusts the

transmission rates by using the time to recover a packet loss as a congestion indicator. The `ADJUST BUFFER THRESHOLD` action defines a congestion threshold for the local packet queue where the incoming packet will be dropped. A small congestion threshold for the packet queue can avoid congestion, but reduces data reliability. Thus, the maximum congestion threshold is demanding. CODA [29] can be used to implement this action for congestion detection and control. In SRM framework, we also define the `SWITCH PROTOCOL` action for network congestion control. When the network is congested, noACK can be used to reduce the number of retransmissions, and the amount of traffic exchanged in the network, while still providing a certain level of data reliability ratio.

In addition, other actions are defined in SRM such as: `ALERT`, send the alarm message to the base station, `RETURN`, return the value after executing the action, etc.

4.4.6 An Algorithm for the `ADJUST_RETRANSMISSION_RATE` Action

We propose an algorithm for the `ADJUST_RETRANSMISSIONS_RATE` action for controlling the E2E delivery ratios. Let us denote R^* and Δ as the desired average reliability requirement and the deviation, respectively, where $0 < \Delta$ and $R^* < 1$. Let us also denote $NR(i)$ as the current number of retransmissions allowed at node i and the MAX_NR as the maximum number of retransmissions allowed at each node. We define $\bar{R}$ as:

$$\bar{R} = \frac{\sum_{i=1}^{i=N} R(i)}{N} \tag{4.1}$$

where $R(i)$ is the actual delivery ratio obtained at node i and N is the number of sensor nodes in the network. The objective of the algorithm is to maintain the delivery ratio R^* such that:

$$R^* - \Delta < \bar{R} < R^* + \Delta \tag{4.2}$$

The algorithm is described as follows. After a predetermined period of time, the evaluation module activates and estimates the delivery ratio $R(i)$ for every sensor node i. If the condition of the Equation (4.2) is held, SRM will do nothing. Otherwise, the decision making module will perform a management action to adjust the reliability. There are two cases:

- If $R(i) \leq R^* - \Delta$, the decision making module will increase the number of retransmissions $NR(i)$ at node i. However, if $NR(i) == MAX\_NR$, node i is no longer capable of improving the reliability. In this case, the decision making module will find a node on the routing path of node i toward the sink, say node j, in which $NR(j) \leq MAX\_NR$ is held and ask node j to increase $NR(j)$. If there is no node j available, it will inform the base station to prevent further requests.

- If $R(i) \geq R^* - \Delta$, the decision making module will check if the node i is required to increase the $NR(i)$ by its children. If it is not, then it will decrease $NR(i)$.

4.5 Implementation

4.5.1 Overview

The implementation has two parts: the base station side and the sensor node side. The base station side was implemented in a high-level programming language, Java, and was run on a PC. On the sensor node side, the SRM was implemented in C programming language under the Fleck Operating System (FOS) in a 30-node Fleck-3 [5] testbed. FOS is an operating system for WSN nodes developed by CSIRO [76]. It provides a priority-based, non-preemptive (cooperative) threading environment with separate stacks for each thread, which has the advantage of providing a simple concurrent programming model which does not require semaphores. The implementation was run on a Fleck-3 platform, but it is straightforward to port it to other platforms such as Mica2 [1] or

Telos [44]. In this work, we only evaluate the network which comprises a base station and sensor nodes. Thus, the base station maintains the management policies at both the base station and the cluster head levels.

4.5.2 Protocol Engine

Surge Reliable was used (for brevity we called Surge) [19] as the routing protocol. Surge is a reliable multi-hop routing protocol that uses link quality as its routing metric. In Surge, each node dynamically selects its parent based on the link quality to construct a stable routing tree to the sink (see Section 2.1.1.1). It is necessary to note that the focus of this work is on designing and implementing a reliable data management for WSNs, but not on designing a reliable data transmission protocol for the downstream direction (from the sink to the sensor nodes). Thus, a flooding protocol is used for the downstream communication. Flooding implements the best-effort flood: each node rebroadcasts packet exactly once and prevents retransmissions by maintaining a duplicate packet list. To avoid packet collision, a packet is rebroadcast with a small random delay. Although we use flooding, any reliable communication protocols from the sink to the sensor nodes can be used such as [32]. To minimize energy consumption, multiple commands are bundled into a single message. Each command consists of nodeID (2 bytes) and commandID (1 byte). Table 4.3 shows the packet header used in FOS. As the packet length in FOS is 32-bytes, each management packet can hold up to 5 commands.

4.5.3 Sensor Node Software Component

In the SRM framework, each sensor node has the responsibility of providing the information requested by the base station, and performing the functions as specified by the command requested from the base station. The sensor node also maintains its own management policies which will be executed when the conditions are fired. The example of an event handler component and a monitoring scheduling component in the

Packet Field	Size	Description
address	uint16_t	The local address
group	uint8_t	The group ID
type	uint8_t	Opcode: 0xfa = data packet, 0xfe= management packet
length	uint8_t	The data length
crc	uint16_t	The cycle redundancy code
ack	uint16_t	The acknowledgment
rssi	uint16_t	The received signal strength indication
time	uint16_t	The time
originaddr	uint16_t	The original packet address
seqNo	uint8_t	The sequence number
nodeID	uint16_t	The destination address
function ID	uint8_t	The function ID

Table 4.3: The Packet Header.

evaluation module are:

```
void * fos_network_rx_thread (void * arg)
{ fos_message_t rxm;
  ....
  while (1) {
    if (fos_macfilter_read (0, f, &rxm) == 1) {
            if (rxm.type  == FOS_MAC_TYPE_MANAGEMENT) {
                fos_management_handler(&rxm);
            }
         }
        }
        ....
}
```

```
void * fos_network_management_send(
uint16_t addr, uint8_t * content, int8_t n, uint8_t flags)
{
  ....
}
```

4.5.4 Base Station Software Component

In the SRM framework, the base station has the responsibility of managing the sensor nodes and providing management information of the sensor network for its users. The base station sensor node is required to provide a bridge between the PC and other sensor nodes. The base station receives the packets from sensor nodes, either management packets or data packets, and forwards them to the PC via a serial port. The network administrator forwards management commands to PC via its TCP port and they eventually reach the sensor nodes via the base station.

4.5.5 Graphical User Interfaces (GUIs)

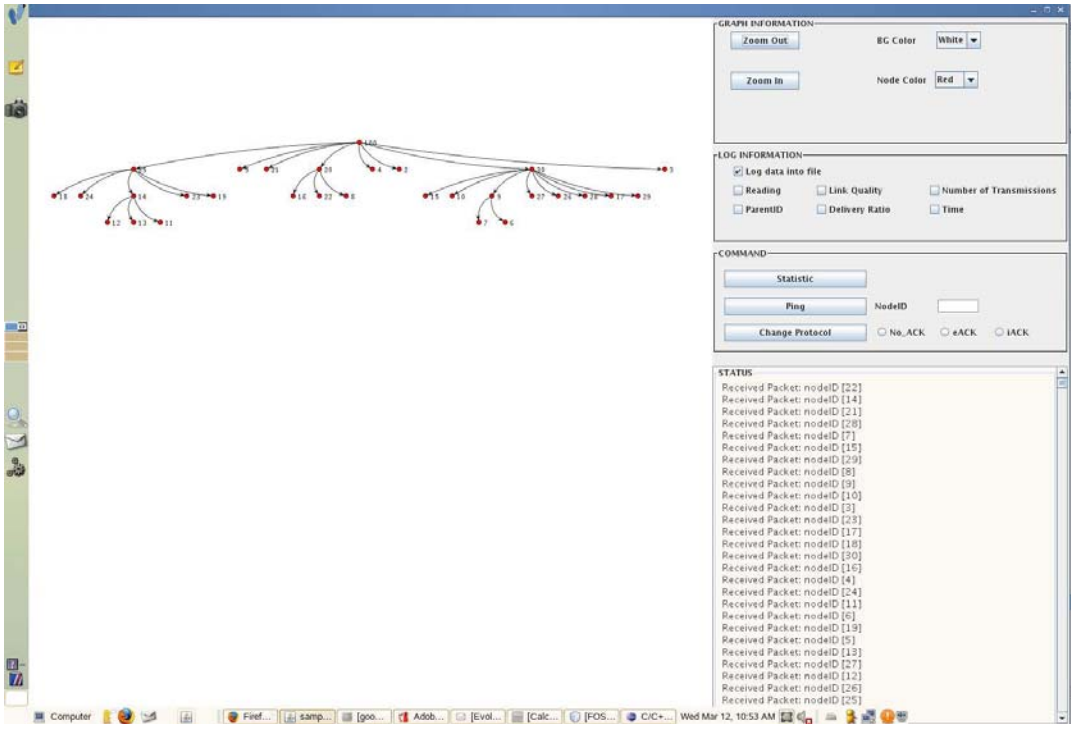
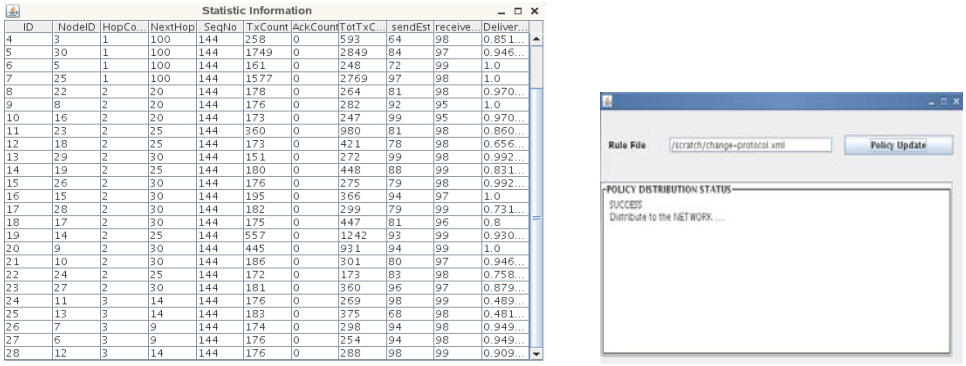


Figure 4.2: Network Topology GUI.

A set of GUIs have been developed at the base station side for the network administrators, as shown in Figure 4.2 and Figure 4.3. Each tool is triggered by a received packet from a sensor node via the TCP port or the requests from the network adminis-



(a) Network Statistic GUI (b) Management Policy GUI

Figure 4.3: Network Statistic and Management Policy GUIs.

trator. The first GUI provides fundamental network functions including displaying the network topology, updating network status, and logging the received data into a file. It also enables the network administrator to ping a node and to change the transport protocol on the fly. The second GUI (Figure 4.3(a)) provides statistical information about the network such as the next-hop ID, the sequence number, the link qualities, etc. The third GUI (Figure 4.3(b)) allows the administrator to load pre-defined management policies in XML format and distribute them to the network.

4.6 Examples

In this section, we provide two examples to demonstrate the data reliability policy in SRM with two actions: `SWITCH_PROTOCOL` and `ADJUST_RETRANSMISSIONS_RATE`. A data streaming application, where the sensor nodes report data to the sink periodically, is considered. The following metrics are used:

- Number of Data Transmissions: is the total number of data packets exchanged in the network and the entire experiment. It includes the retransmitted data packets and the acknowledgment packets.

- Number of Management Transmissions: is the total number of packets used for

Rule ID	Condition	Action	Scope
1	IF TIME $\geq$ 2400	SWITCH_NO_ACK	NETWORK

Table 4.4: An Example of User Reliability Policy.

the management activities over the entire experiment.

- Delivery Ratio: is the ratio between the number of packets received at the base station and the number of packets originally sent from the sensor node.

4.6.1 Example 1: SWITCH_PROTOCOL Action

In data streaming applications, there are usually different interests at each period of time. For example, in the Burdekin water quality monitoring application described in Chapter 3, there is more interest during the period that the farmers pump the water to the field. Thus, it is desirable to have a higher delivery ratio during this period than at other periods. When the farmers pump the water, the SRM decides to use a transport protocol which provides good delivery ratio, i.e. eACK, but possibly with high energy consumption, and uses an energy-efficient transport protocol, i.e. no acknowledgment, at the other period of time to reduce energy consumption, but still provide a certain level of reliability.

In this example, each sensor node periodically sends a data packet to the base station at the rate of one packet every 10 seconds. Assume that the farmer will turn off the water pumping after 40-minute running. In this case, the SRM specifies a data reliability rule that initially uses an eACK transport protocol with the maximum number of retransmissions being 7 and switches to the no-ACK scheme after running 40 minutes (2400 seconds). The rule and its XML schema are shown in Table 4.4 and Table 4.5. The experiment was run for around 1.5 hours (5000 seconds).

Figure 4.4(a) and Figure 4.4(b) show the E2E delivery ratio and the average number of transmissions (including the retransmissions). First, we observe that the network can successfully perform the specified rule and switch the protocol from eACK

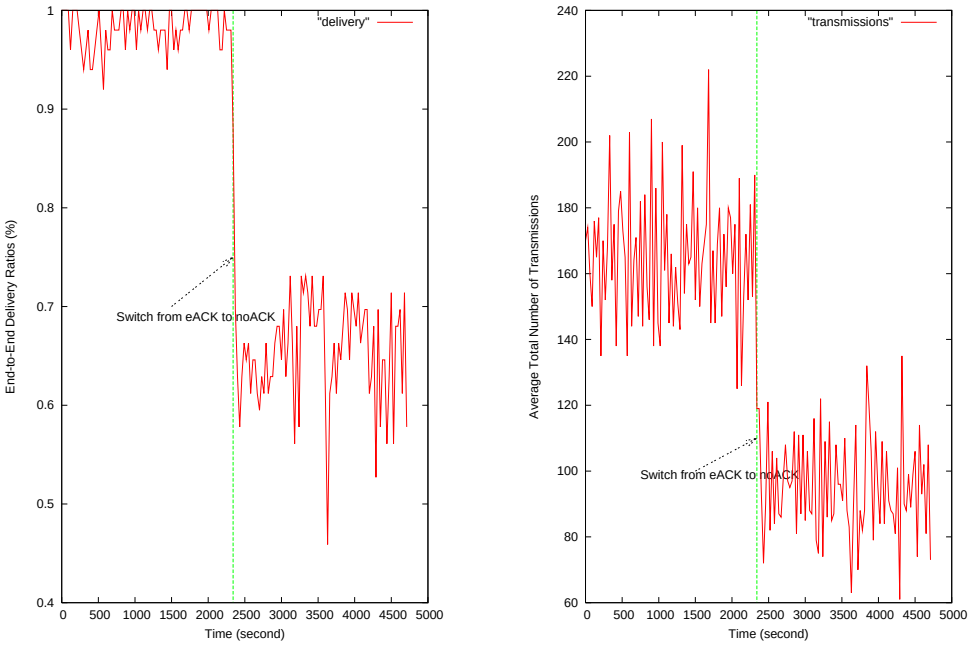
```
<?xml version=‘‘1.0’’ encoding=‘‘UTF-8’’?>
<Rule>
 <CONDITION>
 <VARIABLE> TIME </VARIABLE>
 <OPERATOR> &gt;= </OPERATOR>
 <REFERENCE> 2400 </REFERENCE>
</CONDITION>
 <ACTION>
 <FUNCTION> SWITCH_NO_ACK </FUNCTION>
 </ ACTION>
 <SCOPE SCOPE=‘‘NETWORK’’ />
</Rule>
<Rule>
```

Table 4.5: The XML Schema for the defined Reliability Rule.

to no-ACK on the fly at the defined time. The trade-off between energy consumption and data reliability is observed here. When eACK is used, the delivery ratio is high (on average is more than 95%), but the number of transmissions is large. On the other hand, when the no-ACK is used, the delivery ratio degrades (on average is about 65%), but the number of transmissions are significantly low. For WSN applications where the delivery ratios are not strictly required such as the Burdekin water monitoring application, the hybrid between no-ACK and e-ACK can be used to balance the energy consumption and delivery ratio. This reliability rule can save more than 50% energy consumption if the delivery ratio of 65% is satisfactory.

4.6.2 Example 2: `ADJUST_RETRANSMISSIONS_RATE` Action

In this example, we demonstrate the performance of SRM when the network is dynamic. The `ADJUST_RETRANSMISSIONS_RATE` action proposed in Section 4.4.6 is used to balance the energy consumption and the delivery ratios. The desired data delivery requirement R^* is 60%, 70%, and 90%, respectively, with $\Delta = 0.02$. The algorithm maintains the reliability ratio between $[R^* - \Delta, R^* + \Delta]$. The experiment was run for 35 minutes (2200 seconds). In order to evaluate the performance, artificial losses are introduced for all the links in the network for a period of time. Specifically, the link



(a) Delivery Ratio (b) Average Number of Transmissions

Figure 4.4: The Demonstration of the Switching Protocol Action.

Rule ID	Condition	Action	Scope
1	IF DELIVERY_RATIO < DR_TH1		
	AND RETRANSMISSION < $MAX\_RETRANSMISSION$ THEN	INCREASE_RETRANMISSION_RATE	LOCAL
2	IF DELIVERY_RATIO > DR_TH2 AND RETRANSMISSION > 1 THEN	DECREASE_RETRANMISSION_RATE	LOCAL

Table 4.6: Reliability Rules for Balancing the Delivery Ratio and Energy Consumption.

layer randomly dropped packets with 35% probability during the period between 15 minutes (1000 seconds) and 25 minutes (1600 seconds), as shown in Figure 4.6(a). This effectively increased the expected energy consumption of the sensor nodes.

4.6.2.1 Results

Figure 4.5 is a snapshot of one of routing trees constructed during the experiment. Due to the changes in wireless link quality, the routing tree was dynamic with significant routing variability.

```
<?xml version=‘‘1.0’’ encoding=‘‘UTF-8’’?>
<Rule>
 <CONDITION>
 <VARIABLE> DELIVERY_RATIO </VARIABLE>
 <OPERATOR> &lt </OPERATOR>
 <REFERENCE> DR_TH1 </REFERENCE>
 <VARIABLE> RETRANSMISSION </VARIABLE>
 <OPERATOR> &lt </OPERATOR>
 <REFERENCE> MAX_RETRANSMISSION </REFERENCE>
</CONDITION>
 <ACTION NAME=‘‘INCREASE_RETRANSMISSION_RATE’’ />
 <SCOPE SCOPE=‘‘LOCAL’’ />
</Rule>
<Rule>
 <CONDITION>
 <VARIABLE> DELIVERY_RATIO </VARIABLE>
 <OPERATOR> &gt </OPERATOR>
 <REFERENCE> DR_TH2 </REFERENCE>
 <VARIABLE> RETRANSMISSION </VARIABLE>
 <OPERATOR> &gt </OPERATOR>
 <REFERENCE> 1 </REFERENCE>
</CONDITION>
 <ACTION NAME=‘‘DECREASE_RETRANSMISSION_RATE’’ />
 <SCOPE SCOPE=‘‘LOCAL’’ />
</Rule>
```

Table 4.7: The XML Schema for the defined Reliability Rules.

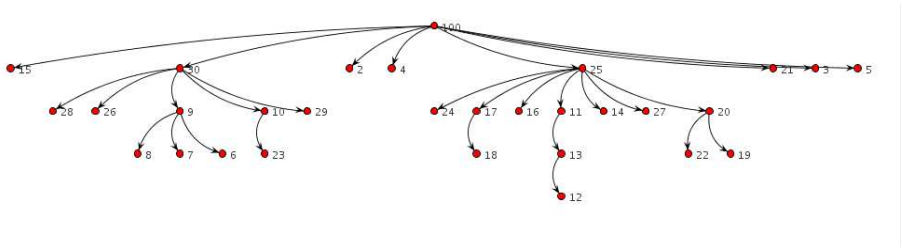
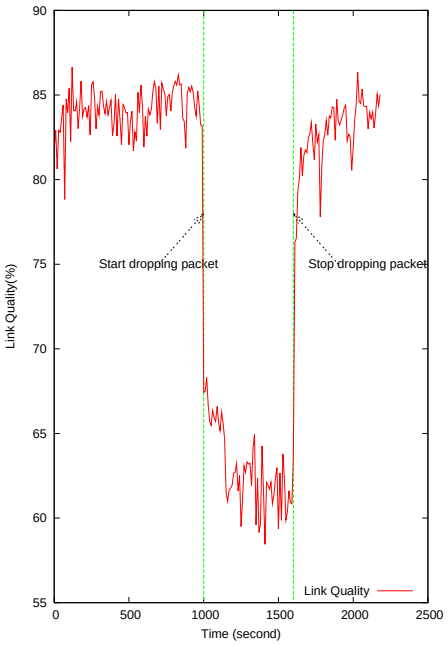
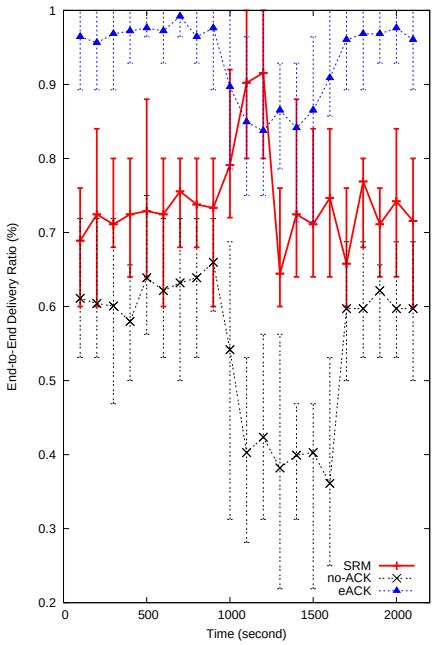


Figure 4.5: Network Topology.

Figure 4.6(b) shows the delivery ratio of 30-node versus time when the required delivery ratio is $R = 70\%$. The error bars in the figure show the minimum, the average, and the maximum of delivery ratio for all 30 nodes over a period of 100 seconds. With no-ACK, the average delivery ratios dramatically reduce when the link loss rates are high whereas with eACK (with 7 maximum number of retransmissions), the average delivery ratios



(a) Link Quality, $R^* = 70\%$

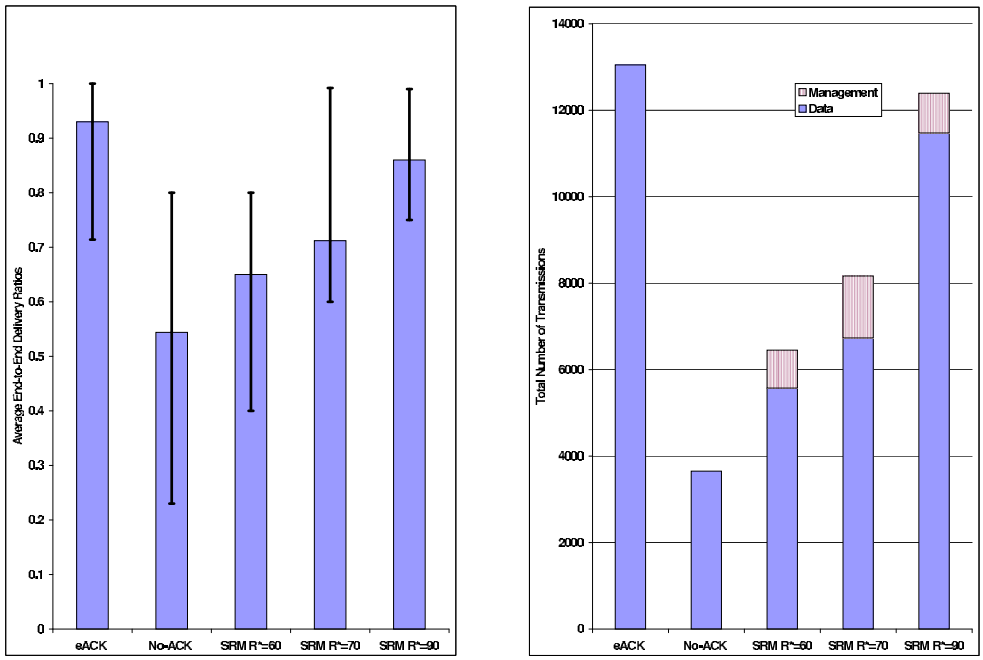


(b) Delivery Ratio, $R^* = 70\%$, The error bar shows the minimum, average, and maximum delivery ratio over a period of 100 seconds

Figure 4.6: The Link Quality and End-to-End Delivery Ratios.

are reasonably high, even under high link loss. The reason is that the high number of retransmissions can compensate the packet loss, thus increases the delivery ratio. It is observed that on average SRM achieves delivery ratios at about 71.2%. When the link loss increases, SRM decides to increase the retransmissions, and is constantly adjusting to maintain the delivery ratio at the rate of 70%.

We also ran the SRM with different delivery ratio requirements of $R^* = 60\%$ and $R^* = 90\%$. Figure 4.7(a) shows the minimum, average, and the maximum delivery ratios achieved for the entire period. It is observed that on average SRM meets the reliability requirement with the small error of $\pm 5\%$. Figure 4.7(b) presents the quantity of data and management traffic transmitted during the experiment with and without management. When $R = 60\%$, SRM reduces the number of transmissions by more than 50%. Although the number of transmissions using no-ACK is small, the delivery ratio of no-ACK is only 54%. When $R = 90\%$, the number of transmissions in SRM



(a) Delivery Ratio (b) Total Number of Transmissions

Figure 4.7: The End-to-End Delivery Ratios and Total Number of Transmissions.

is 13.4% lower than eACK. The total number of transmissions of SRM in these three cases, including both the data and management packets, is still less than for eACK. Thus, SRM can balance the energy consumption and data delivery and adapt to the network dynamics.

Although the SRM can provide adaptivity, it comes at a cost. As shown in Figure 4.7(b), the management traffic is from 7% to 18% of the total messages transmitted. The reason is that in these examples, the monitoring scheduler component is in a passive mode, where the management information is periodically collected. It implies that the management solution may be potentially expensive for the management task which requires a lot of management information. Therefore, our future work will investigate the reactive mode of the monitoring scheduler component, which dynamically selects how often the management information is collected based on network conditions for reducing management traffic.

4.7 Related Work

Cha et al. [77] proposed a policy-based network management approach for sensor nodes to organize and manage themselves. Their work is closest in spirit to our work and both are based on the policy based network management paradigm [17]. However, their work only provides the concepts for designing policy-based sensor network management and primarily focuses on designing an energy-efficient clustering algorithm for policy distribution. In contrast, we provide details of the management system for data reliability management. Further, we evaluated the system in realistic environment. There are many other management systems have been proposed for WSNs such as BOSS [78], Moteview [31], SNMS [32], etc, (see Section 2.2). However, these systems require network administrators to participate in network management tasks and thus, they do not provide self-control capability. SRM is complementary to these work which provides both manual and automatic management services. Kim et al. [40] proposed SenOS, a finite state machine based operating system for WSNs (see Section 2.2.3.2). However, the proposed network management protocol is limited to only SenOS operating system. Ruiz et al. [30] proposed MANNA, a management architecture for WSNs (see Section 2.2.3.1). MANNA provides generic concepts for designing sensor network management. Deb et al. [41] proposed STREAM, a distributed algorithm for sensor topology retrieval at multiple resolutions (see Section 2.2.3.3). The STREAM retrieves network state for multiple resolutions at different communication cost. By selecting a subset of nodes and merging their neighborhood lists, an approximate topology can be constructed. Hsin et al. [35] proposed the two-phase self-monitoring system for WSNs (see Section 2.2.2.2). Two phase self-monitoring system aims to detect malfunctioning nodes in the network. However, none of these work takes data reliability into account.

In the context of data reliability, there are many transport protocols in literature such as CODA [29], ESRT [25], PSFQ [22], RCRT [21], Flush [24], etc (see Section 2.1). However, none of the existing solutions takes the data reliability problem from the

management point of view. Each protocol is developed for a specific application. For example, ESRT [25] is designed for ensuring event reliability; PSFQ [22] is designed for network reprogramming; Flush [24] is designed for delivering a large bulk of data, etc. SRM, on the other hand, provide a framework which allows different reliable transport protocols to run the network.

4.8 Chapter Summary

In this chapter, we proposed, implemented, and evaluated SRM: a Sensor Reliability Management framework for WSNs. SRM is based on a hierarchical management architecture and on the policy-based network management paradigm. Although SRM is designed for data reliability management, it can be easily integrated with other management services as a part of a WSN self-management architecture. In addition, SRM also allows the network administrators to interact with the network by providing management policies. The provided examples show that the SRM can provide enough data reliability while reducing significantly energy consumption.

There are still open issues as far as SRM is concerned. The promising results obtained here motivate a further investigation on other components in SRM. First, to reduce management control traffic, the reactive mode of the monitoring scheduler component is worth to look at. The future work may investigate the learning of the daily trends in data reliability and come up with an adaptive monitoring schedule for reliability management. Another interesting question here is how the decision making module handles the conflicts among management rules. A linear programming approach may be useful to identify how the maximization of number of rules could be satisfied which maximizes the objective benefits.

Chapter 5

ERTP: an Energy-efficient and Reliable Transport Protocol for Wireless Sensor Networks

In this chapter, we discuss a distributed protocol for `ADJUST RETRANSMISSION RATE` action described in Chapter 4 for controlling the E2E delivery ratio. We propose ERTP, an **E**nergy-efficient and **R**eliable **T**ransport **P**rotocol for WSNs. ERTP is designed for data streaming applications, in which sensor readings are transmitted from one or more sensor sources to a base station. ERTP is an adaptive transport protocol based on statistical reliability that ensures the number of data packets delivered to the sink exceeds the defined threshold while reducing the energy consumption when compared to the absolute reliability.

ERTP comprises two components: a HBH reliability component, and a HBH retransmission timeout component. The first component ensures required E2E reliability by dynamically controlling the maximum number of retransmissions for each data packet in all intermediate nodes based on the channel quality. ERTP uses HBH iACK for loss recovery. The HBH iACK mechanism operates by the transmitter overhearing the packet being forwarded by the receiver to its next hop and considers this as an iACK. The transmitter will retransmit the packet if it has not received the iACK after a time-out interval. Determining how long the node should wait for an iACK is non-trivial and depends on the time it takes a packet to be forwarded by the downstream node. A premature Retransmission TimeOut (RTO) value for HBH iACK may increase sensor energy-consumption because transmitters will send duplicate packets. On the other hand, a large RTO value tends to increase transmission latency and thus reduces network

goodputs. In order to achieve energy efficiency, ERTP dynamically adjusts RTO value at each node by observing the channel quality. By combining the statistical reliability and the HBH iACK loss recovery, ERTP can offer sufficient reliability to the application users with the minimal energy expense. Our extensive simulations and experimental evaluations show that ERTP can reduce energy consumption by more than 45% when compared to the state-of-the-art protocol. Consequently, sensor nodes are more energy-efficient and the lifespan of the unattended WSN is increased.

5.1 Chapter Contributions

The primary contributions of the chapter are summarized as follows:

- We present an analysis of the trade-off between energy consumption and E2E reliability for ERTP, in which HBH iACK approach and duplicate detection are used at each sensor node. To balance energy consumption and reliability, ERTP dynamically controls the maximum number of retransmissions at each sensor node.

- We propose a distributed algorithm for RTO estimation in ERTP. Determining how long the node should wait for an iACK is non-trivial since iACK timeout depends on the time it takes a packet to be forwarded by the downstream node. The simulation results in Section 5.4 show that the proposed RTO algorithm is significantly more energy-efficient than other approaches. To the best of our knowledge, ours is the first work which investigates adaptive RTO estimation for the class of HBH iACK protocols in WSNs.

- We design, implement, and evaluate ERTP in TinyOS [59] for real-world sensor networks. Our extensive evaluations show that ERTP can reduce energy consumption by more than 45% when compared to current approaches. Consequently, sensor nodes are more energy-efficient and the lifespan of the unattended WSN is increased.

Table 5.1: Notation

Symbol	Meaning
α	Application layer E2E reliability requirement
β_k	HBH reliability requirement for flow k
$N(\beta_k, i)$	The maximum number of retransmissions for a packet of flow k at node i to be delivered successfully with β_k reliability
$X(\beta_k, i)$	The expected number of transmissions from node i to $i+1$ for a packet of flow k to be delivered successfully with β_k reliability
$Y(\beta_k, i)$	The expected total number of transmissions from node i to $i+1$ for the iACK of a packet of flow k received successfully by node i with β_k reliability
p_i	Link error rate between nodes i and $i+1$
q_i	Link error rate between nodes $i+1$ and i
E_k	The expected total number of transmissions for a packet of flow k received at the sink with α reliability
$\xi(k, i)$	The expected overhearing time for a packet of flow k from node i after sending the packet
$T(k, i)$	The RTO for a packet of flow k at node i

This work could not be completed without the supports of Dr. Zvi Rosberg and Dr. Ren Ping Liu for mathematical formulation.

5.2 Chapter Organization

5.3 ERTP: An Energy-efficient and Reliable Transport Protocol

In this section, we firstly provide an overview of ERTP that includes the requirements and our assumptions. We then discuss the details of the components of ERTP: the HBH Reliability Control, and the HBH Retransmission Timeout Control. Finally, we discuss other details of ERTP that include link quality estimation, duplicate packet detection, and a distributed algorithm for RTO updating.

Definition 3. *The application layer E2E reliability for each sensor node α ($0 < \alpha < 1$), is described by probability of a data packet to be delivered to the sink.*

Definition 4. *The HBH reliability requirement β_k for flow k ($0 < \beta_k < 1$), is described by probability of data packets of node k to be delivered from one node to its next-hop node along the routing path between the source k and sink.*

5.3.1 Overview of ERTP

ERTP is a transport protocol for data streaming applications in WSNs, in which sensor readings are transmitted from one or more sensors (sources) to a base station (or sink). Two requirements of ERTP are:

- **E2E Reliability**: Our primary goal is to achieve an application layer E2E reliability of all data transmitted by each sensor node to a sink.
- **Energy-Efficiency**: While E2E transmission latency is not a pressing concern in many WSN data streaming applications, energy-efficiency often is. For long-term unattended operation of the network, the transport protocol should minimize sensor energy consumption.

ERTP makes three assumptions about the link layer below and the application layer above:

- **Low Data Rate**: ERTP assumes that transmission rate is low such that network congestion is negligible. This is a reasonable assumption for most of deployed data streaming applications in practice [7] [8] [79] [80].
- **Low Cost Snooping**: A node is able to overhear packet transmission within its transmission range. Estimation through snooping comes at a cost, since a node needs to listen for packets that are not addressed to it (idle listening). We assume that a low power listening (LPL) mechanism [81] [82] [83] [84] is used in the underlying MAC layer. LPL MAC protocols are the main stream MAC protocols and have been widely used in many WSN operating systems such as TinyOS [59] and Contiki [85]. LPL MAC protocols operate at a low duty cycle in which sensor nodes periodically sleep, wake-up, listen to the channel, and then return to sleep instead of idle listening. As a result, the snooping cost is very low [19]. Therefore, the communication cost, i.e. the number of transmissions of data packets, is the dominant factor in sensor energy consumption [86].

Although LPL MAC protocol is used here, we believe that ERTP can work with other duty cycles MAC protocols such as TDMA MAC protocols [87] [88]. The performance of ERTP with different MAC-protocols will be investigated in our future work.

- **Low Transmission Contention**: Transmission collisions happen if at least two neighbouring nodes, which lie within the interference range of each other, transmit at the same time. However, for low data rate applications, transmission collisions are negligible because the probability that at least two neighbouring nodes transmit at the same time is small. For example, if there are N interfering neighbor nodes and M number of packets that can be transmitted in a period, the probability that two or more nodes transmit a packet simultaneously is

$$1 - 1.\frac{M-1}{M}\frac{M-2}{M}....\frac{M-N+1}{M} = 1 - \prod_{k=1}^{N-1}\frac{M-k}{M} \tag{5.1}$$

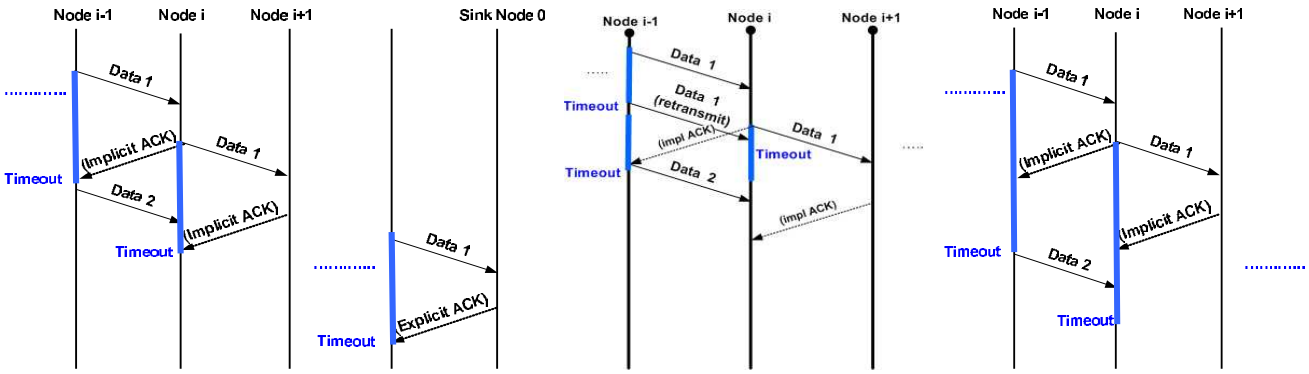
  For example, in our Fleck-3 [5] platform, the bandwidth is $W = 50$ kbps and the size of each data packet is $L = 40$ bytes. The transmission rate at each node is $D = 0.017$ packet per second (1 packet per minute). So, the number of packets that can be transmitted in the period $M = \frac{1}{D}\frac{W}{L} = 9192$ packets. If a node has $N = 20$ neighbouring nodes, for a medium density network, the probability that two or more nodes transmit a packet simultaneously is less than 0.02 (calculated by Equation 5.1).

ERTP consists of two components: ***HBH Reliability***, and ***HBH Retransmission Timeout***.

- ***The HBH Reliability Component*** ensures the required application layer E2E reliability by dynamically controlling the maximum number of retransmissions for each data packet in all intermediate nodes. Obviously, a sensor node can not allow a very large number of retransmissions because of packet freshness and fairness and energy concerns. In most transport protocols, a

pre-set number of retransmissions is used [24] [19] [21]. To achieve both E2E reliability and energy-efficiency, ERTP dynamically determines the maximum number of retransmissions at each node. An insufficient maximum number of retransmissions may cause packet to be lost as it travels to the sink, wasting energy and network resources, as well as degrading E2E reliability. Conversely, there will be energy-inefficiency when the maximum number of retransmissions is too high. To balance energy consumption and E2E reliability, the ***HBH Reliability Component*** dynamically determines and updates a near optimal maximum number of retransmissions for data packets at each node.

- ***The HBH Retransmission Timeout Component*** ensures application layer E2E reliability by dynamically adjusting the RTO at each node. ERTP employs HBH iACK scheme, which operates by the transmitter overhearing the packet being forwarded by the receiver to its next hop and considers this as an iACK. The transmitter will retransmit the packet if it has not received the iACK after a time-out interval. Determining how long the node should wait for an iACK is non-trivial [89] and depends on the time it takes a packet to be forwarded by the downstream node. Figure 5.1(a) shows the normal operation of the HBH iACK protocol. When node i forwards a packet of node $i-1$ to node $i+1$, node $i-1$ overhears this forwarding and considers it as an iACK. A "premature" RTO value for HBH iACK may increase sensor energy-consumption because transmitters will send duplicate packets. This is energy-inefficient since the packet has already been received (Figure 5.1(b)). On the other hand, a large RTO value tends to increase transmission latency and thus reduces network throughputs (Figure 5.1(c)). Therefore, in order to achieve energy efficiency, the ***HBH Retransmission Timeout Component*** of ERTP is responsible for adjusting the RTO dynamically. Obviously, when a packet reaches the sink, there will be no further forwarding. Therefore, the sink node needs to send an



(a) Correct *RTO* (b) Premature *RTO* (c) Large *RTO*

Figure 5.1: HBH iACK operation.

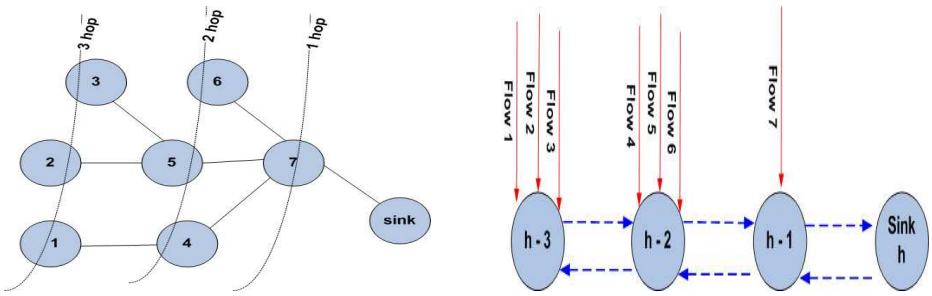
eACK. Since the eACK is sent immediately by the receiver, the eACK timeout is primarily based on the HBH Round Trip Time. Each node maintains a duplicate packet detection list to prevent duplicate packets being propagated over the network.

The remainder of this section describes each component in detail. Let us denote $0 \leq \alpha \leq 1$ as the desired application layer E2E reliability. We first present an idealized model with simplifying assumptions. We then lift these assumptions as we present how ERTP dynamically estimates the maximum number of retransmissions and RTO at each node.

5.3.2 HBH Reliability Component

5.3.2.1 Network Model

We model the network as a graph $G = (V, E)$, where V is a set of nodes and E is a set of edge links. Each sensor node periodically transmits its sensed packet to the sink at the rate of D packets per second. Let W (bits per second) and L (bits) denote the network bandwidth and the size of a packet, respectively. The packets are served by the sensor nodes on first come first serve basis. We assume that sensor nodes are aware of their next-hop neighbors along the routing path to the sink. The network consists



(a) Network Topology (b) The Corresponding Data Flows.

Figure 5.2: An example of Network Topology.

of a set of data flows $k \subset V$ ($k \neq$ the sink node), where k represents the ID of the sensor node from which the flow originated. Figure 5.2(a)-5.2(b) shows an example of a network topology. The data traffic of the network in Figure 5.2(a) can be represented as the graph of 7 data-flows as shown in Figure 5.2(b).

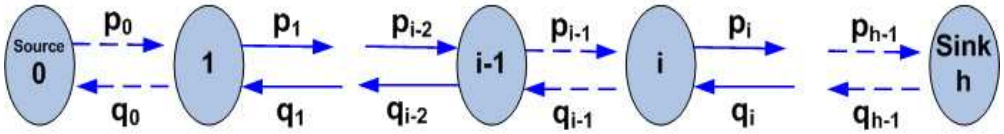


Figure 5.3: Single Data Flow.

5.3.2.2 Maximum Number of Retransmissions

Consider a packet transmission over the link l between node i and node $i+1$. The wireless link quality between node i and node $i+1$, denoted by the Packet Reception Rate (PRR), is described by the probability of a packet from node i being successfully received at node $i+1$. Under the log normal shadowing power model, the received signal power P_r is given by (dBm scale) [90]:

$$P_r = P_t - 10 * n * log(d_r/d_i) + \Psi_i \tag{5.2}$$

where n is the the path loss exponent, d_r is the reference distance and d_i is the transmitter-receiver distance. Ψ_i is a Gaussian random variable with zero mean and

standard deviation σ_{ψ_i}. Therefore, the signal-to-noise ratio (SNR), on the logarithmic scale, is calculated as

$$SNR(dB) = |P_r|_{dBm} - |N|_{dBm} \tag{5.3}$$

where $|P_r|$ is the magnitude of the signal power of a received packet, $|N|$ is the resultant magnitude of any environmental noise or disruption not caused by the network being implemented.

Moreover, PRR is a function of SNR [91]. Therefore, PRR is influenced by the deterministic components such as the distance d_i and transmission power P_t, and the non-deterministic components such as path loss exponent n, and noise Ψ_i.

Given a static deployed network, the distance d_i and transmission power P_t are fixed. With a strong transmission power P_t and a short distance d_i, the non-deterministic components of PRR are insignificant. However, in realistic WSN environments, sensor nodes typically transmit with low transmission power and are deployed at reasonable large distances, the non-deterministic components of PRR become significant and PRR varies over time [92] [93] [94].

Further, let $1 - p_i$ and $1 - q_i$ denote the expected value of PRR for a packet transmission from node i to node $i + 1$ and from node $i + 1$ to node i, respectively. Consider a data flow which involves $h + 1$ nodes, where node h is the sink and node 0 is the source, as depicted in Figure 5.3. Node 0 sends packets to the sink through $\{1, 2, ..., h-1, h\}$. For notational brevity, let $\bar{p}$ and $\bar{q}$ denote $1-p$ and $1-q$, respectively.

To achieve application layer E2E reliability α, the required maximum number of retransmissions N_0^i at node i for a packet of flow 0 is [75]:

$$N_0^i = \frac{log(1 - \alpha^{1/h})}{log(p_i)} \tag{5.4}$$

where $\alpha^{1/h}$ is the HBH reliability requirement for each node. For multi-flow WSNs,

each sensor node could be a source node or a relay node. Similarly, we have,

$$N(\beta_k, i) = \frac{log(1-\beta_k)}{log(p_i)} \tag{5.5}$$

where, $\beta_k = \alpha^{\frac{1}{h-k}}$ is the HBH reliability requirement for flow k, and $N(\beta_k, i)$ is the maximum number of retransmissions for a single packet of flow k at node i.

Assume that node i transmits a packet of flow k to node $i+1$. Let us denote $X(\beta_k, i)$ as the expected number of transmissions made by node i for a packet of flow k so that node $i+1$ receives the packet successfully. This event is a truncated geometric distribution with the successful probability of $\bar{p}_i$ taken from the set $\{1, 2, ..., N(\beta_k, i)\}$. Thus, its expected value $X(\beta_k, i)$ is:

$$X(\beta_k, i) = \sum_{j=1}^{N(\beta_k,i)} j(\bar{p}_i)(1-\bar{p}_i)^{j-1} + N(\beta_k, i)(1-\bar{p}_i)^{N(\beta_k,i)-1} \tag{5.6}$$

By simplifying (5.6) and re-arrange the terms, we have,

$$X(\beta_k, i) = \frac{1 - p_i^{N(\beta_k,i)}}{\bar{p}_i} \tag{5.7}$$

However, it is possible that node $i+1$ receives the packet from i successfully, but the forwarding by $i+1$ is not overheard by node i (the iACK is lost). Let us denote $Y(\beta_k, i)$ as the expected number of transmissions for a single packet of flow k made by node i so that node i overhears the iACK successfully. Therefore, $Y(\beta_k, i)$ is greater than or equal to $X(\beta_k, i)$ (recall that $X(\beta_k, i)$ is the expected number of transmissions made by node i for a packet of flow k so that node $i+1$ receives the packet successfully, and an iACK will be "sent" by node $i+1$ after node $i+1$ receives the packet successfully).

Proposition 1. *For ERTP, the expected number of transmissions $Y(\beta_k, i)$ for a packet of flow k to be delivered successfully from node i to node $i+1$ is*

$$Y(\beta_k, i) = \begin{cases} \frac{1-(1-\bar{p}_i\bar{q}_i)^{N(\beta_k,i)}}{\bar{p}_i\bar{q}_i} & , i = h-1 \\ X(\beta_k, i) + q_i^{Y(\beta_k,i+1)}(N(\beta_k, i) - X(\beta_k, i)) & , k \le i \le h-2 \end{cases} \tag{5.8}$$

Proof

- For $i = h-1$, eACK is used. Node $h-1$ transmits a packet of flow k until both the packet received at the sink h and the ACK from the sink h is received successfully by $h-1$. The probability of this event is $\bar{p}_{h-1}\bar{q}_{h-1}$. This is a truncated geometric distribution with the successful probability of $\bar{p}_{h-1}\bar{q}_{h-1}$ taken from the set $\{1, 2, ..., N(\beta_k, h-1)\}$. Thus, its expected value $Y(\beta_k, h-1)$ is given by the first term of Equation (5.8).

- For $k \leq i \leq h-1$, with the optimal iACK timeouts, the transmitter node, i, transmits a packet, either from itself (when $i = k$) or forwarded from node $i-1$ (when $i \geq k$), until the packet is successfully received by node $i+1$. The probability of success of this event is $\bar{p}_i$ and the expected number of transmissions for this event is given by $X(\beta_k, i)$ (Equation (5.7)). After this event occurs, node $i+1$ will forward the packet to node $i+2$ with the expected $Y(\beta_k, i+1)$ number of transmissions. Note that each sensor node has duplicate detection to prevent redundant packets from being propagated over the network (see Section 5.3.4). Therefore, if node i overhears the forwarding from node $i+1$ to node $i+2$ in one of the $Y(\beta_k, i+1)$ forwarding times, it will transmit the next packet. Otherwise, it will retransmit an additional $(N(\beta_k, i) - X(\beta_k, i))$ times. This event occurs with a probability of $q_i^{Y(\beta_k,i)}$. Therefore, the expected number of transmissions is given by the second term of Equation (5.8).

Proposition 2. *The expected total number of transmissions E_k for a single packet of flow k to be delivered successfully to the sink is*

$$E_k = Y(\beta_k, h-1)(1 + \bar{p}_{h-1}) + \sum_{i=k}^{h-2} (Y(\beta_k, i)) \tag{5.9}$$

Proof The expected total number of transmissions for a packet of flow k is the summation of expected number of transmissions $Y(\beta_k, i)$ at each intermediate node i

$(k \leq i \leq h-1)$ along the routing path from the source k to the sink h. The expected number of transmissions for a packet to be transmitted successfully from node $h-1$ to h is given by $Y(\beta_k, h-1)$ derived from Proposition 1, regardless of the ACK outcome. In addition, the sink h needs to send eACKs to node $h-1$, so the expected number of transmissions of ACKs in the backward route from the sink h to node $h-1$ is reduced by a factor of $\bar{p}_{h-1}$. Therefore, the expected total number of transmissions is given by Equation (5.9).

5.3.3 HBH Retransmission Timeout Component

First, we need to estimate the time required to transmit a packet from node i to node $i+1$. With the channel bandwidth of W and for low data rate applications, the packet collision can be ignored. Thus, the average transmission time for a packet of L bits can be approximated by

$$\overline{T_{tx}} = \frac{L}{W} \tag{5.10}$$

To estimate the RTO value in node $i-1$, assume that node $i-1$ forwards a packet of flow k to node i. Let us denote $\xi(k,i)$ as the expected "overhearing" time in node i. Once node $i-1$ sends a packet of flow k, $\xi(k,i)$ represents the expected time in which node $i-1$ is expected to "overhear" the forwarded packet.

Proposition 3. *For ERTP, the* ***RTO*** $T(k,i)$ *and the* ***expected "overhearing" time*** $\xi(k,i)$ *for a single packet of flow* k *(*$0 \leq k < h-1$*) at node* i *is given by*

$$T(k,i) = \frac{L}{W} + \xi(k,i+1), k \leq i < h-1 \tag{5.11}$$

$$\xi(k,i) = \begin{cases} \frac{1-(1-\bar{p}_i\bar{q}_i)^{N(\beta_k,i)}}{\bar{p}_i\bar{q}_i}\frac{L}{W} & , i = h-1 \\ \frac{1-(1-\bar{q}_{i-1})^{N(\beta_k,i)}}{\bar{q}_{i-1}}T(k,i) & , k \leq i < h-1 \end{cases} \tag{5.12}$$

where, $T(k,i)$ is the ***RTO*** for a packet of flow k at node i.

Proof

- For i = h - 1, since eACK is used from node $h-1$ to the sink h, node $h-1$ is expected to send $Y(\beta_k, h-1)$ transmissions for a packet of flow k as given by Proposition 1. Therefore, the expected overhearing time $\xi(\beta_k, h-1)$ is,

$$\xi(\beta_k, h-1) = Y(\beta_k, i)\overline{T}_{tx} \tag{5.13}$$

  Substituting (5.8) and (5.10) into (5.13), we can obtain the first term of the Equation (5.12). Note that node $h-1$ does not need overhearing time $\xi(\beta_k, h-1)$ since the eACK scheme is used between node $h-1$ and h, but node $h-2$ does need overhearing time $\xi(\beta_k, h-1)$ to estimate its RTO $T(k, h-2)$.

- For $k \leq i < h-1$, if node i receives packet of flow k from node $i-1$ successfully, it will forward the packet to node $i+1$, but no more than $N(\beta_k, i)$ times. Node $i-1$ is expected to overhear the forwarding by node i with the successful probability of $\bar{q}_{i-1}$. This event is the truncated geometric distribution with the successful probability of $\bar{q}_{i-1}$ taken from the set $\{1, 2, ..., N(\beta_k, i)\}$. Thus, its expected value is:

$$\frac{1 - q_{i-1}^{N(\beta_k, i)}}{\bar{q}_{i-1}} \tag{5.14}$$

  Therefore, the expected overhearing time $\xi(\beta_k, i)$ for a packet of flow k is equal to the number of transmissions from node i to node $i+1$, that node $i-1$ can overhear multiplied by the RTO setting at node i ($T(k, i)$). Thus, we can obtain the second term of (5.12).

  The RTO $T(k, i)$ depends on the HBH transmission time from node i to $i+1$ (denoted by $\overline{T_{tx}}$), and the expected "overhearing" time $\xi(\beta_k, i+1)$ for the packet being served at node $i+1$ (denoted by $\xi(\beta_k, i+1)$). Thus, we can obtain (5.11).

5.3.4 Other Details

Equation (5.5) and Proposition 3 provide the estimation for the maximum number of retransmissions and RTO values in each node. We now describe how ERTP

dynamically estimates the maximum number of retransmissions and RTO values in real environments.

5.3.4.1 Link Quality Estimation

Link quality indicates the packet reception rate of the link and is an important parameter in ERTP. One of the main differences between WSNs and wired networks is that the link quality in WSNs may vary greatly with time as a consequence of interference, propagation dynamics, and power depletion, etc.

Link quality can be obtained from the Link Quality Indicator (LQI) defined by IEEE standard 802.15.4 which is readily used on MicaZ and Telos sensor network devices [95] [96]. For those platforms that do not support LQI, link quality can be estimated by observing packet success and loss events. ERTP uses an exponentially weighted moving average (EWMA) for link quality estimation. The EWMA estimator is simple and memory efficient, requiring a constant amount of storage for prior quality estimates [19]. EWMA uses a linear combination of prior estimates, weighted exponentially. The forwarding probability $\bar{p}$ over a link l is calculated using the ratio of the number of data packets received to the total number of data packets transmitted over the link l at the time t. The link quality on the reverse link l, i.e., $\bar{q}$ is calculated as $\bar{p}(\bar{l})$ by the node at the other end of link l. The nodes at both ends of link l exchange the information to obtain the bi-directional link quality of link l.

5.3.4.2 Duplicate Packet Detection and Avoidance

Due to the the existence of asymmetric links, it is possible that the forwarding packet is successfully received but the iACK is lost. For example, in Figure 5.4, node $i+1$ may receive the packet from node i successfully, while node $i-1$ does not overhear this forwarding. In this case, node $i-1$ will retransmit the packet, even though node i has already received it. Following the same principle, node i will also have to repeat the transmission to node $i+1$. Therefore, once an iACK is lost, duplicate packets would be

generated and propagated over the network to the sink.

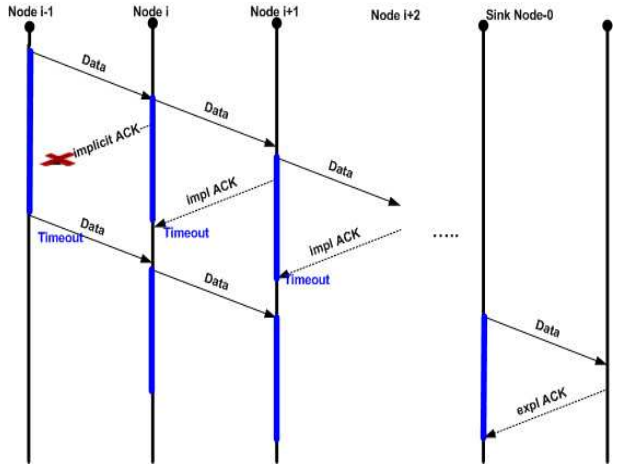


Figure 5.4: The Impact of Loss of Implicit Acknowledgment.

We handle this case by a simple duplicate detection mechanism. Each node maintains a list of the M-most recent packets it has received. If a node receives a duplicate packet, it will drop the packet to reduce unnecessary forwarding. In our implementation, we select $M = 5$.

5.3.4.3 Dynamic Maximum Number of Retransmission Control and Dynamic RTO Control

As the link quality varies with time, sensor nodes need to control the number of retransmissions and RTO values dynamically.

Once the link quality is updated, the HBH reliability component estimates the maximum number of retransmissions using Equation (5.5). These estimates are noisy so we apply a smoothing filter. Each node maintains the most-recent set of m values and the weight w_m is given to each value in the history. Intuitively, the choice should give greater weight to the recent estimations. The smoothed estimate $\bar{x}$ is calculated from the raw estimates x_i by:

$$\bar{x} = \frac{\sum_{i=0}^{4} w_i x_i}{\sum_{i=0}^{4} w_i} \tag{5.15}$$

where x is the new estimation of either the maximum number of retransmissions or RTO. We used $m = 5$ and $w = [1, 1, 0.8, 0.6, 0.4]$ in our protocol.

While the maximum number of retransmissions can be calculated locally, the RTO estimation in a sensor node depends on the RTO of its parent node (Proposition 3). This information could be sent explicitly, but such a mechanism would incur additional communications and therefore energy. We use a distributed algorithm to update the RTO as follows. When a new link quality is estimated, node $h-1$ calculates its new timeout T_{h-1} locally by Equation (5.12). Node $h-1$ does not use the T_{h-1} value, but node $h-2$ does need T_{h-1} to estimate its timeout T_{h-2}. To minimize overheads, node $h-2$ snoops T_{h-1}, which is embedded in the forwarded data packet of node $h-2$ from node $h-1$, and estimates its timeout T_{h-2} by Equation (5.11). Similarly, node $h-3$ snoops the new timeout T_{h-2} and estimates its timeout T_{h-3}. Eventually, all the nodes in the network will update their own timeout values.

Since ERTP uses snooping for all the control and update information, it does not explicitly increase overhead. An extra 2-byte field is used for RTO information in the ERTP header.

5.4 Simulation

In this section, we evaluate the performance of the ERTP through extensive simulations.

5.4.1 Summary

We conducted the simulations for a 200-node network. The nodes are uniform-randomly deployed in an area of 180m×180m, as shown in Figure 5.5. The simulations were run in the discrete-event network simulator ns-2 [63] using a modified-version (to enable iACK) of the Carrier Sense Multiple Access (CSMA) MAC protocol, and DSDV as the routing layer protocol. We selected node 0 as the sink and the other nodes gener-

ate packets of 40 bytes at the rate of 1 packet per minute, which is similar to the traffic pattern in the Burdekin water monitoring application [7]. The simulation parameters are summarized in Table 5.2 based on the Fleck-3 platform [5]. The simulation time is 2 hours, which is sufficient to evaluate the protocol trends. The application layer E2E reliability requirement is $\alpha = 0.95$.

Asymmetric links and link variation over time in sensor networks have been observed and reported by many in the research community [19] [97] including experiments described in Section 5.5. To study the impact of network dynamics on the performance of ERTP, we simulated the link loss rates as follows. For each link, we repeatedly changed the link loss rates every 10 minutes. The link loss rate is randomly assigned to a new value in a pre-define link loss range. The intent was to create asymmetric link characteristics and link variation over time in the network. Specifically, we simulated the following cases:

- Case 1 - Low Link Loss Rate: Each link l in the network dropped packets with p (upstream) and q (downstream) probabilities, where p and q are random values between 5% and 25%, ($5\% \leq p, q \leq 25\%$). After 10 minutes, p and q are randomly assigned to the new values p_1 and q_1, where p_1 and q_1 are lying in the same link loss range. The process is repeated during the entire simulation.

- Case 2 - High Link Loss Rate: Similar to the case 1, but the link loss rate range is between 35% and 55%.

In addition, we also compared the performance of ERTP to a protocol which uses the explicit ACK scheme for different data transmission rates (Section 5.4.3.6). To ensure the reliability requirement, the explicit ACK is modified to dynamically control the maximum number of retransmission (for brevity, we call it as eACK).

Each data point in the simulation figures is the average of 20 simulations. The average values are plotted along with their 95% confidence intervals.

Table 5.2: Simulation setup

Parameter	Value
Bandwidth W	50 Kbps
Packet Size L	40 bytes
Statistical Reliability Requirement α	0.95
Sending Rate	1 packet per minute
Radio Transmit Current	31.8 mA
Radio Receive Current	13.4 mA
Supply Voltage	3.3V
Simulation Time	2 hours

5.4.2 Goals, Metrics, and Methodology

In order to evaluate the performance of the HBH reliability component, each node adjusts the maximum number of transmissions dynamically based on Equation (5.5). We compare the actual achieved E2E delivery ratio to the delivery requirement ($\alpha = 0.95$).

In order to evaluate the performance of the HBH RTO component in ERTP introduced in Section 5.3.3, we compare it with the following algorithms:

- **Theoretical Result:** Proposition 2 shows the expected energy consumption (E_k) required for a packet of flow k ($0 \leq k < h-1$) to be delivered successfully to the sink. The theoretical result provides the lower bound to compare the performance of different RTO algorithms.

- **Fixed Round-Trip Time (RTT):** The RTO value is assigned to a fixed value, i.e. a multiple of RTT. We consider three cases: short timeout such as $RTO = 1{*}RTT$ and $RTO = 2{*}RTT$, medium timeout such as $RTO = 10{*}RTT$ and $RTO = 20 * RTT$, and long timeout such as $RTO = 50 * RTT$ and $RTO = 100 * RTT$. As the obtained results are similar, we only present the results of three cases: $RTO = 1 * RTT$, $RTO = 10 * RTT$, and $RTO = 100 * RTT$.

- **Jacobson's Algorithm:** Jacobson's Algorithm estimates a future RTT by linearly filtering previous measured $RTTs$, and the RTO value is obtained by

adding a scaled mean absolute deviation to the estimated future RTT [98]. The RTO values obtained by Jacobson's algorithm, similar to those obtained by ERTP, change continually as channel conditions vary. Specifically, the estimated RTT g_u for packet u is calculated by

$$g_u = (1 - a) * g_{u-1} + a * h_{u-1} \tag{5.16}$$

where h_{u-1} is the actual RTT, g_{u-1} is the estimated RTT for packet $u-1$, and a is a constant ($0 < a < 1$). The mean absolute deviation of the estimated RTT v_u is then calculated by

$$v_u = (1 - b) * v_{u-1} + b * |g_{u-1} - h_{u-1}| \tag{5.17}$$

where v_{u-1} is the mean absolute deviation for the packet $u-1$, and b is a constant ($0 < b < 1$). The RTO T_u for the packet u is

$$T_u = g_u + 4 * v_u \tag{5.18}$$

In an IP network, $a = 1/8$, and $b = 1/4$ are used [98]. To study the performance of Jacobson's algorithm, we simulated five different cases of (a, b): $(1/8, 1/4)$, $(1/8, 3/4)$, $(1/8, 19/20)$, $(3/4, 1/4)$, and $(3/4, 3/4)$.

We use the following metrics:

- **Reliability (Delivery Ratio):** This metric characterizes the E2E application layer delivery ratio achieved. The application layer reliability requirement is $\alpha = 0.95$.

- **Energy consumption:** This metric characterizes the average energy required for a packet to be delivered to the sink successfully. Ideally, the energy consumption should be as small as possible. Based on the Table 5.2, we can calculate the communication cost, which is 0.477 mJ per packet. To compare sensor energy consumption, we measure the **normalized energy consumption** (R_E) as a ratio of actual energy consumption ($E^{measure}$) and a lower

bound of energy consumption (E^{theory}) achieved from Proposition 2. Namely, $R_E = \frac{E^{measure} - E^{theory}}{E^{theory}}$. Thus, the lower R_E is, the more energy-efficient the RTO estimator is.

- **Average Packet Delay:** This metric characterizes the average latency for a data packet to travel from its source to the sink. Ideally, this metric should be as small as possible to indicate timely data transfer.

5.4.3 Results

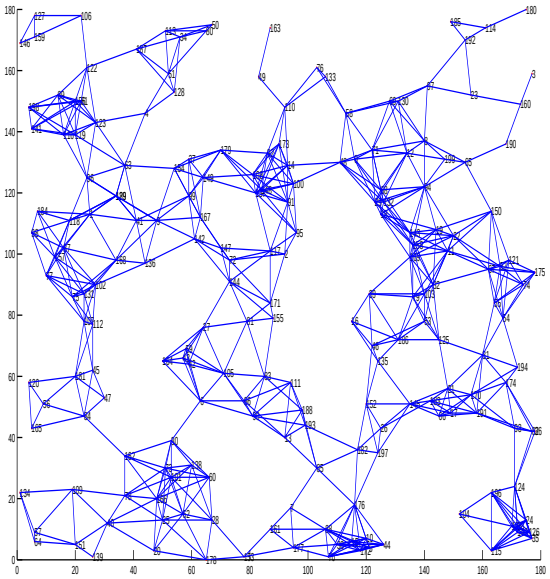
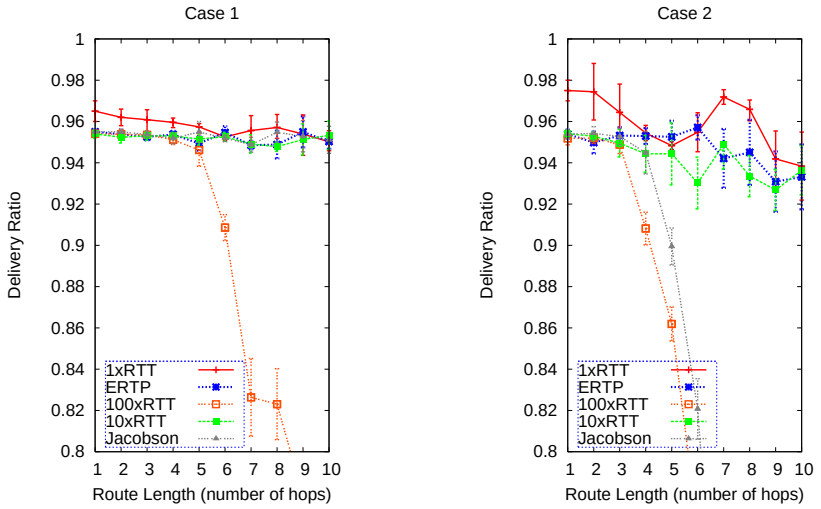


Figure 5.5: Simulation Network Topology.

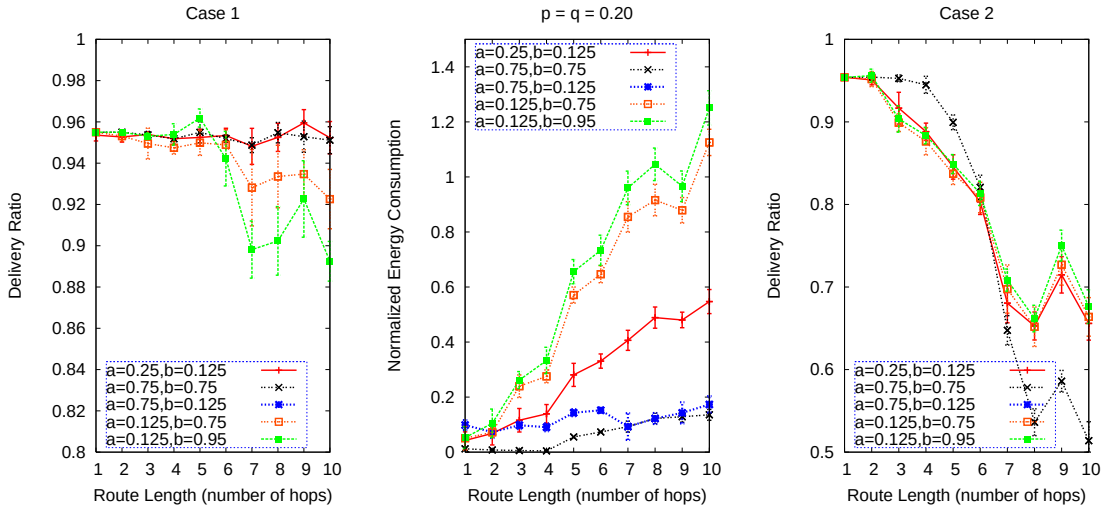
5.4.3.1 Reliability

Figures 5.6(a)-5.6(b) show the simulated E2E delivery ratios. Apart from Jacobson algorithm and $RTO = 100 * RTT$, the other algorithms can achieve 95% E2E reliability with a small error range of ± 0.03. The results validate the Equation (5.5) as well as the HBH reliability component. The obtained delivery ratios are slightly lower



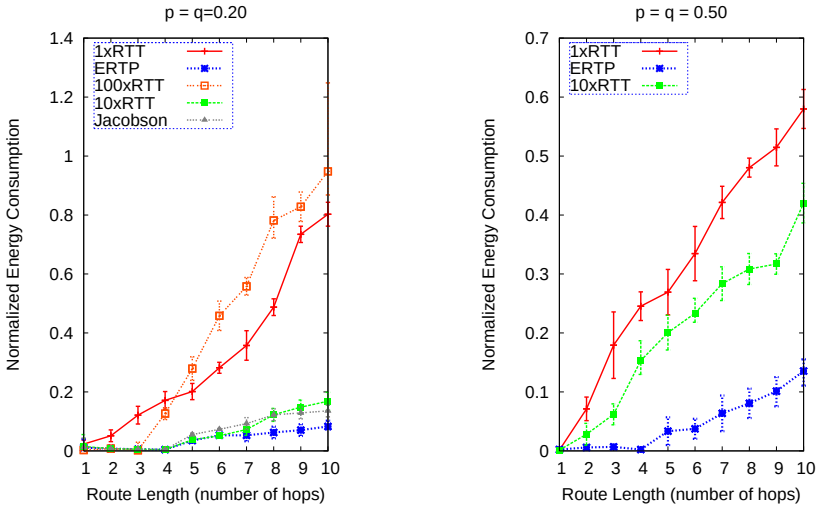
(a) Case 1 - Low Link Loss Rates (b) Case 2 - High Link Loss Rates

Figure 5.6: Average Delivery Ratio.



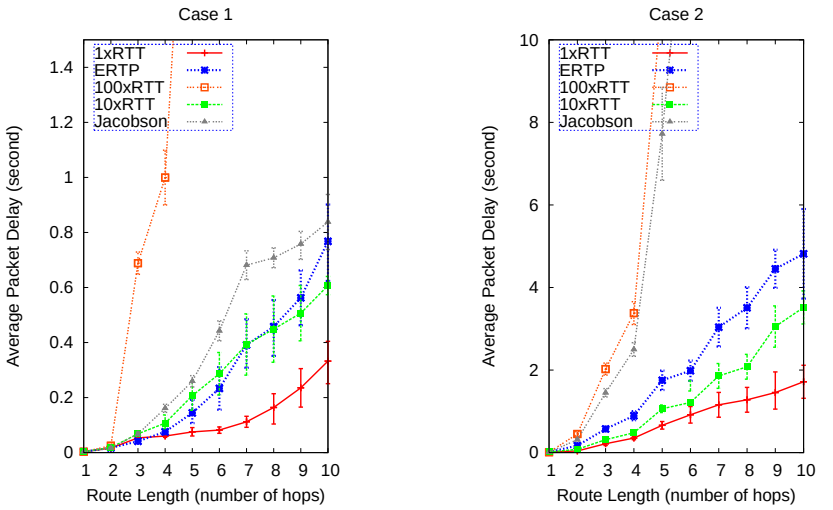
(a) Case 1 - Low Link Loss Rates (b) Case 1 - Low Link Loss Rates (c) Case 2 - High Link Loss Rates

Figure 5.7: The Performance of Jacobson's Algorithm with Different Values of Parameters.



(a) Case 1 - Low Link Loss Rates (b) Case 2 - High Link Loss Rates

Figure 5.8: Normalized Energy Consumption.



(a) Case 1 - Low Link Loss Rates (b) Case 2 - High Link Loss Rates

Figure 5.9: Average Packet Delay.

than the reliability requirement because many packets were dropped during the routing discovery phase. These packets dropped by routing are not considered in our theoretical model. However, the difference is very small (3%) and on average, ERTP can achieve $92 - 96.5\%$ E2E reliability.

Moreover, we observe that the Jacobson algorithm and $RTO = 100 * RTT$ do not satisfy the reliability requirement when the link loss rates are high (case 2) for the reason explained next. Therefore, we do not compare the energy consumption of Jacobson's algorithm and $RTO = 100 * RTT$ to the other algorithms when their reliability is much lower than the design criteria of 95%, i.e. the case 2.

5.4.3.2 Jacobson's Algorithm

Figure 5.7 shows the delivery ratios and the normalized energy consumption of Jacobson's algorithm for different sets of parameters a and b. We observe that the performance of Jacobson's algorithm varies significantly with different parameters. For the low loss rate (case 1), the normalized energy consumption with $a = 0.75, b = 0.75$ is 6.5 times less than the one with $a = 0.125, b = 0.95$ for 10-hop nodes. For the high loss rate (case 2), Jacobson's algorithm no longer satisfies the reliability requirement for the $\geq$ 2-hop nodes because Jacobson's algorithm chooses a very long value of RTO when the link loss rates are high. Since Jacobson's algorithm with $a = 0.75, b = 0.75$ provides the best normalized energy consumption in our simulation, we compare this case to the other algorithms.

5.4.3.3 Energy Consumption

Figures 5.8(a)-5.8(b) show the normalized energy consumption versus the route length for two cases. Not surprisingly, the long iACK timeout is more energy-efficient than the short one. For 10-hop nodes, when compared to $10 * RTT$, the normalized energy consumption in $1 * RTT$ is 60.62% higher for the case 1 and 18.32% higher for the case 2, respectively. This is due to the RTO value of $10 * RTT$ reducing unnecessary

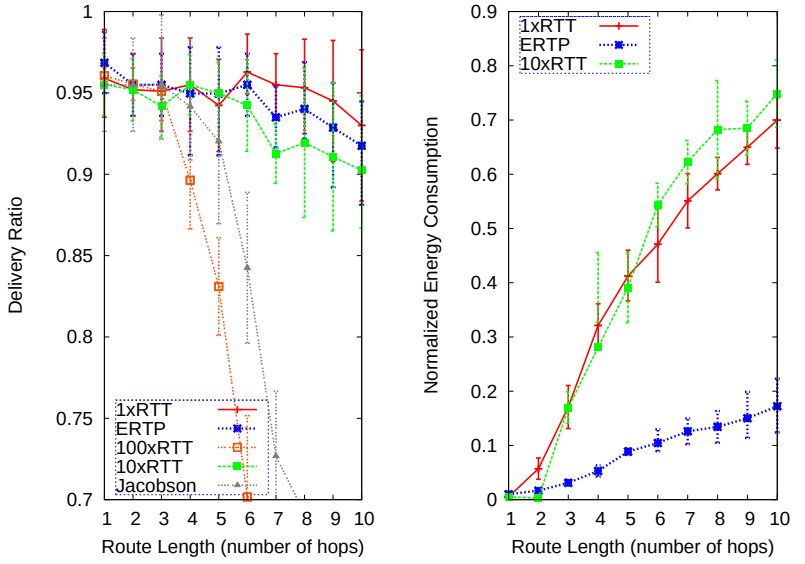
retransmissions.

Counter-intuitively, the very long RTO scheme, i.e. $100*RTT$, is not the most energy-efficient. Figure 5.8(a) shows that the normalized energy consumption of $100 * RTT$ is significantly more than the other approaches. Note that a sensor node will not forward the next packet in its routing queue unless either the current packet is successfully forwarded to the next hop or the number of retransmissions for the current packet exceeds the threshold (a characteristic of the Stop-and-Wait iACK protocol). However, retransmissions and very long RTO setting at a node cause very long serving time for the current packet (significantly longer than the RTO value) when the link loss rates are high. As a result, the forwarding rate of the next packet in the queue is significantly low and the transmitter may not overhear the forwarding packet from the receiver when its timer expiries, even though the packet is successfully received at the receiver. Consequently, the transmitter times out well before its receiver forwards the packet, thus it will retransmit the packet. This results suggest that the adaptive RTO algorithm for iACK protocol is crucial.

We observe that the normalized energy consumption with Jacobson's algorithm is about 10% higher than the theoretical values for $\geq$ 8-hop nodes for the low loss rate (case 1). However, the achieved reliability in Jacobson's algorithm is not satisfactory when the loss rate is high (case 2). The results suggest that the linear filter for RTO value it not energy-efficient in lossy wireless multihop networks.

Finally, Figures 5.8(a)-5.8(b) show that ERTP outperforms other approaches[1] . The normalized energy consumption in ERTP is up to 32.4% less than that of $10 * RTT$ for the high link loss rate (case 2). The results validate the analysis in Section 5.3.

[1] Note that $RTO = 10*RTT$ is a static approach which is not adaptive to changes in the environment such as different topology or traffic patterns. Therefore, it is not always the "best" heuristic solution (as seen in Figure 5.10(b))



(a) Delivery Ratio (b) Normalized Energy Consumption

Figure 5.10: Performance with Asymmetric Links.

5.4.3.4 Average Packet Delay

As shown in Figure 5.9(a) and Figure 5.9(b), a long RTO value has a packet delay penalty. The average packet delay in $RTO = 10 * RTT$ is significantly higher than that of $RTO = 1 * RTT$. For the high link loss rate (case 2), the average packet delay in $RTO = 10 * RTT$ is twice of the one in $1 * RTT$ (3.5 seconds compared with 1.7 seconds, for those nodes that are 10 hops from the sink). Jacobson's algorithm and $RTO = 100 * RTT$ incur significantly high average packet delay because of the long RTO. In addition, we also observe that for the low loss rate (case 1), the average packet delay in ERTP is as low as that in $RTO = 1 * RTT$ for $\leq$ 4-hop nodes and much higher than that in $RTO = 1 * RTT$ from 5-hop nodes. The reason is that to avoid early timeout, ERTP decides to wait long enough for overhearing packet forwarding.

5.4.3.5 The Performance with High Asymmetric Links

In this section, we study the impact of high asymmetric links on the performance of ERTP and other RTO estimation approaches. The link loss model is simulated as

follows. We set upstream link quality $35\% \leq p \leq 55\%$ and downstream link quality $50\% \leq q \leq 70\%$. After 10 minutes, p and q are randomly assigned to the new values p_1 and q_1, where p_1 and q_1 are lying in the same link loss ranges. The process is repeated during the entire simulation. The intent was to create very high asymmetric characteristics (high q) for all the links in the network.

Figure 5.10(a) shows the delivery ratios achieved by different approaches. Similar to the previous cases, the delivery ratios of Jacobson's algorithm and $100 * RTT$ are lower than the requirement for $\geq$ 3-hop nodes. Figure 5.10(b) shows that $10 * RTT$ **is no longer as energy-efficient as** $1 * RTT$ for $\geq$ 5 hop-away nodes when the links are asymmetric, which validates the necessity for the adaptive RTO component in ERTP. We observe that ERTP can achieve from $91\% - 97\%$ reliability. Compared to the results of the best heuristic approach, ERTP can reduce energy consumption by more than 50%.

5.4.3.6 The Performance with High Data Transmission Rates

In order to study the impact of high data transmission rates on the ERTP, we simulated the ERTP and modified eACK (described in Section 5.4.1) with different data transmission rates. We considered a 50-node network and a 200-node network in which nodes are uniform-randomly deployed as shown in Figure 5.11 and Figure 5.5. The reliability requirement is $\alpha = 0.95$. For the 50-node network, we simulated five different cases of data transmission rates: 2 packets every second, 1 packet every 1 second, 1 packet every 3 seconds, 1 packet every 5 seconds, and 1 packet every 7 seconds. For the 200-node network, we simulated seven cases of data transmission rates: 1 packet every 5 seconds, 1 packet every 10 seconds, 1 packet every 20 seconds, 1 packet every 30 seconds, 1 packet every 40 seconds, 1 packet every 60 seconds, and 1 packet every 70 seconds.

Figure 5.12(a) and Figure 5.12(b) show the average delivery ratios of eACK and ERTP with different link loss rates. First, we observe that both eACK and ERTP can

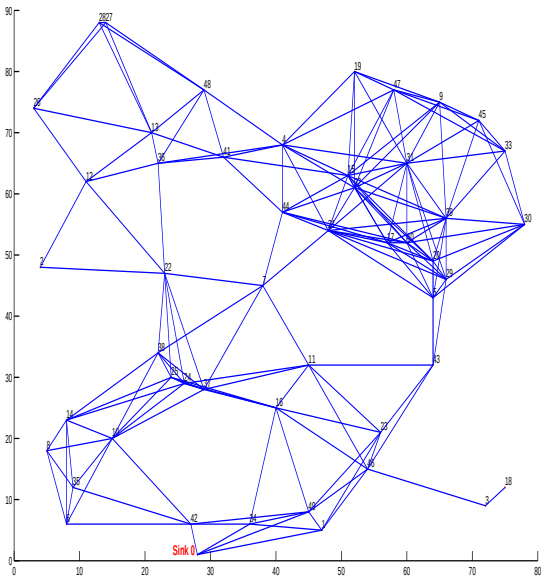
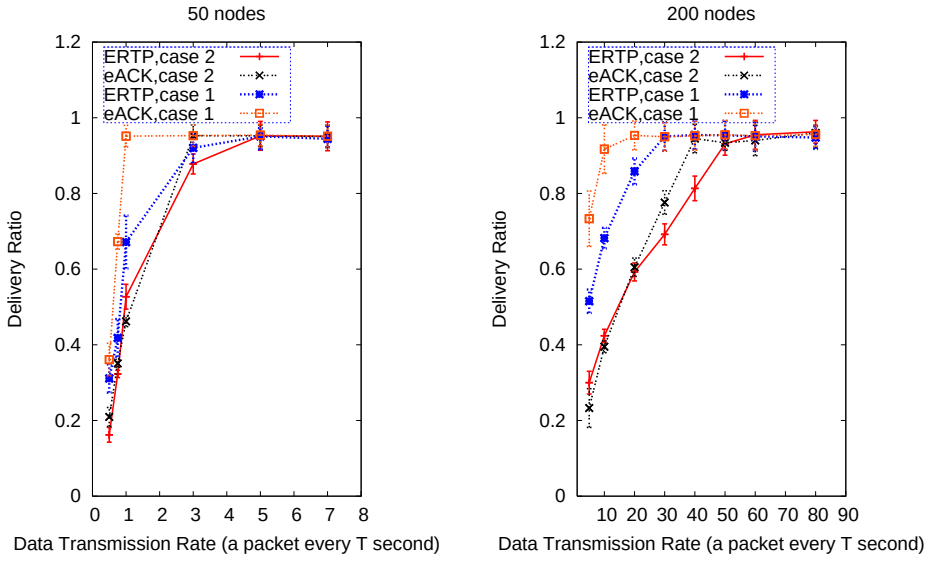


Figure 5.11: Simulation Network Topology.

achieve 95% delivery ratios when the data transmission rates are low, i.e. less than a packet every 5 seconds in the 50-node network. As the transmission rates increase, the delivery ratios are degraded. The limited network bandwidth does not allow the network to handle high data traffic, resulting in significant packet dropping when the data transmission rates are high. In the 50-node network, ERTP is congested at the transmission rate of 1 packet every 3 seconds for the low link loss rate, and at 1 packet every 5 seconds for the high link loss rate. In the 200-node network, ERTP is congested at the transmission rate of 1 packet every 20 seconds for the low link loss rate (case 1), and at 1 packet every 40 seconds for the high link loss rate (case 2). We observe that eACK can handle a slightly higher transmission rate than ERTP. The reason is that the transmitters in eACK do not need to wait as long as the transmitters in ERTP after sending out a packet. As a result, the servce time for a packet in eACK is quicker than in ERTP, thus, it can handle higher data transmission rates. However, these differences are not significant in low data rate streaming applications. Our experimental



(a) Delivery Ratio (b) Average Packet Delay

Figure 5.12: Performance with High Sending Rates.

results in Section 5.5.2 show that ERTP is significantly more energy-efficient than the eACK scheme. The results suggest that ERTP performs well in low data rate streaming applications, but may not be suitable for those applications that require high data rates.

5.5 Implementation and Experimental Evaluation

Having validated the performance of ERTP by simulations in Section 5.4, we implemented ERTP in TinyOS 1.x and compared it to state-of-the-art reliable WSN communication protocol, Surge Reliable [19] in a 16 Fleck-3 [5] real network testbed. We selected node 0 as the sink and the other nodes generate packets of 40 bytes at the rate of one packet every 10 seconds. The application layer E2E reliability requirement is $\alpha = 0.95$. Each experiment was run for 30 minutes and each node logged the energy consumptions for handling each data packet. Each data point in the experiment figures is the average of 5 experiments.

5.5.1 Baseline

We start with a simple baseline experiment that illustrates some of the important features of ERTP. Figure 5.13 is a snapshot of the routing tree in our experiment. Though link changes quality over time, the routing tree also changes.

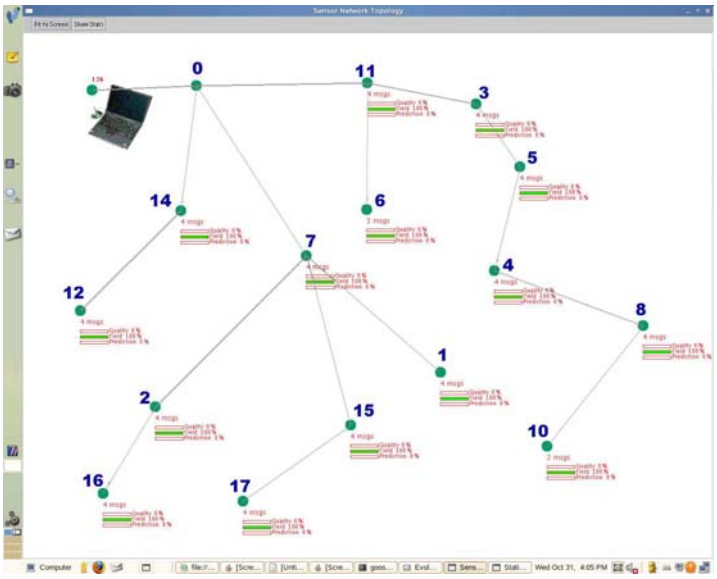


Figure 5.13: Network Topology.

5.5.1.1 Link Quality

We observed the well-known phenomena in wireless communication such as link asymmetry and dynamic link qualities. For example, Figure 5.14 and Figure 5.15 show the link qualities of nodes 8 and 2 during one of our experiments. We discovered that the link quality of node 2 varies a lot because of its low quality antenna.

5.5.1.2 Energy Consumption

The average upstream link quality, the average downstream link quality, and the average number of hops from the sink are obtained for each node. Based on these parameters, we can calculate the expected total energy consumption by Proposition 2.

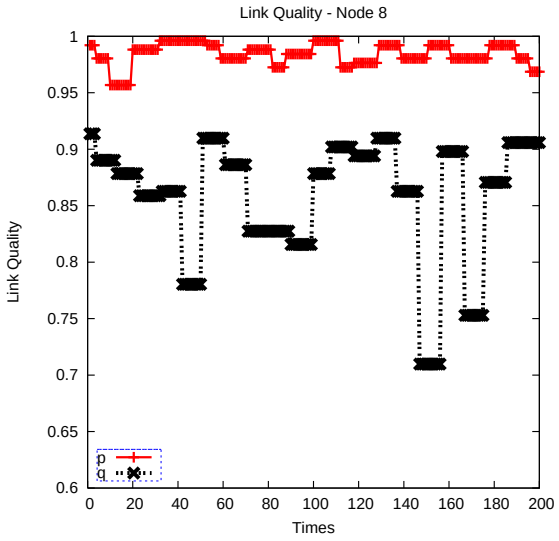


Figure 5.14: Link Quality at Node 8.

Figure 5.16 compares the predicted and the actual average energy consumption for a data packet to be delivered successfully to the sink. There are slight differences between the theoretical results and the actual measurements because the RTO update value is slower than the changes of link qualities. Particularly, the energy consumptions of node 16 and node 2 are much higher than the theoretical results. Because node 2 is downstream of node 16 in the routing tree, the performance of node 16 is impacted by the poor link quality at node 2 (see Figure 5.15). Despite these differences, there is a consistent trend between theoretical results and experimental results.

5.5.1.3 Delivery Ratio

Figure 5.17 shows the average delivery ratio of all nodes. Apart from node 2 and 16, the other nodes achieved more than 93% delivery ratios. The reliability achieved is

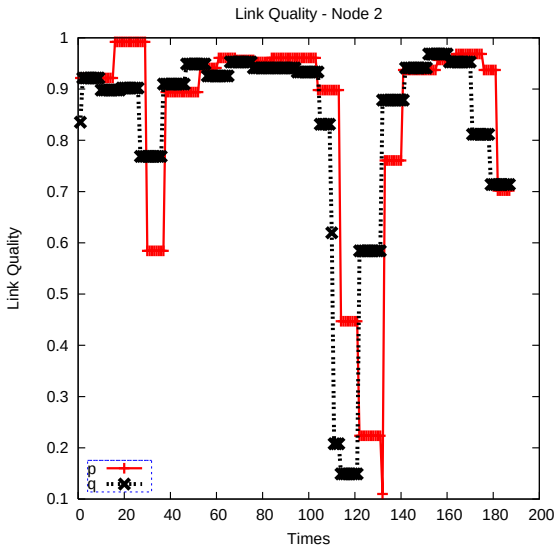


Figure 5.15: Link Quality at Node 2.

slightly lower than the requirement for the following reasons. First, the update of the maximum number of retransmissions is slower than the change in link quality. Thus, old estimates for maximum number of retransmissions are not accurate when link quality changes. Moreover, we observed that the routing path broke down for a small period of time in our experiments and caused packet losses. In addition, we also observed that although affected by poor link quality between node 2 and node 7 (see Figure 5.15), node 2 and node 16 achieved reasonable delivery ratios (83% and 81%, respectively).

5.5.2 The Comparison between ERTP and Surge Reliable

The baseline experiment demonstrates some of the salient features of ERTP. In this section, we compare the performance of ERTP with the state-of-the-art reliable WSN communication protocol, Surge Reliable together with eACK (we call it *Surge*

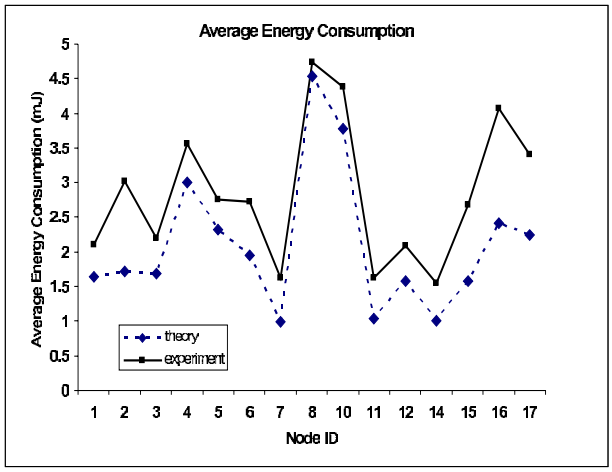


Figure 5.16: Energy Consumption.

for the purpose of brevity). To compare the energy consumption between ERTP and *Surge*, we modified *Surge* so that it can dynamically control the maximum number of retransmissions for ensuring the application layer E2E reliability requirements. Note that for this experiment we changed the antenna of the node 2 so that it provided reasonably stable link quality. Figure 5.18 is a snapshot of the routing tree in one of our experiments. The longest route length was 6 hops from the sink.

5.5.2.1 Energy Consumption

Figure 5.19 shows the average energy consumption of ERTP and *Surge* versus route length. We observe that ERTP outperforms *Surge*. As eACK is used for the 1-hop nodes in both ERTP and *Surge*, the average energy consumption is similar for both protocols. However, for the 6-hop nodes, ERTP has 27% less energy consumption than *Surge*. The average delivery ratios of ERTP and *Surge* are around 93%, which are close to the reliability requirement of 95%.

We also conducted experiments to assess the performance of ERTP when the link error rates are high. We introduced artificial losses for all links in the testbed. The link layer dropped packets with a 25% probability. This effectively increased the expected energy consumption of the sensor nodes.

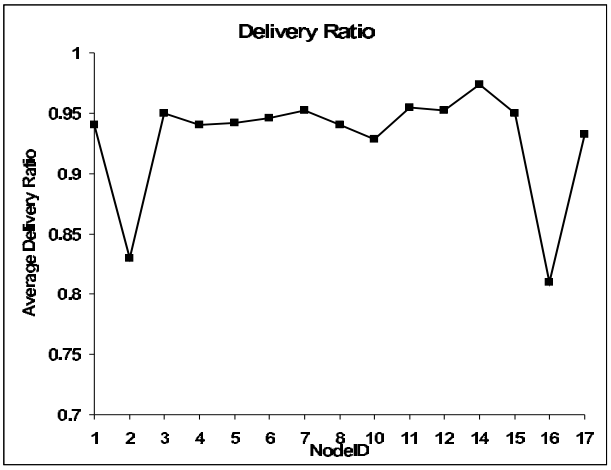


Figure 5.17: Delivery Ratios.

Figure 5.20 shows the average energy consumption of ERTP and *Surge* when the network links are lossy. Similar to Figure 5.19, we observe that ERTP is more energy-efficient than *Surge*. For 6-hop nodes, ERTP reduces energy consumption by more than 35% compared to *Surge*. Furthermore, we also observe that both protocols achieve 95% E2E reliability with the small error rate of ±4%.

5.5.2.2 The Impact of High Data Transmission Rates

Although ERTP is designed for low data rate applications, we would like to evaluate the performance of ERTP when the data rate is high. We ran both ERTP and *Surge* with different data rates: 2 packets every second, 1 packet every second, 1 packet every 2 seconds, 1 packet every 5 seconds, 1 packet every 8 seconds, and 1 packet every 10 seconds. Figure 5.21 shows the average delivery ratio achieved during the entire experiment. We observe that ERTP can achieve around 93% for low data transmission rates, i.e. a packet every 8 seconds, and a packet every 10 seconds. The delivery ratios of ERTP slightly degrade when the data rates increase. The reason is that with high data rates, the packet collision occurs more often and causes significantly packet losses. Moreover, we observe that at 1 packet every 0.5 second, the delivery ratio of ERTP is around 74% because of network congestion. A similar observation has been discussed

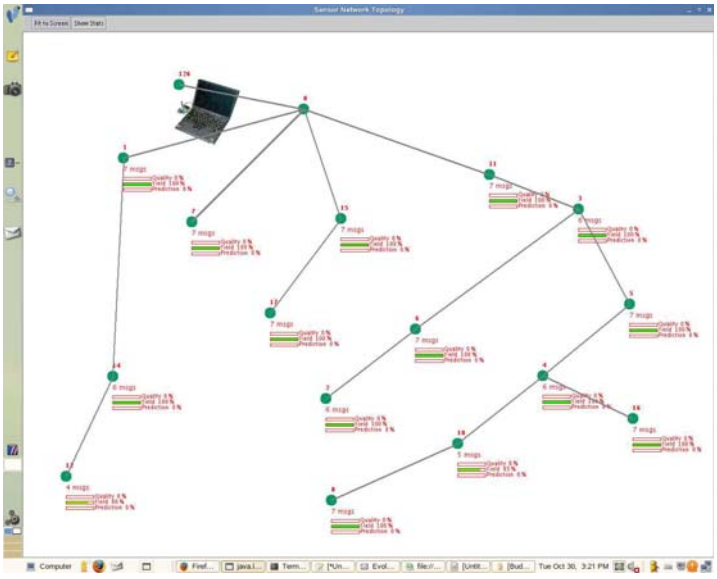


Figure 5.18: Network Topology for Comparison of ERTP and *Surge*.

in the 40-Tmote experiments in [21]. They observed that the network was congested when the data rate of each sensor node was around 0.8 packet per second. Note that the bandwidth of T-mote is 250 kbps, which is 5 times more than the bandwidth of Fleck-3 (50 kbps). We also observe that the obtained delivery ratios of *Surge* is slightly higher than ERTP when the data rate is high. The obtained results are consistent to the simulation results described in the section 5.4.3.6 and the experimental results described in [21].

5.5.2.3 Scalability

Finally, to evaluate the scalability, we ran both ERTP and *Surge* on different network sizes, i.e. 16, 35, and 50 nodes. Figure 5.22 shows the minimum, median, and maximum hop-counts and the delivery ratios achieved for each sensor node in one of our experiments for a 50-node network. The longest route length was 9 hops observed from nodes 22, 24, and 39.

Figure 5.23 shows the total energy consumption and average delivery ratios achieved during the entire experiment. We observe that the network was very dynamic with sig-

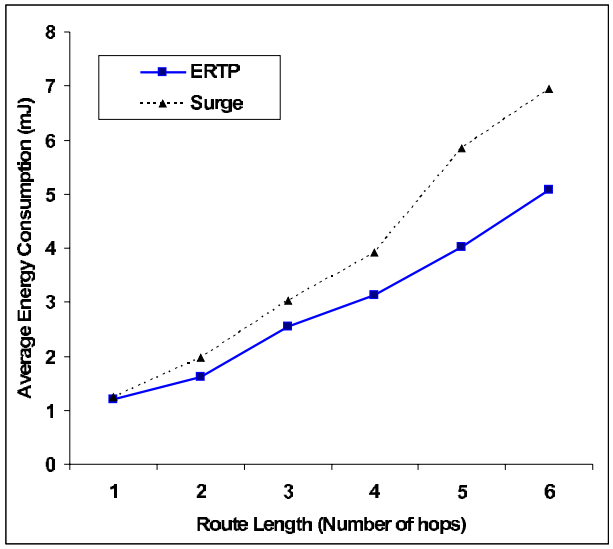


Figure 5.19: Energy Consumption.

nificant routing variability. As a result, packets were dropped more often in the routing maintenance phase. High traffic levels, particularly added traffic for route maintenance, significantly impact the performance of both protocols. Despite the dynamics of the network topology, both ERTP and *Surge* achieve similar delivery ratios (more than 91%) for all the cases while the energy consumption with ERTP is significantly lower than with *Surge*. We observe that ERTP reduces energy consumption by more than 45% when compared to *Surge* for the 50-node network. This highlights the robustness and scalability of ERTP design and implementation.

5.6 Related Work

A summary of relevant related work on transport protocols for WSNs is given in Table 5.3. We distinguish the transport protocols by three different characteristics: reliability, energy-awareness, and the type of data flows that they support (continuous data flows or a bulk data flow). As shown in Table 5.3, ERTP and RMST [20] (see Section 2.1.1.2), to the best of our knowledge, are the only transport protocols for continuous data flow WSNs that takes reliability and energy-constraints into account.

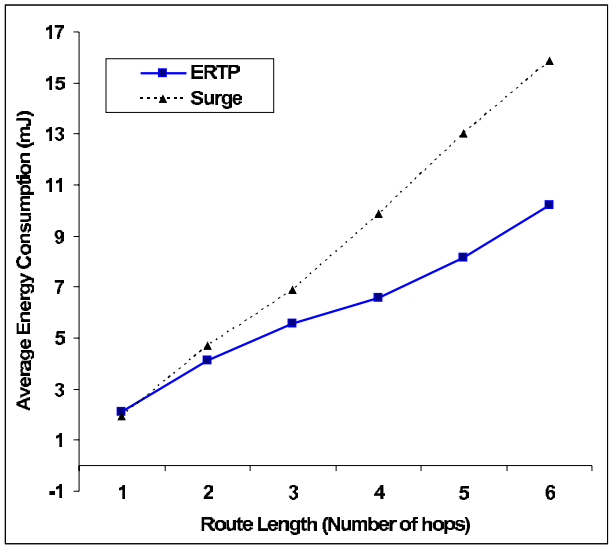


Figure 5.20: Energy Consumption for Lossy Links.

Table 5.3: Sensor Network Transport Protocols

Protocol Name	Main Approach	Reliability	Energy-Aware	Type of data flows
ESRT	Centralized Rate Control	Yes	No	Continuous
RMST	HBH NACK	Yes	Yes	Continuous
PSFQ	HBH NACK	Yes	No	Bulk
RBC	Windowless block ACK	No	No	Bulk
Flush	Distributed Rate Control	Yes	No	Bulk
Wisden	E2E NACK	Yes	No	Continuous
RCRT	Centralized Rate Control	Yes	No	Continuous
ERTP	HBH iACK	Yes	Yes	Continuous

However, RMST uses HBH NACK for loss recovery while ERTP uses HBH iACK. RMST is tightly bound to Directed Diffusion routing protocol [28] in which packet losses are recovered HBH using caches in the nodes along the path to the sink. Furthermore, RMST is not scalable because it requires each intermediate nodes to cache all packets received from each upstream source. Memory limitation on resource-constrained sensor nodes requires intelligent caching strategies to be considered. However, RMST is the closest in spirit to our work in that it attempts to control the HBH reliability to achieves E2E reliability. Unlike RMST, ERTP achieves the E2E reliability through HBH loss recovery using the HBH iACK approach and it is independent to the routing protocols. Thus, ERTP has greater flexibility than RMST.

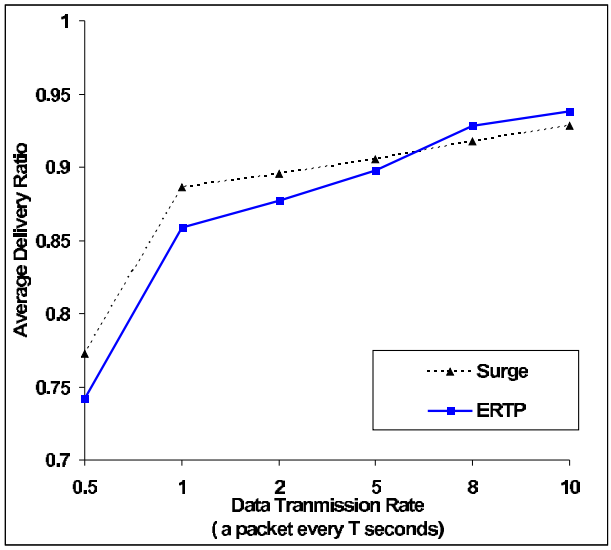


Figure 5.21: The Delivery Ratios of ERTP and *Surge* for different Data Transmission Rates.

Akan et al. proposed Event to Sink Reliable Transport called ESRT [25] (see Section 2.1.3.1) for E2E reliability based on the notion of event-to-sink reliability. ESRT achieves the reliable detection of an event and congestion avoidance by controlling the transmission rate of each source at the sink. Although ESRT does not require packet retransmissions, it is not as energy-efficient as HBH loss recovery schemes since the rate decision is controlled centrally [75]. Moreover, ESRT assumes that the sink can communicate with all sources directly, which may not be a reasonable assumption in practical WSN deployments. PSFQ [22] (see Section 2.1.2.1) is a reliable dissemination protocol aimed for reprogramming WSNs from a sink, i.e., a bulk data flow, a large finite bulk of data packets which needs to be transmitted to the sink, not for the transport of streaming data from the sources to the sink.

To overcome the memory constraints, an E2E NACK loss recovery scheme is used in [14] [24] [21] to provide transmission reliability in WSNs. In this scheme, the sink detects packet losses and requests E2E retransmissions from the source nodes. Although this scheme alleviates the memory burden on sensor nodes, it is not as energy-efficient as HBH loss recovery [75]. Moreover, using E2E NACKs may cause feedback explosion

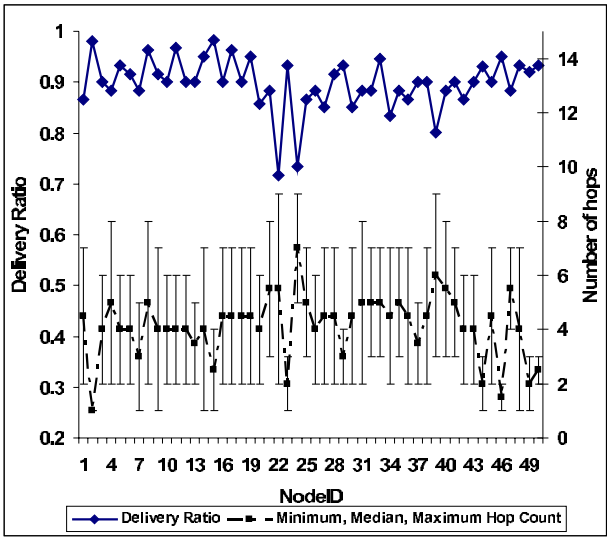


Figure 5.22: Delivery Ratio and Hop Count for 50-node Network.

when the links are lossy. Xu et al. proposed Wisden [14] (see Section 2.1.1.3), a reliable data collection protocol for structural monitoring. However, Wisden uses E2E NACKs for loss recovery, and thus is not energy-efficient. Kim et al. proposed Flush [24] (see Section 2.1.2.3), a reliable, single-flow bulk transport protocol for large diameter WSNs. However, Flush only supports one data flow and targets bulk traffic. Paek et al. proposed RCRT [21] (see Section 2.1.1.4), a rate-controlled reliable transport protocol for WSNs. Both Flush and RCRT focus on achieving 100% reliability and high throughput via congestion control without consideration of energy-efficiency. In contrast, ERTP explores the characteristics of statistical reliability in data streaming applications to reduce energy-consumption in packet transmissions, and achieves E2E reliability through HBH loss recovery using the iACK approach.

Zhang et al. proposed Reliable Bursty Convergecast (RBC) protocol to transport bulk traffic reliably in WSNs [23] (see Section 2.1.2.2). RBC uses a windowless block acknowledgment scheme to improve channel utilization and to reduce the number of nodes competing for channel access. However, RBC is not designed for continuous data flows, and does not guarantee statistical E2E reliability. Rangwala et al. proposed

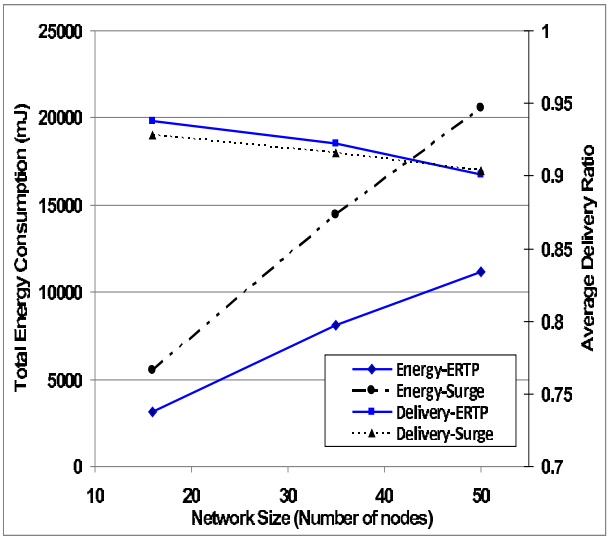


Figure 5.23: Total Energy Consumption and Average Delivery Ratios for Different Network Sizes.

IFRC [99], an Interference-aware Fair Rate Control for WSNs. IFRC is a distributed rate allocation scheme that uses the local queue size to detect congestion.Although IFRC can offer high throughput, it does not guarantee statistical E2E reliability.

Scheuermann et al. presented a HBH congestion control protocol in wireless multihop networks using HBH iACK [89]. The protocol ensures that the input rate of a given flow does not exceed the output rate in all intermediate nodes. To avoid redundant retransmissions, several heuristics to handle packet loss are discussed. However, this work primarily addresses transmission rate and congestion control for absolute reliability and only considers the packet loss caused by buffer overflows whereas in reality, packet loss is mostly caused by the lossyness of wireless channels. Moreover, a fixed RTO scheme (three times the HBH transmission time) is used in this work. Our simulation results in Section 5.4 show that a fixed RTO scheme is not energy-efficient when the link loss rates are high. We propose a distributed algorithm for RTO estimation, which adapts to the environment, i.e., lossy wireless channels. To the best of our knowledge, ours is the first work which investigates adaptive RTO estimation for the class of HBH iACK protocols in WSNs.

Surge Reliable [19] (see Section 2.1.1.1) is the state-of-the-art reliable multi-hop routing protocol for continuous data flows that uses expected number of transmissions as the routing metric. Surge Reliable dynamically forms a spanning tree that covers every node in the network, using link connectivity estimation and neighborhood table management techniques. Each node periodically measures the link qualities between itself and its neighbors by link layer active snooping. A node obtains bi-directional link qualities by exchanging neighborhood tables with its neighbors. Link layer HBH eACK and retransmissions improve E2E transmission reliability. A node selects the best neighbor, the one with the minimum expected number of transmissions, as its parent to which it forwards data packets. The performance of Surge Reliable has been shown to be superior to other routing protocols such as Destination-Sequenced Distance-Vector Routing (DSDV) [60], and Ad hoc On-Demand Distance Vector Routing (AODV) [61] in unreliable wireless environments. However, a fixed number of link layer retransmissions is used, and therefore, it does not guarantee statistical E2E reliability when the link loss rates change. Further, Surge Reliable is not energy-efficient when the link quality is good, since it introduces a significant number of eACKs.

5.7 Chapter Summary

This chapter has presented ERTP, an Energy-efficient and Reliable Transport Protocol for WSNs, designed for WSN data streaming applications. ERTP balances reliability and energy-efficiency by dynamically controlling the maximum number of retransmissions, and exploring the wireless overhearing capability for iACK. The analysis of the trade-off between energy consumption and E2E reliability for ERTP is presented, in which HBH iACK approach and duplicate detection are used at each sensor node. We have also proposed the distributed algorithm for RTO estimation. The challenge in deciding the RTO is that premature timeout will cause redundant transmissions while a large timeout will cause poor capacity utilization. The results show that the pro-

posed RTO algorithm can reduce energy consumption by up to 50% when compared to other approaches. Finally, we have implemented and compared ERTP to *Surge*, the state-of-the-art reliable WSN communication protocol. The results show that ERTP can reduce energy consumption by more than 45% when compared to *Surge*. Consequently, sensor nodes are more energy-efficient and the lifespan of the unattended WSN is increased. The future work may investigate the performance of ERTP with different MAC-protocols such as TDMA MAC protocols [87] [88].

Chapter 6

Conclusion and Future Work

In this thesis, we have investigated different directions for reliable data transport in WSNs. In Chapter 3, we have presented the cross-layer communication protocol for a reliable data transfer in WSN data streaming applications. At the MAC layer, a CSMA MAC protocol with an HBH eACK loss recovery is employed. To ensure the E2E reliability, the maximum number of retransmissions are estimated and used at each sensor node. At the network layer, an E2E NACK with an aggregated positive Acknowledgment mechanism is used. By inspecting the sequence numbers on the packets, the base station can detect which packets were lost. The base station sends a NACK to a source after receiving a preset number of packets from the source. In addition, to increase the robustness of the system, a watchdog process is implemented at both base station and sensor nodes, which enable them to power cycle themselves when a unexpected fault occurs. This has proven to be very helpful for such a remote deployment, particularly when the node behaviors are unexpected. We implemented and evaluated the proposed cross-layer communication protocol in both simulations and field experiment. The designed network system has been working in the deployed field for over a year and has offered relatively good E2E data reliability despite the highly dynamics in the environment. The results show that our system is a promising solution to allow gathering of sufficient water quality data to establish a sustainable irrigation system.

The experimental results indicate that there are complexities in radio propaga-

tion which we do not yet fully understand or have a remedy for. Although the research community has well observed that the "disc" transmission model is not applicable to most wireless transmission scenarios, we failed to find any network deployment methodologies that can model the environment well, and produce high connectivity networks. In particular, the methodology that takes the deployment parameters, such as terrain, humidity, and height of the antennas, into account when calculating the performance of radio links is desirable. In addition, while the upstream traffic can be delivered efficiently, it is inefficient to deliver downstream traffic, e.g., ACKs and NACKs. We observed the receipt of NACKs with high loss in our deployment, in particular to the nodes located deeply in the routing tree. A reliable, scalable, and bi-directional routing protocol for both upstream and downstream is worth in these scenarios to improve the performance of reliable transmission protocols in the transport layer. Finally, during the deployment, we observed that the link connection between sensor nodes and gateway is down frequently during the night time, but we have insufficient data to determine the accurate cause. A possible way to overcome the mid-night crisis is that the sensor nodes can store the packets in the EPROM during the night time and transfer them to the gateway in the morning time when the link quality is good. Thus, an intelligent buffer technique can be investigated in this case.

In Chapter 4, we have proposed, implemented, and evaluated SRM: a Sensor Reliability Management framework for WSNs. SRM is based on a hierarchical management architecture and on the policy-based network management paradigm. Although SRM is designed for data reliability management, it can be easily integrated with other management services as a part of a WSN self-management architecture. In addition, SRM also allows the network administrators to interact with the network by providing management policies. SRM comprises four modules: an evaluation module, a user policy specification module, a decision making module, and an action module. The interaction among these modules enables the management framework to efficiently adapt to the

network dynamics. The evaluation module is responsible for collecting the management information required to estimate the reliability of the network. The user policy specification module allows the users to describe a reliability policy, translate it to the XML format, and distribute it over the network. Based on the estimated reliability from the evaluation module and the policies defined in management policy component, the decision making component determines appropriate actions to be executed and passes the request to the action module. Finally, the action module is responsible for performing the action assigned by the decision making module. The core of the action module is a management function which is executed on the sensor nodes for performing an action. We have implemented and evaluated SRM in a real-testbed. The results show that SRM can offer sufficient data reliability, while significantly reducing energy consumption by more than 50% when compared to no management.

There are still open issues as far as SRM is concerned. The promising results obtained here motivate a further investigation on other components in SRM. First, to reduce management control traffic, the reactive mode of the monitoring scheduler component is worth to look at. The future work may investigate the learning of the daily trends in data reliability and come up with an adaptive monitoring schedule for reliability management. Another interesting question here is how the decision making module handles the conflicts among management rules. A linear programming approach may be useful to identify how the maximization of number of rules could be satisfied which maximizes the objective benefits.

In Chapter 5, we have presented ERTP, an Energy-efficient and Reliable Transport Protocol for WSNs, designed for WSN data streaming applications. ERTP balances reliability and energy-efficiency by dynamically controlling the maximum number of retransmissions, and exploring the wireless overhearing capability for iACK. ERTP has two components: the HBH reliability control, and the HBH retransmission timeout control. The HBH reliability control ensures the data reliability by dynamically control-

ling the maximum number of retransmissions for each data packet at each intermediate node. Obviously, a sensor node can not allow a very large number of retransmissions because of packet freshness and fairness and energy concerns. An insufficient maximum number of retransmissions may cause packet to be lost as it travels to the sink, wasting energy and network resources, as well as degrading E2E reliability. Conversely, there will be energy-inefficiency when the maximum number of retransmissions is too high. To balance energy consumption and E2E reliability, the HBH reliability component dynamically determines and updates a near optimal maximum number of retransmissions for data packets at each node. The HBH retransmission timeout component is responsible for controlling the RTO at each node. A "premature" *RTO* value may increase sensor energy-consumption because transmitters will send duplicate packets. This is energy-inefficient since the packet has already been received. On the other hand, a large *RTO* value tends to increase transmission latency and thus reduces network throughputs. Therefore, in order to achieve energy efficiency, the HBH Retransmission Timeout component of ERTP is responsible for adjusting the RTO dynamically. We have extensively evaluated ERTP in both simulation and experiments. The results show that ERTP can reduce energy consumption by more than 45% when compared to *Surge*, the current state-of-the-art reliable protocol for WSNs. Consequently, sensor nodes are more energy-efficient and the lifespan of the unattended WSN is increased.

Future work in this area includes the investigation of the hybrid protocols between eACK and iACK. Although the iACK is more energy-efficient than eACK, it may incur high packet latency when the transmission rates are high, because the sensor nodes need to wait for overhearing the packet. SRM framework can be used here to adaptively determine which protocols should be used to balance the energy consumption and the packet latency.

Another piece of work on a distributed protocol for MObile SEnsor Relocation called Moser for maintaining the network connectivity and coverage has been presented

in the Appendix A. This work although handles a very important aspect of reliability, it is peripheral to the reliability protocol work described in rest of this thesis and hence has been added as an appendix. We consider a hybrid sensor network where a subset of the nodes have ability to move movement at a high energy expense. When a node is low remaining energy and it is a critical node such as a cluster head, it will seek a replacement. If a redundant node is located in the transmission range of the dying node and can fulfill the connectivity and coverage requirement, it can be used for substitution. Otherwise, a protocol should be in place to relocate the redundant nodes for replacement. Moser protocol works in three phases. In the first phase, the dying node determines if network partition occurs, finds an available mobile node, and asks for replacement by using flooding algorithm. The dying node also decides the movement schedule of the available redundant node based on certain criteria. The second phase of the protocol involves the actual movement of the mobile node to approach the location of the dying node. Finally, when the mobile node has reached the transmission of the dying node, it communicates to the dying nodes and moves to a desired location, where the network connectivity and coverage to the neighbors of the dying nodes are preserved. Unlike existing solutions using the assumptions of precise location information is available, Moser protocol can be performed without any localization algorithm needed. The simulation shows that the Moser protocol can perform the replacement successfully with minimal energy consumption.

Future work in this area is the implementation of the protocol. The experimental study of the relationship between RSSI and distance may be investigated and used in the Moser protocol.

Publication List

1. **Tuan Le**, Wen Hu, Peter Corke, and Sanjay Jha, "ERTP: Energy-efficient and Reliable Transport Protocol for Data Streaming in Wireless Sensor Networks", Elsevier Computer Communications, 2008 (under submission) (Chapter 5).

2. **Tuan Le**, Wen Hu, Sanjay Jha, and Peter Corke, "Design and Implementation of a Policy-based Management System for Data Reliability in Wireless Sensor Networks", The Third IEEE International Workshop on Practical Issues in Building Sensor Network Application (SenseApp), Canada, 2008 (Chapter 4).

3. Z. Rosberg, R. Liu, A. Dong, **Tuan Le**, and Sanjay Jha, "ARQ with Implicit and Explicit ACKs in Sensor Network", IEEE Global Communications Conferences (Globecom), USA, 2008.

4. **Tuan Le**, A.Y. Dong, R. Liu, Sanjay Jha, and Z. Rosberg, "Implementation Aspects of Reliable Transport Protocols in Wireless Sensor Networks", The Third IEEE International Conference on Communication System Software and Middleware (Comsware), India, 2008 (Chapter 5).

5. **Tuan Le**, Wen Hu, Pavan Sikka, and Peter Corke, "Design and Development of a Remote Robust Sensor Network: Experiences from an Outdoor water quality Monitoring", The Second IEEE International Workshop on Practical Issues in Building Sensor Network Application (SenseApp), Ireland, 2007 (Chapter 3).

6. **Tuan Le**, Nadeem Ahmed, and Sanjay Jha, "Location-free Fault Repair in Hybrid Sensor Networks", The First ACM International Conference on Integrated Internet Ad-hoc and Sensor Networks (Intersense), France, 2006 (Appendix A).

7. C. Sreenan, S. Nawar, **Tuan Le**, and Sanjay Jha, "On the Sensitivity of sensor network simulations", The 63rh IEEE Vehicular Technology Conference (VTC), Australia, 2006.

8. **Tuan Le**, Nadeem Ahmed, Nandan Parameswaran, and Sanjay Jha, "Fault Repair Framework for Mobile Sensor Networks", The First IEEE International Conference on Communication System Software and Middleware (Comsware), India, 2006 (Chapter 4).

Bibliography

[1] "Crossbow Technology, Mica2." http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf.

[2] "Tmote." http://www.moteiv.com/products/tmotesky.php.

[3] G. Sibley, M. Rahimi, and G. Sukhatme, "Robomote: A tiny mobile robot platform for large-scale sensor networks," in ICRA '02: Proceedings of the 2002 IEEE International Conference on Robotics and Automation, (Washington, DC, USA), pp. 1143–1148, 2002.

[4] "CSIRO ICT Centre." http://www.ict.csiro.au.

[5] "Fleck Platform." http://www.sensornets.csiro.au/fleck1.htm.

[6] P. Sikka, P. Corke, L. Overs, P. Valencia, and T. Wark, "Fleck: A platform for real-world outdoor sensor networks," in ISSNIP '07: Proceedings of the third International Conference on Intelligent Sensors Networks and Information Processing, (Melbourne, Australia), pp. 709–714, 2007.

[7] T. L. Dinh, W. Hu, P. Sikka, P. Corke, L. Overs, and S. Brosnan, "Design and deployment of a remote robust sensor network: Experiences from an outdoor water quality monitoring network," in SenseApp '07: Proceedings of the second IEEE Workshop on Practical Issues in Building Sensor Network Applications, (Dublin, Ireland), pp. 799–806, 2007.

[8] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," Communications of the ACM, vol. 47, no. 6, pp. 34–40, 2004.

[9] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: application driver for wireless communications technology," SIGCOMM Computer Communication Review, vol. 31, no. 2 supplement, pp. 20–41, 2001.

[10] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, and J. Porter, "Luster: wireless sensor network for environmental research," in SenSys '07: Proceedings of the 5th International Conference on Embedded Networked Sensor Systems, (Sydney, Australia), pp. 103–116, ACM, 2007.

[11] A. Ledeczi, P. Volgyesi, M. Maroti, G. Simon, G. Balogh, A. Nadas, B. Kusy, S. Dora, and G. Pap, "Multiple simultaneous acoustic source localization in urban terrain," in IPSN '05: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, (Los Angeles, CA, USA), p. 69, IEEE Press, 2005.

[12] R. Chellappa, G. Qian, and Q. Zheng, "Vehicle detection and tracking using acoustic and video sensors," in ICASSP '04: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, (Montreal, Canada), pp. 793–796, 2004.

[13] T. Bokarev, W. Hu, S. Kanhere, B. Ristic, N. Gordon, T. Bessell, M. Rutten, and S. Jha, "Wireless sensor networks for battlefield surveillance," in LWC '06: Proceedings of the Land Warfare Conference, (Brisbane, Queensland, Australia), 2006.

[14] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, (Baltimore, MD, USA), pp. 13–24, ACM, 2004.

[15] L. Schwiebert, S. K. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," in MobiCom '01: Proceedings of the 7th annual International Conference on Mobile Computing and Networking, (Rome, Italy), pp. 151–165, ACM, 2001.

[16] E. Jovanov, A. Milenkovic, C. Otto, and P. C. de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," Neuro-Engineering and Rehabilitation, vol. 2, no. 11, pp. 2–6, 2005.

[17] "IETF, Terminology for Policy-based Management." `http://www.ietf.org/rfc/rfc3198.txt`.

[18] "RFC 793 - Transmission Control Protocol." `http://www.faqs.org/rfcs/rfc793.html`.

[19] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, (Los Angeles, CA, USA), pp. 14–27, ACM Press, 2003.

[20] R. Stann and J. Heidemann, "Rmst: Reliable data transport in sensor networks," in SNPA '03: Proceedings of the first IEEE International Workshop on Sensor Network Protocols and Applications, (Anchorage, Alaska, USA), pp. 102–112, 2003.

[21] J. Paek and R. Govindan, "Rcrt: Rate-controlled reliable transport for wireless sensor networks," in SenSys '07: Proceedings of the 5th International Conference on Embedded Networked Sensor Systems, (Sydney, Australia), pp. 305–319, ACM, 2007.

[22] C.-Y. Wan, A. Campbell, and L. Krishnamerthy, "Psfq: A reliable transport protocol for wireless sensor networks," in WSNA '02: Proceedings of the first ACM International Workshop on Wireless Sensor Networks and Applications, (Atlanta, Georgia, USA), pp. 862–872, 2002.

[23] H. Zhang, A. Arora, Y. ri Choi, and M. G. Gouda, "Reliable bursty convergecast in wireless sensor networks," in MobiHoc '05: Proceedings of the 6th ACM International Symposium on Mobile Adhoc Networking and Computing, (Urbana-Champaign, IL, USA), pp. 266–276, ACM, 2005.

[24] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. E. Culler, P. Levis, S. Shenker, and I. Stoica, "Flush: A reliable bulk transport protocol for multihop wireless network," in SenSys '07: Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems, (Sydney, Australia), pp. 351–365, 2007.

[25] Özgür B. Akan and I. F. Akyildiz, "Esrt: Event-to-sink reliable transport in wireless sensor networks," IEEE/ACM Transactions on Networking, vol. 13, no. 5, pp. 1003–1016, 2005.

[26] Y. Zhou and M. R. Lyu, "Port: A price-oriented reliable transport protocol for wireless sensor networks," in ISSRE '05: Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering, (Washington, DC, USA), pp. 117–126, 2005.

[27] Y. G. Iyer, S. Gandham, and S. Venkatesan, "Stcp: A generic transport layer protocol for wireless sensor networks," in ICCCN '05: Proceedings of the 14th IEEE International Conference on Computer Communications and Networks, (San Diego, CA, USA), 2005.

[28] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," IEEE/ACM Transactions on Networking, vol. 11, no. 1, pp. 2–16, 2003.

[29] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "Coda: congestion detection and avoidance in sensor networks," in SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, (New York, NY, USA), pp. 266–279, ACM, 2003.

[30] L. B. Ruiz, MANNA: A management architecture for wireless sensor networks. PhD thesis, 2003.

[31] M. Turon, "Mote-view: a sensor network monitoring and management tool," in EmNets '05: Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors, (Washington, DC, USA), pp. 11–17, IEEE Computer Society, 2005.

[32] G. Tolle and D. Culler, "Design of an application-cooperative management for wireless sensor networks," in EWSN '05: Proceedings of Second European Workshop on Wireless Sensor Networks, (San Diego, CA, USA), pp. 121–132, 2005.

[33] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler, and D. Estrin, "Sympathy for the sensor network debugger," in SenSys '05: Proceedings of the 3rd

International Conference on Embedded Networked Sensor Systems, (San Diego, CA, USA), pp. 255–267, ACM, 2005.

[34] C. Frank and K. Römer, "Algorithms for generic role assignment in wireless sensor networks," in SenSys '05: Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, (San Diego, CA, USA), pp. 230–242, ACM, 2005.

[35] C. fan Hsin and M. Liu, "A distributed monitoring mechanism for wireless sensor networks," in Wise '02: Proceedings of the 1st ACM workshop on Wireless Security, (Atlanta, GA, USA), pp. 57–66, ACM, 2002.

[36] G. Mainland, D. C. Parkes, and M. Welsh, "Decentralized, adaptive resource allocation for sensor networks," in NSDI '05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, (Berkeley, CA, USA), pp. 315–328, USENIX Association, 2005.

[37] "Fixed point iteration." `http://math.fullerton.edu/mathews/n2003/FixedPointMod.html`.

[38] 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, (Wailea Maui, Hawaii, USA), p. 8020, IEEE Computer Society, 2000.

[39] O. Younis and S. Fahmy, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," IEEE Transactions on Mobile Computing, vol. 3, no. 4, pp. 366–379, 2004.

[40] T.-H. Kim and S. Hong, "Sensor network management protocol for state-driven execution environment," in ICUC '03: Proceedings of the International Conference on Ubiquitous Computing, (Seoul, South Korea), pp. 197–199, 2003.

[41] B. Deb, S. Bhatnagar, and B. Nath, "Stream: Sensor topology retrieval at multiple resolutions," Kluwer Journal of Telecommunications Systems, vol. 26, pp. 285–320, 2004.

[42] W. Liu, Y. Zhang, W. Lou, and Y. Fang, "Rrp: Managing wireless sensor networks with supply chain strategy," in QSHINE '04: Proceedings of the 1st International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, (Dallas, TX, USA), pp. 59–66, IEEE, 2004.

[43] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," IEEE Design and Test, vol. 18, no. 2, pp. 62–74, 2001.

[44] "Moveiv, Telos." `http://www.moteiv.com/`.

[45] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," ACM Wireless Networks, vol. 8, no. 2–3, pp. 169–185, 2002.

[46] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," MobiCom '01: Proceedings of the 7th annual International Conference on Mobile Computing and Networking, pp. 85–96, 2001.

[47] A. Cerpa and D. Estrin, "Ascent: Adaptive self-configuring sensor network topologies," SIGCOMM Computer Communication Review, vol. 32, no. 1, pp. 62–62, 2002.

[48] S. Ganeriwal, A. Kansal, and M. B. Srivastava, "Self aware actuation for fault repair in sensor networks," in ICRA '04: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, (New Orleans, LA, USA), pp. 5244–5249, 2004.

[49] G. Wang, G. Cao, T. Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in Infocom '05: Proceedings of the 24th Conference of the IEEE Computer and Communications Societies, (Miami, Florida, USA), pp. 2302–2312, 2005.

[50] A. Sekhar, B. S. Manoj, and C. S. R. Murthy, "Dynamic coverage maintenance algorithms for sensor networks with limited mobility," in PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications, (Washington, DC, USA), pp. 51–60, 2005.

[51] N. Li and J. C. Hou, "Improving connectivity ofwireless ad hoc networks," in MOBIQUITOUS '05: Proceedings of the second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, (San Diego, CA, USA), pp. 314–324, 2005.

[52] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," IEEE Transactions on Mobile Computing, vol. 5, no. 6, pp. 640–652, 2006.

[53] A. Howard, M. J. Matarić, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," Autonomous Robots, Special Issue on Intelligent Embedded Systems, vol. 13, no. 2, pp. 113–126, 2002.

[54] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in MobiHoc '04: Proceedings of the 5th ACM International Symposium on Mobile Adhoc Networking and Computing, (Roppongi Hills, Tokyo, Japan), pp. 187–198, ACM, 2004.

[55] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Deployment and connectivity repair of a sensor net with a flying robot," in ISER '04: Proceedings of the 9th International Symposium on Experimental Robotics, (Singapore), p. 04, 2004.

[56] "Water resources - management and development - victoria." `http://www.anra.gov.au/topics/water/management/vic/gmu-lillimur-kaniva.html`.

[57] "Water resources observation network." `http://wron.net.au/`.

[58] "North burdekin water board." `http://www.nbwb.com.au/`.

[59] "TinyOS." http://www.tinyos.net/.

[60] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," ACM SIGCOMM Computer Communication Review, vol. 24, no. 4, pp. 234–244, 1994.

[61] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in WMCSA '99: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, (New Orleans, Louisiana, USA), pp. 90–100, 1999.

[62] NRF905 Specification. http://www.semiconductorstore.com/pdf/newsite/nordic/nRF905.pdf.

[63] "The Network Simulator - ns-2." http://www.isi.edu/nsnam/ns/.

[64] "Elders Weather Statisics." http://www.eldersweather.com.au/.

[65] M. Srivastava, R. Muntz, and M. Potkonjak, "Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments," in MOBICOM '01: Proceedings of the 7th annual International Conference on Mobile Computing and Networking, (Rome, Italy), pp. 132–138, 2001.

[66] W. Hu, V. N. Tran, N. Bulusu, C. T. Chou, S. Jha, and A. Taylor, "The design and evaluation of a hybrid sensor network for cane-toad monitoring," in IPSN '05: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, (Los Angeles, CA, USA), p. 71, IEEE Press, 2005.

[67] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture," in WPDRTS '06: Proceedings of the 14th International Workshop on Parallel and Distributed Real-Time Systems, (Rhodes, Greece), p. 8, 2006.

[68] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in Proceedings of the International Conference on Acoustics, Speech and Signal Processing, (Salt Lake City, Utah, USA), pp. 2033–2036, 2001.

[69] T. Wark, P. Corke, P. Sikka, L. Klingbeil, Y. Guo, C. Crossman, P. Valencia, D. Swain, and G. Bishop-Hurley, "Transforming agriculture through pervasive wireless sensor networks," IEEE Pervasive Computing, vol. 6, no. 2, pp. 50–57, 2007.

[70] T. Wark, C. Crossman, W. Hu, Y. Guo, P. Valencia, P. Sikka, P. Corke, C. Lee, J. Henshall, K. Prayaga, J. O'Grady, M. Reed, and A. Fisher, "The design and evaluation of a mobile sensor/actuator network for autonomous animal control," in IPSN '07: Proceedings of the 6th International Conference on Information Processing in Sensor Networks, (Cambridge, MA, USA), pp. 206–215, ACM Press, 2007.

[71] W. Hu, S. Rothery, and P. Corke, "An empirical study of data collection protocols for wireless sensor networks," in REALWSN 08': Proceedings of 3rd ACM Workshop on Real-World Wireless Sensor Networks, (Glasgow, Scotland), 2008.

[72] "Policy Framework Definition Language." http://www.ietf.org/proceedings/98dec/I-D/draft-ietf-policy-framework-pfdl-00.txt.

[73] P. Biron, K. Permanente, and A. Malhotra, "a XML schema part 2: Datatypes," 2001. http://www.snmp.com/protocol/.

[74] "Simple Network Management Protocol." http://www.snmp.com/protocol/.

[75] Z. Rosberg, R. Liu, T. Le, S. Jha, A. Y. Dong, and J. Zic, "Energy efficient statistically reliable hybrid transport protocol for sensed data streaming," Tech. Rep. 213, CSIRO ICT Centre, 2007.

[76] P. Corke, "FOS a new operating system for sensor networks," in EWSN '08: Proceedings of 5TH European Workshop on Wireless Sensor Networks, (Bologna, Italy), 2008.

[77] S. H. Cha, J. E. Lee, M. Jo, H. Y. Youn, S. Kang, and K. H. Cho, "Policy-based management for self-managing wireless sensor networks," IEICE Transactions on Communications, vol. 90-B, no. 11, pp. 3024–3033, 2007.

[78] H. Song, D. Kim, K. Lee, and J.Sung, "UPnP-based sensor network management architecture," in ICMU '05: Proceedings of 2nd International Conference on Mobile Computing and Ubiquitous Networking, (Osaka, Japan), 2005.

[79] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, "Firewxnet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments," in MOBISYS '06: Proceedings of the 4th International Conference on Mobile Systems, Applications and Services, (Uppsala, Sweden), pp. 28–41, ACM Press, 2006.

[80] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, "Deploying a wireless sensor network on an active volcano," IEEE Internet Computing, vol. 10, no. 2, pp. 18–25, 2006.

[81] T. V. Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, (Los Angeles, CA, USA), pp. 171–180, ACM, 2003.

[82] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, (Baltimore, MD, USA), pp. 95–107, ACM, 2004.

[83] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated, adaptive sleeping for wireless sensor networks," ACM/IEEE Transactions on Networking, vol. 12, no. 3, pp. 493–506, 2004. A preprint of this paper was available as ISI-TR-2003-567.

[84] T. Zheng, S. Radhakrishnan, and V. Sarangan, "Pmac: An adaptive energy-efficient mac protocol for wireless sensor networks," in IPDPS '05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, (Washington, DC, USA), p. 237.1, IEEE Computer Society, 2005.

[85] "Contiki Operating System." http://www.sics.se/contiki/.

[86] Q. Cao, T. Abdelzaher, T. He, and R. Kravets, "Cluster-based forwarding for reliable end-to-end delivery in wireless sensor networks," in Infocom '07: Proceedings of the 26th Conference on Computer Communications, (Alaska, USA), IEEE, 2007.

[87] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient collision-free medium access control for wireless sensor networks," in SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, (Los Angeles, CA, USA), pp. 181–192, ACM, 2003.

[88] J. Li and G. Y. Lazarou, "A bit-map-assisted energy-efficient mac scheme for wireless sensor networks," in IPSN '04: Proceedings of the third International Symposium on Information Processing in Sensor Networks, (Berkeley, CA, USA), pp. 55–60, ACM, 2004.

[89] B. Scheuermann, C. Lochert, and M. Mauve, "Implicit hop-by-hop congestion control in wireless multihop networks," Elsevier AdHoc Network, vol. 6, no. 2, pp. 260–286, 2008.

[90] T. S. Rappaport and T. Rappaport, Wireless Communications: Principles and Practice (2nd Edition). Prentice Hall PTR, 2001.

[91] B. Z. Ares, C. Fischione, A. Speranzon, and K. H. Johansson, "On power control for wireless sensor networks: System model, middleware components and experimental evaluation," in ECC '07: Proceedings of European Control Conference, (Kos, Greece), 2007.

[92] G. Anastasi, E. Borgia, M. Conti, E. Gregori, and A. Passarella, "Understanding the real behavior of 802.11 and mote ad hoc networks," Elsevier Pervasive and Mobile Computing, vol. 1, pp. 237–256, 2005.

[93] H. Lee, A. Cerpa, and P. Levis, "Improving wireless simulation through noise modeling," in IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks, (Cambridge, MA, USA), pp. 21–30, ACM, 2007.

[94] Q. Cao, T. He, L. Fang, T. Abdelzaher, J. Stankovic, , and S. Son, "Efficiency centric communication model for wireless sensor networks," in Infocom '06: Proceedings of the 25th Conference on Computer Communications, (Barcelona, Spain), pp. 1–12, IEEE, 2006.

[95] "IEEE Standard 802.15.4." http://en.wikipedia.org/wiki/IEEE_802.15.4.

[96] "CC2420 Product Information and Data Sheet." http://www.chipcon.com/.

[97] K. Srinivasan and P. Levis, "RSSI is under-appreciated," in EMNETS '06: Proceedings of the Third Workshop on Embedded Networked Sensors, (Cambridge, USA), 2006.

[98] V. Jacobson, "Congestion avoidance and control," in SIGCOMM '88: Symposium proceedings on Communications architectures and protocols, (Stanford, CA, USA), pp. 314–329, ACM, 1988.

[99] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-aware fair rate control in wireless sensor networks," SIGCOMM Computer Communication Review, vol. 36, no. 4, pp. 63–74, 2006.

[100] G. Wang, G. Cao, and T. L. Porta, "A bidding protocol for deploying mobile sensors," in 11th IEEE International Conference on Network Protocol ICNP '03, pp. 315–324, November 2003.

[101] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," in INFOCOM '04: Proceedings of the 23rd Conference of the IEEE Computer and Communications Societies, (Hong Kong, China), 2004.

[102] C. H. Liu and D. J. Fang, "Propagation. in antenna handbook: Theory, applications, and design," Van Nostrand Reinhold, vol. Chapter 29, pp. 1–56, 1988.

[103] M. Beke and L. Gurvits, "Mobile robot localization using landmarks," IEEE Transactions on Robotics and Automation, vol. 13, p. 2, 1997.

[104] R. Grossmann, "Localization in zigbee-based wireless sensor networks," Tech. Rep. 213, University of Rostock, Germany, April, 2007.

[105] J. S. Esteves, A. Carvalho, and C. Couto, "Generalized geometric triangulation algorithm for mobile robot absolute self-localization," in ISIE 03': IEEE International Symposium On Industrial Electronics, (Rio De Janeiro, Brazil), 2003.

[106] J. Borenstein, H. Everett, and L.Feng, "Where am I? Sensors and methods for mobile robot positioning," tech. rep., The University of Michigan, 1996.

Appendix A

Moser: a Mobile Sensor Relocation Protocol for Mobile Wireless Sensor Networks

In WSNs, sensor node failures can create network partitions or coverage loss which can not be solved by providing reliability at higher layers of the protocol stack. Sensor nodes are prone to failure and may fail for many reasons such as battery depletion, fire or extreme heat, accidentally damage, malicious activity, or simply from extended use, etc. These failures may occur upon deployment or over time after deployment. If the failed nodes are the critical nodes, i.e. cluster heads or bridge nodes, the network connectivity is greatly degraded or even the network may split into two or more disconnected partitions. Moreover, in many applications, repairing this kind of breakdown may not be feasible by use of higher layer protocols unless we are looking at a densely deployed network with large redundancy.

In static WSNs, a common solution for maintaining connectivity and coverage is to deploy redundant sensor nodes. When failed nodes cause the network to be disconnected, redundant nodes can be used for repairing network connectivity. However, deploying the redundant nodes for maintaining network connectivity is an expensive solution because a large number of backup nodes must be deployed together with the actual required sensor nodes, unless the nodes become extremely cheap. Moreover, in many cases it is difficult to ensure that redundant nodes are available for the replace-

ment, especially for a network in which the sensor nodes are randomly deployed.

Using mobility to repair the failed nodes is a potential solution for maintaining the connectivity and coverage. When there are node failures, mobile nodes can be relocated to replace the failed nodes. Mobile nodes can also relocate themselves from a densely deployed area to a sparse area for improving network connectivity. One example of a mobile node is the Robomote [3]. These sensors are smaller than 0.000047 m^3 and cost less than 150 US dollars [3].

In the final part of this thesis, we investigate the problem of maintaining the network connectivity and coverage when the sensor nodes are failed. We propose a distributed protocol for **Mo**bile **Se**nsor **R**elocation problem called **Moser**. We consider a hybrid WSN where a subset of the nodes has the ability to move at a high energy expense. When a node has low remaining energy (dying node) but it is a critical node which constitutes the network such as a cluster head, it will seek a replacement. If a redundant node is located in the transmission range of the dying node and can fulfill the network connectivity and coverage requirement, it can be used for substitution. Otherwise, a protocol should be in place to relocate the redundant sensor node for replacement.

A.1 The Contribution

The primary contribution of this chapter is the **Moser** protocol that assists dying nodes to locate redundant mobile sensor nodes for replacement. Unlike previous works which either present centralized algorithms or rely on the fact that sensor nodes are aware of their true coordinates such as GPS cordinates, the proposed protocol does not require any location information. The Moser protocol works in three phases. In the first phase, the dying node determines if network partition occurs, finds an available mobile node, and asks for replacement by using flooding algorithm. The dying node also decides the movement schedule of the available mobile node based on certain criteria.

The second phase of the Moser protocol involves the actual movement of the mobile nodes to approach the location of the dying node. Finally, when the mobile node has reached the transmission of the dying node, it communicates to the dying nodes and moves to a desired location, where the network connectivity and coverage to the neighbors of the dying nodes are preserved. Our contribution is a light, distributed, scalable, and energy efficient protocol called **Moser** for sensor relocation in the location free environment.

A.2 System Model and Problem Statement

A.2.1 Our Assumptions

We make three assumptions as stated below.

- A mobile sensor node has capability to move in a static 2-dimensional obstacle-free environment [100] [52] [101] [49]. We model the movement of the node in the environment by a point p moving in the plane.
- A mobile sensor node is able to move with a certain orientation (with the compass) in a plane. Mobile node velocity is constant.
- Nodes are able to determine the distances to the neighbors by using Received Signal Strength Indicator (RSSI) values. Let d denote the distance between the the mobile sensor node m to a neighbor s. We have [102],

$$P_r = \frac{P_t}{d^\alpha} \tag{A.1}$$

where P_t is the transmission power, P_r is the received signal power, and α is a constant between 2 and 4, depending on the wireless channel condition. When a transmitter sends a packet using a fixed transmission power, the receiver can obtain the signal attenuation, and therefore, can estimate the distance d between them.

A.2.2 Problem Statement

We consider a hybrid WSN in which sensor nodes are randomly deployed in a two dimensional area A. Each sensor node does not have its absolute location information. Moreover, each sensor node has a transmission range of R_i, and is able to communicate to its neighbors within this transmission range.

There are M redundant mobile sensor nodes (for brevity we call mobile nodes),

$$S_M = \{S_{m_1}, S_{m_2}, \ldots, S_{m_M}\}; \tag{A.2}$$

and N low-energy sensor nodes (for brevity we call dying nodes),

$$S_D = \{S_{d_1}, S_{d_2}, \ldots, S_{d_N}\}; \tag{A.3}$$

The objective is to replace the dying sensor nodes S_D by the mobile nodes S_M within a permitted time constraint with the minimal energy consumption.

A.3 Moser: a Mobile Sensor Relocation Protocol

In this section, we discuss the Moser protocol in detail. Moser consists of three phases. In the first phase, the dying node determines if network partition occurs, finds an available mobile node, and asks for replacement by using flooding algorithm. In the second phase, the selected mobile node moves to the transmission range of the dying node for replacement. Finally, the mobile node establishes the connectivity and coverage to the neighbors of the dying node.

A.3.1 Phase I: Determining the Network Partition and Locating a Redundant Mobile Sensor Nodes

Let us denote S_m as the mobile node from the set S_M, S_d as a low-energy static node from the set S_D that needs to be replaced, and E_{TH} as an energy threshold. When the remaining energy of a node is less than the threshold E_{TH}, it needs to determine if

a replacement is required. A sensor node is required to be replaced if its death creates a partition in the network. Formally, assume that that the dying node S_d has k neighbors $\{S_1, S_2, ..., S_k\}$, node S_d is required to be replaced if there exists a pair of neighbors (S_i, S_j) $(0 \leq i, j \leq k)$ which can not communicate to each other without S_d.

The network partition detection requires a global communication mechanism, because a neighboring node S_i needs to find all the possible routing paths to the neighboring node S_j. To reduce overhead, Moser uses the following heuristic approach. Each node S_i finds its neighbors within n-hop by flooding a **HELLO** message. In addition, during flooding, if an intermediate node receives the **HELLO** message but it currently has low remaining energy, it does not forward the message. This mechanism prevents the case that the S_d finds a routing path that contains a dying node. If there is a n-hop neighboring node of S_i is also a neighboring node of S_d, say S_u $(0 \leq u \leq k)$, the pair node S_i and S_u is "safe". If there exists one or more pairs of neighboring nodes (S_i, S_j) which is not safe, the dying node S_d is required to be replaced. High value of n results in more accurate of the network partition detection, but also high energy consumption.

When the dying node S_d decides that the replacement is required, it broadcasts a **HELP** message to find a mobile node. A mobile node S_m may receive one or more **HELP** messages from different dying nodes, however, it only selects one $S_d \in S_D$ based on a certain criteria and replies with an **ACCEPT** message to the selected node. Similarly, the dying node S_d may receive multiple **ACCEPT** messages from different dying nodes, but it only selects one S_m and sends a **SELECT** message to the selected node. When the mobile node S_m receives a **SELECT** message to confirm the replacement, the movement process will take place.

This **HELP-ACCEPT-SELECT** mechanism is described in detail in the following sub-sections.

A.3.1.1 HELP message

The dying node S_d (energy below E_{TH}) floods a **HELP** message in the network to find a mobile node. A **HELP** message contains the nodeID, the remaining energy, and the approximate available time that S_d can tolerate for replacement (explain in the Section A.3.1.4).

The broadcast **HELP** message is either distance-based or hopcount-based. In the distance-based broadcasting, the **HELP** message is propagated in the network to find the shorest path to a mobile node. In the hopcount-based broadcasting, the **HELP** message is propagated to a mobile node with the least number of hops. When an intermediate node receives a **HELP** message, it stores the distance/hopcount information if the **HELP** message has not been received previously. Otherwise, it will compare the new distance/hopcount information to the previous one, and only forwards the new information if it is smaller than the previous one. This scheme can reduce significant number of messages exchanged in the network when compared to the traditional flooding algorithm in which a received message is re-broadcasted to all neighbors.

In addition, each **HELP** message that is forwarded by an intermediate node is appended with the intermediate nodeID and the distance/hopcount information. In the hopcount-based broadcasting, the hopcount information is incremented by one. In the distance-based broadcasting, the estimated distance is added to the message. This processing is repeated at every intermediate node until the **HELP** message reaches the mobile node. This scheme allows the mobile node to have the routing path information toward the dying node S_d and the estimated travel cost, either the hopcount cost or distance cost.

A.3.1.2 ACCEPT message

Upon the reception of the **HELP** message, the mobile node S_m builds a path table to select the best candidate for replacement. The path table contains the dying

node ID, the estimated cost, and the intermediate nodeIDs toward the dying node. Moreover, it is possible that there are more than one dying nodes in the network. To ensure that the mobile node S_m receives enough **HELP** messages, the mobile node S_m also waits for a small random interval of time before making a decision.

After the timer expires, the mobile node will decide which candidate is selected for replacement. The mobile node S_m selects the node S_d, which has the least hopcount/distance and unicasts an **ACCEPT** message to the selected S_d based on the routing path in the **HELP** message it has received. In the case of tie, the dying node S_d with the lowest nodeID is selected. The **ACCEPT** message contains mobile nodeID, hopcount/distance, and routing path node IDs.

A.3.1.3 SELECT message

The dying node S_d may receive **ACCEPT** messages from different mobile nodes. Among these offers, the dying node S_d only selects the mobile node with the least hopcount/distance. Again, in the case of tie, the mobile node with the lowest nodeID is selected. The dying node S_d then unicasts the **SELECT** message to the selected mobile node.

However, in the worst scenario, all mobile nodes may only reply to a particular dying node S_d, but S_d only selects one S_m. In this case, other mobile nodes are still available while other dying nodes are waiting for an **ACCEPT** message. To overcome this problem, after replying an **ACCEPT** message, the mobile node needs to ensure the acceptance of the offer by maintaining a timer. When the timer expires but the mobile node still does not receive any **SELECT** messages, it realizes that its offer has not been accepted. In this case, it selects the next candidate dying node in the path table and sends an **ACCEPT** message to this node.

A.3.1.4 Ensuring the Replacement Time Constraint

Some applications may require a timely replacement of dying nodes for ensuring the continuous network operation such as target tracking applications [11–13]. These applications dictate time constraints on the replacement process. To ensure the replacement time constraint, Moser requires that the **HELP** message is only propagated within R radius (unit of distance).

Let T_{max} denote the maximum available time for replacement. T_{max} depends on two factors. First, the remaining energy of the dying node determines the left over working life, denoted by t_1. Second, there is the bound on total down time in replacement process t_2 by the application requirement. The maximum response time T_{max} can be estimated by:

$$T_{max} = t_1(remainingenergy) + t_2(downtime)$$

This T_{max} is included in the **HELP** message, **HELP** message is propagated only in the R_{max} radius (unit of distance). Each intermediate node compares the total distance to the R_{max} distance and only forwards the message if the total estimated distance to the intermediate node less than R_{max}, where:

$$R_{max} = k_r * v * T_{max} \tag{A.4}$$

R_{max} is the maximum distance that satisfies the time constraint T_{max} and v is the velocity of the mobile node.

The coefficient k_r is the movement coefficient factor ($0 \leq k_r \leq 1$). T_{max} gives the time estimation if the mobile node moves with the shortest directed distance. Since the location information is not available, the mobile node is expected to move more distances than the ideal shortest distance. The coefficient k_r takes into account of the approximation for an additional time taken by the mobile node to reach the dying node in the absence of location information. The bound on this movement coefficient factor is described in the Section A.3.2.2. It is noticed that R_{max} only applies to the distance-based broadcasting. This R_{max} mechanism ensures that the **HELP** messages are only received by the mobile nodes that satisfy the T_{max} time constraint.

A.3.2 Phase II: Sensor Movement

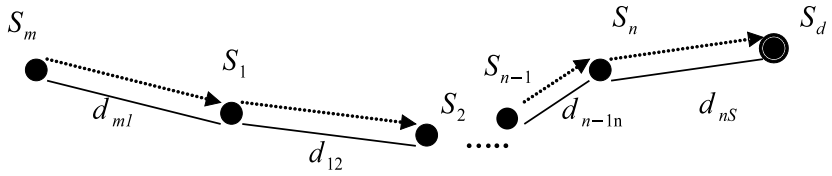


Figure A.1: An Example of a Mobile Node Movement.

Assume that the mobile node S_m needs to move to the location of the dying node S_d. Because the mobile node S_m has the routing path information along with the distance estimates to the dying node S_d (in the **HELP** message), Moser uses these routing path information to assist the mobile node S_m to move to the dying node S_d by following the routing path toward the S_d for the actual movement. For example, in Figure A.1, the mobile node S_m, which is initially within the transmission range of S_1, will first try to approach closer to S_1 so as to establish communication to the node S_2, then try to reach to the transmission range of S_3. This process is repeated until the mobile node S_m can reach to the transmission range of the target node S_d. Once S_m has established the communication with S_d, the goal is to replace the dying node S_d by the mobile node S_m so that the network connectivity and coverage with all of the existing neighbors of S_d are preserved.

However, although the mobile node S_m can measure the distance to its neighbors, it does not know the direction. Moser applies a triangulation based-approach [103] for the sensor movement. By performing additional movements, the mobile node S_m is able to work out the correct direction to the node S_i. The mobile node S_m initially chooses an arbitrary direction and moves a certain distance. It then estimate the new distance to S_i by exchanging a **BEACON** message. These distances form a triangular, which are used to determine the correct direction.

In addition, when S_m moves, it may get out of the transmission range of S_i.

Therefore, the movement algorithm needs not only to ensure that the mobile node S_m to approach the target node with minimum additional movements, but also to ensure the connectivity during movement. Next, we will discuss the movement phase of Moser in detail.

A.3.2.1 The First Stage - Find the Relative Angle to the Target Sensor Node

Node S_m determines the relative angle to the target S_i by moving a certain distance and solving the triangle formed. Let d_0 represent the estimated distance from S_m to S_i, S_m moves dm_1 unit of distance in an arbitrary direction, where,

$$dm_1 = k_1 * Min(d_0, TR_{S_i} - d_0) \tag{A.5}$$

TR_{S_i} is the transmission range of S_i and k_1 is the accuracy coefficient that controls magnitude of distance S_m will move ($0 < k_1 < 1$). The higher the k_1 is, the more accurate is the triangulation formed and the corresponding distance estimates. But higher value of k_1 also results in higher energy consumption in excessive movement for estimating the direction. We will explain how dm_1 is calculated next.

In Figure A.2, S_m, originated at SM_1, will move dm_1 distance with a random direction to the position of SM_2. It then estimates the new distance d_2 to the target node S_i by exchanging a **BEACON** message.

From the Law of Cosines: For a triangle with sides d_0, dm_1, and d_2, the angle α opposite the side d_0 is calculated by,

$$cos\alpha = \frac{{d_2}^2 + {dm_1}^2 - {d_0}^2}{2 * d_2 * dm_1} \tag{A.6}$$

where the angle obtained from measuring $\alpha = \angle(SM_2SM_1, SM_2S_i)$ is the relative angle between two landmarks SM_1 and SM_2.

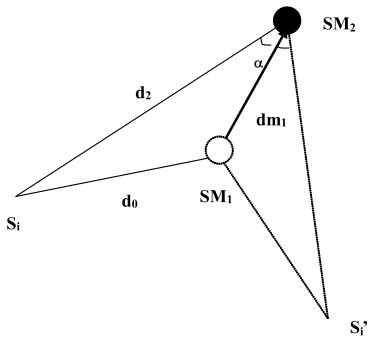


Figure A.2: First Stage - Angle Calculation.

However, although the angle to the target node can be estimated, there is not enough information to find the exact location to the target. In Figure A.2, when the angle is determined, there are still two possible mirror locations of the target S_i and $S_i^{'}$. Therefore, additional movements are required in order to determine the correct direction to the target.

A.3.2.2 The Second Stage - Resolving the Direction

To determine the direction, node S_m will select either of the two direction to S_i or to $S_i^{'}$ and moves dm_2 unit of distance in the selected direction (Figure A.3), where

$$dm_2 = k_2 * Min(d_2, TR_{Si} - d_2) \tag{A.7}$$

k_2 is the accuracy coefficient that controls the movement in this stage ($0 < k_2 < 1$). Coefficient k_1 and k_2 are the tunable accuracy coefficient that depend on the accuracy distance estimates based on the RSSI values. The lower values of k_1 and k_2 can be applied if we have higher confidence in distance estimates while the higher values of these coefficients reflect lower confidence in distance estimates. The more accurate the distance estimate is, the lower the coefficient values are.

The mobile node then exchanges **BEACON** message with the target node and estimates the new distances d_3.

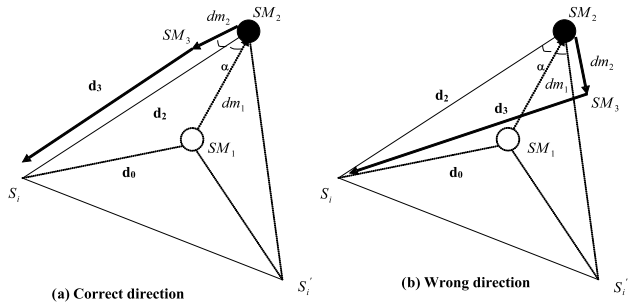


Figure A.3: Second Stage - Direction Calculation.

Both movement cases are considered here. In Figure A.3a, if S_m has moved in the correct direction, the total distance of d_3 and dm_2 is equal to the distance d_2

$$dm_2 + d_3 = d_2 \tag{A.8}$$

In this case, to reach the target, S_m needs to move d_3 distance on the same direction. On the other hand, if the total is greater than d_2

$$dm_2 + d_3 > d_2 \tag{A.9}$$

S_m has selected the wrong initial direction for movement. S_m calculates the new angle to the target node and moves to the destination with the new orientation as in Figure A.3b. It is noted that the total distance is never less than d_2, because of triangle inequality theory.

This algorithm enables both the distance and the direction estimation with triangulation. It is iteratively applied by the node S_m to approach the next hop target node and eventually, reach the target node S_d.

Proposition 4. *If we choose*

$$dm_i = k_i * Min(d_i, (TR_{Si} - d_i)) \tag{A.10}$$

Algorithm 1 Sensor movement algorithm

1: $d_0 = RSSI.measure(S_m, S_i)$;
2: **if** $d_0 >= TR_{Si} - d_0$ **then**
3: $dm_1 = k_1 * (TR_{Si} - d_0)$
4: **else**
5: $dm_1 = k_1 * d_0$;
6: **end if**
7: $angle = rand(360)$;
8: $MS.move(dm_1, angle)$; {First move}
9: $d_2 = RSSI.measure(S_m, S_i)$;
10: **if** $d_2 >= TR_{Si} - d_2$ **then**
11: $dm_2 = k_2 * (TR_{Si} - d_2)$
12: **else**
13: $dm_2 = k_2 * d_2$
14: **end if**
15: $angle = acos((d_2 * d_2 + dm_1 * dm_1 - d_0 * d_0)/(2 * d_2 * dm_1))$;
16: $MS.move(dm_2, angle)$; {Second move}
17: $d_3 = RSSI.measureDist(S_m, S_i)$;
18: **if** $d_3 + dm_2 - d_2 < Error$ **then**
19: angle = last-angle;{correct direction}
20: **else**
21: $angle = acos((d_3 * d_3 + dm_2 * dm_2 - d_2 * d_2)/(2 * d_3 * dm_2))$;{wrong direction}
22: **end if**
23: MS.move(d_3,angle);{Third move}

where $0 < k_i < 1$ *is the accuracy coefficient,* TR_{Si} *is the transmission range of node* S_i *and* $d_i < TR_{Si}$*,* S_m *can never get out of transmission range of* S_i *in our movement algorithm.*

PROOF:

The First Move: Based on triangle inequality, we have:

$$d_2 \leq d_0 + dm_1 \tag{A.11}$$

The longest distance d_2 between S_m and S_i occurs when S_m moves to the opposite direction of S_i. In this case, as $dm_i = k_i * Min(d_0, (TR_{Si} - d_0)) \leq 1 * (TR_{Si} - d_0)$. Thus,

$$d_2 \leq d_0 + (TR_{Si} - d_0) = TR_{Si} \tag{A.12}$$

Therefore, in the first movement, S_m never gets out of transmission range of S_i.

The Second Move: the proof is similar to above.

The Third Move: As this movement does not involve any random movements, S_m never moves out the transmission range of S_i.

Proposition 5. *The movement coefficient factor in equation (3)* $k_r = \frac{1}{d_0+(2*k_1+2*k_2+2*k_1*k_2)*d_0}$

PROOF:

The total distance of movement is:

$$d = dm_1 + dm_2 + dm_3 \tag{A.13}$$

where $dm_1 \leq k_1 * d_0$; and $dm_2 \leq k_2 * d_2$; $0 < k_1, k_2 < 1$

We also have: $d_2 \leq d_0 + dm_1 \leq (1 + k_1) * d_0$ and $dm_3 < d_2 + dm_2 \leq (1 + k_2) * d_2 \leq (1 + k_1) * (1 + k_2) * d_0$

Therefore,

$$\begin{aligned} d &= dm_1 + dm_2 + dm_3 \\ &\leq k_1 * d_0 + k_2 * (1 + k_1) * d_0 + (1 + k_1) * (1 + k_2) * d_0 \\ &= d_0 + (2 * k_1 + 2 * k_2 + 2 * k_1 * k_2) * d_0 \end{aligned} \tag{A.14}$$

In the worst case scenario, mobile sensor S_m may need to move $d_0 + (2 * k_1 + 2 * k_2 + 2 * k_1 * k_2) * d_0$. Therefore, the movement coefficient factor $k_r = \frac{1}{d_0+(2*k_1+2*k_2+2*k_1*k_2)*d_0}$.

In the worst case scenario, mobile sensor S_m may need to move $d_0+(2*k_1+2*k_2+2*k_1*k_2)*d_0$. Therefore, the movement coefficient factor $k_r = \frac{1}{d_0+(2*k_1+2*k_2+2*k_1*k_2)*d_0}$.

A.3.2.3 Special Case Analysis

In some situations, S_m may be located on the boundary of transmission range S_i (see Figure A.4). In this case, an arbitrary movement can get S_m out of transmission range of S_i. Moser protocol handles this case as follows. If S_m can not communicate to S_i after the first movement, S_m needs to get back in the transmission range of S_i. To ensure this, S_m needs to move twice distance of the last movement in the opposite direction (Figure A.4). There are three possible cases after this movement. In Figure A.4a, after turning back, if S_m can communicate to S_i, it proceeds the second stage of finding the direction. However, if S_m still can not communicate to S_i after turning back, S_m is moving on the tangent line of the circle. In this case, S_m needs to determine the center of the circle, at which the target S_i is located. In Figures A.4b and A.4c, this stage is similar to the second stage of finding the direction, as S_m can calculate the angle to the target. However, the movement distance in this case should be large enough so that S_m will be in the transmission range of S_i if it moves on the correct direction. S_m moves a certain unit of distance in the selected direction and exchanges the **BEACON** message to estimate the distance to S_i. If S_m can communicate to S_i, it is moving on the correct direction (Figure A.4b). In this case, it will keep moving on the selected direction to reach the target S_i. Otherwise, S_m is moving on the wrong

direction. In this case, it then needs to turn back the previous position and then move to the other direction (Figure A.4c).

Analysis for the Special Case We need to find out the minimum distance of each movement. In the first move, we choose $dm_1 = k_1 * Min(d_0, (TR_{Si} - d_0))$. As the distance in the second move is twice of the distance in the first move,$dm_2 = 2 * dm_1$.

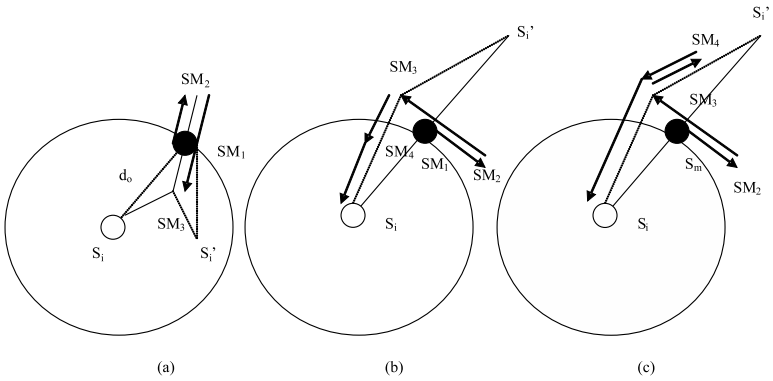


Figure A.4: Special Case.

In the third move, we need to make sure that S_m is in the transmission range of S_i if it chooses the correct direction. In Figure A.4b, as $\Delta SM_1SM_3S_i$ is the right triangle, $SM_3S_i = \sqrt{(SM_1S_i)^2 + (SM_1SM_3)^2} \leq \sqrt{(d_0)^2 + (k_1 * d_0)^2} = \sqrt{1 + k_1^2} * d_0$ Therefore, $dm_3 = SM_3SM_4 \geq (\sqrt{1 + {k_1}^2} - 1) * d_0$.

A.3.2.4 Optimization

The **BEACON** message exchanges the content list of all the nodes in the path to the dying node S_d. A node that found its nodeID listed in the **BEACON** message needs to reply back. In some cases when S_m is moving toward to S_i, it may get into the transmission range of another node S_k in the routing path toward to S_d where $k > i$. In this case, it can skip the rest of movement for the current target node, and starts moving toward to the next target node. This shortcut will result in less movement.

A.3.3 Phase III: Establishing the Network Connectivity and Coverage to the Neighbors

To avoid network partitioning, our goal is to replace the dying node S_d by the mobile node S_m so that the communication to all the existing neighbors of S_d is preserved. Unlike the previous phase in which S_m only tries to approach the transmission range of the target sensor, S_m also needs to maintain the connectivity to the neighbors of S_d. Our approach is that when S_m gets into the transmission range of S_d, S_m will request the list of neighborIDs of S_d by exchanging a **NEIGHBOR** message to S_d and check the connectivity to all the neighbors in the list by broadcasting a **HELLO** message. If S_m can not communicate to any node in the neighbor list, S_m needs to perform the movement again to get closer to S_d. Moreover, to preserve the coverage of S_d, S_m also needs to approach as close to S_d as possible. Thus, if the connectivity of S_m to all neighbors of S_d has not established yet or if the distance between them is still large, S_m needs to perform the movement again to reach closer to S_d. The movement is repeated until both constraints are satisfied.

A.3.4 Practical Considerations

In realistic scenarios, the distance estimation based on RSSI information may not be accurate, which may effect on the movement decision, because the distance and direction are required to calculate in each step of movement. We propose a heuristic solution in the movement phase to ensure the mobile node can reach the destination successfully. Our heuristic solution works as follows. At each step of movement, if the triangle inequality theory is violated (because of the error in distance estimation), a mobile node needs to move to a location that the error in distance estimation is smaller, and thus, distance and direction can be estimated. Assume that S_m, originated at the location of SM_1 (which is d_0 distance from S_i), moves to the location of SM_2 and estimate the new distance d_2 to S_i. As shown in Figure A.5, the angle can not be

calculated after this movement, as the triangle inequality is not satisfied. There are two cases. If $d_2 + dm_1 < d_0$, the error in d_0 estimation is the dominant factor. To overcome this error, the sum of $d_2 + dm_1$ should be increased. Thus, S_m will move dm'_1 more in the same direction (Figure A.5a). If $d_0 + dm_1 < d_2$, the distance d_2 should be reduced to overcome the large error in the d_2 estimation. S_m thus needs to move back dm'_1 distance in the opposite direction (Figure A.5b). The process is repeated until the triangle inequality theory is satisfied. The magnitude of distance dm'_1 is randomly assigned in the range of (0, dm_1). It is noted that $d_2 + d_0 < dm_1$ never occurs, because dm_1 is always less than d_0. The heuristic is also applied to the other stages of the movement phase.

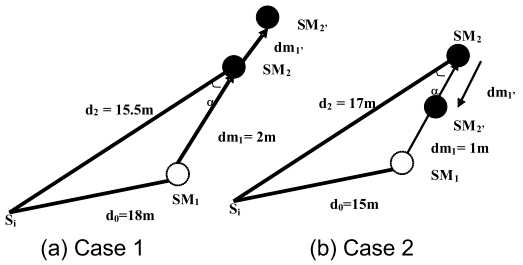


Figure A.5: Overcome the errors in RSSI.

A.4 Simulation Results

A.4.1 Simulation Setup

In order to evaluate the performance of the Moser protocol, we conducted the simulations of 100 nodes are randomly deployed in a 100m * 100m square region. The simulations were run in the Discrete Event Simulator (ns2) [63]. Although Moser protocol allows different transmission ranges, for simplicity, in these simulations, we set the transmission range of all the sensor nodes to be 20 meters. 20 redundant mobile nodes are randomly selected. The speed of the mobile node is 2 m/s. Base on [3], the

calculation of energy consumption per meter is 8.275 J/m. We randomly chose 10 static nodes, depleted their energies, and started the replacement process. The simulation was repeated 10 times and we calculated the average results.

A.4.2 Simulation Results

A.4.2.1 Ideal Conditions - No Error in RSSI Estimation

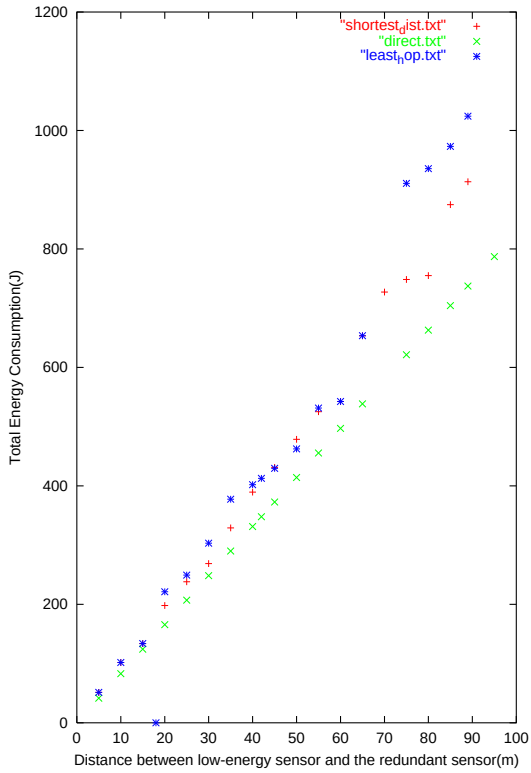


Figure A.6: Total Energy Consumption, k=0.1

Although the shortest directed movement is not possible in our case, we use it as the baseline comparison. We compare the energy consumption between the shortest directed movement and Moser protocol. We simulated two different cases when accurate coefficient $k_1 = k_2 = 0.1$, and $k_1 = k_2 = 0.5$, respectively.

Figure A.6 and Figure A.7 show the total energy consumption versus the distance between the mobile node and the dying node. First, we observe that the mobile nodes

can reach to the location of the dying nodes successfully in all of the simulations, which validate Moser protocol. For $k_1 = k_2 = 0.1$, the energy consumption of Moser prtocol is very similar to the shortest direct movement. The results show that Moser is energy-efficient. For $k_1 = k_2 = 0.5$, the differences in energy-consumption between the shortest direct movement and Moser are significant when the distance between them is ≥ 50 meters (Figure A.7). Further, the results show that total energy consumption in the shortest path route is likely better than the one in the least hop route in most of the cases.

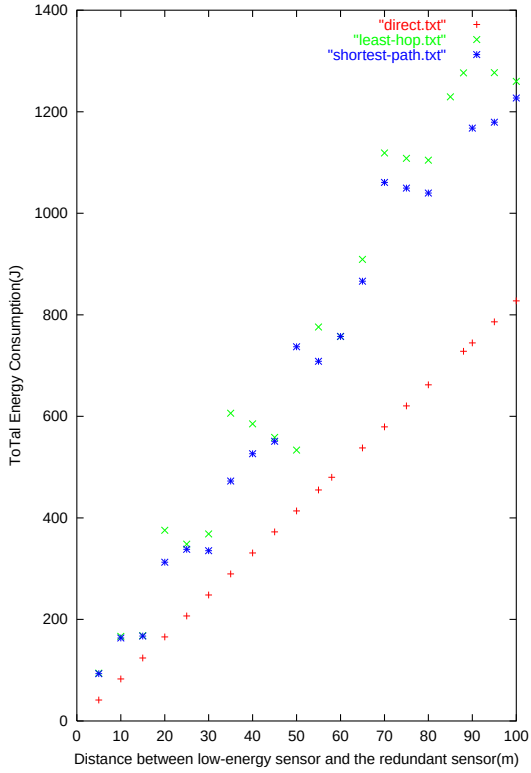


Figure A.7: Total Energy Consumption, k=0.5

A.4.2.2 Practical Considerations

In realistic scenarios, the RSSI distance estimation may not be accurate because the propagation of the radio signal is interfered with a lot of influencing effects such

as reflections on metallic objects, superposition of electro-magnetic fields, diffraction at edges, refraction by media with different propagation velocity, polarization of electro-magnetic fields, etc [104]. These effects degrade the quality of the determined RSSI significantly. To incorporate these errors, we use the following error model.

$$d = d * (1 \pm e\%) \tag{A.15}$$

where e represents the noise in RSSI distance measurement ($0 \leq e \leq 1$).

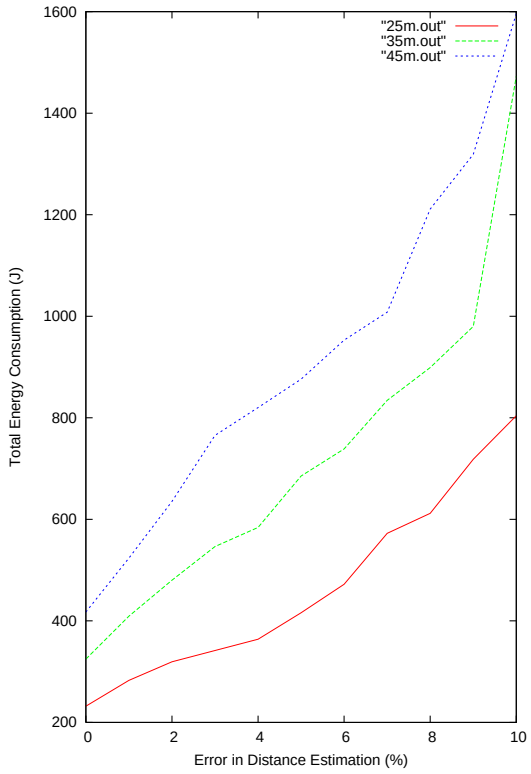


Figure A.8: Total Energy Consumption with the Error in Distance Estimation.

Figure A.8 shows the total energy consumption in the presence of the error in distance estimation for different distances between mobile nodes and dying nodes such as 25 meters, 35 meters, and 45 meters, respectively. In each distance estimation d, there contains error e ($d = d * (1 \pm e\%)$), which is randomly generated between 0% and 10%. The results show that the mobile nodes always reach the location of the dying

nodes successfully. As the error in distance estimation increases, the mobile nodes consume more energy for movement. When the error is 10%, the mobile node consumes about three to four times energy consumption when compared to that of no error in distance estimation. Thus, large error in distance estimation require addition energy consumption on movement.

A.5 Related Work

In static WSNs, the common solution for maintaining connectivity is to deploy redundant sensor nodes. When sensor nodes fail or the network is disconnected, the redundant nodes can be used for repairing connectivity [46] [47] (see Section 2.3.1). However, deploying redundant nodes for maintaining network connectivity is an expensive solution because a large number of backup nodes must be deployed together with the actual required sensor nodes. Moreover, in many cases it is difficult to ensure that redundant nodes are available for replacement, especially for a network in which the sensor nodes are randomly deployed.

Using mobility for maintaining connectivity has been discussed in [48] [49] [50] [52] [53] [54] (see Section 2.3.1). When there are node failures, mobile nodes can be relocated to replace the failed nodes [48] [49] [50]. Mobile nodes can also relocate themselves from a densely deployed area to a sparse area for improving network connectivity [52] [53]. However, these works require all the sensor nodes in the network are mobile. This requirement is expensive and may not suit to practical WSN deployments. Another solution has been proposed in [54] in which mobile nodes are used as data carriers and forward data between disconnected components of the network to the base station. In contrast, Moser protocol is designed for a hybrid network, which consists of static and mobile nodes. Moreover, the existing works require knowledge of accurate location information and do not discuss how the mobile sensor nodes moving to the dying nodes. Therefore, Moser is more flexible.

Moser protocol uses the triangular approach for assisting the mobile nodes moving to a target location. The triangular approach was used in the previous works [103] [105] [106], which aimed to find a position bearing to landmarks with known positions for localization. However, these works use triangular approach to find the exact location of a robot and require the anchor nodes. They are different from our work, in which we use triangular approach to assist the movement of the mobile sensor nodes. Moreover, Moser protocol also ensures that the sensor nodes do not get out of their transmission range and take the energy efficiency into account.

A.6 Summary

We proposed a protocol called Moser for mobile sensor relocation problem. Moser is a light, distributed, and energy efficient, which is applicable for many applications such as chemical leak monitoring or mining detection, where human intervention is restricted. Unlike existing solutions using the assumptions of precise location information is available; Moser can perform without any localization algorithm needed. The simulation results show that the Moser performs very close to the shortest directed movement in low error conditions and achieves considerably good performance under noisy conditions. This work although handles a very important aspect of reliability, it is peripheral to the reliability protocol work described in rest of this thesis and hence has been added as an appendix.

Future work in this area is the implementation of the protocol. To address the practical challenges of RSSI measurement, the experimental study of the relationship between RSSI and distance may be investigated and used in the Moser protocol.