



**ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES  
COLLEGE OF NATURAL SCIENCES  
DEPARTMENT OF COMPUTER SCIENCE**

***Railway Infrastructure Health Monitoring Using Wireless Sensor  
Networks***

**Yemisrach Asnake**

**A THESIS SUBMITTED TO  
THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE IN COMPUTER SCIENCE**

**April, 2015**

**ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES  
COLLEGE OF NATURAL SCIENCES  
DEPARTMENT OF COMPUTER SCIENCE**

*Railway Infrastructure Health Monitoring Using Wireless Sensor  
Networks*

**Yemisrach Asnake**

**Advisor: Mulugeta Libsie (PhD)**

**APPROVED BY:**

**BOARD OF EXAMINERS:**

**Name**

**Signature**

1. Dr. Mulugeta Libsie, Advisor

\_\_\_\_\_

2. Dr. Fekade Getahun

\_\_\_\_\_

3. Dr. Mesfin Kifle

\_\_\_\_\_

## **Acknowledgement**

I would like to take this opportunity to first and foremost thank God for being my strength and guide in the writing of this thesis. Without Him, I would not have had the wisdom or the physical ability to do so. This thesis appears in its current form due to the assistance and guidance of several people. I would therefore like to offer my sincere thanks to all of them.

I express my gratitude to my thesis advisor Dr. Mulugeta Libse. In his capacity as my advisor, Dr. Mulugeta, whose work ethic is worth modeling, closely followed my progress corrected and improved the manuscript with valuable comments. I would not hesitate to state that the merit that is in this paper is on account of the supervision of my advisor. I am deeply grateful for the encouragement and help offered by him.

I am forever grateful to my parents for their love, prayers, caring and sacrifices for educating and preparing me for my future. Without their patience and timeless assistance, much less would have been possible. Also I extend my thanks to my sister, brothers and God mother.

Finally, my heartfelt thanks goes to Ato Gashaw Mengistu and Tsion Elias for their assistance, encouragement, cooperation and friendship.

## Table of Content

<b>List of Tables</b> .....	<b>i</b>
<b>List of Figures</b> .....	<b>ii</b>
<b>List of Algorithms</b> .....	<b>iii</b>
<b>List of Acronyms</b> .....	<b>iv</b>
<b>Abstract</b> .....	<b>v</b>
<b>Chapter One: Introduction</b> .....	<b>1</b>
1.1 Background.....	1
1.2 Statement of the Problem.....	2
1.3 Objective.....	3
1.3.1 General Objective .....	3
1.3.2 Specific Objectives .....	4
1.4 Methodology.....	4
1.4.1 Literature review .....	4
1.4.2 Fact Finding .....	4
1.4.3 Software and Hardware Tools.....	5
1.4.4 Design and Simulation.....	6
1.5 Scope and Limitation .....	7
1.6 Application of Results.....	7
1.7 Organization of the Thesis .....	8
<b>Chapter Two: Literature Review</b> .....	<b>9</b>
2.1 Wireless Sensor Networks .....	9
2.2 Wireless Sensor Nodes.....	10
2.3 Applications of Wireless Sensor Networks.....	12
2.4 Operating System.....	14
2.5 Wireless Networking .....	15
2.5.1 Communication Structure of a Wireless Sensor Network .....	15
2.5.2 Physical Layer.....	17
2.5.3 Medium Access Control (MAC) Layer.....	18
2.5.4 Network Layer Protocols .....	19
2.6 Clustering for Data Aggregation.....	20
2.6.1 Low Energy Adaptive Clustering Hierarchy (LEACH).....	21

2.6.2	Hybrid Energy-Efficient Distributed Clustering (HEED).....	22
2.7	Railway System .....	24
2.7.1	Railway in Ethiopia.....	24
2.7.2	Addis Ababa Light Railway Transport (LRT).....	24
2.7.3	Railway Security.....	26
2.7.4	Supervisory Control and Data Acquisition (SCADA) System .....	27
2.8	Summary.....	29
<b>Chapter Three: Related Work.....</b>		<b>30</b>
<b>Chapter Four: Proposed Solution .....</b>		<b>36</b>
4.1	Introduction.....	36
4.2	System Architecture.....	36
4.3	Railway Infrastructure Monitoring .....	38
4.3.1	Railway Infrastructure Monitoring (RAIM) Application.....	40
4.4	Data store and Access Tools .....	43
4.5	Summary.....	44
<b>Chapter Five: Prototype Implementation .....</b>		<b>45</b>
5.1	Introduction.....	45
5.2	Development and Simulation Tools.....	45
5.3	Prototype Implementation.....	46
5.3.1	Implementation Detail of Main Components of the application.....	48
5.3.1.1	Routing Protocol .....	48
5.3.1.2	Sensing.....	52
5.3.1.3	Collecting.....	54
5.4	Data Management .....	55
<b>Chapter Six: Experiment and Results .....</b>		<b>56</b>
6.1	Simulation Setup.....	56
6.2	Simulation process .....	57
6.3	Evaluation and Results.....	58
6.4	Node Information.....	62
6.5	Evaluation of Routing Protocol.....	62
6.6	Summary.....	64

<b>Chapter Seven: Conclusion and Future Work .....</b>	<b>65</b>
7.1 Conclusion .....	65
7.2 Future Works .....	66
<b>References .....</b>	<b>67</b>

## List of Tables

<i>Table 1.1: Evaluation metric for Railway Infrastructure monitoring.....</i>	<i>7</i>
<i>Table 5.1: Example result of routing protocol output.....</i>	<i>50</i>
<i>Table 6.1: Simulation parameters that are used for this work.....</i>	<i>56</i>
<i>Table 8.1: Node information with selected simulated value of each node.....</i>	<i>62</i>

## List of Figures

<i>Figure 2.1: Components of a Sensor Node</i> .....	11
<i>Figure 2.2: Architectural Layers of a WSN</i> .....	16
<i>Figure 2.3: East-West Transit of the AALRT Line</i> .....	25
<i>Figure 2.4: South-North Transit of the AALRT Line</i> .....	26
<i>Figure 4.1: Proposed System Architecture</i> .....	37
<i>Figure 4.2: Software Tools for the application</i> .....	39
<i>Figure 5.1: Component Graph</i> .....	47
<i>Figure 5.2: Schedule Frame Format</i> .....	49
<i>Figure 5.3: Data Frame Format</i> .....	49
<i>Figure 5.4: Routing Protocol Simulation with Six Motes</i> .....	52
<i>Figure 5.5: Simulation of Sensing Component</i> .....	53
<i>Figure 5.6: Simulation of Collecting</i> .....	55
<i>Figure 5.7: Sensor Database</i> .....	55
<i>Figure 6.1: Simulation Process Flow</i> .....	57
<i>Figure 6.2: Packets Received Over Time</i> .....	58
<i>Figure 6.3: Estimated Result of Packets Lost</i> .....	59
<i>Figure 6.4: Received and Duplicated Packet per Node</i> .....	60
<i>Figure 6.5: Latency Result of Each Node</i> .....	60
<i>Figure 6.6: Average Power Consumption</i> .....	61
<i>Figure 6.7: Duty Cycle of the Routing Protocol</i> .....	64

## List of Algorithms

<i>Algorithm 4.3: Pseudo Code for Mobile Node Advertisement</i> .....	42
<i>Algorithm 4.4: TDMA Scheduling</i> .....	42
<i>Algorithm 4.5: Pseudo Code for Routing Protocol</i> .....	43

## **List of Acronyms**

AALRT: Addis Ababa Light Railway Transport

CH: Cluster Head

CIP: Critical Infrastructure Protection

CSMA-CA: Carrier Sense Multiple Access with Collision Avoidance

HEED: Hybrid, Energy-Efficient, Distributed clustering protocol

LEACH: Low Energy Adaptive Clustering Hierarchy

mA: MilliAmp

MAC: Medium Access Control

MEMS: Micro Electro-Mechanical System

MNRP: Mobile Node Routing Protocol

QoS: Quality of Service

RAIM: Railway Infrastructure Monitoring

RF: Radio Frequency

SCADA: Supervisory Control and Data Acquisition

SF: Serial Forwarder

SHM: Structural Health Monitoring

TDMA: Time Division Multiple Access

TinyOS: Tiny Operating System

WSNs: Wireless Sensor Networks

WiFi: Wireless Fidelity

## **Abstract**

The continuous miniaturization process of computing devices featuring wireless technologies influences our everyday life. With the popularity of laptops, cell phones, PDAs, GPS devices, RFID, and intelligent electronics in the post-PC era, computing devices have become cheaper, more mobile, more distributed, and more pervasive in daily life. The emergence of wireless sensor networks (WSNs) is essentially the latest trend of Moore's Law toward the miniaturization and ubiquity of computing devices.

Wireless Sensor Networks are composed of devices of reduced size, self-powered and with wireless transmission capabilities. Because of these features this technology has been recognized as promising for a large variety of monitoring and surveillance applications.

In this thesis, we presented the details of an application that makes use of WSNs to monitor railway infrastructures. The WSN collects information about the structural health and behavior of the infrastructure when a train travels along it and relays the readings to a base station. The base station uses the next train as a data carrier to upload the information. The use of a train as a data carrier is especially suitable to collect information from remote or inaccessible places which do not have a direct connection to the internet.

We have implemented a railway infrastructure monitoring application and its application specific routing protocol on top of TinyOS operating system using nesC programming. We have also implemented data management method to make sure the sampled data is being stored for later purpose We have evaluated the performance of our application through simulation using COOJA simulator on TelosB sensor nodes. COOJA is the default simulator for Contiki operating system which can be used for TinyOS applications. Finally, we demonstrated the feasibility of our application and routing protocol. The experiment showed encouraging results.

### **Keywords:**

Wireless Sensor Networks, critical infrastructure protection, infrastructure health monitoring, Railway Infrastructure Monitoring, Sensors, Routing Protocol, Simulation

# Chapter One: Introduction

## 1.1 Background

Recent advances in wireless communications and electronics have enabled the development of low-cost, low-power, small-size, and multi-functional sensor nodes. These sensors consist of a microprocessor, a few kilobyte of RAM, a short-range radio transmitter, a small power source (e.g., a battery), and a few sensors to interact with the environment [1, 2]. Such tiny sensor nodes, which cooperate on sensing different physical phenomena, have led to the appearance of wireless sensor networks (WSNs). In [3] it is stated that sensor networks hold the promise of revolutionizing sensing in a wide range of application domains because of their reliability, accuracy, flexibility, cost effectiveness, and ease of deployment.

There are a great number of applications where WSNs can be applied and their advantages have long been acknowledged in the research community. Their main application domain is the monitoring and controlling of large scenarios. Applications where WSNs have been applied include pollution detection, agriculture, health monitoring, structural health monitoring, intrusion detection, motion tracking, among many others [4].

From the many applicable application areas Structural Health Monitoring (SHM) is a relatively new scientific field. SHM using wireless sensor networks has drawn considerable attention in recent years. The ease of deployment of tiny wireless devices that are coupled with sensors and actuators enhances the data collection process and makes prognostic and preventive maintenance of an infrastructure much easier [5].

Structural Health Monitoring (SHM) is a technology that allows the estimation of the structural state and detection of structural change that affects the performance of a structure. Two discriminating factors in SHM are the time-scale of the change (how quickly the state changes) and the severity of the change. These factors represent two major sources of system change: alarm warnings (e.g. disaster notification for earthquake, explosion, etc.) and continuous health monitoring (e.g. from ambient vibrations, wind, etc.). The general approaches taken to SHM are

either direct damage detection (visual inspection, x-ray, etc.) or indirect damage detection (detecting changes in structural properties or system behavior) [6, 7].

SHM systems lead to the function of safety maintenance and extending the life time of real time buildings and bridges. In general structural monitoring includes the change of structures in properties at the geometrical and material level. In addition to that structural monitoring system should reduce the energy consumption. Damage monitoring in any civil structures includes, corrosion monitoring, crack monitoring, vibration and strain measurement [6].

One of the critical parts of many nations' infrastructure is a railway system. Currently Ethiopia is constructing a railway transport system, which is considered as a solution for the existing transportation problem. Railway infrastructures, as any other kind of infrastructure, are affected by the aging process. Therefore, accuracy and reliability of such infrastructure inspections are imperative, and advanced quantitative methods to locate the damage are of pressing importance.

In many other nations modern security systems used by infrastructure protection applications include a set of different sensing technologies integrated by appropriate management systems. Such systems are highly dependent from human operators for supervision and intervention. One of the challenging goals of the research community in this field is the automatic detection of both natural and malicious threat scenario. Therefore to monitor the safety of a railway it is important to adopt an automated monitoring system of both natural and malicious threats. For such cases WSNs introduced an innovative way of monitoring and interacting with such environment [8].

This thesis aims to employ wireless sensor network technology for the continuous health monitoring of railway infrastructure.

## **1.2 Statement of the Problem**

WSNs have been identified as having the potential to become an integral part of the protection of critical infrastructures (CIP). A railway infrastructure health monitoring system using WSNs should consist of wireless sensor nodes which have high amount of energy and processing power, a technique for data collection, routing, energy efficient data processing and data management techniques to take appropriate action.

On the other hand, research today mainly focuses on sensor processing, data management, safety analysis and visualization [9, 10], development, optimization and improvement of physical, Medium Access Control (MAC) and routing layer issues, parameters adjustment, in order to minimize the energy consumption and maximize the lifetime [11].

These results indicate the increasing need for QoS (Quality of Service) support. These are the most important challenges that need to be tackled while developing any WSN application.

This thesis aims to investigate QoS (in our case the reliability, integrity and deadline are the necessary parameters) that hinder WSNs from becoming widely used; reliability is the ability of the network to ensure reliable data transmission between nodes checking whether there are lost packets or checking the amount of packets received overtime or per node. Information in CIP needs to be reliably transmitted to make sure that the possible error or important notifications generated in the system reach the desired destination. The integrity parameter is about the absence of improper system alteration. It is obvious that integrity is expected in CIP. In fact, maintaining the integrity of a critical infrastructure is one of the main functions of the CIP system. The other QoS requirement is deadline where it defines a maximum length of time in which some sort of task must be accomplished, i.e., maximum latency. All this can be achieved by designing a system that monitors the health of the railway infrastructure using a wireless sensor network where the wireless sensor nodes perform real-time infrastructure monitoring, communication and synchronization, in-network processing in a distributed fashion along with the techniques like data management. All these are carried out in this work.

## **1.3 Objective**

### **1.3.1 General Objective**

The general objective of this thesis is to propose a way of monitoring a railway infrastructures' health using WSNs.

### **1.3.2 Specific Objectives**

The specific objectives are:

- Study the works and current practices being undertaken in the area of monitoring critical infrastructure problems (CIP) especially railway systems.
- Model the framework of railway infrastructure monitoring system using a wireless sensor network.
- A routing protocol for WSNs that support a many to many communication pattern, and that can handle QoS requirements (reliability, deadline and integrity) and energy efficiency.
- Simulate the approach
- Test validity of the application

## **1.4 Methodology**

In order to achieve the objective of the study, the following methods are employed.

### **1.4.1 Literature review**

We have reviewed published related works and also literatures to gather information about the study area relevant to this research work.

### **1.4.2 Fact Finding**

In order to design a railway infrastructure monitoring system based on a wireless sensor network, it is important to understand the structural formation of the railway infrastructure, that is, how many spans could be found on the railway bridge and the distance between them since our application scenario is based on the railway bridge as it is one of the infrastructures of a railway system. We achieve these tasks by interviewing civil engineers and engineers working on the construction site of the railway infrastructures

### 1.4.3 Software and Hardware Tools

Software and hardware tools, those that are relevant for development and simulation, have been identified and studied.

**TinyOS:** is a free and open source component based operating system and platform targeting Wireless Sensor Networks (WSNs) and it is an embedded operating system written in the nesC programming language. It is an operating system environment designed to run on embedded devices used in distributed wireless sensor networks [44].

**nesC:** is a component based, event driven programming language used to build applications for the TinyOS platform. It is built as an extension to the C programming language with components “wired” together to run applications on TinyOS [45].

**COOJA:** The COOJA simulator is an event based simulator that uses a combination of Java code for the front-end interface and platform specific emulators to carry out the simulations. COOJA allows for cross-level simulation, a novel type of wireless sensor network simulation that enables holistic simultaneous simulation at different levels. In COOJA one simulation can contain nodes from several different abstraction levels. These are the network level, the operating system level, and the machine code level. We demonstrate a few different cross-level simulation scenarios using the COOJA simulator [47].

**Telosb:** Crossbow’s TelosB mote is an open source platform designed to enable cutting edge experimentation for the research community. The TelosB mote bundles all the essentials for lab studies into a single platform including: USB programming capability, an IEEE 802.15.4 radio with integrated antenna, a low-power MCU (Micro Controller Unit) with extended memory and an optional sensor suite (TPR2420). TPR2400 offers many features, including [46]:

- IEEE 802.15.4/ZigBee compliant RF transceiver
- 2.4 to 2.4835 GHz, a globally compatible ISM band
- 250 kbps data rate
- Integrated onboard antenna
- 8 MHz TI MSP430 microcontroller with 10kB RAM

- Low current consumption
- 1MB external flash for data logging
- Programming and data collection via USB
- Optional sensor suite including integrated light, temperature and humidity sensor (TPR2420)
- Runs TinyOS 1.1.10 or higher

#### **1.4.4 Design and Simulation**

##### **Wireless sensor node simulation design**

This is the objective of wireless sensor network based railway infrastructure monitoring system. To accomplish this task there are two alternatives. The first alternative is designing the sensor node or use on-the-shelf sensor. The second alternative is to model the wireless sensor node using simulation tools. Currently, since it is not possible to get (buy) the wireless sensor node (the hardware) in the local market, the second option of modeling the node using simulation tool is to be used.

##### **Explore Communication and synchronization protocol**

Since the wireless sensor nodes used in the application need to communicate and synchronize with each other, they have to support IEEE 802.15.4/Zigbee implementation. So there is a need to explore the network topology and protocols.

##### **Simulation and performance evaluation**

After the system is implemented, its proper functioning is checked using the simulation tools with different values of parameters to be tested. In order to validate the capability and effectiveness of the designed wireless sensor network based railway infrastructure monitoring system, there must be an evaluation metrics. There are evaluation metrics that will be used to evaluate the wireless sensor network and also to evaluate the performance of individual nodes. Evaluation metrics are dependent on the high-level objective of the application [48]. Thus, based

on our application objective, we will use the evaluation metric that are listed in Table 1.1 for our application.

*Table 1.1: Evaluation Metric for Railway Infrastructure Monitoring*

<b>Evaluation Metric</b>	<b>Description</b>
Average Power consumption	It is the average of consumed energy among all the nodes in the Network
Received and Lost Packet	It is the metric to measure the packets received and lost over time
Latency	It is performance metric is used to measure the average end-to-end delay of data packet transmission.
Received packets per node	It is to measure the number of packets received per node (to check it there are duplicates or not)

## **1.5 Scope and Limitation**

The main focus of this thesis is to apply WSN technology to the CIP problem, more specifically the monitoring of railway infrastructures. It only focuses on the monitoring process that is, monitoring the physical properties of the infrastructure. This thesis does not include the monitoring of the impact of the external events on the infrastructure.

## **1.6 Application of Results**

Railway systems are critical in many regions and it is important to have a system to monitor the health of these infrastructures and report when and where maintenance operations are needed.

The result of the work presented in this thesis is to provide a system to monitor the health of railway infrastructure using WSNs, by providing reliable information about the real behavior of the infrastructure, its condition and changes to enable intelligent decision making, and damage detection at early stage. Thus immediate repair and corrective actions can be made before any collapse occurs and allow avoiding considerable risk of further damage. It tackles the transfer of large quantities of data in a reliable manner.

## **1.7 Organization of the Thesis**

This thesis work comprises of Seven Chapters. Chapter 2 covers the study of a WSN, the architecture, the protocols and applications, about railway, railway in Ethiopia, railway types and monitoring techniques to be applied. In Chapter 3 we deal with reviews of related works.

Chapter 4 deals with the proposed solution, and the main design issues and Chapter 5 discusses the prototype implementation and evaluation are discussed such as development and simulation tools that are used for this work and the result of the thesis is presented. Finally, Chapter 6 explains about the experiment and results. Finally Chapter 7 summarizes the thesis by proposing the recommendations and future works.

## **Chapter Two: Literature Review**

### **2.1 Wireless Sensor Networks**

The continuous miniaturization process of computing devices featuring wireless technologies influences our everyday life. With the popularity of laptops, cell phones, PDAs, GPS devices, RFID, and intelligent electronics in the post-PC era, computing devices have become cheaper, more mobile, more distributed, and more pervasive in daily life [9, 10]. The emergence of wireless sensor networks (WSNs) is essentially the latest trend of Moore's Law toward the miniaturization and ubiquity of computing devices.

Researchers see WSNs as an exciting emerging domain of interconnected node devices that sense and exchange the data gathered from the natural environment, to help reveal and better understand the perceptual properties of the world around us. The natural principles are governed by the laws of physics and mathematics; however, WSNs allow us to observe and monitor these parameters upfront using electronic hardware and physical sensors [10].

Because WSNs are composed of devices of reduced size, self-powered and with wireless transmission capabilities they have been recognized as a promising technology for a large variety of applications. Moreover, sensor networks hold the promise of revolutionizing sensing in a wide range of application domains because of their reliability, accuracy, flexibility, cost-effectiveness, and ease of deployment [11].

As with many other technologies, the military has been a driving force behind the development of wireless sensor networks. In the recent years the fields like environmental studies, medicine, agriculture, industry, etc., have also manifested a high level of distinctive trend for the support of complex processes with various sensor data.

In order to enable wireless sensor network applications using sensor technologies, the range of tasks can be broadly classified into three groups; the system, the communication protocol and service. Each sensor node is an individual system and in order to support different applications on a sensor system, development of new platforms and operating systems are needed. The communication protocols are the ones that enable communication between the applications and sensors. They also enable communication between the sensor nodes. Finally the network services

are the ones that will help in enhancing the application and improve system performance and network efficiency. So going further we will look into these three common groups in detail.

## **2.2 Wireless Sensor Nodes**

Wireless sensor nodes are the central element in a wireless sensor network. A sensor node, also known as a mote, is a node in a wireless sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network [12].

When a WSN application development is underway, it is first and foremost essential to select the hardware components to be equipped in the wireless sensor node. The application's requirements are most of the time with regard to size, costs and energy consumption of the nodes communication facilities as such are often considered to be of acceptable quality, but the trade-offs between features and costs is crucial.

The nodes are made up of four basic components: a sensing unit, processing unit, transceiver unit and power unit. They may also have application dependent additional components such as a location finding system, a power generator and a mobilizer [1]. Sensing units are usually composed of two subunits: sensors and analog to digital converters (ADCs). The analog signals produced by the sensors based on the observed phenomenon are converted to digital signals by the ADC, and then fed into the processing unit. The processing unit, which is generally associated with a small storage unit, manages the procedures that make the sensor node collaborate with the other nodes to carry out the assigned sensing tasks. A transceiver unit connects the node to the network. One of the most important components of a sensor node is the power unit. There are also other subunits, which are application dependent. Typical wireless sensor node components are shown in Figure 2.1 [1].

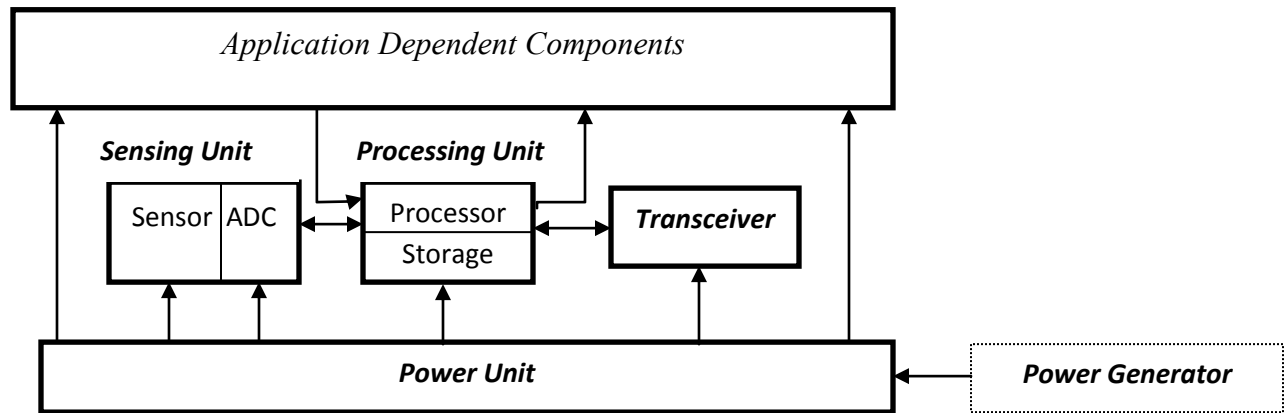


Figure 2.1: Components of a Sensor Node

Further description of the four basic components is as follows [1]:

**Sensing Unit:** When motes are under construction, their intended purpose often dictates the sensors that are added to the mote. A sensor is an electronic component that measures the physical quantity and converts to the type that can be read by the user or other electronic devices such as, for example, a microprocessor. Integrated into numerous devices, machines, and environments, sensors provide a tremendous social benefit. They can help to avoid catastrophic infrastructure failures, conserve precious natural resources, increase productivity, and enhance security.

**Processing Unit:** in the past WSN-era the processor was known to be an electronic “brain”. The basic functions of the processing unit are to make decisions and deal with collected data. The processing unit stores collected data in its memory until enough information has been collected. Once this point is reached, the microprocessor portion of the processing unit then puts the data in “envelopes,” or packages of data formatted for greatest transferring efficiency. These envelopes are then sent to the transceiver for broadcast. The processing unit also communicates with other motes to maintain the most effective network in much the same way it deals with data.

**Transceiver:** The transceiver consists of a radio transmitter and a radio receiver. Both of these parts must exist for any node to fully communicate with the other nodes. The transceiver, when transmitting, receives information from the processor and broadcasts the data to other nodes according to the network connections. In the other direction, when receiving, the transceiver receives information from another node’s radio and passes it to the processor.

**Power Unit:** The power source for the node also depends on the node's intended use. If the node is designed to last a very long time, say one year, it will have a larger power source than a node that is only meant to run for a month. The power sources usually range between a couple of AA batteries, and a watch battery, but with the new smart-dust nodes, also called "Spec," they can collect enough energy to sustain themselves from ambient light, or even vibrations. The power source provides energy required to run the sensors, processor, and transceiver.

## 2.3 Applications of Wireless Sensor Networks

Wireless sensor networks have gained considerable popularity due to their flexibility in solving problems in different application domains and have the potential to change our lives in many different ways. These ever-increasing applications of WSNs can be mainly categorized into five categories, military, environment, health, home and industrial. It is possible to expand this classification with more categories such as space exploration, chemical processing and disaster relief [1]. For our case we are going to discuss about the five main categories.

### **Military Applications**

Wireless sensor networks can be an integral part of military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting systems. The rapid deployment, self-organization and fault tolerance characteristics of sensor networks make them a very promising sensing technique for military targeting systems. Since sensor networks are based on the dense deployment of disposable and low-cost sensor nodes, destruction of some nodes by hostile actions does not affect a military operation as much as the destruction of a traditional sensor, which makes sensor networks a better approach for battlefields. Some of the military applications of sensor networks are monitoring friendly forces, equipment and ammunition; battlefield surveillance; reconnaissance of opposing forces and terrain; targeting; battle damage assessment; and nuclear, biological and chemical (NBC) attack detection and reconnaissance [1].

**Environmental Applications:** Some environmental applications of sensor networks include tracking the movements of birds, small animals, and insects; monitoring environmental conditions that affect crops and livestock; irrigation; macro instruments for large-scale earth

monitoring and planetary exploration; chemical/biological detection; precision agriculture; biological, earth, and environmental monitoring in marine, soil, and atmospheric contexts; forest fire detection; meteorological or geophysical research; flood detection; bio-complexity mapping of the environment; and pollution study [13, 14].

**Health Applications:** Some of the health applications for sensor networks are providing interfaces for the disabled; integrated patient monitoring; diagnostics; drug administration in hospitals; monitoring the movements and internal processes of insects or other small animals; telemonitoring of human physiological data; and tracking and monitoring doctors and patients inside a hospital [1].

**Home Applications:** As technology advances, smart sensor nodes and actuators can be buried in appliances such as vacuum cleaners, microwave ovens, refrigerators, and DVD players as well as water monitoring systems. These sensor nodes inside domestic devices can interact with each other and with the external network via the Internet or satellite. They allow end-users to more easily manage home devices both locally and remotely. Accordingly, WSNs enable the interconnection of various devices at residential places with convenient control of various applications at home [15].

**Industrial Applications:** Networks of wired sensors have long been used in industrial fields such as industrial sensing and control applications, building automation, and access control. However, the cost associated with the deployment of wired sensors limits the applicability of these systems. Therefore WSNs became the promising alternative solution for these systems as they offer significant cost savings and enable new functionalities. Some of the commercial applications are monitoring material fatigue, building virtual keyboards, managing inventory, monitoring product quality, constructing smart office spaces, environmental control of office buildings, robot control and guidance in automatic manufacturing environments [1, 13, 14, 15, 16, 17] and so many others.

## 2.4 Operating System

In most cases an operating system is defined as the master program existing between the computer hardware and the user. Its main purpose is the management and coordination of various system resources, which may include hardware resources such as hard disk, memory, peripheral device management, or non-hardware resources such as managing processor timeslots.

In most wireless sensor networks, sensors implement a basic operating system containing the minimum functions necessary to execute their tasks. This minimalist approach is of great importance because of the highly resource constrained nature of these sensing devices; operating system code and application code must reside in devices having much less than one megabyte (1 MB) of memory. There are several sensor-based operating systems in use today; some of them are TinyOS, Contiki, Mantis OS, BTnut, and Nano-RK. In the following Section we will be reviewing TinyOS, since it is the de-facto standard and very mature Operating System for wireless sensor networks [11].

### **Tiny Operating System (TinyOS)**

TinyOS, as its name implies, can be described as a miniature framework designed for embedded systems that require very aggressive resource management due to the highly constrained nature of their resources such as power and available memory [18]. It implements the hardware abstraction layer and scheduler of a conventional operating system, allowing generic programs that may have no knowledge of the intricate details of the operations supported by the underlying hardware components (such as sensors) to use well-defined interfaces to interact with these components. Its C-language-like framework provides an interface to core system components, allowing a programmer to manage various services of the system.

TinyOS provides software abstraction for hardware components such as its communication, routing, sensing, and storage subsystems. Software components in TinyOS refer to abstractions of specific services provided by either another software component or a hardware component. A software component consists of any number of the following: Modules and Configurations. Modules are the lowest form of abstraction provided by the TinyOS operating system. They implement program logic that directly addresses a software component and can also provide a particular set of services, thereby enabling the reuse of software components. Other software

components can interact with a module through its defined interface, which specifies a set of operations/services that a particular module implementation provides [11, 18].

On the other hand, abstractions of multiple modules and other configurations grouped together form a newly abstracted component referred to as a configuration. A configuration can be visualized as a super component consisting of several subcomponents to provide a single unified interface. Configurations wire a set of components defined in a component signature, this is the set of interfaces that a component uses and provides to another component thereby allowing two or more components to communicate with each other [18].

In the next Section, we are going to discuss about the foundation of many important concepts which helped WSNs to be recognized as one of the most promising technologies of today and for future generations to come.

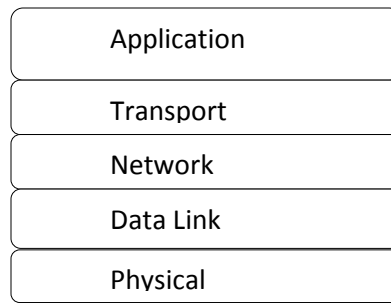
## **2.5 Wireless Networking**

### **2.5.1 Communication Structure of a Wireless Sensor Network**

The sensor nodes are usually scattered in a sensor field. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink and the end users. Data are routed back to the end user by a multi-hop infrastructure-less architecture through the sink. The sink may communicate with the task manager node via Internet or Satellite.

The protocol stack used by the sink and the sensor nodes combines power and routing awareness, integrates data with networking protocols, communicates power efficiently through the wireless medium and promotes cooperative efforts of sensor nodes. The protocol stack consists of the application layer, transport layer, network layer, data link layer, physical layer, power management plane, mobility management plane, and task management plane [1, 16].

A simplified protocol stack for a WSN is shown in Figure. 2.2.



*Figure 2.2: Architectural Layers of a WSN*

**Application layer:** It defines a standard set of services and interface primitives available to a programmer independently on their implementation on every kind of platform.

**Transport layer:** It helps to maintain the flow of data if the sensor networks application requires it. This layer is especially needed when the system is planned to be accessed through Internet or other external networks. Unlike protocols such as TCP, the end-to-end communication schemes in sensor networks are not based on global addressing. Therefore, new schemes that split the end-to-end communication probably at the sinks may be needed.

**Network layer:** It takes care of routing the data, directing the process of selecting paths along which to send data in the network.

**Data Link layer:** It provides the multiplexing of data streams, data frame detection and medium access control (MAC).

**Physical layer:** it is responsible for frequency and power selection, modulation, and data encryption.

Communication paradigms in WSNs differ from those associated to traditional wireless networks, triggering the need for new communication protocols. In this context, the recently standardized IEEE 802.15.4 protocol presents some potentially interesting features. It provides enough flexibility for fitting different requirements of WSN applications by adequately tuning its parameters, even though it was not specifically designed for WSNs. In fact, low-rate, low-power consumption and low-cost wireless networking are the key features of the IEEE 802.15.4 protocol, which typically fit the requirements of WSNs.

IEEE 802.15.4 protocol defines both physical and medium access layers. The physical layer and media access layers are called PHY and MAC, respectively. In the WSN domain, development of the IEEE-802.15.4 standard protocol has revolutionized this technology [18, 19, 20].

Having pin pointed the common points in the communication protocol stack it is very important that we now look into the details of IEEE 802.15.4. As mentioned before the main focus of this protocol is to provide low power consumption and reliable throughput for short range communications. WSNs commonly adhere with the 802.15.4 specifications for the physical and MAC layers [19].

In the next Sections, we review IEEE 802.15.4 standard which covers the Physical and Medium Access Control (MAC) layer of Wireless Sensor Networks and the issues of routing protocols for wireless sensor networks. Different clustering protocols will also be discussed in regard to the application.

## **2.5.2 Physical Layer**

IEEE 802.15.4 physical layer is the first and lowest layer consisting of basic hardware transmission technologies of a network. It also provides an interface between the medium access control (MAC) sub-layer and the physical radio channel. Two services are provided, the physical data service and the physical management service.

IEEE 802.15.4 physical layer provides an interface for transmitting bit streams over the physical communication medium. It is responsible for the following tasks [17]:

- Activation and deactivation of the radio transceiver
- Channel frequency selection
- Data transmission and reception.

The physical layer (PHY) of the reference model specifies the network interface components, their parameters, and their operation. Furthermore, to support operation of the MAC layer, the PHY layer includes a variety of features, such as receiver energy detection (RED), link quality indicator (LQI), and clear channel assessment (CCA). The PHY layer is also specified with a

wide range of operational low-power features, including low-duty-cycle operations, strict power management, and low transmission overhead [21].

### **2.5.3 Medium Access Control (MAC) Layer**

The MAC layer is the second lowest layer, offering a management interface for the physical channel. In WSN, nodes usually have to share a common channel. Therefore, the MAC sub-layer task is to provide fair access to channels by avoiding possible collisions [22, 23, 24]. The main goal in MAC protocol design for WSN is energy efficiency in order to prolong the lifetimes of sensors. The prime role of the MAC is to coordinate access to and transmission over a medium common to several nodes.

The IEEE 802.15.4 MAC-layer specification is designed to support a vast number of industrial and home applications for control and monitoring. These applications typically require low to medium data rates and moderate average delay requirements with flexible delay guarantees. Furthermore, the complexity and implementation cost of the IEEE 802.15.4 standard compliant devices must be low to minimize energy consumption and enable the deployment of these devices on a large scale. To address the needs of its intended applications while enabling the deployment of a large number of monitoring and control devices at a reduced implementation cost, the IEEE 802.15.4 MAC-layer specification embeds in its design several unique features for flexible network configurations and low-power operations [21].

The authors in [24] made a survey on the different MAC protocols including Sensor-MAC (S-MAC), WiseMAC, Spatial TDMA (Time Division Multiple Access) and CDMA (Code Division Multiple Access), Traffic-Adaptive MAC protocol (TRAMA), Node Activation Multiple Access (NAMA), etc. and presented the pluses and limitations of each MAC protocol.

The IEEE 802.15.4 MAC Standard provides information about type and association of devices, channel access mechanism, packet delivery, frame structure, guaranteed packet delivery, possible network topologies and security issues.

## 2.5.4 Network Layer Protocols

In this Section we will look into some of the design issues, challenges and techniques that are relevant when designing a routing layer protocol for WSNs.

WSNs are extremely versatile and can be deployed to support a wide variety of applications in many different situations, whether they are composed of stationary or mobile sensor nodes. The way these sensors are deployed depends on the nature of the application. In environmental monitoring and surveillance applications, for example, sensor nodes are typically deployed in an ad hoc fashion so as to cover the specific area to be monitored. In health care related applications, smart wearable wireless devices and biologically compatible sensors can be attached to or implanted strategically within the human body to monitor vital signs of the patient under surveillance.

Despite the disparity in the objectives of sensor applications, the main task of wireless sensor nodes is to sense and collect data from a target domain, process the data, and transmit the information back to specific sites where the underlying application resides. Achieving this task efficiently requires the development of an energy-efficient routing protocol to set up paths between sensor nodes and the data sink. The path selection must be such that the lifetime of the network is maximized. The characteristics of the environment within which sensor nodes typically operate, coupled with severe resource and energy limitation, make the routing problem very challenging.

In order to design an efficient routing protocol, several challenging factors that influence its design should be addressed carefully. These factors are discussed below [20]:

**Network dynamics:** Three main components of a sensor network, i.e., sink, sensor nodes and monitored events can be either stationary or mobile. Routing to a mobile node (sink or sensor node) is more challenging as compared to a stationary node. Similarly, keeping track of a dynamic event is more difficult.

**Node deployment:** Node deployment can be either deterministic or random. Deterministic deployment is simple as sensors are manually placed according to a plan and data is routed along pre-determined paths. Whereas random deployment raises several issues as coverage, optimal clustering, etc., which need to be addressed.

**Multi-hop vs. single-hop network:** As the transmission energy varies directly with the square of distance therefore a multi-hop network is suitable for conserving energy. But a multi-hop network raises several issues regarding topology management and media access control.

**Data reporting models:** In wireless sensor networks data reporting can be continuous, query-driven or event-driven. The data delivery model affects the design of network layer, e.g., continuous data reporting generates a huge amount of data therefore, the routing protocol should be aware of data aggregation.

**Node heterogeneity:** Some applications of sensor networks might require a diverse mixture of sensor nodes with different types and capabilities to be deployed. Data from different sensors can be generated at different rates, network can follow different data reporting models and can be subjected to different quality of service constraints. Such a heterogeneous environment makes routing more complex.

**Data Aggregation:** Data aggregation is the combination of data from different sources according to a certain aggregation function, e.g., duplicates suppression, minima, maxima, and average. It is incorporated in routing protocols to reduce the amount of data coming from various sources and thus to achieve energy efficiency.

Below we will discuss about one of the most important techniques in routing that addresses the sensor network-specific issues in the routing layer.

## **2.6 Clustering for Data Aggregation**

As sensor networks are expected to scale to large numbers of nodes, protocol scalability is an important design criterion. If the sensors are managed directly by the base station, communication overhead, management delay, and management complexity become limiting factors in network performance. Clustering has been proposed by researchers to group a number of sensors, usually within a geographic neighborhood, to form a cluster that is managed by a cluster head. A fixed or adaptive approach may be used for cluster maintenance. In a fixed maintenance scheme, cluster membership does not change over time, whereas in adaptive clustering scheme, sensors may change their associations with different clusters over time [25].

Clustering provides a framework for resource management. It can support many important network features within a cluster, such as channel access for cluster members and power control, as well as between clusters, such as routing and code separation to avoid inter-cluster interference. Moreover, clustering distributes the management responsibility from the base station to the cluster heads, and provides a convenient framework for data fusion, local decision making and local control, and energy savings [26, 27, 28].

Owing to a variety of advantages, clustering is becoming an active branch of routing technology in WSNs. In the following Sections, we present a fine grained explanation on the two most popular clustering routing protocols.

### 2.6.1 Low Energy Adaptive Clustering Hierarchy (LEACH)

Low energy adaptive clustering hierarchy (LEACH) proposed in [27, 29] is a routing protocol, which includes distributed cluster formation. This cluster formation occurs by the selection of a certain percentage of sensor nodes as cluster heads. Among other functions, the cluster heads compress data arriving from various sensors within the cluster and send it to the base station, hence reducing the amount of information that flows towards the base station. LEACH uses CDMA/TDMA to reduce the amount of inter and intra-cluster collision. The protocol operates in two phases. The first phase is the setup phase that involves organization of clusters and formation of cluster heads. It starts by each node generating a random number between 0 and 1. If the generated number turns out to be less than a threshold value  $T(n)$  the node becomes a cluster head. The threshold is calculated as [29]:

$$T(n) = \begin{cases} \frac{p}{1 - p * (r \bmod \frac{1}{p})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

where,  $p$  is the predetermined fraction of nodes that are planned to be cluster heads,  $r$  is the current round,  $n$  is the number of nodes and  $G$  is the set of all nodes that are participating in election process.

The new elected cluster-heads send an advertisement message to all the nodes in the network. The nodes join the cluster-head from which they can receive maximum signal strength and cluster-heads calculate a TDMA schedule for the members of the cluster. The second phase involves sensing and transmitting data to the cluster heads, which aggregate the data and send it to the base station. After certain predetermined time the network again goes in the setup phase.

Because there is no interaction between nodes when deciding roles, the cluster heads may be chosen such that there is no uniformity throughout the network and certain sensors are forced to join clusters located at large distances from them. To mitigate this problem, a centralized version of LEACH called LEACH-C has been developed. LEACH-C uses simulating annealing to choose the cluster heads for a given round so that the average transmission power between sensors and their cluster heads is minimized.

In some cases it has been shown using simulation that LEACH outperforms conventional routing protocols, including direct transmission and multi-hop routing, minimum transmission-energy routing, and static clustering based routing algorithms.

### **2.6.2 Hybrid Energy-Efficient Distributed Clustering (HEED)**

Nodes in LEACH independently decide to become cluster heads. While this approach requires no communication overhead, it has the drawback of not guaranteeing that the cluster head nodes are well distributed throughout the network. While the LEACH-C protocol solves this problem, it is a centralized approach that cannot scale to very large numbers of sensors. Many papers have proposed clustering algorithms that create more uniform clusters at the expense of overhead in cluster formation. One approach that uses a distributed algorithm that can converge quickly and has been shown to have low overhead is called HEED [25, 30].

HEED uses an iterative cluster formation algorithm, where sensors assign themselves a “cluster head probability” that is a function of their residual energy and a “communication cost” that is a function of neighbor proximity. Using the cluster head probability, sensors decide whether or not to advertise that they are a candidate cluster head for this iteration. Based on these advertisement messages, each sensor selects the candidate cluster head with the lowest “communication cost” (which could be the sensor itself) as its tentative cluster head. This procedure iterates, with each

sensor increasing its cluster head probability at each iteration until the cluster head probability is one and the sensor declares itself a “final cluster head” for this round.

As mentioned before in HEED, CHs are periodically elected based on two important parameters: residual energy and intra-cluster communication cost of the candidate nodes. Initially, in HEED, a percentage of CHs among all nodes,  $C_{prob}$ , is set to assume that an optimal percentage cannot be computed a priori [30].

The probability that a node becomes a CH is [30]:

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}}$$

Where  $E_{residual}$  is the estimated current energy of the node, and  $E_{max}$  is a reference maximum energy, which is typically identical for all nodes in the network.

The value of  $CH_{prob}$ , however, is not allowed to fall below a certain threshold that is selected to be inversely proportional to  $E_{max}$ . Afterwards, each node goes through several iterations until it finds the CH. If it hears from no CH, the node elects itself to be a CH and sends an announcement message to its neighbors. Each node doubles its  $CH_{prob}$  value and goes to the next iteration until its  $CH_{prob}$  reaches 1.

If a node completes HEED execution without selecting a cluster head that is final CH, it considers itself uncovered, and announces itself to be a cluster head with state final CH. A tentative CH node can become a regular node at a later iteration if it finds a lower cost cluster head. Note that a node can become a cluster head at consecutive clustering intervals if it has high residual energy and low cost.

## **2.7 Railway System**

As an important infrastructure of the country, as the artery of national economy and as a popular traffic tool, railway plays an important role in modern logistics system. Railways have been a means of commercial transportation for both passengers and freight since 1825 [43].

### **2.7.1 Railway in Ethiopia**

Ethiopia has a more than 100 year old meter gauge 781 km diesel railway owned jointly with the Government of Djibouti and operated by CDE (Chemin de fer Djibouti Ethiopien) [31]. This railway, which was established during the reign of Emperor Menelik II, in addition to its deterioration and malfunctioning due to age, is almost abandoned due to its incapability of supporting the current demand of freight and passenger mobility. The railway has a significant place in the cultural songs, proverbs, poetry, etc., of the country. Therefore, the people and the land of Ethiopia are not new to the service of railway. However, due to the current growth momentum of the country, it has been decided that the country needs a modern and reliable railway system that can accommodate and facilitate the growth.

In the next Section we will be covering some details about the current railway system in Ethiopia specifically the Addis Ababa Light Railway Transport (LRT), and the techniques to be used in order to monitor the railway system.

### **2.7.2 Addis Ababa Light Railway Transport (LRT)**

Addis Ababa Light Rail Transit Project is a semi-closed urban rail transit system. To effectively solve the problem of urban transportation, especially that of the downtown area, the government of Ethiopia decided to build a light rail in the city of Addis Ababa. Currently this project has planned two lines, the east-west (E-W) line and the south-north (S-N) line. About 3 km is the sharing Section for both E-W route and N-S route, which has the greatest passenger [32].

For the project, modern trolley car (DC750V power supply type) is used as the passenger train; DC750V diversified power supply system is adopted for the power supply system; most of the tracks are constructed on the ground, and some Sections are built on overhead bridges or in

underground tunnels as shown in Figures 3.1 and 3.2. Where the line has level crossing with the municipal roads, signal system is adopted for traffic control. The planned line has a total length of about 75 km which is being constructed in two phases. The main line of E-W line is about 17.4 km in full length, and 16.689 km is the full length of the N-S line.

### **The East-West Line**

The E-W line starts from Ayat and ends at Torhailoch. The total length is 17.4 km. There are 22 stations, among which 5 are elevated stations, 1 underground station and 16 ground stations. The control center (commonly used by both lines) is temporarily considered to be placed inside the depot.



*Figure 2.3: East-West transit of the AALRT line*

### **The South-North Line**

The S-N line phase starts from Menelik II Square and ends at Kaliti. The total length is 16.689 km. There are 22 stations, among which 9 are elevated stations (5 common stations at the common line), 2 underground stations and 11 ground stations.



*Figure 2.4: South-North Transit of the AALRT line*

In both cases the project covers all systems of both lines, which mainly includes: rolling stock, marshaling, passenger flow forecast, route, clearance limit, track, station construction, sub grade structure, elevated structure, underground structure, power supply, communication, signal, automatic fare collection, ventilation and air condition, water supply and drainage and firefighting, depot, control center, environmental protection and cost estimate [32].

### **2.7.3 Railway Security**

As mentioned before, railways are large infrastructures and are the prime mode of transportation in many countries. The railways have become a prime means of transportation owing to their capacity, speed, and reliability. Even a small improvement in performance of railways has significant economic benefits to the rail industry. Thus, a proper maintenance strategy is required to govern optimization of inspection frequency and/or improvement in skill and efficiency. Accidents happening due to track breaking have been a big problem for railways for life security and timely management of services. This breakage needs to be identified in real time before a train actually comes near to the broken track and get subjected to an accident. Also Railroad Bridge is one of the most important infrastructures of railway; therefore it is imperative to monitor the safety of the bridge so that the effect on railway operation efficiency and safety won't be catastrophic.

Having pointed out the importance of railway, and emphasizing on the point that a proper maintenance is necessary we reach to the point where we need to ask the question, "How well are rail systems protected?" Secondarily it must be asked, "What has been done to this point to

secure a vital link in the infrastructure, i.e., if there are any ways or mechanisms to be applied to protect the infrastructure?”

Conventionally the first rail inspections were done visually. Many sources cite that the need for better rail inspections came after a derailment at Manchester, New York, in 1911. That particular accident resulted in the death of 29 people and injuries to 60 others [43]. The investigation of the accident revealed that the cause was a transverse fissure (a critical crack which lies perpendicular to the length of the rail) in the rail. Non-destructive testing (NDT) methods are used as preventative measures against track failures and possible derailment.

The methods used to detect flaws in rails include: [43]

- **Ultrasound** - the most popular method
- **Eddy current inspection** - great for surface flaws & near surface flaws
- **Magnetic Particle Inspection (MPI)** - used for detailed manual inspection
- **Radiography** - used on specific locations (often predetermined) such as bolt holes and where termite welding was used
- **Magnetic induction or Magnetic flux leakage** - earliest method used to locate unseen flaws in the railway industry
- **EMAT** (Electromagnetic Acoustic Transducer): used for railroad and wheel inspection.

Currently the company that is undertaking the project of AALRT is considering applying a system called Supervisory Control and Data Acquisition (SCADA) system. In the next Section we will give an overview of the system and for what purpose will it be used in the LRT.

#### **2.7.4 Supervisory Control and Data Acquisition (SCADA) System**

SCADA systems are widely used nowadays to monitor and control dispersed hardware components in industrial settings such as power plants, gas or water treatment. Often, they are used in critical infrastructures where security is a vital factor for the environment. Due to this, they have to satisfy strict regulatory standards. Traditionally, SCADA systems have been deployed in a monolithic way with a central datacenter and where a large amount of wiring is required to connect the different hardware elements with the datacenter. SCADA normally comprises isolated and proprietary hardware, software and protocols. Due to the restricted

knowledge about these elements, previous security efforts were minimal and focused primarily on physical measures [4].

In the Addis Ababa LRT project the SCADA system is composed of three parts in the control center including power dispatching system, comprehensive automatic system of the controlled station, and data transmission channel linking the previous two systems, through which remote control, remote signaling, remote surveillance, and remote regulation of the power supply equipment are realized [32]. Generally the SCADA system is expected to meet the requirements for real time supervision of the power supply system equipment along the whole line. This shows that currently there is no means taken into consideration for monitoring the safety of the infrastructure.

Recently, the use of WSN technology has been acknowledged as promising for the CIP field. WSNs find in SCADAs, a mature technology that can be used to collect the information sensed by the network, to analyze it and to show it to users in elegant way [4]. WSNs exhibit some unique characteristics that are interesting in the context of SCADA systems:

**Dynamism:** WSNs are not static networks and allow connecting new nodes to the network and disconnecting or deactivating nodes deployed. The lack of wiring between the devices allows node mobility and eases the task of changing the network topology.

**Retrofit:** SCADA systems are usually deployed in a large scenario. That means that updating and changing the architecture once it has already been deployed is not simple. Reprogramming techniques can be used, but since the location of the devices is static and normally those devices require wiring, the updates are costly. On the other hand, diverse reprogramming techniques have been proposed in the context of WSNs that can be used to update the behavior of the monitoring application deployed in the WSN.

**Ease of installation:** WSNs can be easily deployed in the electrical power grid scenario since a separate energy source is not needed to power the nodes. Energy harvesting from the power grid can be done to feed the sensors. Furthermore, the wireless capabilities also help to reduce the cost of deploying a monitoring network.

The authors in [10, 33] proposed the numerous advantages of merging SCADA systems and WSN with the Internet such as wide area connectivity and pervasive access, redundancy, large addressing range, etc. In their solution, security is taken into account both at the service level and in the SCADA. Otherwise by exposing a SCADA system to the Internet becomes a potential subject of attack by hackers.

In [34] the authors proposed a way to replacing existing SCADA systems with a WSN. They share some of our points of view: advantages of WSN, simplicity. Hence, after looking into the SCADA system and comparing it with the WSN technology we decided to use an easy to deploy, low-cost, low-power, and multifunctional WSNs as a safety monitoring system for the current ongoing railway infrastructure.

## **2.8 Summary**

This Chapter pointed out the necessary points that act as a backbone in the wireless sensor network technology. It started with introducing this revolutionizing technology by providing an overview and description of wireless sensor networks and continued explaining about the enabling technologies, wireless sensor nodes and the operating system. The different standards and protocols in the area of wireless sensor networks and their specific application domain were also explained. An introduction about railway system specifically about railway in Ethiopia, railway security and monitoring systems that will be used once the new railway construction is finished were also discussed. Moreover, the Chapter explicitly described the currently proposed monitoring system, SCADA and for what purpose it will be used. We pointed out that our alternative to railway infrastructure monitoring system is based upon wireless sensor networks composed of inexpensive distributed sensor nodes. We tried to clearly point out the main reasons why WSNs are preferred more than SCADAs. The overall aim of this Chapter was to provide the theoretical background knowledge that is necessary for the design and modeling of our specific application.

## Chapter Three: Related Work

The use of WSNs for infrastructure health monitoring has been extensively studied. This Chapter covers some of the existing proposals that focus on WSNs monitoring the infrastructure health of bridges.

In [35], the author proposed a distributed railway bridge monitoring platform based on the Internet of Things (IOT) to monitoring the state of bridge safety. The author divided the railroad monitoring system into two parts, outfield and monitoring center. The outfield refers to the railroad bridges and railway line where sensors are deployed to collect data. The monitoring centers consist of the monitoring server and database, which are used to perform centralized data processing thus to achieve real-time monitoring of the railway bridge safety state. The author, not only describe about the railroad bridge monitoring platform but also pointed out deployment of sensor nodes, data collection and uploading, and early warning data processing mechanisms thus to ensure design rationality. The sensor deployment of the wireless sensor network in this work is divided into two parts: the sensor node deployed on the bridge and the sensor node deployed around railway line. The main tasks are to monitor the security state of the bridge and to deliver the data monitored by the bridge sensors safely and quickly to the nearest monitoring center, respectively. For the purpose of load balancing and network live extending, the sensors are deployed symmetrically on both sides of the railway line.

In this work information collection is realized by deploying corresponding sensors on the bridge or around the railway line depending on the monitoring requirements. Based on the preset threshold value the data collected by the sensors is divided into two types: emergency data and the daily data. The emergency data is the data that indicates a serious threat to the bridge's safety and the daily data is the data which don't indicate a threat to the security of the bridge.

Finally, the proposed monitoring system uses a hybrid deployment of multimedia sensor nodes, which are used to collect image information such as bridge cracks and ordinary sensor nodes, used to sense data such as pressure of bridge deck, soil temperature, and piers displacement information. Communications between nodes of one type or of different type is also applicable. After data compression, encryption and fusion, the data are sent to the sink node through multi-

hop communication, and finally sent to the monitoring host via train or Internet depending on the urgency.

Overall the work can help researchers as a starting point. But the problem here is that there is no clearly stated procedure to be taken such as the number of sensors to be deployed. The author also did not explain on how data will be transferred via train, whether a WiFi will be used or other mechanisms. Overall it lacks clear explanation on the approaches the author has used and also no experimental results are provided.

In [37] a wireless sensor network composed of Tmote-sky devices is deployed on a railway bridge. Information is collected by nodes and transmitted to the train that acts as a mobile sink node. The routing protocol forms a tree rooted at the head node of the WSN by periodically transmitting a message which is flooded down the WSN. The design of this work is driven by two important factors: application requirements, and detailed measurement studies of several pieces of the architecture. In comparison with prior bridge monitoring systems and sensor network prototypes, their contributions are three-fold. First, they have designed a novel event detection mechanism that triggers data collection in response to an oncoming train. Next, they employed a simple yet effective multi-channel data transfer mechanism to transfer the collected data onto a sink located on the moving train. Third, the architecture is designed with consideration of the interaction between the multiple requisite functionalities such as time synchronization, event detection, routing, and data transfer. Based on a prototype implementation. Overall their work focuses on delay tolerant network for bridge vibration monitoring using accelerometers where gateway mote collects data and forwards opportunistically to a mobile base station attached to a train passing by. The problem in this work is that they used a specific node as a solution for detecting an incoming train which is a problem when the number of nodes increase in number, instead they could have used a radio which can be attached to the head node so that this specific head node won't be an overhead to the network. The other issue is that they did not present any general test bed to support their work.

A monitoring application with a wireless sensor network that is performed on a 95 years old riveted steel railway bridge is presented in [38]. In order to perform an accurate assessment, strains were monitored on critical elements to catch the real loading during the passage of heavy freight trains. The wireless sensor network deployed on the bridge consisted of 8 nodes supplied with resistance strain gages and the root node connected to a solar energy rechargeable, battery powered base station. The monitoring system is operated in event-based mode to achieve an energy efficient operation to prolong the lifetime of the sensor network. The event detection is carried out with ultra-low power MEMS (Micro-Electro Mechanical Systems) acceleration sensors, which measured continuously the accelerations of the bridge and detected an approaching train. All the data recorded by the eight nodes is sent to the base station. To avoid packet collision, a sending policy is introduced where each node has its timeslot when it can transmit its data. The wireless monitoring system recorded temperature, humidity, and supply voltage with a time interval of 2 minutes. The paper also demonstrates that this procedure provides sufficiently accurate input data for use in cycle counting based fatigue assessment of steel bridges. Even though the work gives a glimpse of what a WSN monitoring should be it still misses detail explanation of the overall monitoring system such as no view of the software and system architecture.

Applicability of WSN technology to CIP problem is described in [39]. Their work more specifically concentrates on the monitoring of railway infrastructure. The WSN collects information about the structural health and behavior of the infrastructure when a train travels along it and relays the readings to the base station. The application is built using a middleware called PS-QUASAR over Tmote Sky nodes. PS-QUASAR is a publish/subscribe middleware used to automatically handle the QoS requirements and to simplify the task of developing the application. It also uses a directed acyclic graph based routing protocol that supports many-to-many communication and can handle priority, deadline, and reliability requirements in the communication between nodes. In this work the authors also applied an interesting mechanism that can be used to organize the network such as clustering, which will help in avoiding packet collision and packet loss, data fusion to minimize the number of sent packets and QoS support. This can be used as a ground work for our application. However, they did not give out a clear preview of which specific sensors to use in the sensor nodes and the way in which they are deployed can affect the accuracy of the readings. Moreover, they used a middleware as a solution

for simplifying the task of developing the application which enables them design a generic based routing protocol. Therefore, the limitation here is we do not see having a middleware as a solution to simplify the task of the application as a good way out because the middleware is there to design a generic based routing protocol rather than application specific protocol. Having application specific routing protocol will enable us to get results as our expectation but that is not the case in generic based routing protocol.

In [40] an early warning system based on WSN for railway infrastructure monitoring system is proposed. It exploits already available research results, frameworks and tools for the integration of heterogeneous sensors with the aim of continuously and automatically checking environmental parameters. Two frameworks allowing the integration of different and heterogeneous sensor networks and the detection of events are developed. The system uses clusters of sensors interconnected by WSN and communicates events to remote control centers using GSM-R/GPRS mobile terminals. They developed two frameworks SeNsIM (Sensor Networks Integration and Management), integration platform for heterogeneous sensor networks, and DETECT, a reasoning module which operates by performing a model-based correlation of events retrieved by sensors and trying to detect or, if possible, prevent threats against the physical infrastructure. They also exploited the tenet system, architecture for tiered sensor networks, which has been preliminary for rail bridge monitoring. The limitation in this work is that since they are using GSM-R/GPRS means of communication it does not state how they can be able to handle when some nodes are out of communication range.

A monitor system of railway track status using WSN is proposed in [36]. The primary technical and scientific objectives of the system introduced in this work are to generate innovative solutions for a number of issues facing the railroad community through the development of a system based on WSN. The overall overview of their system consists of one or more control centers (sink nodes) connected through a wire line connection, and many nodes scattered across railway track where each node is capable to collect the necessary data and forward the data back to the sink. In this work to improve the reliability of the Internet by connecting to multiple service providers they utilized a multi-homing scheme which allows every node in the network to be associated with two “homes”.

Furthermore, sensor devices are hierarchically organized with redundant paths so that multi-path routing is used to send the information to the remote base station. Fuzzy logic techniques are also employed to aggregate the data collected.

The monitoring of structural health of historical heritage buildings is presented in [41]. The authors pointed out that monitoring of such structure may be a daunting task for civil engineers due to the lack of a pre-existing model for the building stability, and to the presence of strict constraints on monitoring device deployment. Since such structures require periodic interventions in order to preserve them from deterioration, and in order to fix damages caused by natural phenomena, the authors proposed a novel approach to SHM (Structural Health Monitoring) of such buildings that involves the use of an innovative framework based on WSNs.

The overall goal of this work is to monitor the restoration works carried out on a historical building as a consequence of the damages suffered after a light earthquake. In particular, it is required both to analyze the response of the structure to vibrations on the fly, in order to promptly signal potential alarms, and to collect the corresponding measurements for further, more refined analysis. The system provides real-time feedback to the civil engineer that may promptly steer the functioning of the monitoring network, also remotely accessing sensed data via web interfaces.

An integrated structural health monitoring (SHM) system for highway bridges is presented in [42]. The monitoring system aims to collect dynamic response data of the bridge when it is subjected to random and moving vehicle loads to help evaluate the state of the highway bridge during its life span.

The system is based on a customized wireless sensor network platform with a flexible design that provides a variety of sensors typical in SHM. These sensors include accelerometers, strain gauges, and temperature sensors with ultra-low power consumption. An S-Mote node, an acceleration sensor board, and a strain sensor board are developed to satisfy the requirements of bridge structural monitoring. Communication software components were integrated within TinyOS operating system to provide a flexible software platform whereas the data processing software performs analysis of acceleration, dynamic displacement, and dynamic strain data. The prototype system comprises a nearly linear multi-hop topology and is deployed on an in-service

highway bridge. Data acquired from the system are used to examine network performance and to help evaluate the state of the bridge. Experimental results showed that the system assists continuous or regular interval monitoring for in-service highway bridges.

The limitation here is that packet loss is observed during the tests. A good solution to this problem could be to incorporate a data recovery mechanism into sink nodes and also more strain sensors should be considered, including vibrating steel wire transducers, which are as widely used as resistance strain gauges.

### **Summary**

Even if we have some commonalities with the other works there are still some issues they did not cover. The other proposals are more engaged on the sensor processing part and a great number of them lack application specific routing protocol. Furthermore, they do not state how they handled the nodes that are out of communication range. When one decides to develop a WSN application it is always a wise choice to consider the backbone infrastructures that a WSN may contain such as components for data storage, visualization and network control and none of these have been mentioned in any of the works.

Unlike other proposals, in this thesis to simplify the task of developing the application and to handle the QoS, application specific routing protocol will be used. Also as mentioned before, a mobile sink based technique will be taken into account to cover the nodes that are out of communication range. Furthermore, Modeling an application that integrates the most important services which are the pillar for any wireless sensor network application, those are, data collection, routing, and data management services, they will be incorporated for our specific application.

## **Chapter Four: Proposed Solution**

### **4.1 Introduction**

In the previous Chapter we have stated that currently applicable railway infrastructure monitoring techniques in most areas of the world are costly and require visual inspection to check if the infrastructure needs maintenance or not which makes it very tedious. In the case of Ethiopia, the system to be employed is more of on the monitoring of the power supply rather than the infrastructure itself for example, monitoring the Railway Bridge, or the rail track. Consequently we consider using wireless sensor network for the monitoring of railway infrastructure as a wise choice since it helps in mitigating the problems mentioned above. These WSNs utilize small, inexpensive and smart nodes for data acquisition, routing and data management purpose to achieve the goals of the monitoring application.

Following the introduction, in Section 4.2, we present the details about the proposed system architecture for railway infrastructure monitoring using wireless sensor networks. The overview of the railway infrastructure monitoring application and routing protocol are presented in Section 4.3. The data management and Access tools are also explained in Section 4.4. Finally, Section 4.5 summarizes the Chapter.

### **4.2 System Architecture**

The general architecture of the railway infrastructure monitoring is depicted in Figure 4.1. The application scenario consists of a WSN deployed on a railway bridge and a mobile node mounted on the train passing through which will collect information sensed by WSN on the bridge. The WSN gathers important data about the structural health of the infrastructure. The architecture houses sensors to sample the different physical parameters of the railway bridge such as temperature sensors for sensing the temperature value around the railway tracks and vibration caused by the external sources on the infrastructure using vibration sensors and temperature. These two sensors are the necessary when monitoring any civil infrastructure. Vibration and temperature monitoring has been used for decades as a performance evaluation technique. The

vibration sensors will help in detecting the forced and free vibration caused by the train when travelling along the bridge.

The architecture also requires the sensor network to be organized in a multi-hop cluster based tree where sensors with similar communication channel, i.e., sensors communicating within channel 26, are placed in the same cluster. Clustering helps in solving the difficulties that hamper wireless sensor nodes from becoming widely used, by reducing the cost of transmitting data to the base station, reducing the power consumption of the devices, maximizing the routing process execution and allowing scalability. Furthermore the most important concept in clustering is the minimization of data redundancy which is sensed by similar cluster members, that is, data aggregation which is performed by the cluster heads.

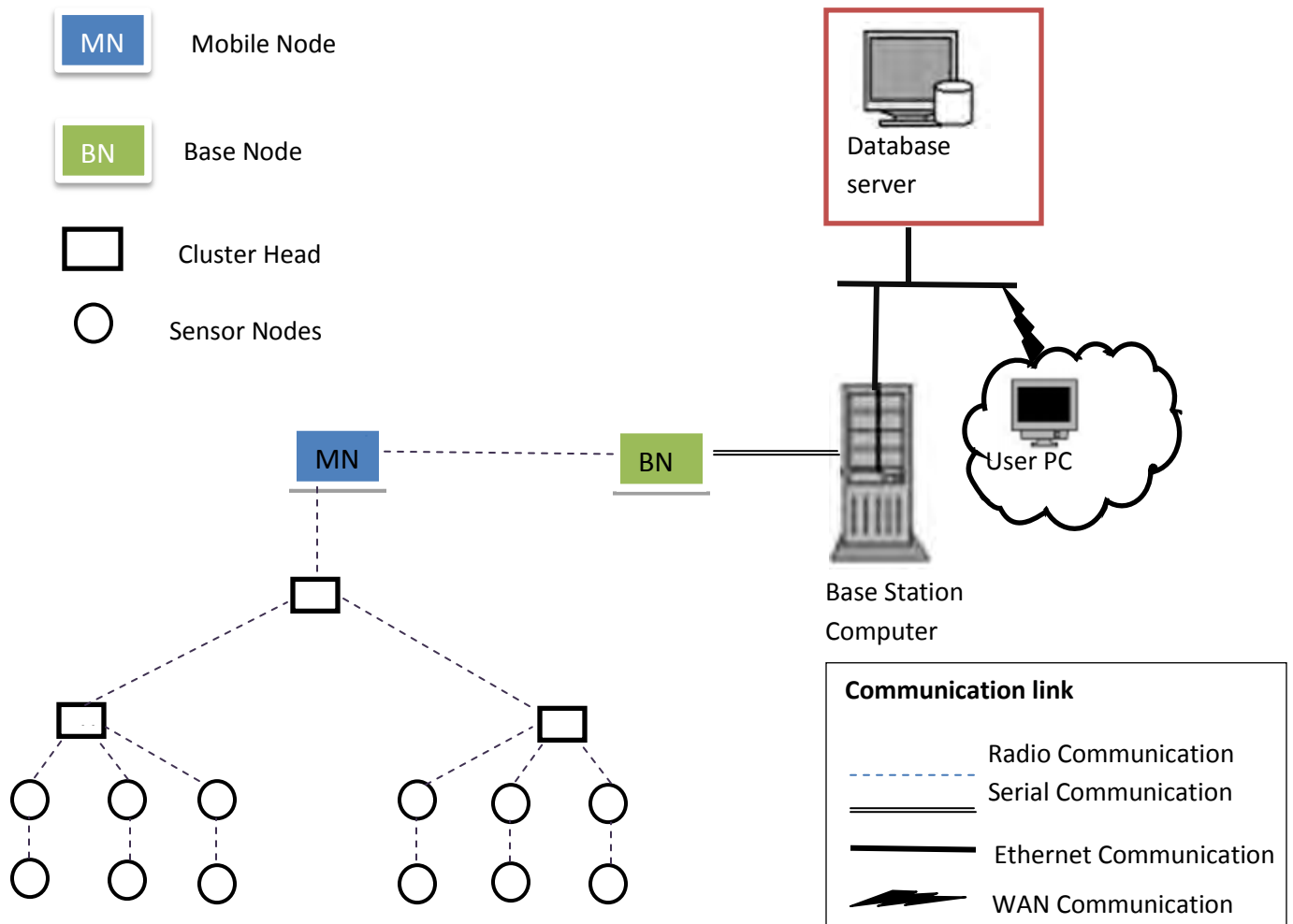
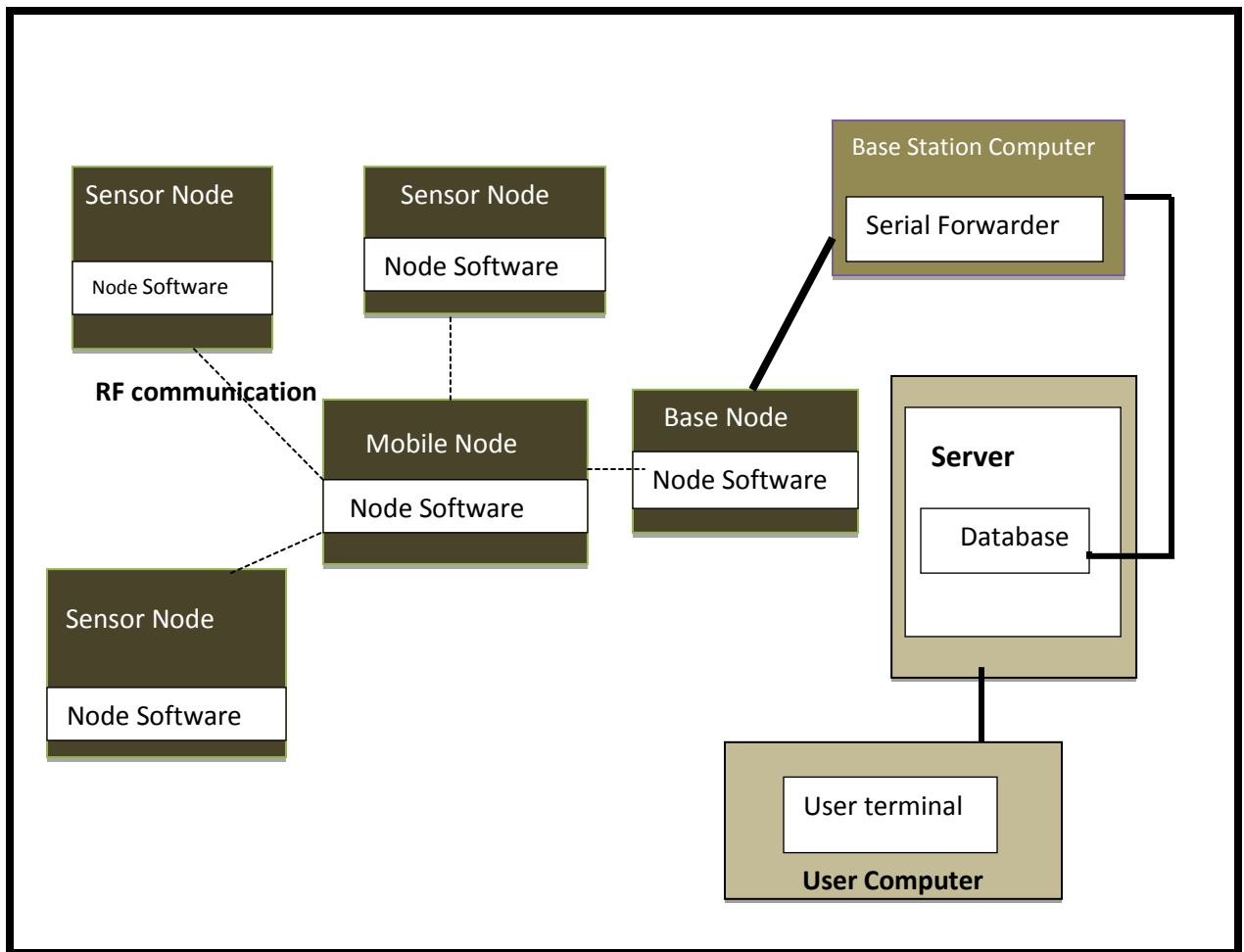


Figure 4.1: Proposed System Architecture

As shown in Figure 4.1 the nodes are deployed, as it is intended to mimic the way we expect nodes to be placed on a bridge span for each component to talk to one another and the routing protocol as explained in Chapter 5 will form the tree. Each node uses its own communication method. In our case the communication methods are not the same for all components. Communication by the sensor nodes, the cluster heads, the mobile node, and the base node (sink node) is through radio communication. The base node then communicates with the base station computer to download sampled data received from the mobile node through a serial port. The base station computer then forwards it to the database server via Ethernet communication. Finally the monitoring personnel in the command station can retrieve the sampled records through its terminal window. Therefore, what this difference in communication methods implies is that, for the WSN application to become effective and to reliably acquire the required data, choosing the best communication method is one of the necessary steps when constructing a WSN for a particular infrastructure.

### **4.3 Railway Infrastructure Monitoring**

When one application development is being undertaken there are pillars that enable it to become fully functional. Following, we give an overview of the software tools to be used in this application. In each sensor node in the WSN there is software, that is, node software which entails both the monitoring application and routing protocol. One important point to note here is that the mobile node and the base node are sensorless. There is no need for them to run application dependent software; they only provide a means of collecting the data and interface between the base station computer and the WSN, respectively. The base station computer is also one part of the application that runs a program to receive and forward the sampled data in the database. The software tools to be used are as shown in Figure 4.2



*Figure 4.2: Software tools for the Application*

The CH is equipped with 802.15.4 radio as the mobile node it can detect the arrival of the train. Therefore, in response to an oncoming train, the sensor nodes start sensing the physical environment and send the information they hold to the cluster head (CH) and they will go to sleep mode to save energy so later on when the other train with mobile node arrives only the CH will communicate with it so that the mobile node will be able to collect the sampled data. One point to note here is that there are two separate trains being used one for triggering the sensor nodes to get ready to start sensing and the second one is for transferring of data from the bridge nodes to the data analysis center.

The data from these nodes are transferred to a base station computer whenever a train passes a nearby station where the base node that is attached to the base station computer resides. The sensor nodes, the CH, the mobile node and the base node integrate through a wireless communication (i.e., Radio Frequency (RF) communication) as shown in Figure 4.2. The base

node is attached to the base station computer through its USB serial port and the base station computer comprises of serial forwarder tool that forwards the packet to the database. For our case our database and base station computer are in the same place. Later on users read the database records using their terminal window.

### **4.3.1 Railway Infrastructure Monitoring (RAIM) Application**

In the above Section, we gave explanation on the software tools that make the monitoring application operational. Following is about the proposed application, RAIM, wireless sensor network software that is designed for the purpose of monitoring the railway bridge. The sensor nodes deployed in the bridge run data acquisition application that determines how the sensor should sample data about the physical phenomena and when they should start sampling and a routing protocol that determines how the information should be routed from the WSN to the control center.

#### **Routing Protocol: Mobile Node Based Routing Protocol (MNRP)**

The protocol is designed to meet the requirements of the architecture in Figure 4.1, that is, applicability of a mobile node (MN) as a data collector from the WSN, which will help in maintaining the energy efficiency of the sensor nodes. In MNRP energy efficiency is considered as the core design for data gathering in WSN. In our case we consider a mobile node as a data collector in clustered based WSN where the whole network is divided into small regions known as clusters. In each cluster, one node is elected as a cluster head (CH). Elected CH is responsible for aggregating sensed data from its cluster member nodes(s) and forwards it to the mobile node. As mentioned earlier the mobile node is mounted on the train moving in the network and in its mobility when it reaches the vicinity of the cluster head the CH at a particular instance of time forwards its data to the mobile node. Following is how forwarding of data to the mobile node is handled in MNRP.

Data gathering in mobile node based clustered WSN is divided into two main phases, setup and steady phase. Explanation of MNRP is given using these phases. The workflow of each phases such as the mobile node advertisement phase and TDMA scheduling phase, and the working flow of the general routing protocol is shown in Algorithms 4.3, 4.4, and 4.5 respectively.

**Setup phase:** In this phase clustering is done and nodes send their data to the CH and the CH waits for the mobile node to come in its communication range to send sensed data. The mobile node equipped with 802.15.4 radio broadcasts beacon message when it reaches a new destination. The beacon message contains location information of the mobile node. CHs in its vicinity come to know about its presence and send their data, this is the process handled in the mobile node advertisement phase as shown in Algorithm 4.4. Finally the CH that received the mobile node beacon message responds to that advertisement by sending back registration message to the mobile node.

**Steady phase:** in this phase, a mobile node performs actual work for which it is moved in the network, i.e., data gathering. Just like the setup phase it handles tasks in two different phases and the two sub phases are, Neighbor discovery phase: In this phase, the head periodically issues HELLO message; the head is the node that is the parent of every node in the routing tree. Nodes hearing the HELLO message themselves transmit HELLO periodically. This results in each node knowing its good neighbors. Tree construction phase is the other phase where root queries its one hop neighbor for link state information. After having this information root then queries the two hop nodes for the link state and so on, until all the nodes detected in the first phase are covered. This information can then be used to construct the whole tree.

Following, we present the algorithms of the routing protocol and the clustering protocol. In Chapter 2 we have explained about the clustering techniques applicable in WSN application. At first, we considered using the LEACH protocol but faced a disadvantage that cluster head selection is a difficult problem to optimize. Therefore, in our setting this dynamic clustering is out of question since we want the nodes on a particular span to be in a cluster, since it is their data which is correlated.

One thing to include is that when mobile node reaches a new place during its mobility, it needs to inform the CH in its neighborhood about its presence. Thus, a mobile node broadcasts beacon message to the sensor nodes in its vicinity. Beacon message contains the location information of the mobile node and information of its moving velocity.

**Input:** *Beacon Message, Timeperiod*

**Process:**

*Mobile node sends the beacon message and sets the time interval 'Ti' based on the communication delay*

*IF mobile node receives message from any other sensor node within The timeperiod 'Ti'*

*IF message received is CHregister*

*Send CHack message to that node*

*Include the node in registered CH list*

*END IF*

*END IF*

**Output :***Cluster Head registration*

*Algorithm 4.3: Pseudo code for Mobile Node Advertisement*

As mentioned before when the mobile node sends its beacon message the CH that received the advertisement message will respond back by sending a CH registration message. Therefore, after the mobile node has registered the CH in its current neighborhood it is the responsibility of mobile node to assign the time slots to the registered CH, i.e., slots when the registered CH can send the sensed data to the mobile node. Consequently, with this phase the mobile node sends the TDMA schedule to the registered CH as shown in Algorithm 4.4.

**Input:** *Registered Cluster Head*

**Process:**

*Check the CH registered in previous phase and accordingly arranges time slot for the registered CH.*

*IF(CH is in previous phase)*

*Mobile node sends TDMA schedule to registered CHs*

*Mobile node waits for the sensed data from the registered CHs and receives the sensed data from CH.*

*ELSE*

*Wait for cluster head to register*

*ENDIF*

**Output:** *Assign time slot (TDMA schedule) to send data and at the same time receives sensed data from CH*

*Algorithm 4.4: Time Division Multiple Access(TDMA) Scheduling*

```

Input: Registered Cluster Head
Process:
  Check if routing has started
  IF(routing is started)
    Find Neighbors
    update neighbor vector
    update tree
    IF ( tree is complete)
      Dispatch tree
    ELSE
      Query nodes
      IF (nodes are to be queried)
        Query node for its link state
        Update neighbor vector
      ELSE
        Dispatch Tree
    END IF
  END IF
  END IF
  END IF
Output: Routing Tree

```

*Algorithm 4.5: pseudo code for the routing protocol*

#### **4.4 Data store and Access Tools**

The data obtained from the sensor network has to be accessible in a suitable way for system users or operators. The most important requirement of our application is that the data items acquired from the sensor network have to be stored appropriately and users have to be able to access the stored data to perform the needed analysis by looking at the recorded samples.

Data from the sensors of different types (temperature and vibration) are collected and monitored by looking into the data store containing the records. With the help of the serial forwarder tool, that is, the default extension of the simulator it helps to extract spurious data from the transmitted packets before passing the data to the database. The recorded data to be included are timestamp of the event, message, and node identification.

## **4.5 Summary**

Overall, we have discussed about the proposed system architecture and also gave an overview of the chain of software tools for the application. Following, this we have explained the railway infrastructure monitoring application and the routing protocol. Lastly, we described about the data management tools to be applicable to see the recorded values of the sampled data.

## **Chapter Five: Prototype Implementation and Evaluation**

### **5.1 Introduction**

In this Chapter, we explain our work on implementing RAIM, a wireless sensor network based system for railway infrastructure monitoring. The project involved coding in the nesC dialect of C and working with Telosb motes, a sensor node platform used in this work. The main idea is to sense the acceleration induced by a train passing over the bridge. The sensed data is routed to a particular node designated as the head node, which transfers the data to a receiver on the next train that passes on the bridge. In this Chapter, we describe the various modules of the RAIM code implemented in nesC using TinyOS platform for programming Telosb motes.

To accomplish the monitoring application we selected components which we think are necessary to accomplish the specific task of the monitoring application. We implemented the following components: Routing protocol, Sensing component, collecting component, and transferring component. The implementation detail description of these components is presented in various Sections of this Chapter. In Section 5.2 we present the development and simulation tools to be applied throughout this work. Section 5.3 explains the development environment employed to implement the system and in Section 5.4 the implementation description of various components used in the application is given. We present simulation experiment and evaluation in Section 5.5 and finally the Chapter is summarized in Section 5.6.

### **5.2 Development and Simulation Tools**

The development environment, the hardware platform, programming language and simulation tool that were used for the implementation and evaluation of the proposed solution is described in this Section.

We have used Tinyos 2.1.2, an open source operating system for a WSN application, which can be downloaded from [44]. It is written in nesC [45] programming language and this language is also used in our monitoring application. COOJA [47] simulator is used which is a default simulator for contiki operating system and can also be used for TinyOS based applications.

In our specific application we used a Telos family hardware platform. The Telos family of sensors consists of TelosA and TelosB motes. They are a newer generation of motes when

compared with the Mica family, as they have a universal serial bus (USB) interface for data collection and programming. It is a versatile platform for scalable wireless sensing applications. Harnessing the power of TinyOS, an event-driven operating system designed specifically for extremely low power sensor networks, the Telos platform can be used with almost any sensor with either digital or analog outputs. We choose TelsoB mote as the ideal platform to prototype our application primarily because it was successfully used in previous works [4, 37, 39] done on similar applications. It has the following features:

- 250 kbps 2.4GHz IEEE 802.15.4 Chipcon Wireless Transceiver
- 8MHz Texas Instruments MSP430 microcontroller (10k RAM, 48k Flash)
- 1 MB external flash for data storage
- Integrated ADC, DAC, Supply Voltage Supervisor, and DMA Controller
- Integrated onboard antenna with 50m range indoors / 125m range outdoors
- 16-pin expansion support and optional SMA antenna connector
- TinyOS support : mesh networking and communication implementation

Collectview and Serial Forwarder (SF) are the tools used in this application to collect the samples from the nodes and to read the packets from sink sensor node serial port to base station computer in order to allow transfer of data between the sensor network and data management unit, respectively. We used MYSQL as a database tool to store samples taken from the sensor nodes.

### **5.3 Prototype Implementation**

In the previous Section, the necessary tools used for designing and implementing the RAIM application were identified. This Section describes the implementation details of the various components of RAIM which are implemented in nesC using TinyOS platform for programming Telosb motes.

We begin with a description of the components implemented in this application and how they work. Therefore, the most important modules (components) implemented to achieve the functionality of the application are routing, sensing, collecting and transferring. Other common components used while implementing the prototype are

- TimerMilliC: A component which defines the virtualized millisecond timer abstraction.
- MainC:A component that defines the task of booting the sensor node.
- LedsC: A component that manipulates the LEDs.
- SerialActiveMessageC: A generic component that defines the task of sending packets over the serial port.

The component graph that is used to represent the wiring of components is as shown in Figure 5.1. The wiring of components is useful to accomplish the necessary tasks of the monitoring application.

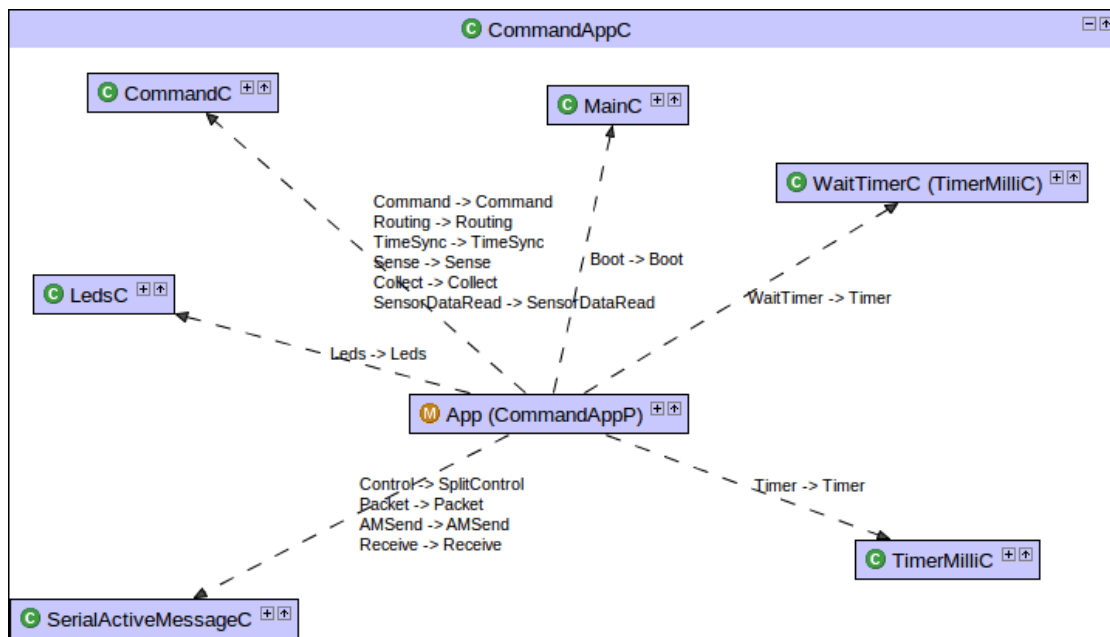


Figure 5.1: Component Graph

Before starting the different functionality of the application the sensor nodes need to boot first, thus the commandAppP is wired to the MainC component, which is the component responsible for booting the sensor nodes. Until the sensor boot the nodes will be in a waiting state their waking up state is defined using WaitTimerC, which is the virtualized millisecond timer abstraction.

When a boot event occurs, CommandP turns on the radio of the sensor node using RadioControlinterface(*RadionControl.nc*) which wires the application with radio of sensor node,

once the radio is on, the application starts the routing subsystem using RoutingC interface. When the sampled data packet reaches to the sink node, it converts the received radio packet to the serial packet format before sending it to the base station computer through serial port. The application uses a component, SerialActiveMessage, for sending serial packet using an interface SerialControl. The SerialActiveMessage component has defined AciveMessageSend(AMsend) and Receive components in its SerialActiveMessage.nc file and CommandAppP wire both in its configuration file with its defined read and send interfaces. The application samples data by ordering a generic component, DemoSensor.nc component. Overall the application uses the interfaces provided by each component and the interface that it provides such as RoutingC.nc, CollectC.nc, Sense.nc, timesyncC.nc and it provides command.nc interface.

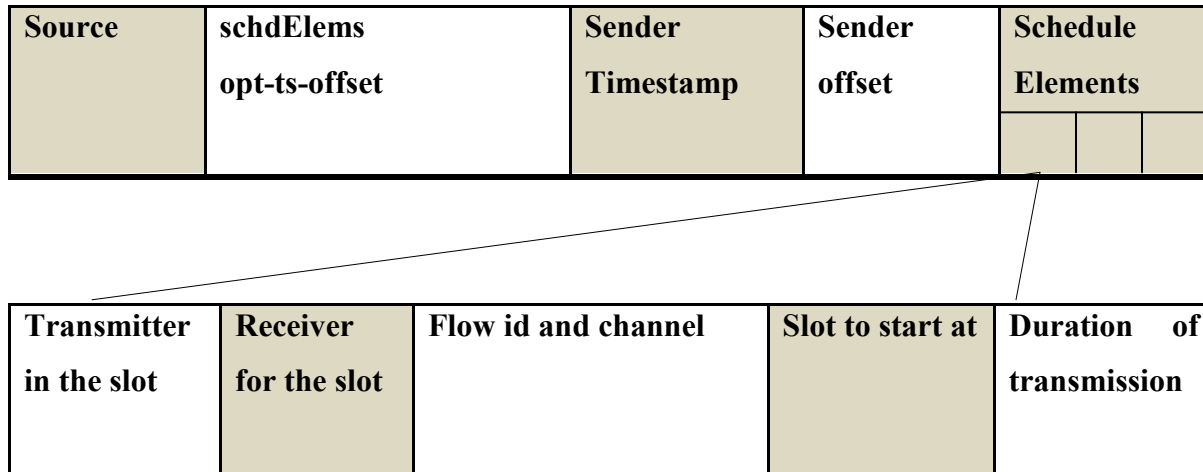
### **5.3.1 Implementation Detail of Main Components of the application**

In this Section we will explain the main components implemented which are the ones that play the big role for the prototype to become applicable. Those are: Routing, Sensing and Collecting.

#### **5.3.1.1 Routing Protocol**

The protocol works in two phases. Neighbor discovery phase: In this phase, the head periodically issues HELLO message. The head is the node that is the parent of every node in the routing tree. Nodes hearing the HELLO message themselves transmit HELLO periodically. This results in each node knowing its good neighbors. Tree construction phase is the other phase where the root queries its one hop neighbor for link state information. After having this information the root then queries the two hop nodes for the link state and so on, until all the nodes detected in the first phase are covered. This information can then be used to construct the whole tree.

Once the routing tree is constructed each of the nodes knows its parent and child, and it only accepts data or schedule information coming from the parent (downstream) or child (up-stream). The timeline is divided into frames. The frame starts whenever the head node broadcasts a schedule packet (also called as schedule frame). This packet carries information about allotment of the slots in the frame.



*Figure 5.2: Schedule Frame Format*



*Figure 5.3: Data Frame Format*

Now we explain the fields of the structure of the schedule frame as shown in Figure 5.2 and 5.3 [37]

- Source (1 byte): To check whether data is received from parent or child with the help of routing tree
- No of scheduling elements (6 bits): 64 possible schedule elements
- Optional timestamp and offset bit (opt-ts-offset): indicates whether timestamp and offset is present or not. Timestamp and offset are used for synchronization purpose. Offset is used to synchronize the nodes along the branch of the tree.
- Repeat bit: To repeat the previous schedule to get rid of schedule elements.

Each scheduling element consists of the following fields. Transmitter and Receiver in the slot, flow id to identify a flow uniquely with channel to be used for this slot, slot to start transmission and duration to transmit for. Furthermore, all other components implemented for the purpose of the application depend on the routing component. They all use the routing tree built by the component. This component is controlled by the head node and it sends the routing tree to all

nodes once the routing is done. This specific functionality of the component makes it the most crucial element in RAIM.

The overall functionality of the Routing component is that, first and foremost it is important to define the requirements of the component such as number of nodes in the topology, threshold for good and bad link, number of hello messages, and timeout value as an input. It takes all this inputs in the header file, *Routing.h*. By broadcasting a Hello Message any node can initiate routing and when a node receives this message, it assumes that some node has started routing. Each node in the topology sends 3 HelloMessages, which is also specified in the header file. In our case node with id 1 initiates the routing by sending the first hello message. As mentioned earlier in the head file we set two threshold values of the RSSI (Received Signal Strength Indicator), which is used to measure link quality as good or bad link. The bad link can be used in case no good link exists. Thus head node creates good link children and bad link children. Table 5.1 shows the result of the routing protocol with the formed tree.

*Table 5.1: Example Result of the Routing Protocol Output*

Node	Good Link	Bad Link	Tree Formed
1	1—0—1—1—0—1	0—0—0—0—0—1	0—1—1—2—1—1
2	1—0—1—1—0—0	0—0—0—0—0—1	
3	1—1—0—0—0—1	0—0—0—1—1—0	
4	1—1—0—0—0—0	0—0—1—0—0—1	
5	0—0—0—0—0—0	0—0—1—0—0—1	
6	0—0—1—0—0—0	1—1—0—0—1—1	

We make use of bit vectors to maintain the neighborhood information. The root node starts with setting the appropriate bits of its own neighborhood bit vectors. For this, it will have to decide whether a node is a good or bad neighbor to it, based on a threshold RSSI and then set the appropriate bit of good/bad neighbor vector. It may occur to us that there is no need to maintain a bad neighbor vector separately, since we are anyway setting the good neighbor vector bit only if current  $RSSI > R_{th}$ , i.e., RSSI was given in the header file of the routing component(*Routing.h*).

But there may be cases where a node may not be heard at all during the neighborhood discovery, meaning that it is not a neighbor at all. So, it is required to maintain a separate bad neighbor vector to distinguish a bad neighbor from a non-neighbor node. The root node uses the good neighbor vector to update the routing tree, which is represented as a list of node addresses against their parent node addresses. Thus the root node proceeds forming the tree with good neighbors.

When all the intermediate nodes which are part of the tree receive request for RSSI packet, they forward it to all their children nodes. If they receive RSSI reply, they forward it to their parent node. Thus every request reaches to destination and every reply reaches to head node. Head node checks whether routing is done after adding nodes to the tree. If the tree is built, it floods routing done message in the tree or lights up all the LEDs when simulating. In our case to make it simple we only light up all the LEDs when it is done. Each node receiving routing done message will forward it to all its children nodes and signal routing done event to the program. Few things to test in routing were successful completion of routing, failure of routing after some time. All these aspects are tested using six motes. Also we tested the routing by removing one mote from the tree and it gave failure message by lighting all the LEDs on the mote, the routing protocol formed the tree as expected and even if one of the nodes is out of the transmission range there will be no problem with the routing protocol.

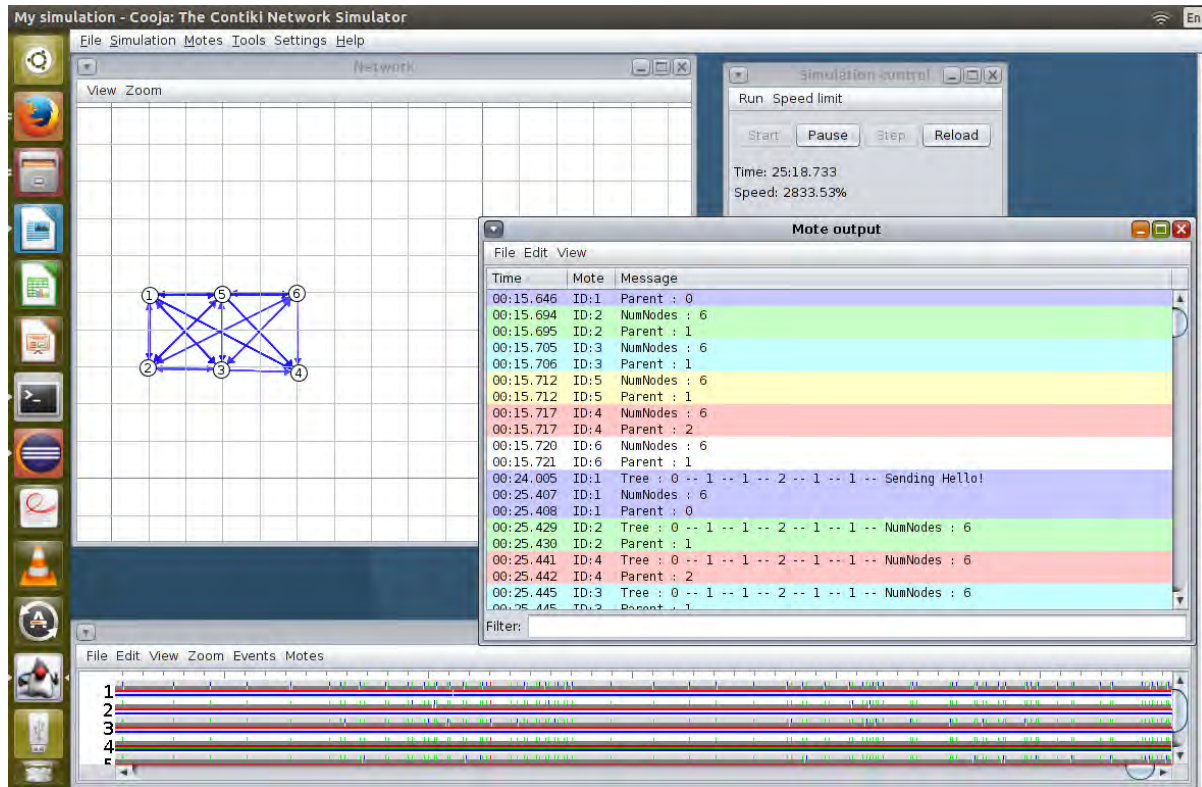


Figure 5.4: Routing Protocol Simulation with Six Motes

### 5.3.1.2 Sensing

The sensing component is used to sense periodically and write out the sampled values to a logfile as shown in Figure 5.5. The sensing operation begins when the train on the bridge is arriving and detected by the head node and then the head node will notify its children to get ready for sampling. The following sequences of steps are followed during the sensing operation:

Before the sensing is about to begin from the three LEDs only LED1 will light up informing that sampling is about to start. Once sampling begins LED0 and LED2 will toggle in between making sure the sampling is still going on. Whenever the head node wants to send a ‘start sampling’ command it uses the Sense interface and calls the Sense.sense() command. Sense is implemented by *SenseC* and included in the *SenseAppC* configuration. On receiving the command, the *SenseC* component disseminates the command to its children via broadcasting. A node acts on a Sense packet only if it is received from the parent. This ensures that packets propagate from head node down to the leaves of the routing tree and do not loop in the network. *SenseC* then disables the

radio and reinitializes all parameters to start the sense operation. It also starts the periodic timer for the purpose of sensing periodically. The sensing timer is fired every 2.5 milliseconds.

We used the resource interface provided by Msp430AxisC component to request access to the ADC. Once the resource has been granted, the sampling is done and the resource is released to be used for the next sample. During sampling the array of structures are written out to flash storage. The flash storage is accessed by using the LogRead and LogWrite interfaces provided by LogStorageC module. At the end of the sensing operation, the log is synced to ensure that all content has been written so that later on the content can be read by other components like data collection which must transfer data to the head node.

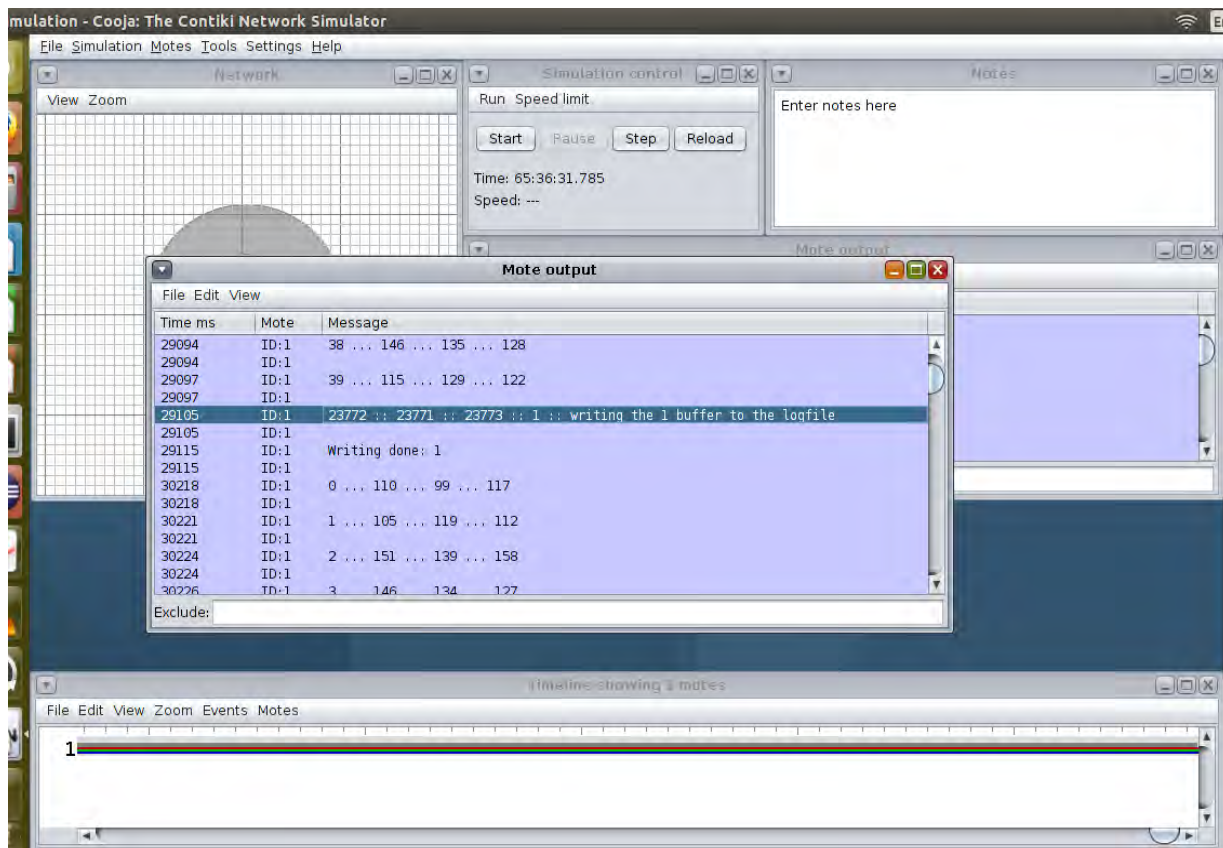


Figure 5.5: Simulation of Sensing Component

### 5.3.1.3 Collecting

The collect component (collectC) is implemented for a means of collecting sampled data from the sensor nodes. The application collects data by default where there are no trains on the bridge to make sure the status of the infrastructure is stable enough for a train to travel through. The collecting module in each node sends the information stored to the head node of the Section. Data collection is required after sensing is done. Head node will initiate data collect and it will store all the data till the next train comes. Basically the aim of data collection is to collect all the data sensed by all nodes and store it on the head node so that only one node needs to send the data to the mobile node on the train. Data collection uses routing tree created by routing interface. Just like the head file of the routing protocol the total number of nodes, size of data block, size of data on each node and size of each packet are expected for collection. All these inputs are specified in the header file, *Collect.h*.

In the implementation, we integrated the collect component with the transfer component, transfer, since transfer is the simpler version of data collection. Data transfer component involves transferring data from head node to the mobile node. It uses similar mechanism as data collection. It collects data only from one node which makes it simpler.

We tested the CollectC using six nodes, and the result is shown in Figure 5.6. The collect view tool lets us send commands to the nodes. Therefore we gave the collecting duration of every 60 seconds and when the collect command is sent to the nodes the nodes will give out their sampled data to the head node. Later on the head node will transfer it to the mobile node when the second train arrives at the bridge.

In the simulation process to actually simulate the movement of the train we installed a plug-in called Mobility since it won't be installed with the first time installation. This plug-in allows us to create and control the movement of the train passing through the bridge it has a '*position.data*' structure file that depicts the mobility of the nodes that are being simulated. Therefore we changed the original position data structure file to the point where it can satisfy our specific case. We then simulated a train passing over the bridge, once every 60 seconds thus the head node will start collecting samples from the nodes and transfer every 60 seconds.

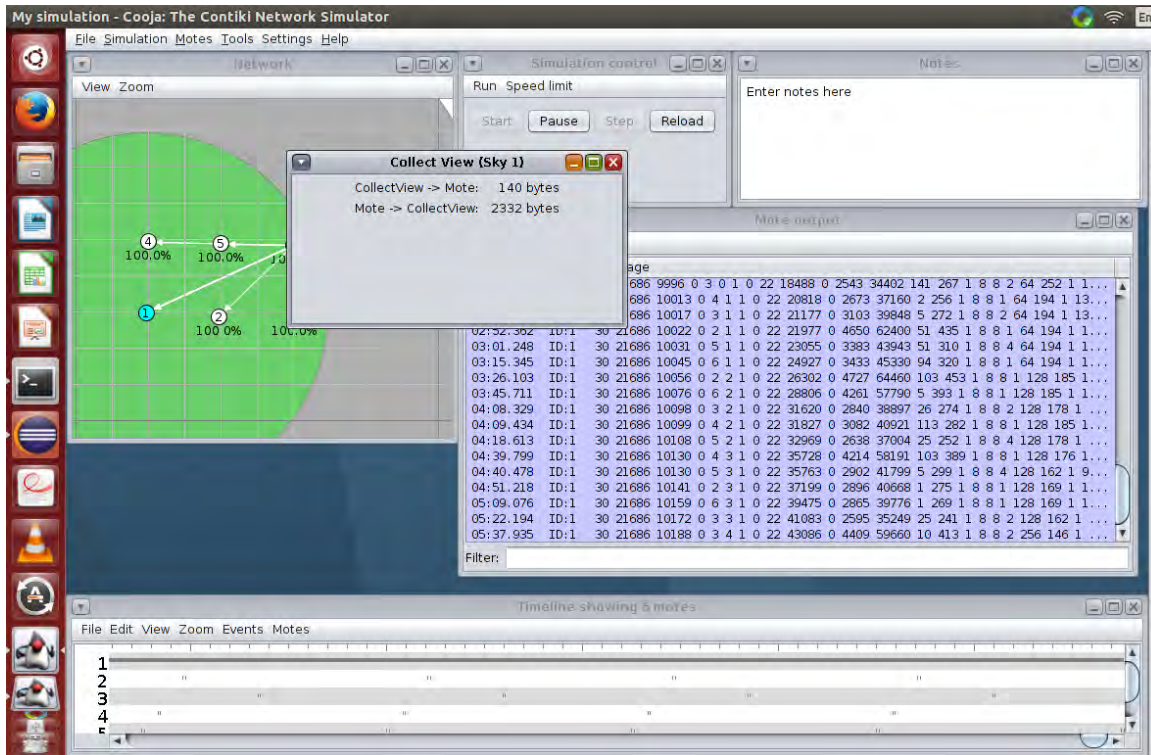


Figure 5.6: Simulation of Collecting and Transferring Sampled Data

## 5.4 Data Management

A table that holds the sampled data from the sensor network which is used for storing the sampled data. The table is contained with an event with field values of Node id, Time Stamp, and Message. The information kept by the Node id column is the id of the deployed sensor node, time stamp is the value when the sensor node sends the sampled value and the message is the digital value of the sensed value.

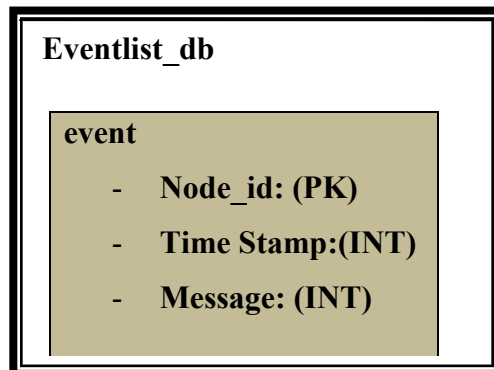


Figure 5.7: Data Table

## Chapter Six: Experiment and Results

To test our wireless sensor network based railway infrastructure monitoring application and to investigate its performance we performed a simulation experiment and evaluation using different metrics. To accomplish these tasks we followed the following procedures: first we set up the simulation environment with the necessary tools that enables us to achieve our objective. Second we simulated individual components of RAIM that we have implemented. Lastly we evaluated our wireless sensor network based application and each component's behavior.

### 6.1 Simulation Setup

Our application scenario is based on a railway bridge and in our network topology we deployed a total number of 12 nodes, 6 on each span of the bridge, the simulation parameters used in this work are as shown in Table 6.1. Each of the nodes samples the accelerometer and temperature data of the bridge. The nodes on different span of the bridge communicate on a different channel and this way we clustered the sensor nodes.

We have chosen to use IEEE 802.15.4 as medium accesses control protocol because as explained in Chapter 2 this protocol is the standard for low rate personal area networks. The default implementation of the MAC protocol used by the COOJA simulator is CSMA-CA. Thus we employ CSMA-CA channel access mechanism in our simulation experiment.

*Table 6.1: Simulation Parameters that are used for This Work*

Parameter	Value
No. of sensor nodes	12
Simulation time	~1000 seconds
Transmit energy(TX)	17.4 mA(Milliamp)
Received energy(RX)	19.7 mA(Milliamp)

## 6.2 Simulation process

Once the simulation setup is defined we proceeded with conducting the simulation experiment using a computer running Ubuntu 14.10 operating system. The computer is contained with COOJA simulator that is running our WSN based application. Figure 6.1 is the process flow of the simulation.

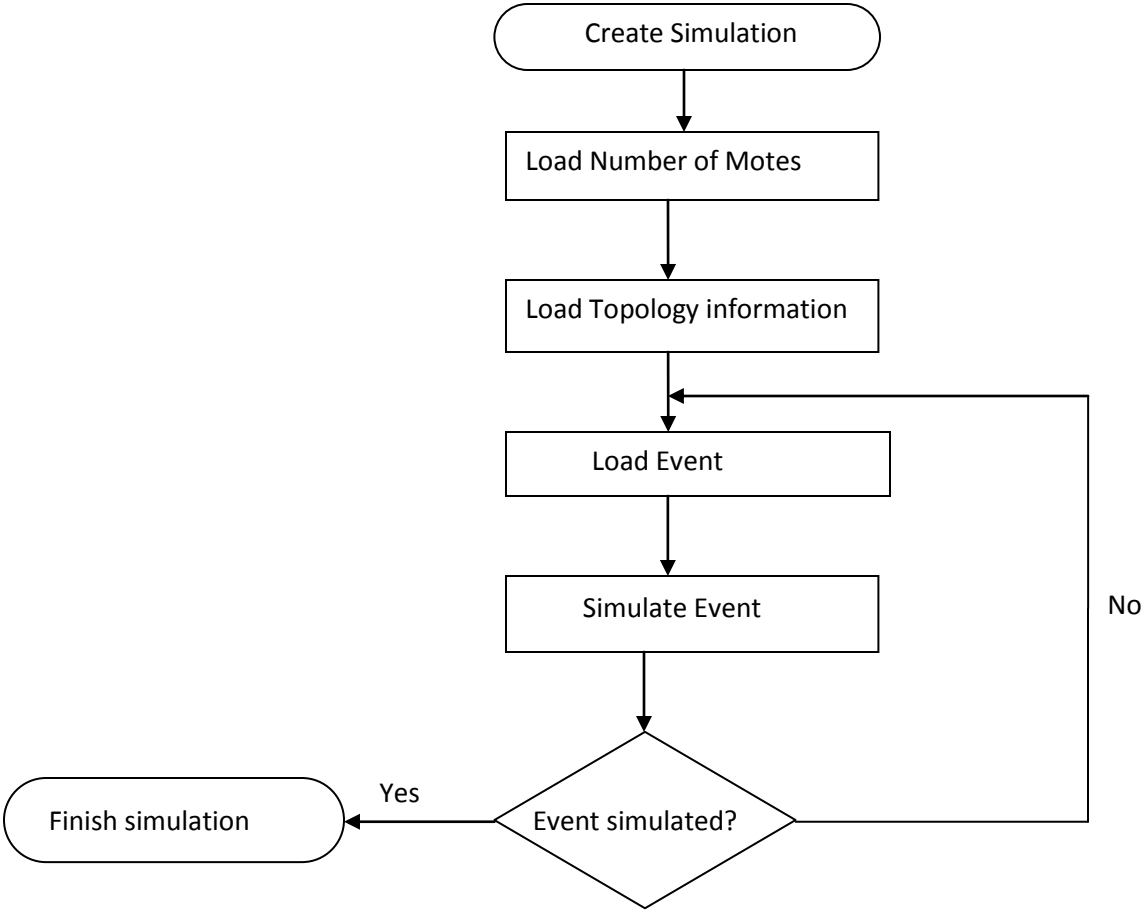


Figure 6.1: Simulation Process Flow

### 6.3 Evaluation and Results

We evaluated the performance of our application through a set of metrics that can summarize the most significant features of the application. Following we will look into the evaluation metrics that will describe the application.

**Packet Received (over time):** this metric is important because it is necessary to know whether there is a packet lost during the transferring of the sampled data because when a head node transfers data to the mobile node on the train it is important to make sure that there are no packets dropped or lost over time. Therefore we need to check the number of packets received from each of the nodes in their specific time. As Figure 6.2 shows within the time that every packets are being received, 80 packets from the 12 nodes deployed were received within the time of 09:20:00 ms proving that within in this time there are no dropped packets. It is shown in Figure 6.3 to have a clear view whether there are packets dropped or lost.

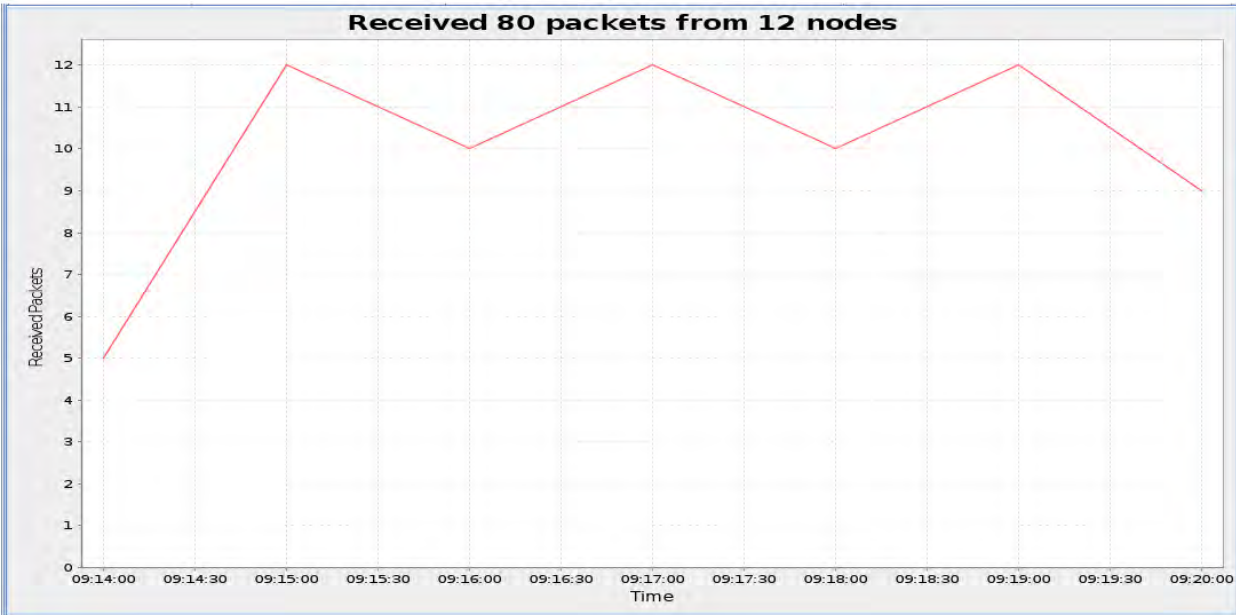
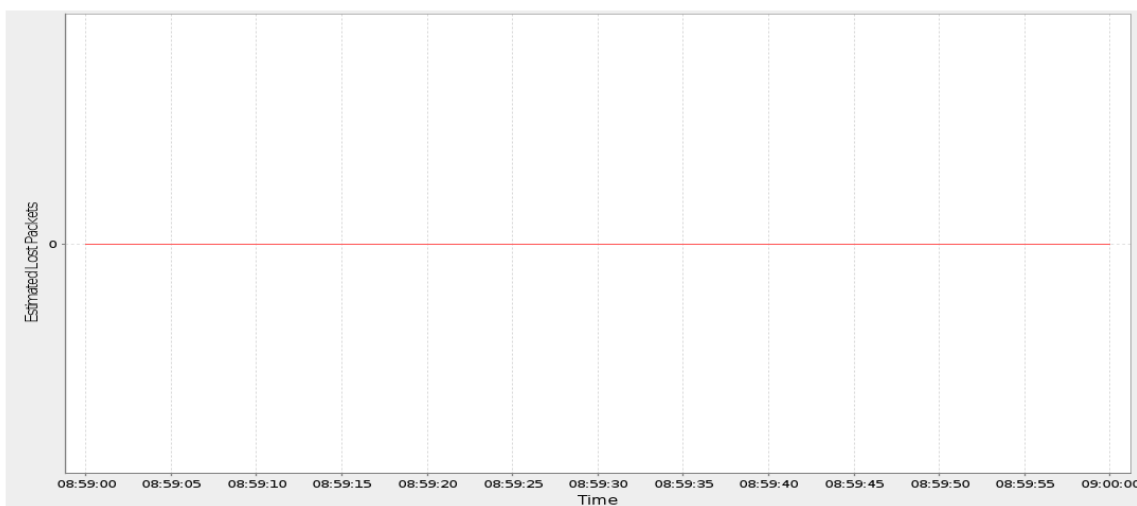


Figure 6.2: Packets Received Over Time

**Packet Lost (over time):** we have now already checked that packets are received over time and now we need a way to make sure that there are no packets lost over time. Therefore in this case with the work of our application specific routing protocol we can see that there are no packets lost at any time of the transferring or collecting. Having no packets lost over time ensures that our WSN based application and routing protocol which help in achieving this reliability of packet deliverance is consistent. Figure 6.3 shows the expected result of the simulation, the line in the graph is showing that the estimated packets lost within the time 09:00:00 ms is 0. What this number implies is that our application is a reliable one.



*Figure 6.3: Estimated Result of Packets Lost*

**Packet Received (Per Node):** In this case we check two things; packets delivered by each node and duplicated packets per node. Therefore Figure 6.4 depicts that in our specific prototype simulation it shows that there are no duplicated packets deliver to the head node.

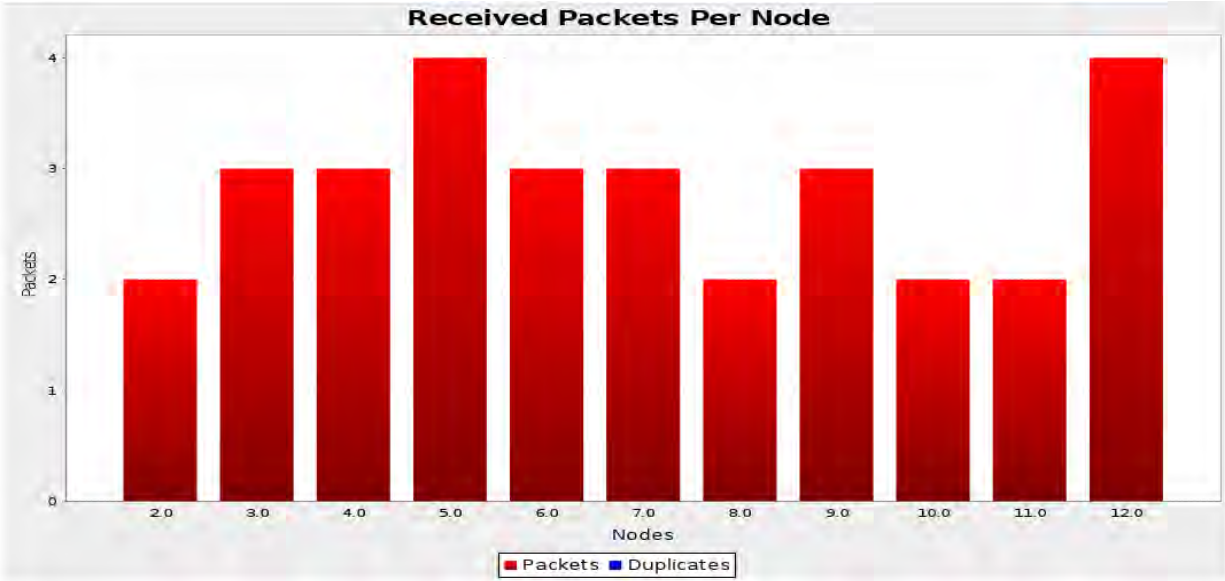


Figure 6.4: Received and Duplicated Packets per Node

**Latency:** It might take a long time for each packet to reach its destination, because it gets held up in long queues, or it takes a less direct route to avoid congestion. Latency is one of the challenges and issues in WSN design. In our simulation of prototype the output concerning latency is very low. Every node is at a constant level indicating that there is no packets held up in long queues. Figure 6.5 depicts the out result of the simulation:

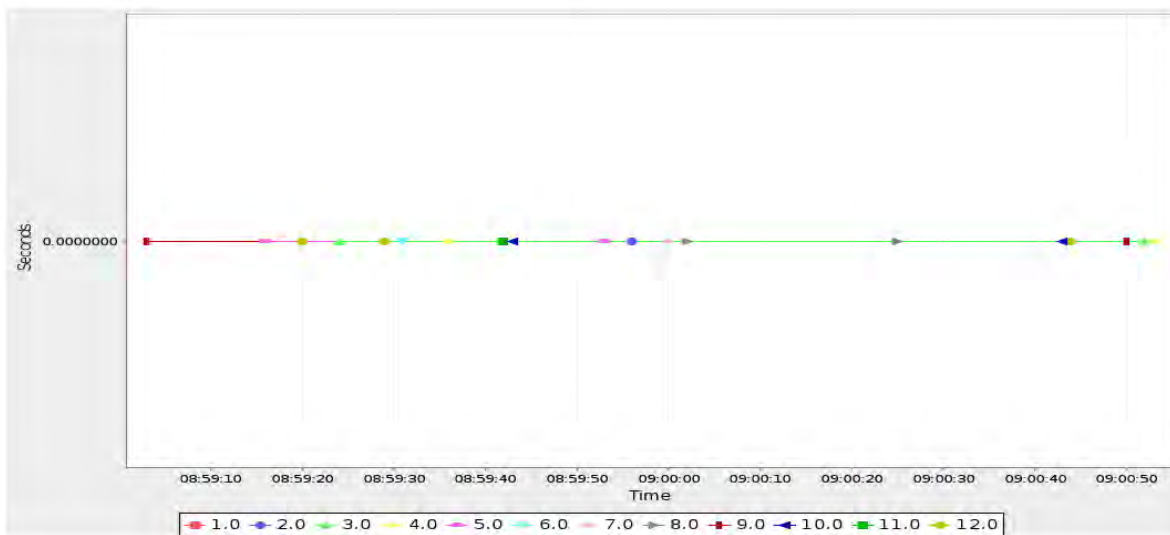
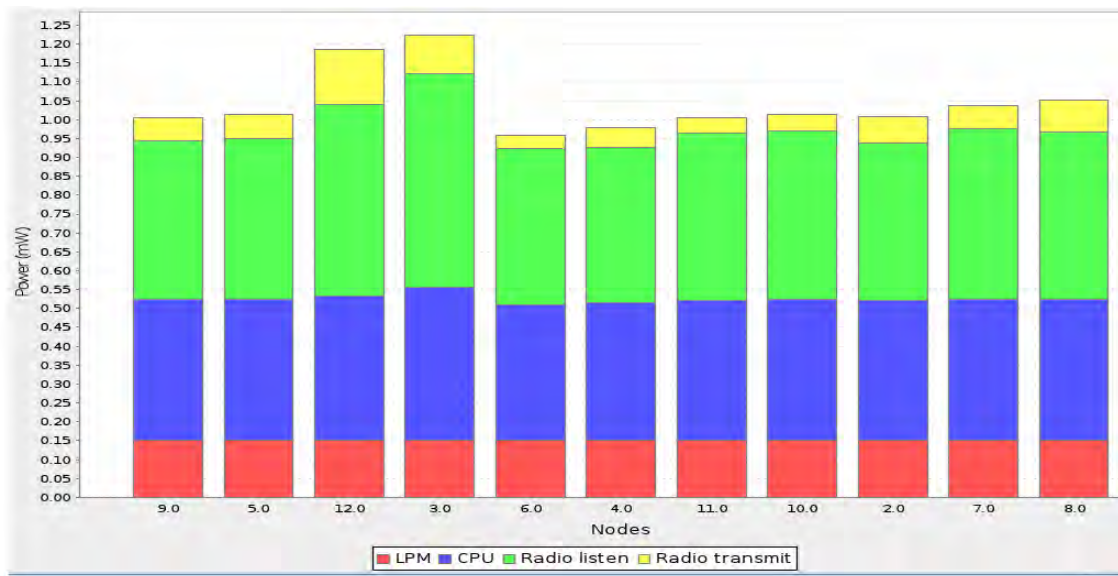


Figure 6.5: Latency Result of Each Node

**Power:** We chose power as one of the evaluation metrics because as it has been stated that power consumption is one of the most important factors in wireless sensor network research, and it is indicated that the sensor nodes that play the big role in any wireless sensor network application have the case where their batteries die very quickly than expected. Therefore it is important to evaluate the power consumption of our application figuring how much power it consumes for it to transfer and collect the sampled data and whether we should somehow manipulate our code to enable it to minimize the power drainage. But in our case Figure 6.6 depicts the average power consumption during radio listen, radio transmit and the power consumed by the CPU are reduced and this are the most important parts of the sensor node that usually drain the battery. What we are trying to show is that our application specific routing protocol is energy efficient because the amount of energy consumed by each of the components is below the maximum milli watt power set, that is, 1.25mW. In Figure 6.6 we will see the average power consumption of each node.



*Figure 6.6: Average Power Consumption*

## 6.4 Node Information

This is the overall node information that is captured during the simulation of the application. The above evaluation metrics, i.e., LPM, CPU, Radio listen and Radio transmit, are the chosen ones that need to be given attention. Below we have different types of information each node complied with:

For instance if we see the power consumed by the CPU node with id of 1.0 its CPU power with the value of 0.300% showing that it is the minimum consumption of power required. Again if we look into the power consumption during the listening for the arrival of packets node 1.0 consumed power of 0.01%, which satisfies our one basic objective regarding energy consumption. Same is for the rest of the nodes.

*Table 8.1 Node Information with Selected Simulated Value of Each Node*

Node	CPU Power	LPM Power	Listen power	Listen Duty Cycle
1.0	0.300	0.000	0.010	0.012
2.0	0.370	0.152	0.120	0.700
3.0	0.382	0.152	0.513	0.855
4.0	0.371	0.152	0.420	0.700
5.0	0.388	0.152	0.522	0.871
6.0	0.380	0.152	0.430	0.717
7.0	0.380	0.152	0.479	0.799
8.0	0.380	0.152	0.457	0.761
9.0	0.377	0.152	0.466	0.776
10.0	0.369	0.152	0.443	0.738
11.0	0.368	0.152	0.459	0.765
12.0	0.367	0.152	0.457	0.792
<b>Average</b>	<b>0.376</b>	<b>0.152</b>	<b>0.462</b>	<b>0.770</b>

## 6.5 Evaluation of Routing Protocol

To evaluate our application specific routing protocol we used MSPSIM simulator which can also be found in COOJA and as a standalone. We used the standalone version of MSPSIM sensor node emulator. MSPSIM [47] is a Java-based instruction level emulator of the MSP430 microprocessor series. In contrast with CPU level emulators, it emulates complete sensor

networking platforms such as the Tmote Sky platforms. MSPSIM targets both realistic simulation with accurate timing for use as a research tool, and good debugging support for use as a development tool. MSPSIM combines cycle accurate interpretation of CPU instructions with a discrete event based simulation of all other components, both internal and external. MSPSIM uses an event-based execution kernel that enables accurate timing while keeping the host processor utilization as low as possible. When evaluating the routing protocol using MSPSIM before proceeding into evaluating the protocol we need to first change the into *.elf* format, ELF is a format known as executable and linkable file that defines the structure of the file. In our case it enables the emulator to understand the file. What is done is that we first compiled the component RoutingC and turn it to .elf using the command `'mv build/telosb/main.exe build/telosb/main.elf'` and finally we compile our component using `'java-jar /opt/mspsim/mspsim.jar build/telosb/main.elf'` in the terminal.

From this experiment we would like to know the amount of energy that will be consumed during the routing process. During simulation the setup is intended to mimic the way we expect nodes to be placed on a bridge span. The nodes go through a sleep-wake up cycle; with sleep duration of 60 second (corresponding to the result of the arrival of train on bridge). Routing was made to run at every wakeup and the head node logs all the routing trees formed. This is so as to know the hop distribution during the course of the experiment. The cycle of routing followed by sleep-wakeup was run continuously without any interruption for hours.

What we observed is that

- The routing protocol can be run quite infrequently, only when a node fails or when a node gets disconnected. Thus, our routing protocol meets the application requirement.
- In the most frequently formed tree, all the links were good, i.e., the link RSSIs were greater than -75dBm, i.e., given in the head node (*Routing.h*).

When looking into the motes radio duty cycle result of the routing protocol implies that the protocol regarding the energy efficiency battery of the sensor nodes is not that drained. To save energy the sensor node is at sleep until the hello message is sent and after receiving it goes to sleep to again respond to hello message if received this is also true for transmitting. Figure 6.7 depicts the duty cycle output during the listening and transmitting. It also displays the stack

monitoring including the maximum stack and minimum stack monitoring during the routing but our concerning is regarding the radio's duty cycle. As an example for the case of telosb mote 1 as the diagram below shows the radio during transmission, i.e., Radio TX(%) result was 0.30% and the radio during receive i.e., Radio RX was 0.19% showing results with less energy being consumed.

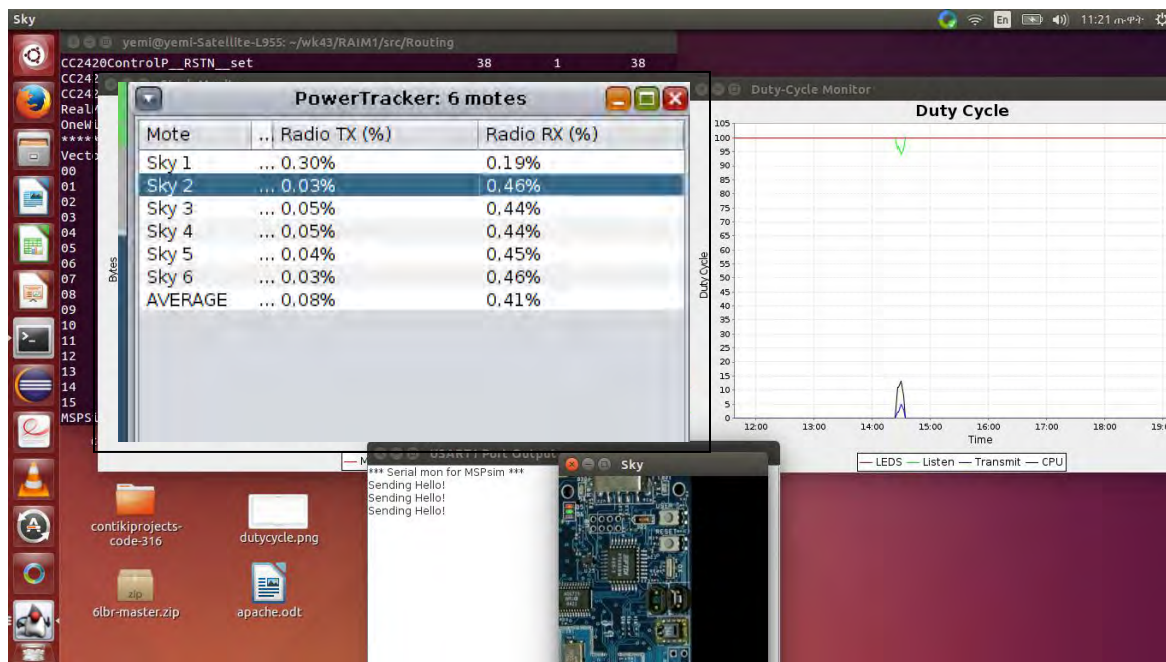


Figure 6.7: duty cycle of the routing protocol

## 6.6 Summary

In this Chapter, we described the implementation details, simulation experiment, and evaluation result of our railway infrastructure for monitoring application. We implement the wireless sensor application for railway infrastructure monitoring and also application specific routing protocol. We also explained the individual components of the application and how they function when simulated. To evaluate the performance of our application and application specific routing protocol, we performed simulation experiment using COOJA simulator. The simulation experiment result shows our application meets the railway infrastructure monitoring requirement with no packet loss and low power consumption.

## **Chapter Seven: Conclusion and Future Work**

### **7.1 Conclusion**

In Ethiopia railway construction is now one of the biggest infrastructural developments being undertaken, since Railway Bridge is one part of the infrastructure. As it is a critical civil infrastructure, keeping track of its safety by obtaining useful information about the railway infrastructure is the most important task.

Traditionally railway infrastructure monitoring is performed by examining the health of the railway using visual inspection conducted by a team of persons who are qualified either by experience or education to evaluate a particular structure. It is based on a review of existing data and information, first hand input from the field and operational personnel, and site inspection. This is a labor intensive activity and may reduce the quality of the data gathered. As a result, the reliability and accuracy of the railway infrastructure monitoring task will be doubtful. To monitor the safety of a railway in real time, it is necessary to adopt an automated monitoring system which reduces the probability of catastrophic failures and saves lives.

Currently Ethiopia is considering to incorporate a system known as SCADA for the Addis Ababa LRT project which is widely used nowadays to monitor and control dispersed hardware components in industrial setting such as power plants, gas or water treatment. They are mostly used in critical infrastructures where security is a vital factor for the environment. We then make a comparison between WSNs and SCADA and came into conclusion that WSNs are promising technology for the CIP field.

In this thesis, we proposed a railway infrastructure monitoring system using a wireless sensor network that consists of wireless sensor nodes which consist of wireless sensor nodes which are highly energy and processing power constraint devices, a technique for data collection, routing, and energy efficient data processing. We propose a model/framework for a system to perform automated monitoring of Railway Bridge.

The system designed in this thesis contributes to the field of wireless sensor networks for the railway infrastructure monitoring, the special case of Ethiopia. The main accomplishments of

this work are: modeled a framework for railway safety monitoring using wireless sensor network, developed a railway infrastructure monitoring application running in TelosB sensor nodes under TinyOS operating system that allows measuring of different parameters of the railway, developed and adopted clustering and routing protocols and implemented them using COOJA simulator. We have evaluated, analyzed, and proved that the monitoring system using a wireless sensor network and the application specific routing protocol and concluded that there were no lost packets during transmission and no duplicated files were there and less power consumption is observed in the case of the routing protocol.

## **7.2 Future Works**

Although we tried our best to realize the proposed architecture for railway infrastructure monitoring application using a wireless sensor network with the objective of addressing that there is no currently applicable monitoring method, we do not believe that the architecture is generic enough to incorporate potential issues in railway infrastructure monitoring. For instance, we have not included the safety analysis tools and the monitoring of the impacts caused by external forces since it is out of the scope of the work. Therefore, we believe that the proposed architecture can be enhanced in such a way that analysis tools and measurement of impact of external sources can be taken into account.

Our work focused on railway infrastructure monitoring using a wireless sensor network specifically on reliable data collection and transferring, and routing. However if there is an integrated analysis tool it would make monitoring personnel life much easier. Therefore, it is important to deal on this issue to make this work more complete.

The other line of improvement is regarding testing it on physical sensors to assess its practical significance. Hence, by taking this prototype as a platform, a more realistic implementation of the proposed architecture is another area of improvement so as to maximize its usability in a real world scenario setting in the future.

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless Sensor Networks: A Survey”, Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA.
- [2] Carlos F. García-Hernández, Pablo H. Ibarguengoytia González, JoaquínGarcía-Hernández, and Jesús A. Pérez-Díaz, “Wireless Sensor Networks and Applications: a Survey”, Electric Research Institute (IIE), ITESM, Cuernavaca Campus, Mexico, March 2007.
- [3] EmadAboelela, William Edberg, Christos Papakonstantinou, and VinodVokkarane, “Wireless Sensor Network Based Model for Secure Railway Operations”, Computer and Information Science Department, University of Massachusetts Dartmouth, North Dartmouth, USA, 2006.
- [4] Jaime H. Chen Gallardo, “A Dependable Middleware for the Development of Applications for Wireless Sensor and Actor Networks “, University of M’alaga, M’alaga, Spain, 2014.
- [5] Tat S. Fu, AmitabhaGhosh, Erik A. Johnson, and BhaskarKrishnamachari, “Energy-efficient deployment strategies in structural health monitoring using wireless sensor networks”, Department of Civil Engineering, University of New Hampshire, Durham, 2012.
- [6] Hivya. A, Hemalatha. M, “Structural health monitoring system- an embedded sensor approach”, M.Tech-Embedded Systems, SASTRA University Thanjavur, Tamil Nadu, India.
- [7] Sukun Kim, ShamimPakzad, David Culler, James Demmel, Gregory Fennes, Steven Glaser, and Martin Turon, “Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks”, Electrical Engineering and Computer Sciences and Civil and Environmental Engineering University, California, Berkeley.
- [8] A. M. Mejri, G. R. Rekaya& J-C.B. Belfiore, “Physical Layer Network Coding For Bridge Wireless Monitoring”, Telecom-ParisTech, Paris, France, 2012.
- [9] MadhavBoakre, and AnaghaRalegaonkar, “Wireless Sensor Network”, Department of Computer Science, ITM, Nanded, Maharashtra, India, March, 2012.

- [10] MadhavBokare, and AnaghaRalegaonkar, “Wireless Sensor Network: A Promising Approach for Distributed Sensing Tasks”, SSBES’s Institute of Technology and Management, Nanded, December-January, 2012.
- [11] S.Tilak, N. Abu-Ghazaleh, and W. Heinzelman, “A Taxonomy of Wireless Micro-Sensor Network Models”, ACM SIGMOBILE Mobile Computing and Communications Review, April, 2002.
- [12] WalteneusDargie and Christian Poellabauer, “Funrailwayentals of Wireless Sensor Networks: Theory and Practice”, Springer, 2010.
- [13] D. Steere, A.Baptista, D. McNamee, C. Pu, and J. Walpole, “Research Challenges in Environmental Observation and Forecasting Systems”, in proc. ACM/IEEE MOBICOM’00, Boston, August, 2000.
- [14] Ian F. Akyildiz, and Mehmet Can Vuran, “Wireless Sensor Networks”, USA, 2010.
- [15] Bharathidasan, A., Anand, V., and Ponduru, S., “Sensor Networks: An Overview”, Department of Computer Science, Technical Report, University of California, Davis, 2001.
- [16] M. A. Matin and M. M. Islam, “Overview of Wireless Sensor Networks”, 2012
- [17] Luís M. L. Oliveira and Joel J. P. C. Rodrigues, “Wireless Sensor Networks: A Survey on Environmental Monitoring”, April, 2011.
- [18] AnisKoubaa, MárioAlves, and Eduardo Tovar, “Time Sensitive IEEE 802.15.4 Protocol”, IPP-HURRAY Research Group, Polytechnic Institute of Porto, Portugal, DATE.
- [19] AmroQandour, “Application Framework for Wireless Sensor Networks”, Unpublished Master’s Thesis, Edith Cowan University, YEAR.
- [20] Nitaigour P. Mahalik, “Sensor Networks and Configuration”, Springer, 2007.
- [21] KazemSohraby, Daniel Minoli, and TaiebZnati, “Wireless Sensor Networks: Technology, Protocols, and Applications”, USA, Springer, 2007.
- [22] PawelPiotrCzapski, “A Survey: MAC Protocols for Applications of Wireless Sensor Networks”, School of Computer Engineering, Nanyang Technological University, Singapore, 2006.
- [23] UrošPešovi’c, AleksandarPeuli’c, and Žarko’ Cu’ cej, “MAC Protocols for Wireless Sensor Networks”, University of Maribor, Faculty of Electrical Engineering and Computer Science, 2008.

- [24] Abdelmalik Bachir, MischaDohler, Thomas Watteyne, and Kin K. Leung, “MAC Essentials for Wireless Sensor Networks”, IEEE Members, March 12, 2009.
- [25] Mark A. Periillo and Wendi B. Heinzelman, “Wireless Sensor Network Protocols”, Department of Electrical and Computer Engineering, University of Rochester, NY, USA, DATE
- [26] P. Varshney.,”Distributed Detection and Data Fusion”, Springer, New York, 1997.
- [27] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “An Application-Specific Architecture for Wireless Microsensor Networks”, IEEE Transactions on Wireless Communications, October, 2002.
- [28] J. Deng, Y. Han, W. Heinzelman, and P. Varshney., “Balanced-Energy Sleep Scheduling Scheme for High Density Cluster-Based Sensor Networks”, In Proceedings of the 4<sup>th</sup> workshop on Applications and Services in Wireless Networks (ASWN), 2004.
- [29] Ankit Gupta, Sangeeta Malik, Monika Goyal, and Pankaj Gupta, “Clustering Approach for Enhancing Network Energy using LEACH Protocol in WSN”, Department of CSE, Vaish College of Engineering, Rohtak, October, 2012.
- [30] Younis, O. and Fahm, S., “HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-Hoc Sensor Networks”, IEEE, Mobile Computing, 2004.
- [31] Project Brief Outline. [http://www.et.emb-japan.go.jp/comp\\_photo\\_for\\_news\\_articles/03\\_1.pdf](http://www.et.emb-japan.go.jp/comp_photo_for_news_articles/03_1.pdf), Ethiopian Railway Corporation, last accessed on October 23, 2014
- [32] China Railway Group Limited, “Addis Ababa LRT Project Study Report”, May, 2009.
- [33] T. hoonKiml, “Integration of Wireless SCADA through the Internet,” International Journal of Computers and Communications, 2010.
- [34] C. Amarawardhana, K. Dayananada, H. Porawagama, and C. Gamage, “Case Study of WSN as a Replacement for SCADA”, in Industrial and Information System (ICIIS), International Conference, December, 2009.
- [35] Ying Sun, “Research on the Railroad Bridge Monitoring Platform Based on the Internet of Things”, Department of Electronic and Information Engineering, Liaoning Railway Vocational and Technical College, 2014.

- [36] Aboelela, E., Edberg, W., Papakonstantinou, C., and Vokkarane, “Wireless Sensor Network Based Model for Secure Railway Operations”, Performance Computing and Communications Conference, 21<sup>st</sup> IEEE International, 2002.
- [37] Chebrolu, K., Raman, B., Mishra, N., Valiveti, P. K., and Ku-mar R.,”BriMon: A Sensor Network System for Railway Bridge Monitoring”, The 6th Annual International Conference on Mobile Systems, Applications and Services (MobiSys), 2008.
- [38] Bischoff, R., Meyer, J., Enochsson, O., Feltrin, G., and Elfgren, L., “Event-based Strain Monitoring on a Railway Bridge with a Wireless Sensor Network”, 4th International Conference on Structural Health Monitoring of Intelligent Infrastructure (SHMII-4), 2009.
- [39] Jaime Chen, Manuel D’iaz, Bartolom’e Rubio, and Jos’e M. Troya , “RAISE: Railway Infrastructure Health Monitoring using Wireless Sensor Networks “, University of M’alaga, M’alaga, Spain, 2011.
- [40] Francesco Flammini, Andrea Gaglione, Francesco Ottello, AlfioPappalardo, ConcettaPragliola, and Annarita Tedesco, “Towards wireless Sensor Networks for Railway Infrastructure Monitoring”, Department of Computer and Systems Engineering, Naples, Italy, 2010.
- [41] Giuseppe Anastasi, Giuseppe Lo Re, and Marco Ortolani, “WSNs for Structural Health Monitoring of Historical Buildings”, Italy, DATE.
- [42] Xiaoya Hu, Bingwen Wang, and Han Ji, “A Wireless Sensor Network-Based Structural Health Monitoring System for Highway Bridges”, Huazhong University of Science and Technology, China, 2013
- [43] Rail Inspection, [http://en.wikipedia.org/wiki/Rail\\_inspection](http://en.wikipedia.org/wiki/Rail_inspection), last accessed on 28 October, 2014.
- [44] TinyOS: <http://www.tinyos.net/>,last accessed on 22December, 2014
- [45] NesC: <http://nesc.sourceforge.net/>,last accessed on 22 December, 2014
- [46] Telosb:[http://www.willow.co.uk/html/telosb\\_mote\\_platform.php](http://www.willow.co.uk/html/telosb_mote_platform.php), last accessed on 23 January, 2015
- [47] COOJA:[http://anrg.usc.edu/contiki/index.php/COOJA\\_Simulator](http://anrg.usc.edu/contiki/index.php/COOJA_Simulator), last accessed on 23 January, 2015
- [48] Hill, J., “System Architecture for Wireless Sensor Networks.” PhD Thesis, US Berkeley, 2003.

## **Declaration**

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Name: **Yemisrach Asnake**

Signature: \_\_\_\_\_

Place: Addis Ababa

Date: \_\_\_\_\_

This thesis has been submitted for examination with my approval as a university advisor.

Name: **Dr. Mulugeta Libsie**

Signature: \_\_\_\_\_

Place: Addis Ababa

Date: \_\_\_\_\_