USING INTERNET OF THINGS TO MONITOR VOLCANIC ACTIVITY

A PROJECT REPORT

Submitted by

A.GNANA SELVA KUMAR

Register No: 13MCO10

in partial fulfillment for the award of the degree

of

MASTER OF ENGINEERING

in

COMMUNICATION SYSTEMS

Department of Electronics and Communication Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

(An autonomous institution affiliated to Anna University, Chennai)

COIMBATORE - 641049

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2015

BONAFIDE CERTIFICATE

Certified that this project report titled **"USING INTERNET OF THINGS TO MONITOR VOLCANIC ACTIVITY"** is the bonafide work of **A.GNANA SELVA KUMAR [Reg. No. 13MCO10]** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr.G.AMIRTHA GOWRI

PROJECT SUPERVISOR

Department of ECE

Kumaraguru College of Technology

Coimbatore-641 049

SIGNATURE

Dr. RAJESWARI MARIAPPAN

HEAD OF THE DEPARTMENT

Department of ECE

Kumaraguru College of Technology

Coimbatore-641 049

The Candidate with university **Register No. 13MCO10** was examined by us in the project viva –voice examination held on .....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First I would like to express my praise and gratitude to the Lord, who has showered his grace and blessing enabling me to complete this project in an excellent manner. He has made all things in beautiful in his time.

I express my sincere thanks to the management of Kumaraguru College of Technology and our beloved Joint Correspondent, **Shri.Shankar Vanavarayar** for his kind support and for providing necessary facilities to carry out the project work.

I would like to express my sincere thanks to our beloved Principal **Dr.R.S.Kumar M.E., Ph.D.,** who encouraged me with his valuable thoughts.

I would like to express my sincere thanks and deep sense of gratitude to our HOD, **Dr.Rajeswari Mariappan M.E., Ph.D.,** for her valuable suggestions and encouragement which paved way for the successful completion of the project.

I am greatly privileged to express my deep sense of gratitude to the Project Coordinator **Ms.R.Hemalatha M.E., (Ph.D)**, Associate Professor, for her continuous support throughout the course.

In particular, I wish to thank and express my everlasting gratitude to the Supervisor **Dr.G.Amirtha Gowri M.E., Ph.D,** Associate Professor, for her expert counselling in each and every steps of project work and I wish to convey my deep sense of gratitude to all teaching and non-teaching staff members of ECE Department for their help and cooperation.

Finally, I thank my parents and my family members for giving me the moral support in all of my activities and my dear friends who helped me to endure my difficult times with their unfailing support and warm wishes.

ABSTRACT

Internet of Things technology would connect billions of sensors to the internet and applications will be developed utilizing all the sensor information to enhance the quality of life. Platforms built using Internet of Things would help us to monitor things remotely and avail different service. In this paper, we propose a flexible volcano monitoring system whereby Internet of Things technology is implemented to monitor volcanic activities and alert people in case of volcanic eruptions. We use wireless Tmote sky sensor node to monitor the temperature of the volcano and stream the data wirelessly to a gateway. We use the Contiki OS to implement a new mechanism to minimize the power consumption of the wireless sensor node.

The new mechanism is based on contikiMAC radio duty cycle. We develop a new application to collect data from the various wireless sensor nodes and display various parameters of the volcano in real time. All the collected data is graphically presented. An automated alert system is also deployed based on the temperature reading from the various sensor nodes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CCA	Clear Channel Assessment
CPU	Central Processing Unit
CSMA	Carrier Sense Multiple Access
DHCP	Dynamic Host Configuration Protocol
DODAG	Destination-Oriented Directed Acyclic Graph
Hz	Hertz
IoT	Internet of Things
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
LLN	Low-power and Lossy Network
LPM	Link Power Management
MAC	Media Access Control
OSI	Open System Interconnection
RDC	Radio Duty Cycle
RPL	Routing Protocol
RSSI	Received Signal Strength Indicator
WSN	Wireless Sensor Node

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF INTERNET OF THINGS

The Internet of Things (IoT) is a scenario which connects the physical world to the internet. IoT connects billions of sensors, objects, devices, industrial components onto networks. Enhancing amounts of data produced by those sensors and connected devices are hence acquired, stored and logged onto networks. A new dimension has been added to the world of information and communication technologies: from anytime,any place connectivity for anyone as shown in Fig 1.1. Then networked inputs are combined in a two-way communication system for better decision making, increasing efficiency, environmental benefits, new services, etc. IoT based services will provide more automation to build a smart world ineverywhere such as home, office, industry etc. Fig1.2 shows the roadmap to Internet of Things. It is a network which connects nodes/objects into the network with new technologies.

The following are some of the applications of IOT,

- Environmental Monitoring
- Smart Cities: Smart Agriculture/Animal Farming
- Smart Transport/Logistics
- Home Automation and Health Care Monitoring
- Industrial control

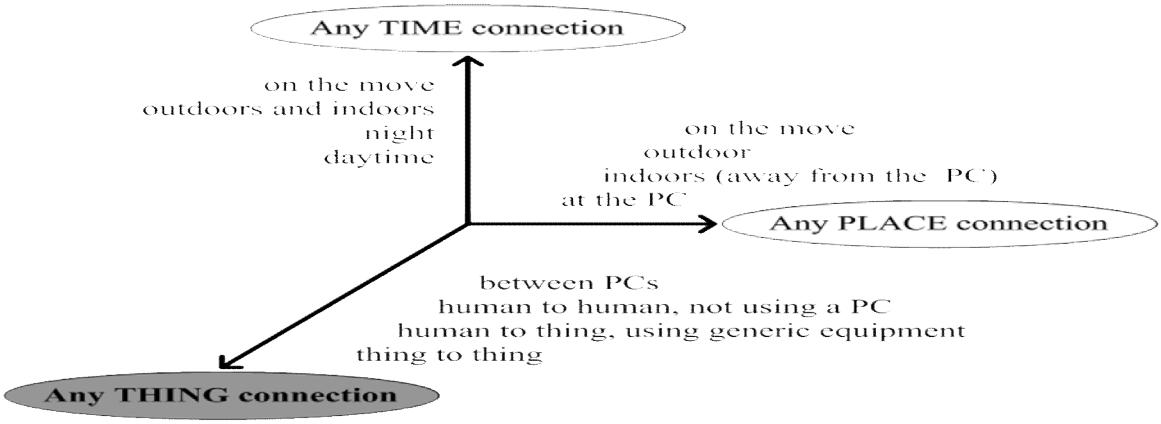

Figure 1.1 A new dimension

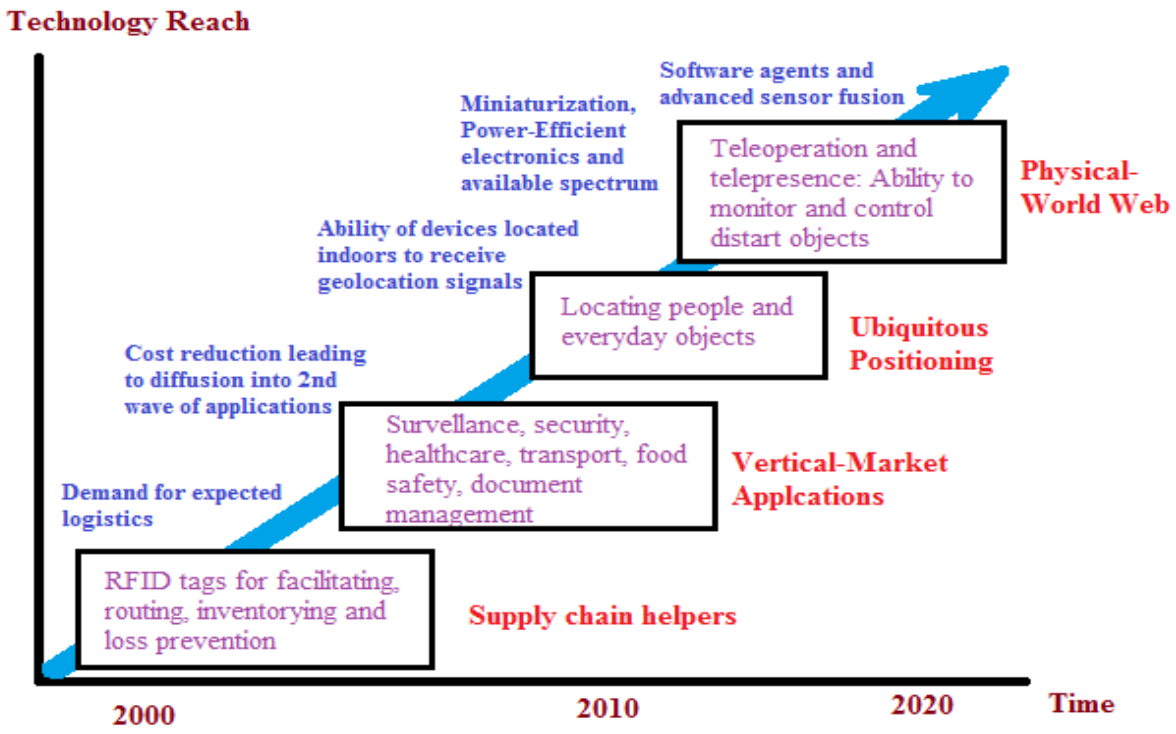

Figure1.2 Roadmap: Internet of Things

1.1.1FEATURES OF THE IOT

A number of significant technology changes have come together to enable the rise of the IoT. These include the following.

- **Cheap sensors** – Sensor prices have dropped to an average 60 cents from $1.30 in the past 10 years.
- **Cheap bandwidth** – The cost of bandwidth has also declined precipitously, by a factor of nearly 40X over the past 10 years.
- **Cheap processing** – Similarly, processing costs have declined by nearly 60X over the past 10 years, enabling more devices to be not just connected, but smart enough to know what to do with all the new data they are generating or receiving.
- **Smartphones** – Smartphones are now becoming the personal gateway to the IoT, serving as a remote control or hub for the connected home, connected car, or the health and fitness devices consumers are increasingly starting to wear.

- **Ubiquitous wireless coverage** – With Wi-Fi coverage now ubiquitous, wireless connectivity is available for free or at a very low cost, given Wi-Fi utilizes unlicensed spectrum and thus does not require monthly access fees to a carrier.
- **Big data** – As the IoT will by definition generate voluminous amounts of unstructured data, the availability of big data analytics is a key enabler.
- **IPv6** – Most networking equipment now supports IPv6, the newest version of the Internet Protocol (IP) standard that is intended to replace IPv4. IPv4 supports 32-bit addresses, which translates to about 4.3 billion addresses – a number that has become largely exhausted by all the connected devices globally. In contrast, IPv6 can support 128-bit addresses, translating to approximately 3.4 x 1038 addresses – an almost limitless number that can amply handle all conceivable IoT devices.

1.2 VOLCANO HAZARDS

The Volcano is one of the most dangerous natural hazards.The word, ‘volcano’ comes from the name Vulcan, who was the Roman god of fire. From 79A.D to now, it caused risk to human life, property and economic growth. Many volcanoes are found in heavily populated areas.Volcanic eruptions are one of Earth's most dramatic and violent agents of change. Not only can powerful explosive eruptions drastically alter land and water for tens of kilometers around a volcano, but tiny liquid droplets of sulfuric acid erupted into the stratosphere can change our planet's climate temporarily. Eruptions often force people living near volcanoes to abandon their land and homes, sometimes forever. Fig 1.3 shows primary effects of a volcanic eruption. Those living farther away are likely to avoid complete destruction, but their cities and towns, crops, industrial plants, transportation systems, and electrical grids can still be damaged by tephra, ash, lahars, and flooding.

1.2.1 TYPES OF VOLCANIC HAZARDS

The following is a list of volcanic hazards.

- Volcanic Earthquakes
- Directed Blast
- Tephra
- Volcanic Gases

- Lava Flows
- Debris Avalanches,Landslides and Tsunamis
- Pyroclastic Surge
- Pyroclastic Flows

Hence, it is necessary to monitor the volcano activities of a future eruption or any other possible changes. However, the available monitoring system consists of expensive, complex components, high power consumption and wired equipments which are hard to move. Dissimilarity with available volcanic monitoring system is that our wireless sensor nodes are considerably low power consumption, inexpensive, small, compact and lighter.

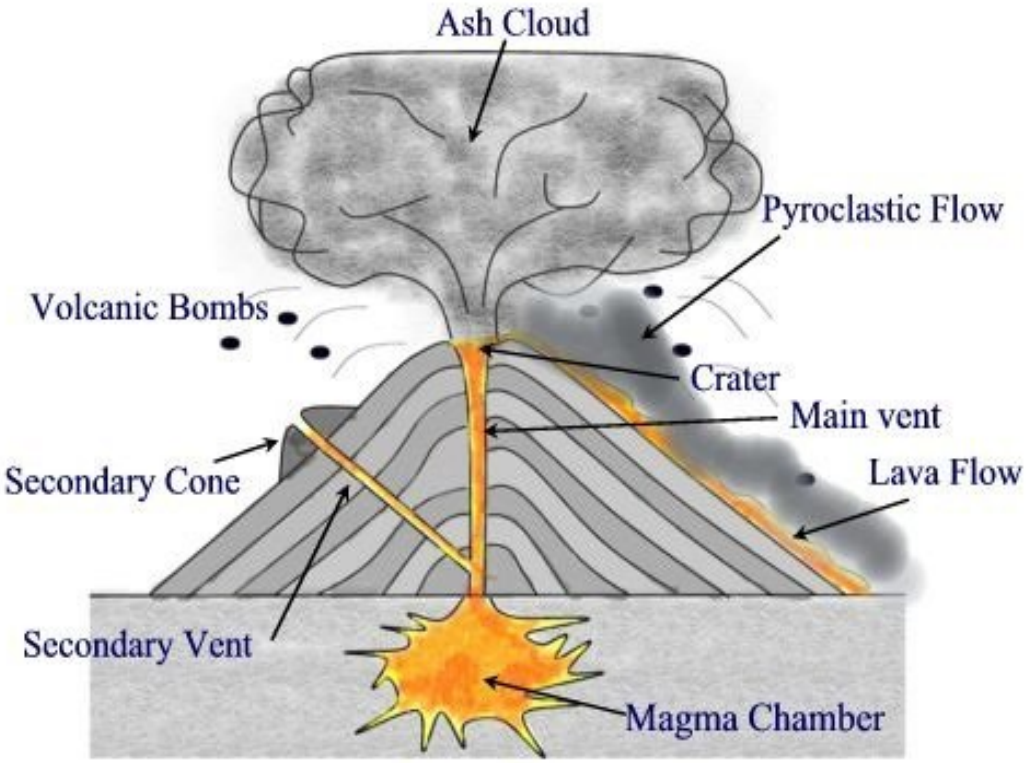

Figure 1.3 Volcano Hazards

1.3 WIRELESS SENSOR NETWORK

Volcano monitoring is one of the extreme applications where continuous human access is not possible so WSN have been used in such environments.

The rapid development in the fields of microelectronics, communication/networks and other related technologies enabled us to develop various kinds of wireless sensors. These sensor nodes are consisted of spatially distributed devices using sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants at different locations. They are capable of doing actuation, communication and computation while enabling us to sense and measure the data more efficiently and accurately independent from wire.

Wireless sensor network (WSN) can be described as a collection of these low power sensor nodes which are connected wirelessly. It is a network system that enables to communicate sensor nodes between each other.Each sensor node is capable of only a limited amount of processing and power. But when they are coordinated with other nodes in the network, they have the ability to communicate, measure and actuate in great detail.

With the help of combination of these nodes, ad-hoc networks can be created. For example, the nodes can be distributed to an environment and wireless ad-hoc networks can be formed. These distributed and formed nodes constitute a sensor network system as shown in Fig 1.4.

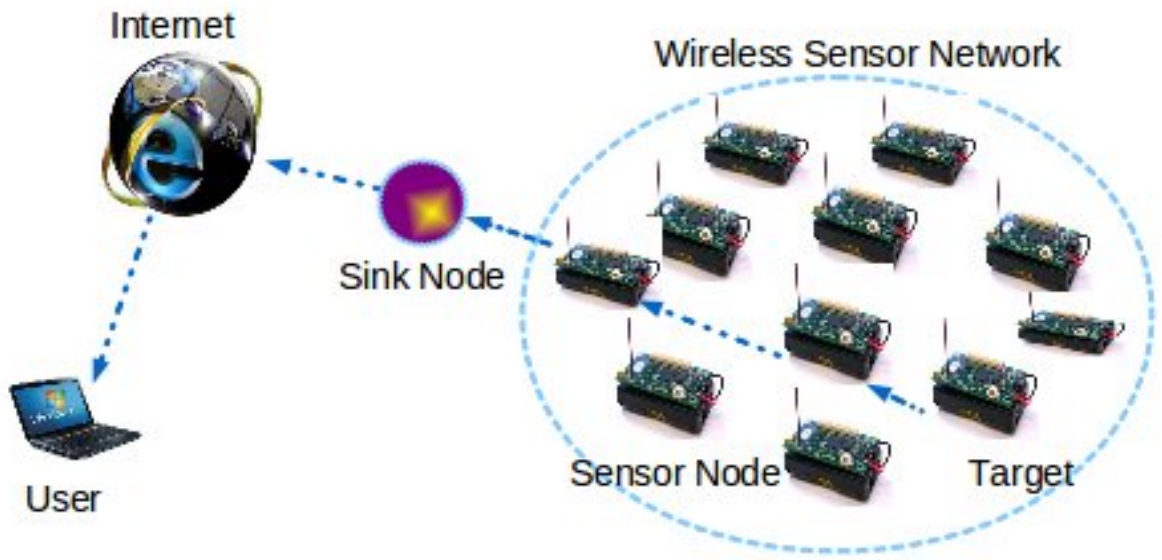

Figure 1.4 Wireless Sensor Network

A sensor network provides easy access to information from anywhere at every time. This functionality is achieved by collecting, processing, analyzing and spreading data. So that, wireless sensor network plays an important role in creating smart environments effectively.

WSN provides the capability of revolutionary detection over a wide range of different applications. Because the sensor networks have features such as:

- Reliability
- Accuracy
- Flexibility
- Cost efficiency
- Ease of Installation

1.3.1 BASIC CHARACTERISTICS OF WSN

They are:

- Self-configuration,Self-healing, Self-optimization, and Self-protection capabilities
- Short range broadcast communication and multi-hop routing
- Dense deployment and cooperative effort of sensor nodes
- Frequently changing topology due to fading and node failures
- Severe limitations in energy capacity,computing power, memory and transmit power

1.3.2 OPERATING SYSTEMS FOR WSN

The operating systems developed for WSN are

- TinyOS
- SOS
- CORMOS
- ContikiOS
- EYES
- PEEROS
- MantisOS
- SenOS
- LiteOS
- Nano-RK

The intelligent sensors may serve prudential supervision and collect information from machine crashes, earthquakes, floods and even about terrorist attacks. Sensor network makes possible,

- Information gathering
- Information processing
- Environment monitoring for a variety of civilian and military applications.

CHAPTER-2

LITERATURE SURVEY

2.1INTRODUCTION

This chapter presents the literature surveyed in the area of Internet of Things for Volcano monitoring system.Several methods were proposed by many research groups.The purpose of this is to design the best volcano monitoring system based on Internet of Things.

[1] Towards a New Volcano Monitoring System Using Wireless Sensor Networks [2006]

In this paper, Roman Lara Cueva, Rodolfo GordilloOrquera, Iv´anLondo˜no have presented that Wavelet Transform was the most effective tool to analyze signals acquired by the WSN deployed onCotopaxi Volcano, through this signal processing, a spectralpattern of seismic events was determined.A volcanic eruption is a natural event in which monitoringshould consider technical, scientific and social aspects. Bythis purpose they emphasise just in designing a networkplatform and signal processing issues.A reliable transmission system is required, it must berobust and remotely managed, therefore, they used WiFi forlong distance technology, obtaining a throughput close to 1Mbps.

[2] TheContikiMAC Radio Duty Cycling Protocol [2011]

In this paper, Adam Dunkels explainedthe ContikiMAC radio duty cyclingmechanism or low-power wireless networks, the defaultradio duty cycling mechanism in Contiki 2.5. ContikiMAC is designed to be simple to understand and implement, uses only asynchronous and implicit synchronization, and requires no signaling messages or additional headers. ContikiMAC uses a simple but elaborate timing scheme to allow its wake-up mechanism to be highly power efficient, a phase-lock mechanism to make transmissions efficient, and a fast sleep optimization to allow receivers to quickly go to sleep when faced with spurious radio interference.Measurements show that thewake-up mechanism is significantly lower than for existing duty cycling mechanisms and that the phase-lock and fast sleep mechanisms reduce the network power consumption between 10% and 80%, depending on the wakeup frequency of the devices in the network.

[3]Power saving and energy optimization techniquesfor Wireless Sensor Networks [2011]

In this paper, Sandra Sendra, Jaime Lloret, Miguel García and José F. Toledo have presented the main causes of energy loss in wireless sensor nodes. The main characteristics required to make a wirelesssensor node and the factors to be considered whenimplementing a WSN or ad-hoc network have been discussed. They discussed the energy wastage given by the electronic circuit. Therefore, counting on a soundelectronic design that includes the right components for the sensor device is absolutely essential. Finally, they show and compare several MAC and routing protocols that have been designed to optimize the power consumption without compromising the data delivery in WSNs.

[4] Integrating Wireless Sensor Networks with the Web

In this paper,Walter Colitti, Kris Steenhaut,Niccolò De Caro explained that the integration of WSNs with the Web. Thisis being facilitated by the development of CoAP, an IETF protocol providing LLNs with a RESTful architecture. CoAPoffers the same methods for the resource manipulation as HTTP. In addition, CoAP supports additional functionalities typical of IoT and M2M applications, such as multicast, asynchronous communication and subscriptions. Unlike HTTP, CoAP is built on top of UDP and has a compact packet overhead. The introduction of UDP and the packet overhead compression drastically reduce the mote's power consumption and consequently increase the battery lifetime. It also described the design and development of an end-to-end IP based architecture integrating a CoAP over 6LowPAN Contiki based WSN with an HTTP over IP based application. The application allows a user to access WSN data directly from a Web browser. It described the main building blocks of the gateway connecting the Web client with the WSN. The gateway is still inprototype phase and it requires the development of proxy and observation functionalities. The database performance needs to betested for scalability purpose. The application will be deployed and tested in a greenhouse monitoring testbed.

[5] Monitoring Volcanic Eruptions with aWireless Sensor Network [2005]

In this paper, Geoffrey Werner-Allen, Jeff Johnson, Mario Ruiz, Jonathan Lees, and Matt Welsh have demonstrated thefeasibility of using wireless sensors for volcanic studies.Their deployment at Volc´an Tungurahua provided a wealthof experience and real data from which they can developmore sophisticated tools for volcanic instrumentation.

[6]Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: two cases studies [2011]

In this paper, Leila Ben Saad, Cedric Chauvenet, Bernard Tourancheau evaluate the performance of the RPL protocol in two cases dealing with low power WSN and low power PLC. Their research provides new functionalities either in WSNetwith the implementation of RPL for our needs in the context of sink mobility and in COOJA with the support of a new networking hardware, namely low power PLC. With these improved simulators, the conducted experiments show the interest of RPL simulation in order to improve WSN lifetime by managing the sink mobility and to provide coherent routing in LLN heterogeneous platforms with wireless and PLC sensor networks.

[7] Fidelity and Yield in a Volcano Monitoring Sensor Network [2006]

In this paper, Geoff Werner-Allen, KonradLorincz, Jeff Johnson, Jonathan Lees , and Matt Welsh have attempted to understand how well a wireless sensor network can serve as a scientific instrument for volcano monitoring. They have presented an evaluation of the data fidelity and yield of a real sensor network deployment, subjecting the system to the rigorous standards expected for geophysical instrumentation. They find that wireless sensors have great potential for rapid and dense instrumentation of active volcanoes, although challenges remain including improving reliability and validating the timing accuracy of captured signals. The network was able to detect and retrieve data for a large number of seismic events, although our event detection parameters require tuning to capture more signals. In terms of reliability, base station outages affected the network about 27% of the time during our deployment, with a single software failure causing a 3-day outage.

However, nodes appeared to exhibit an uptime of 96%, which is very encouraging. Clearly, work is needed to improve the robustness of the base station and system software infrastructure.

Their most difficult challenge was correcting the timing in the captured signals and validating timing accuracy. A comparative analysis against a GPS-synchronized standalone data logger shows very good correlation: 23 out of 28 events correlated against the Reftek broadband station exhibited lag times within the expected 47 ms window. Across the sensor array, only 2 out of 124 P-wave arrivals fall outside of an expected velocity envelope, suggesting that our timing rectification is very consistent.

[8] Cross-Level Sensor Network Simulation with COOJA [2006]

In this paper, Fredrik O¨ sterlind, Adam Dunkels, Joakim Eriksson, NiclasFinne, Thiemo Voigthave presented COOJA, a cross-level simulator for the Contiki operating system. COOJA enables simultaneous simulations at the network, operating system and machine code instruction set level. They have shown that cross-level simulation has advantages in terms of effectiveness and memory usage. It allows a user to combine simulated nodes from several different abstraction levels. This is especially useful in heterogeneous networks where fine-grained execution details are only needed for a subset of the simulated nodes.

CHAPTER-3

VOLCANO MONITORING SYSTEM DESIGN

3.1 EXISTING SYSTEM

In 2005, a sixteen number of Tmote sky sensor nodes using multi-hop communication were used for volcano activities, volcanReventador, in northern Ecuador around the duration of 19 days [12]. Collected sensor data were transmitted to the base station attached to the laptop, about 4km away from the sensing platform. During this time the network successfully detected 230 events including, earthquakes, eruptions and other seismo-acoustic events. The Tmote sky sensor was designed to run by TinyOS. A laptop equipped with a directional antenna was used as a sink to collect information and manage the network remotely as shown in Fig 3.1.

The main goal of the application was to collect seismic information based on earthquakes that occur near the volcanoes. Since these earthquakes usually last less than 60 seconds, a high sampling rate is employed for the seismic sensors,i.e., 100Hz, which limits the amount of locally stored information to 20 minutes of volcanic activity. Each sensor node uses the local storage as a cyclic buffer and filters the collected information through a short-term average/long-term average threshold detector to determine the events related to volcanic activities. When an event is generated, it is reported to the sink. If high confidence of an event is established through multiple event reports, the sink sends back a data collection command to the network. As a response to this command, each node transmits the information stored in its memory. Consequently, a large volume of data is transmitted to the sink only if an event of interest is detected by a smaller number of nodes. This reduces the bandwidth wastage and energy consumption at individual nodes.With reference to this work[12] proposed work is carried out to reduce the power consumption in Tmote sky sensor nodes.

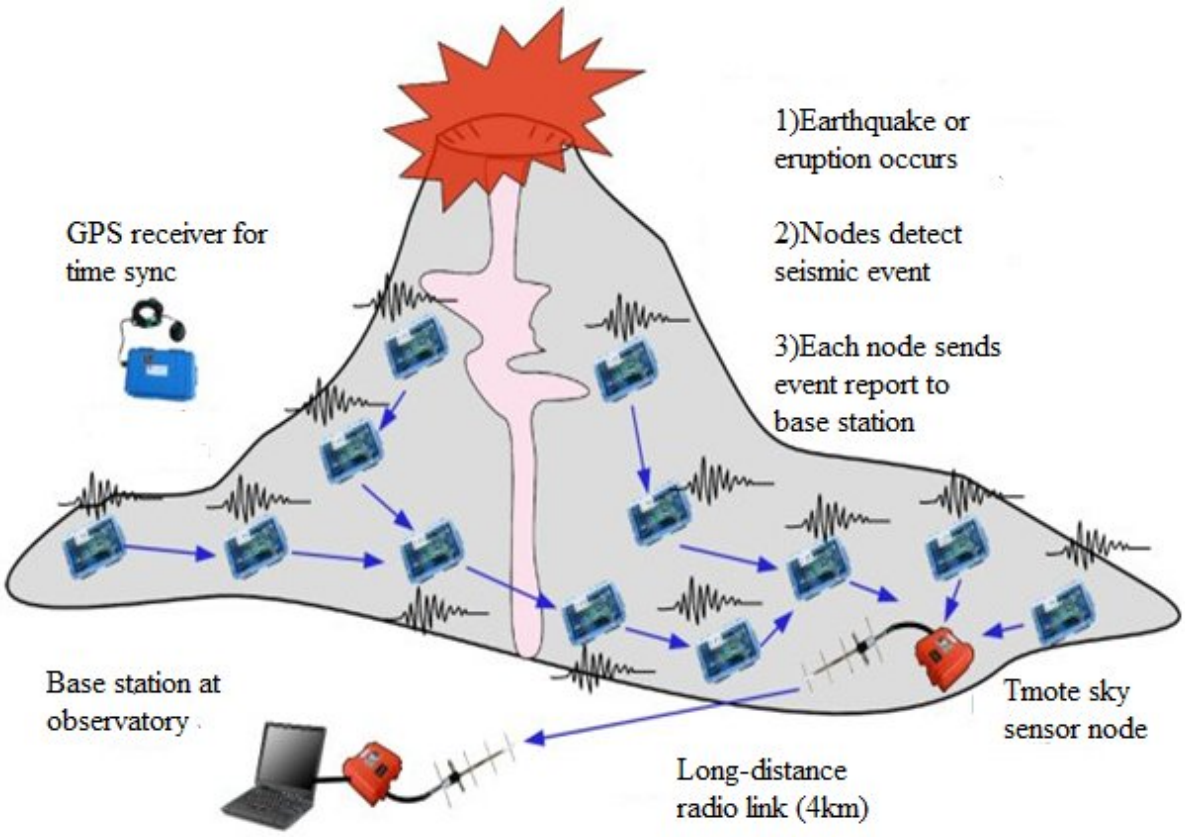

Figure 3.1Volcano monitoring sensor network architecture

3.2 SYSTEM OVERVIEW

The sensor network delivers the detected data to the border router which act as a gateway and forward to the internet through IPv6 communication protocol as shown in Fig 3.2. IPv6 communication protocol is used in Tmote sky sensor nodes, where every sensor node has a unique IP address and also provides scalable, versatile, manageable, open and flexible standard, light weight and low power. Hence, by adopting this technique sensor nodes are frequently accessible through internet.

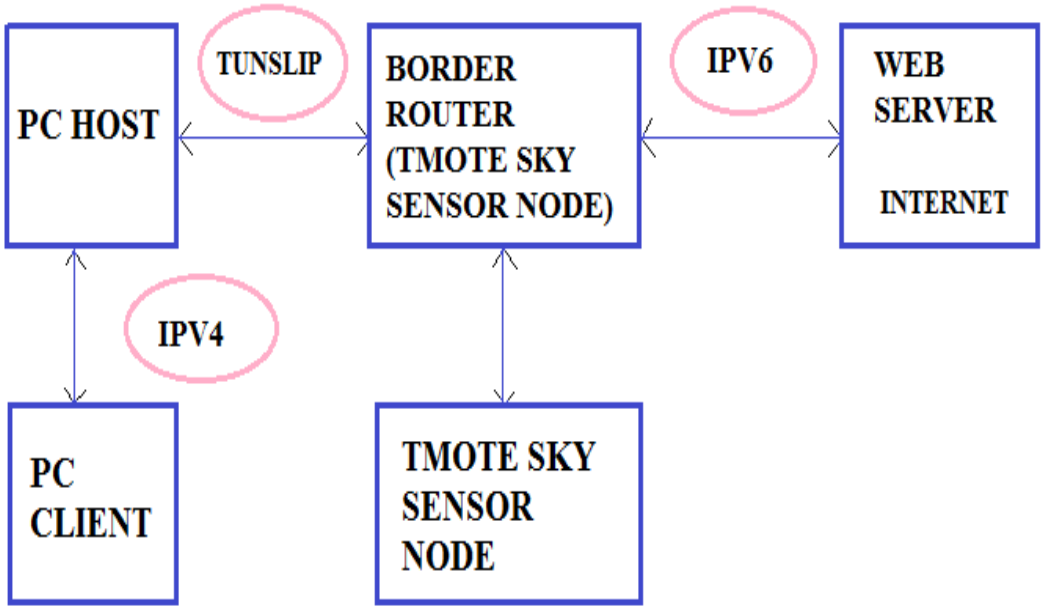

Figure 3.2 Block diagram

3.3 TMOTE SKY SENSOR NODE

Thesensor node chosen for this project is a readily available commercial type derived from the original TelosB platform design (also known as the Tmote Sky) which was developed by the University of California at Berkeley. Its low-cost hardware design, shown in Fig 3.3, is open-source and it has been used for many academic and scientific projects. The design is based on a 2.4GHz IEEE 802.15.4 compatible transceiver.

Figure3.3 CM5000 Sensor node

The Tmote sky platform is a wireless sensor board from Moteiv. It is an MSP430 based board with a 1 megabyte external serial flash memory.It is the wireless sensor module that has many things to offer like high data rate sensor network applications requiring ultra low power. Some other features to look out for are high reliability and ease of development. It is widely proven platform for wireless sensor systems deployments.

Key features of Tmote sky sensor node

- Inter operability with many different IEEE 802.15.4 devices
- Integrated on-board antenna with 50m range indoors and 125m range outdoors
- Optional Integrated humidity, temperature and light sensors
- Ultra low current consumption
- Fast wakeup from sleep
- Programming data fed via USB and data collected via USB too
- IEEE 802.15.4 WSN mote fully compatible with TelosB platform
- SMA antenna connector and 16 pin expansion support

3.4 IPv6 COMMUNICATION PROTOCOL

IPv6 is the communication protocol for this project.IPv6 protocol stack is an excellent choice to provide standardised and reliable communications in a smart node network providing its implementation can be made sufficiently small to accommodate the constrained environment. Fig 3.4 shows header of IPv6.

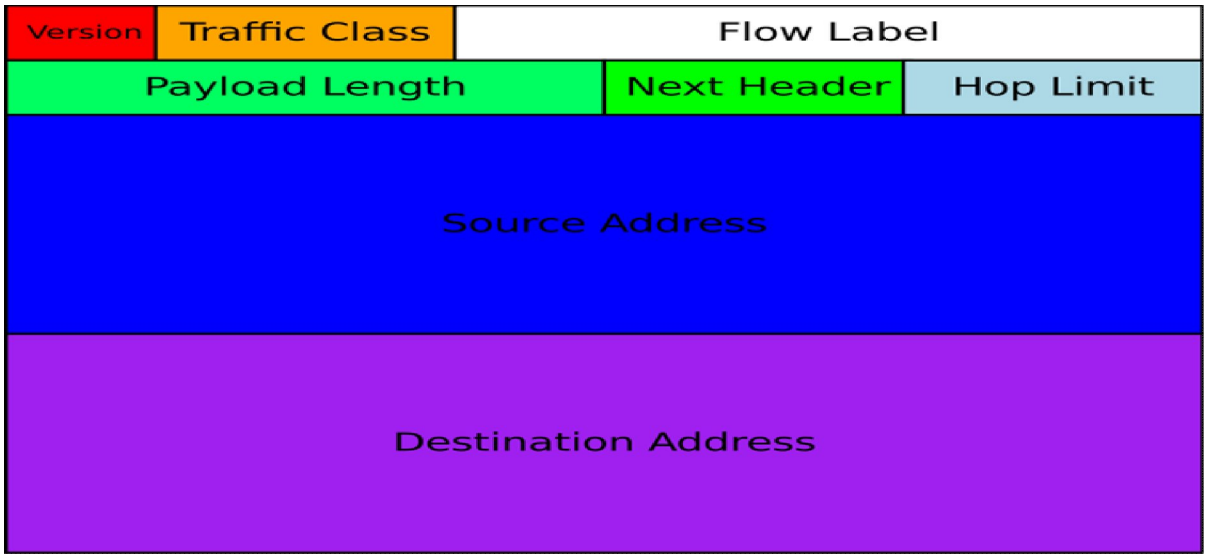

Figure 3.4 IPv6 Header

It is the successor to IP version 4 and tries to alleviate its disadvantages; in particular its address space limitations. However, it also offers many other improvements. Like IPv4, IPv6 allows a best-effort, unacknowledged,connectionless datagram delivery service. Like IPv4, the IPv6 protocol is well known and described in the literature. Here, I will concentrate on its importance within a LLN environment. If any disadvantages are retained from IPv4, it would be that IPv6 is also a large and demanding protocol.

The advantages to the use of IPv6 in an LLN can be listed as follows:

- It has been developed based on IPv4.

The versatility, reliability, scalability and independence of upper and lower stack layers have been fully retained and even extended.Scalability for example has been greatly extended by the larger address range of IPv6.

- The range of possible addresses is now 2^{128}.

Compared to version 4 which had an address range of 2^{32}, the extended range is truly enormous. “This can be compared to providing multiple IP addresses to every grain of sand on the planet” (Hagen, 2006 p.36). For LLN devices which are expected to grow into many millions and by far surpass the number of PCs on the present internet, this is an essential feature.It also means that there will no longer be a need for Network Address Translation (NAT) and will restore the original end-to-end modelwhich is important e.g. for implementation of IPSec and to simplify routing.

- Stateless auto-configuration.

Centralised and managed host configuration of IP addresses by DHCP is no longer necessary. Host nodes can configure themselves with one or more valid IP addresses based on their MAC address (established as a unique number during manufacturing of the device) or as a random number. This Host ID will be combined with a prefix advertised by a router. If so desired, a form of DHCP can still be used to provide network options such as DNS server addresses.

- Simpler header format.

Instead of the variable length IPv4 header format (20-60 bytes), IPv6 has a fixed length header of 40 bytes. Options are taken care of by extension headers that can follow the IP header if and when required.

- Broadcasts are no longer used.

Instead of broadcasts, multicasts are now extensively used. This avoids man problems with routing of messages (routers cannot forward broadcast messages).

3.5 CONTIKI OS FIRMWARE

Contiki OS is a pioneer open source operating system for sensor networks, also known as Internet of Things OS. It has several features such as a small memory footprint, IP networking, protothreads etc. A popular open-source software package for smart nodes is Contiki. This package has been designed and implemented on various platforms by the Swedish Institute for Computer Science (SICS). It is constantly maintained and extended and now forms a very complete solution. The software is written in ‘C’ and comprises a small but capable operating system, an IPv6 communication stack, routing software, implementing the IETF ROLL WG recommendations, named RPL (IETF ROLL, 2011) as well as a physical and data link layer implementation of the IEEE 802.15.4 specification. Fig 3.5 gives architecture of ContikiOS .Furthermore, a 6LoWPAN adaptation layer is implemented (IETF 6LoWPAN, 2005).

3.5.1 CONTIKI OPERATING SYSTEM

The Contiki operating system is a lightweight system designed for use with resource-constrained smart nodes. It connects tiny low-cost,low-power microcontrollers to the internet. Its kernel is event-driven rather than multithreaded in order to limit memory usage (multi-threading needs a stack maintained in memory for each thread). However, a multi-threaded programming abstraction called ‘protothreads’ is provided to make it possible to code in a multi-threading style

while only using an overhead of two bytesper protothread (ACM Sensys 2006).A file system named ‘Coffee’ is part of the OS. It allows accessing the node’s flash memory in a style similar to the normal ‘C’-language file access constructs.

3.5.2 DATA LINK LAYER

The OSI data link layer (sometimes labeled ‘Layer 2’) enables medium access control functionality based on Carrier Sense Multiple Access (CSMA). This means that packets will be retransmitted for a limited number of times when sensing packet collisions or in the presence of radio interference (Shelby & Bormann 2010 p.129). The RIME library of communication functions provides low-level communication functionality that can be used for fast and efficient data collection from the nodes. The RIME functions are also used by the higher level IPv6 stack. Although this technique breaks the isolation between layers, it does make for an efficient structure.

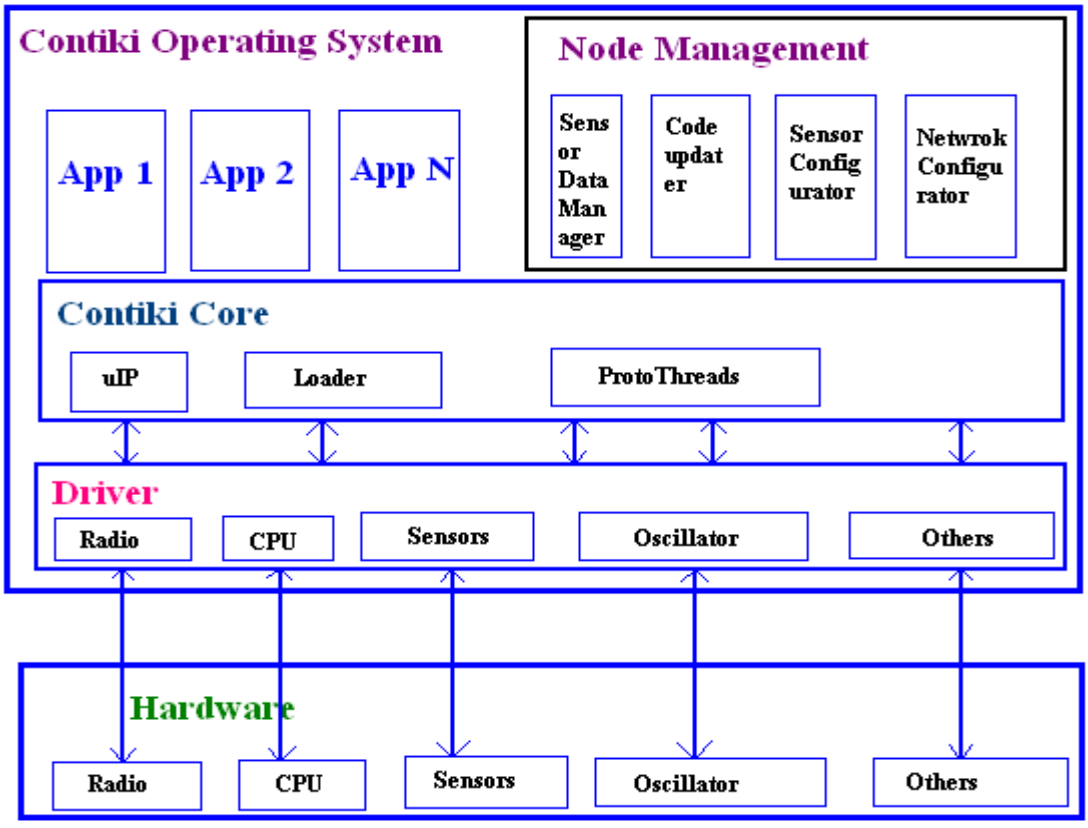

Figure 3.5 ContikiOS Architecture

The data link layer must conform to the IEEE 802.15.4 specifications. The link messages are by default secured by a built-in 128-bit AES encryption feature. As this would be a very processor-intensive operation, encryption takes place in the hardware of the CC2420 transceiver.

3.5.36LoWPAN ADAPTATION LAYER

The IEEE 802.15.4 frame Maximum Transmission Unit (MTU) is 127 bytes minus the MAC frame overhead of 25 bytes and a security header of up to 21 bytes. This leaves only 81 bytes for the IPv6 payload. After removal of the IPv6 header, which has a fixed size of 40 bytes, and the upper layer protocol header, 8 bytes for UDP and 20 bytes for TCP, very little available space is left for the final application payload. Furthermore, IPv6 specifies a minimum MTU size of 1280 bytes which cannot be fulfilled by the IEEE 802.15.4 frames.

The 6LoWPAN adaptation layer can be seen as an additional layer between the IPv6 stack and the data link layer. It provides functionality to enable relatively large IPv6 packets to be adapted to the constraints of the IEEE 802.15.4 interface.

To bring the incompatible message sizes to match, the 6LoWPAN adaptation layer provides for packet fragmentation and reassembly as well as compression of the IPv6 main and extension headers.

3.5.4 RPL

RPL is an open routing protocol for IPv6 smart node networks. It is subject to an ongoing specification process by an IETF working group: Routing Over Low-power and Lossy networks (ROLL). The RPL routing protocol functions at the IP layer and communicates with the network protocol using ICMPv6 (IETF ROLL, 2011). It will be discussed in more detail in a later section.

3.5.5 POWERTRACE

For instrumentation purposes, a built-in application for measurement of energy usage of the different hardware and transceiver components has been added named PowerTrace.

3.6 COOJA SIMULATOR

Cooja is the Contiki network simulator. Cooja allows large and small networks of Contiki motes to be simulated. Motes can be emulated at the hardware level, which is slower but allows

precise inspection of the system behavior, or at a less detailed level, which is faster and allows simulation of larger networks.The carried out work is executed in Cooja simulator, which is a cross-level network simulator available in Contiki OS. The Cooja Simulation for volcanic activity monitoring is shown in Fig 3.6

Cooja is a flexible Java-based simulator designed for simulating networks of sensors running the Contiki operating system.It simulates networks of sensor nodes where each node can be of a different type; differing not only in on-board software, but also in the simulated hardware. It is flexible in that many parts of the simulator can be easily replaced or extended with additional functionality. Example parts that can be extended include the simulated radio medium, simulated node hardware, and plug-ins for simulated input/output.

A simulated node in COOJA has three basic properties: its data memory, the node type, and its hardware peripherals. The node type may be shared between several nodes and determines properties common to all these nodes. For example, nodes of the same type run the same program code on the same simulated hardware peripherals. And nodes of the same type are initialized with the same data memory. During execution, however, nodes' data memories will eventually differ due to e.g. different external inputs. COOJA currently is able to execute Contiki programs in two different ways. Either by running the program code as compiled native code directly on the host CPU, or by running compiled program code in an instruction-level TI MSP430 emulator. COOJA is also able to simulate non-Contiki nodes, such as nodes implemented in Java or even nodes running another operating system. All different approaches have advantages as well as disadvantages. Javabased nodes enable much faster simulations but do not run deployable code.

Hence, they are useful for the development of e.g. distributed algorithms. Emulating nodes provides more fine-grained execution details compared to Javabased nodes or nodes running native code. Finally, native code simulations are more efficient than node emulations and still simulate deployable code. Since the need of abstraction in a heterogeneous simulated network may differ between the different simulated nodes, there are advantages in combining several different abstraction level in one simulation..

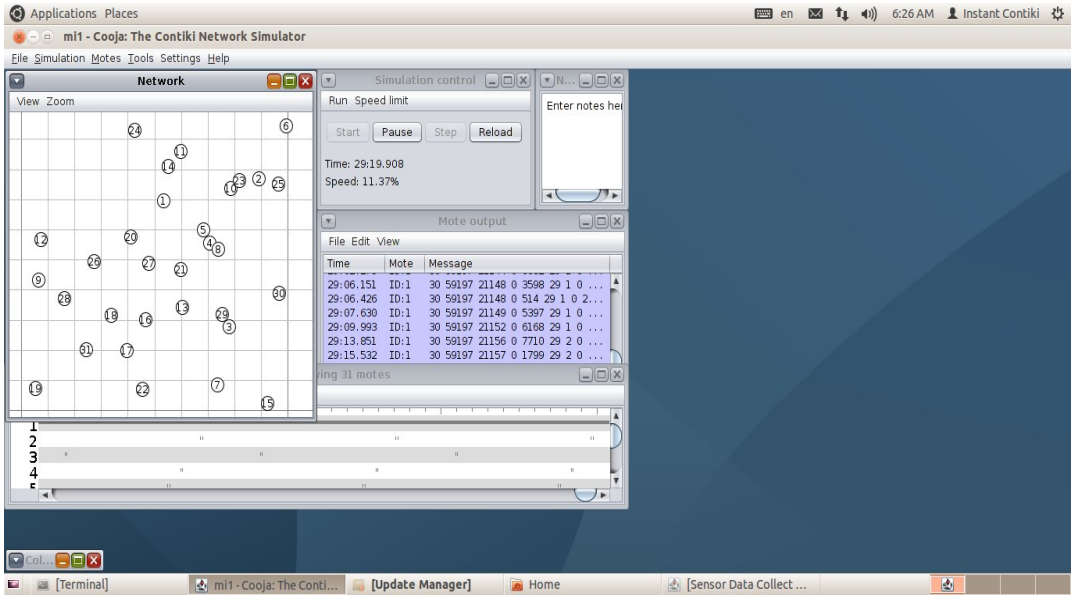

Figure 3.6Cooja Simulator

3.6.1 NETWORKING LEVEL

During design and implementation offor example routing protocols, the specific hardware is oftennot as important as the networking itself. The most importantfactors may instead concern the radio medium, radiodevices and perhaps sleep duty cycles of the sensor nodes.When performing such a design and implementation taskit may be possible, but not necessary, to use a fine-grainedsimulation environment such as an instruction-level simulator.Cooja supports code development by enabling the userto easily exchange certain simulator modules such as devicedrivers or radio medium modules. A simulation can be saved and reloaded using other more or less detailed modules, still with the other simulation parameters unaltered. Fig 3.7 shows Cooja simulation at different levels (Networking, Operating system and Machine code instructon set level). Furthermore, new radio mediums and interfaces such as radiodevices can easily be developed in Java and be added to the COOJA simulation environment.

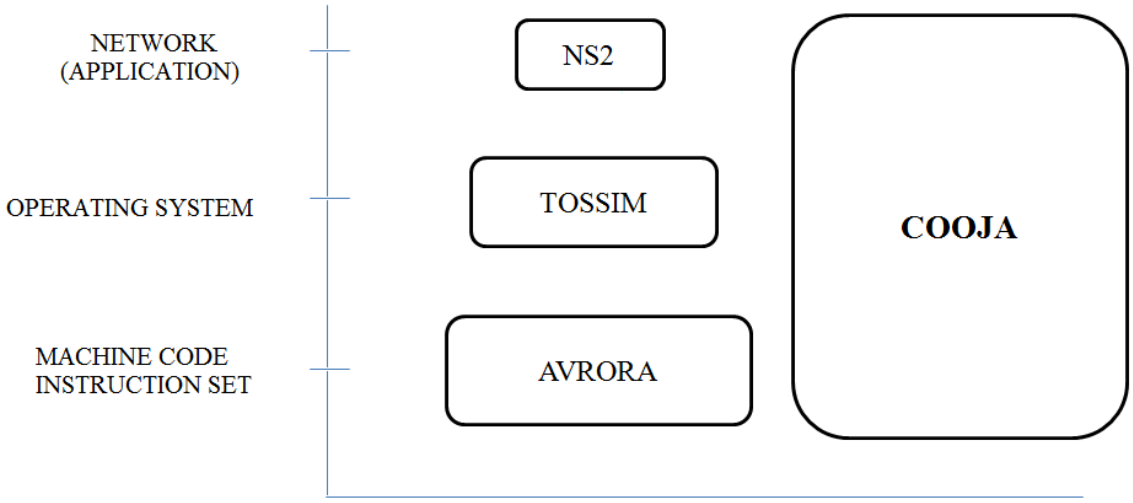

Figure 3.7 Coojasimulation at several levels

To further simplify and speed up development in such scenarios, COOJA also supports adding pure Java code nodes. Without any connection to Contiki, these can be useful when developing high-level algorithms which when tested and evaluated will be ported to deployable sensor node code. Pure Java nodes can also be used in heterogeneous networks where the user only needs to focus on asubset of all the simulated nodes. Since such Java nodes require less memory and processing power, larger heterogeneous network can be simulated more efficiently. For example, using Java nodes, users may rapidly implement the functionality of several different nodes together forming a network. And then later users can, node by node, port the Java code to deployable Contiki node code, while still maintaining full functionality of the network.

3.6.2 OPERATING SYSTEM LEVEL

COOJA simulates the operating system by executing native operating system code as described in the previous section. As the entire Contiki OS, including any user processes, is executed it is also possible to alter Contiki core functionality. This is useful for example to test and evaluate changes in included Contiki libraries.

3.6.3 MACHINE CODE INSTRUCTION SET LEVEL

Using COOJA, it is possible to create new nodes with a very different underlying structure than the typical nodes. We have evaluated this statement by adding nodes connected to a Java-based microcontroller emulator instead of a compiled Contiki system.

The emulator represents an ESB node], emulating at the bit level. The emulated nodes are controlled in a similar way as the native code nodes. Each simulated node is allowed to run for maximum a fixed period of time or long enough to handle one event. Events are then, by using the current node memory, transferred via the hardware interfaces to and from the simulator.

3.7 TUNSLIP

An IP traffic bridges between a host and other network element is obtained by a tool called tunslip, typically a border router, over a serial line. It creates a virtual network interface (tun) on the host side and uses *SLIP* (serial line internet protocol) to encapsulate and pass IP traffic to and from the other side of the serial line.Fig 3.8 shows cooja simulation before tunslip.

Now, it is need to create a bridge between the RPL network simulated on Cooja and the local machine.This can be done by right clicking on the mote which is programmed as the border router. Select 'More tools for border router' and then select 'Serial Socket (SERVER)'. It shows the following message on the successful completion of this step.Note that the message says 'Listening on port 60001'.

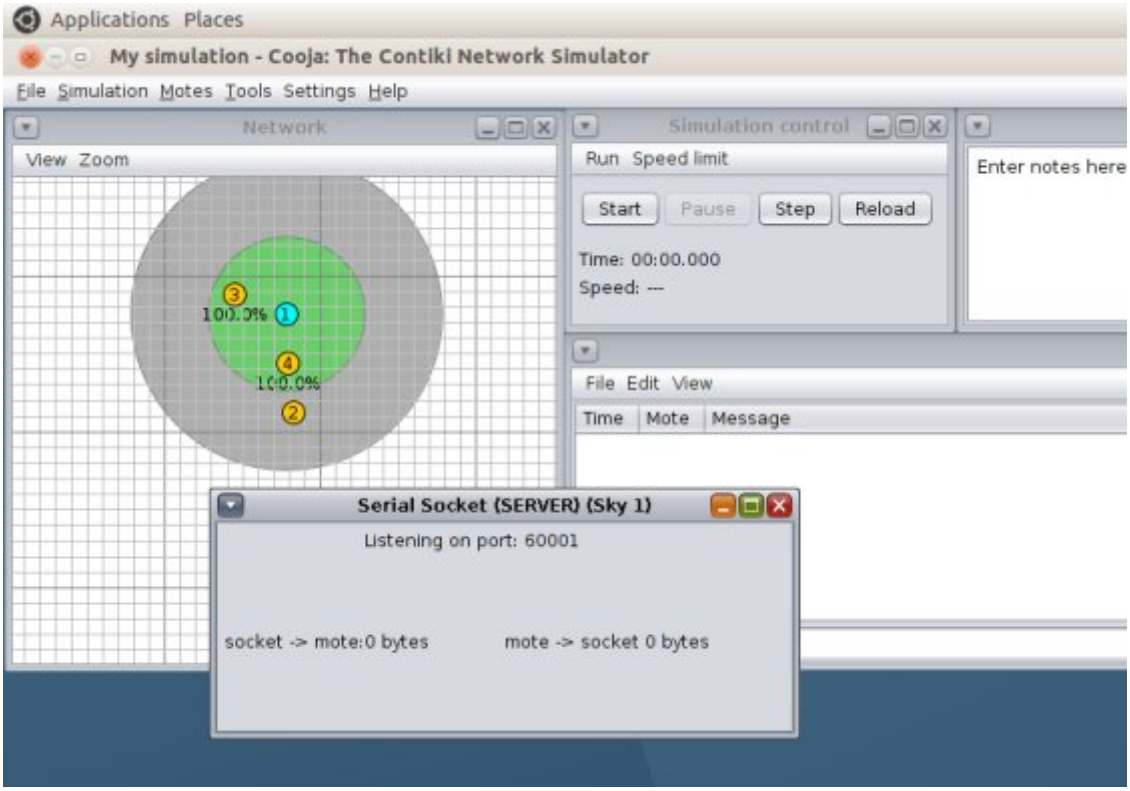

Figure 3.8 Screenshot before Tunslip

Command :Sudo ./tunslip6 –a 192.168.150.131 aaaa::1/64

Using above command, a bridge between PC host and border router is obtained as shown in Fig 3.11. At the end of this command we are providing a prefix of aaaa::1/64 which enables the device to pick its own IPv6 address using stateless address autoconfiguration. Because at initial device picks link-local address (fe80::212:7401:1:101) which is not routable, therefore a router never forwards these addresses outside the link as shown in Fig 3.9. The link-local address is restricted to the segment. It is used for auto-address configurationand neighbor discovery. For global access, it is necessary to change from link-local to global unicast address. It is obtained by tunslip by changing the first prefix 16 bits of fe80 to aaaa. Thus the border router is accessible globally. Fig 3.10 illustrates global unicast IPv6 address.

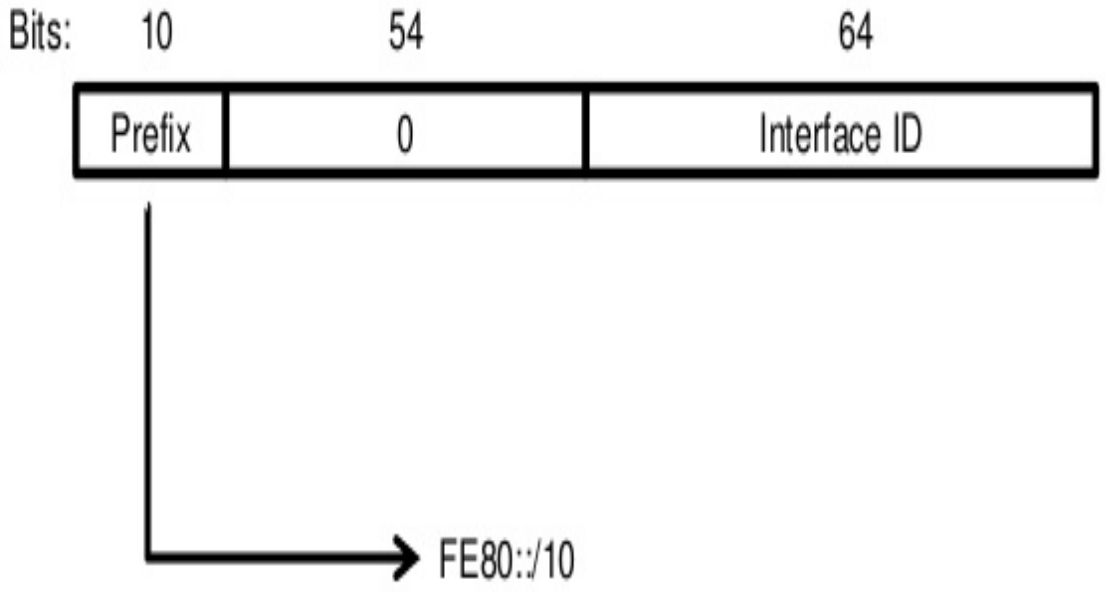

Figure 3.9 Link-local unicast IPv6 address

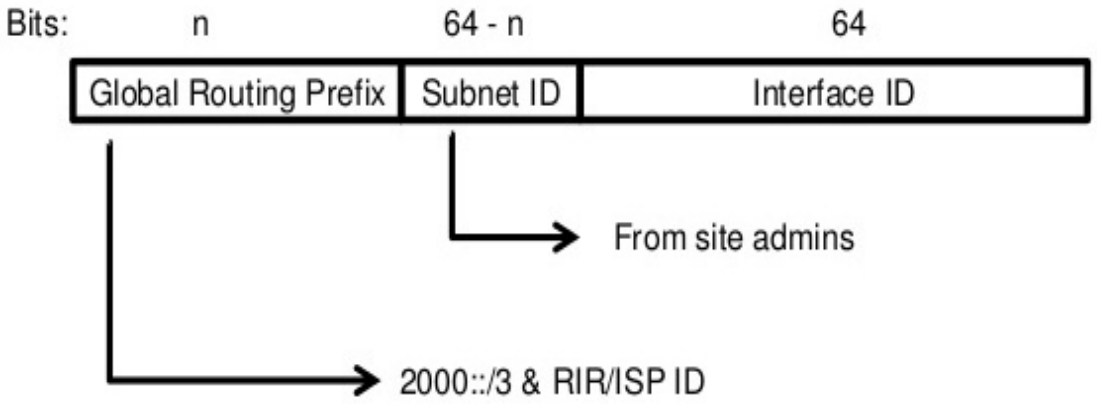

Figure 3.10 Global unicast IPv6 address

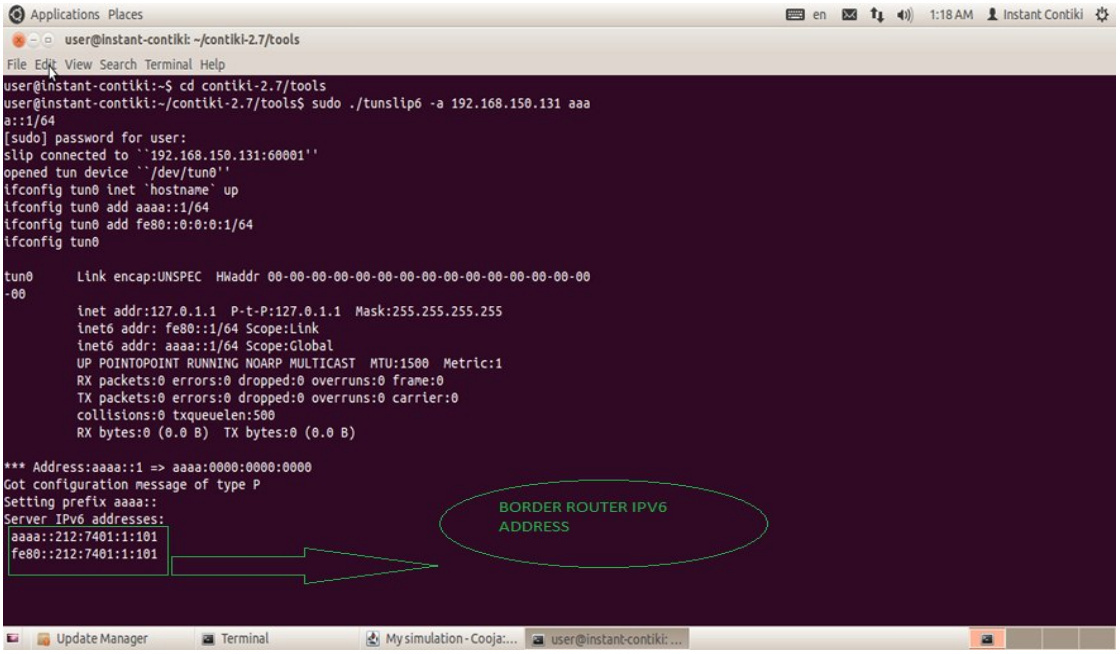

Figure 3.11Screenshot of terminal command shows bridge between border router and local host machine

Cooja simulator GUI and look at the dialogue box. The message has now changed to 'Client connected: /127.0.0.1’

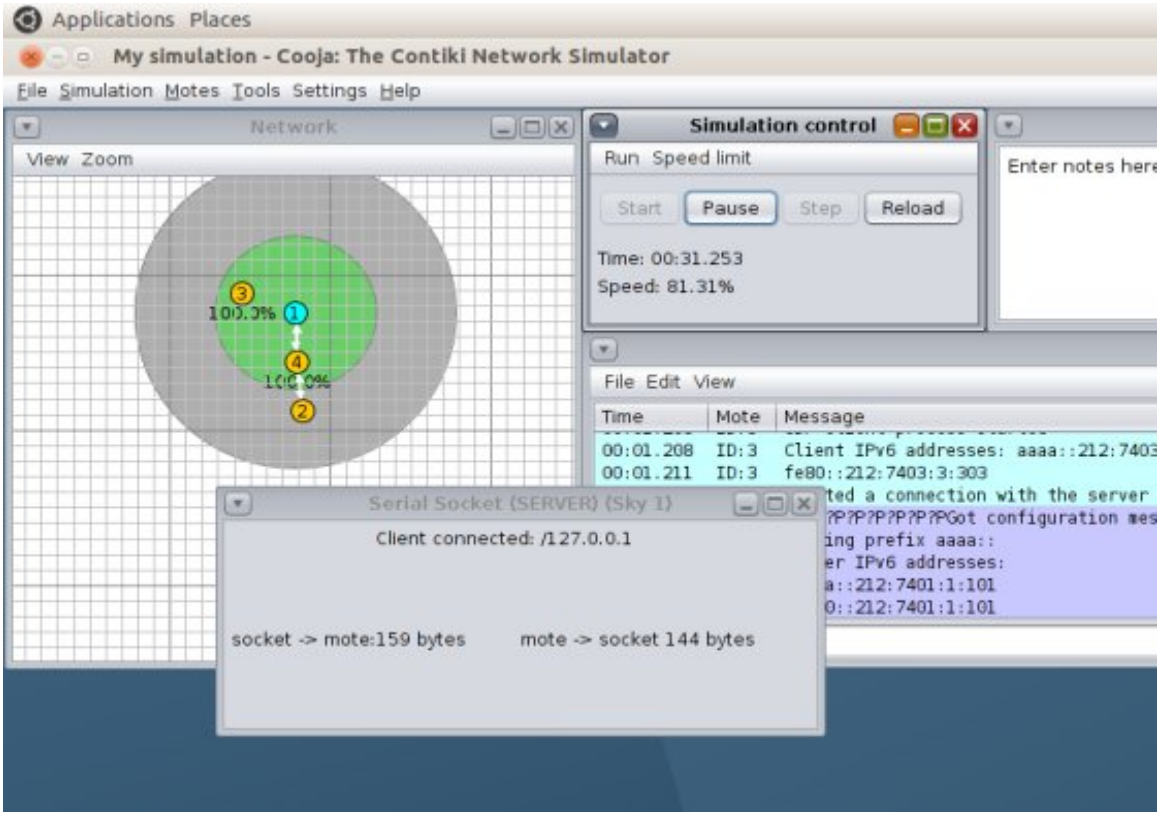

Figure 3.12 Screenshot after TunSlip

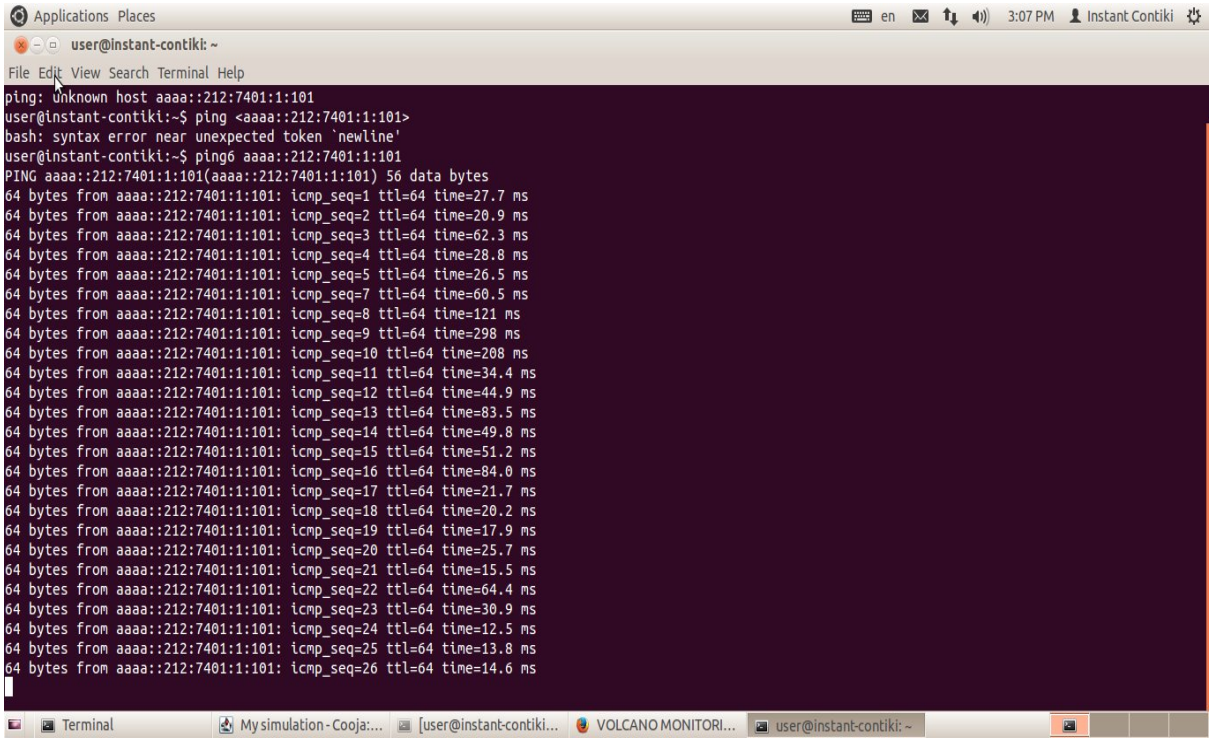

Figure 3.13 Screenshot of ping command

Thus the address of the border router has been verified by the ping command as shown in Fig 3.13. The command is **ping6 –aaaa::212:7401:1:101.**

In order to view the sensor values on the internet by entering the address of the border router in any suitable browser. The border router hosts a page which contains the continuous detected values of Tmote sky sensor values(Temperature and light sensor) will be displayed on the browser as shown in Fig 3.14.

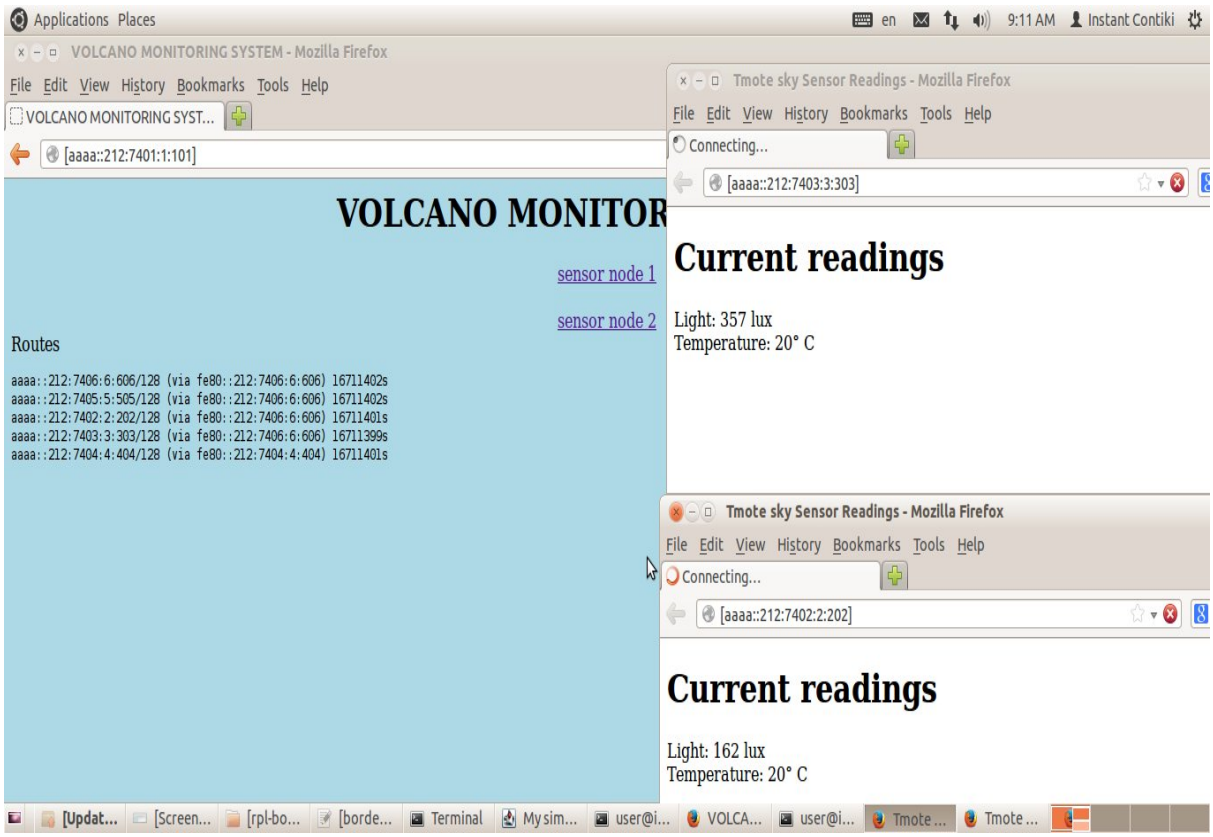

Figure 3.14Sensor values are displayed in the web browser (Mozilla Firefox)

CHAPTER 4

POWER REDUCTION

4.1 CONTIKIMAC RADIO DUTY CYCLE

Tmote sky sensor consists of a processing unit, communication unit (transceiver), power unit and sensing unit. A transceiver consumes 90% of power than other units. To achieve a long network lifetime, power consumption plays a vital role for sensor nodes. Tmote sky sensor node consists of the CC2420 radio transceiver, which draws 60mW of power, when it is listening to the radio traffic and consumes higher power when radio data is transmitting.The radio transceiver should turn off as much as possible to achieve a long period time.

Contiki OS consists of three duty cycling MAC protocols such as ContikiMAC, Contiki X-MAC and Contiki LPP.ContikiMAC is a protocol based on the principles behind low-power listening but with better power efficiency. Contiki's X-MAC is based on the original X-MAC protocol, but has been enhanced to reduce power consumption and maintain good network conditions. Contiki's LPP is based on the Low-Power Probing (LPP) protocol but with enhancements that improve power consumption as well as provide mechanisms for sending broadcast data**.**

Both the Contiki X-MAC and the Contiki LPP provides a streaming functionality where packets can be tagged with a stream flag. Packets tagged with the stream flag are part of a higher layer stream of packets, and the MAC layer treats them differently than others. For example, in a stream, it is likely that more packets arrive quickly. Thus the MAC protocol can keep its radio on for a while longer, knowing that more packets are about to arrive.

ContikiMAC gives a very better power efficiency than other types. In ContikiMAC, Radio Duty Cycle (RDC) mechanism keeps the radio switch off as much as possible and check the radio medium for radio activity. When it is detected, the radio is on to receive the full packets. Channel check rates are given in Hz such as 2, 4, 8 and 16Hz.

4.1.1 CONTIKIMAC TIMING

ContikiMAC radio duty cycling protocol uses periodically wake-ups to listen for packet transmissions from neighbor node.

To receive a packet, during a wake-up

- The receiver is on to receive the packet.
- When the full packet is received successfully, the receiver sends a link layer acknowledgment.

ContikiMAC has a power-efficient wake-up mechanism that relies on precise timing between transmissions as shown in Fig 4.1. ContikiMAC wake-ups use CCA that uses RSSI of the radio transceiver to give an indication of radio activity on the channel.

- If the RSSI is below a given threshold, the CCA returns positive, indicating that the channel is clear.
- If the RSSI is above the threshold,the CCA returns negative, indicating that the channel is in use.

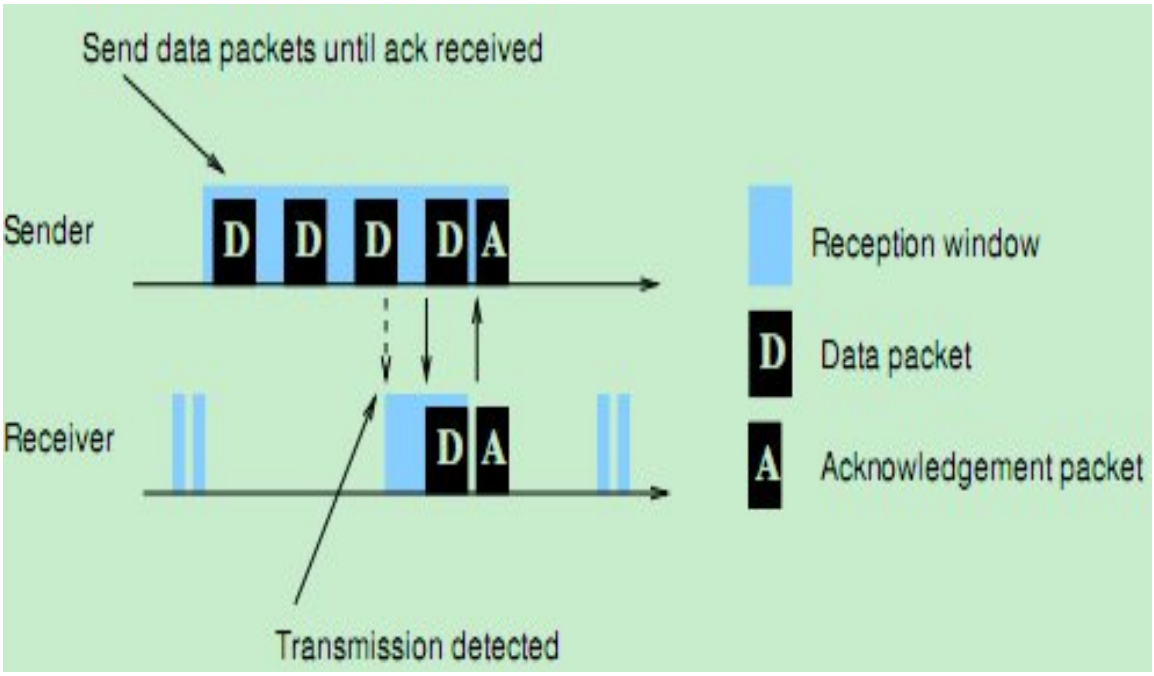

Figure 4.1 Wake-up Mechanism

To transmit a packet, a sender periodically sends its packet until it receives a link layer acknowledgment from the receiver.

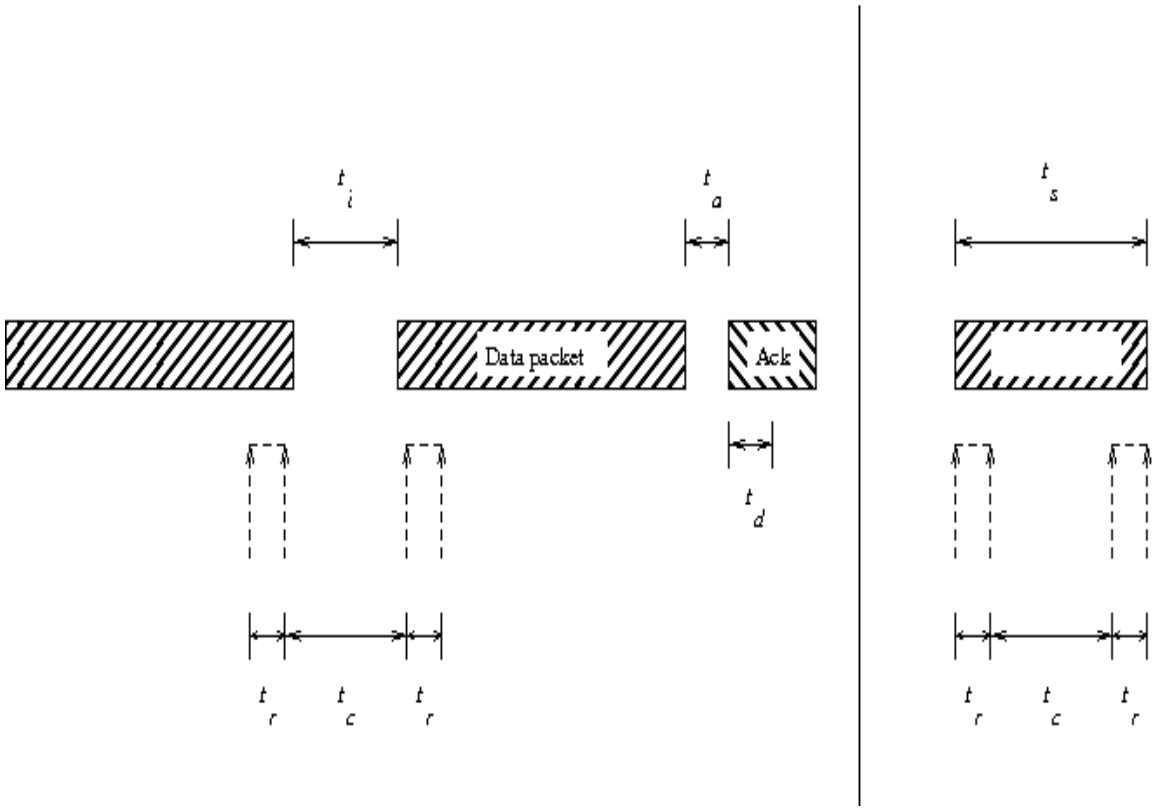

Figure 4.2 ContikiMAC RDC mechanism

The wake-up mechanism depends on timing between transmissions. For ContikiMAC, the timing must satisfy the following constraints.

$$t_a + t_d < t_i < t_c < t_c + 2t_r < t_s$$

Where,

- t_i-the interval between each packet transmission
- t_r-the time required for a stable Received Signal Strength Indication (RSSI)
- t_c-the interval between each channel check
- t_a-the time between receiving a packet and sending the acknowledgment packet
- t_d-the time required for successfully detecting that an acknowledgment is being transmitted

• $\mathbf{t_s}$-the time for transmitting the shortest data packet

The ContikiMAC radio duty cycling mechanism is illustrated in the Fig 4.2. The figure illustrates the timing constraints under which ContikiMAC operates.

For ContikiMAC to work, the timing must satisfy the following constraints. First, t_i, the interval between each packet transmission, must be smaller than t_c, the channel check interval. This is to ensure that either the first or the second channel check is able to see the packet transmission. If t_c would be smaller than t_i, two channel checks would not be able to reliably detect a packet train. Conversely, $t_r+t_c+t_r$ must be smaller than the transmission time of the shortest packet, t_s. Table 4.1 gives the value for timing variables.

Variable	Value
$\mathbf{t_i}$	0.4 ms (Implementation dependant)
$\mathbf{t_r}$	0.192 ms (CC2420 dependant)
$\mathbf{t_c}$	0.5 ms (Implementation dependant)
$\mathbf{t_a}$	0.192 ms
$\mathbf{t_d}$	0.16 ms

t_s	0.884 ms (Implementation dependant)

Table 4.1 Timing Variables

When a packet transmission has been detected, ContikiMAC keeps the radio on to be able to receive the full packet. When a full packet has been received, a link-layer acknowledgment is transmitted. ContikiMAC requires that the underlying link layer is able to transmit acknowledgment packets in response to received packets. If the link layer does not provide this by itself, it is possible to implement this as a layer between the link layer and ContikiMAC.

The time it takes for an acknowledgment packet to be transmitted, t_a, and the time it takes for an acknowledgment packet to be detected, t_d, establishes the lower bound for the check interval t_c.

Channel Check Rates	**CPU Power (mW)**	**LPM Power (mW)**	**Listen Power (mW)**	**Transmit Power (mW)**	**Total (mW)**
RDC at 4Hz	0.205	0.256	0.478	0.123	1.062
RDC at 8Hz	0.195	0.258	0.658	0.066	1.177
RDC at 16Hz	0.195	0.258	0.989	0.041	1.483

Table 4.2Average power values at different RDC frequencies

Table 4.2 shows the comparison of average power values at different RDC frequencies. The result shows that 4Hz is lowest RDC frequency and most suitable for battery-powered nodes, but it also results in lower network responses which gives high network latency. 16Hz RDC frequency consumes higher power than other. So 8Hz RDC frequency is better than other two types and suitable for volcano monitoring system to produce low power.

Fig. 4.3, Fig. 4.4 and Fig.4.5show the simulation result of volcano monitoring system at 16Hz, 8Hz and 4Hz respectively. An average power usage is shown for 40 nodes. The first simulation has setup a smart object data collection network with a sink and 40 sender nodes from the network with 16 Hz RDC frequency. The remaining test is followed by second and third with an RDC frequency of 8 Hz and 4 Hz.The result of the three different RDC frequencies gives power below 2mW.

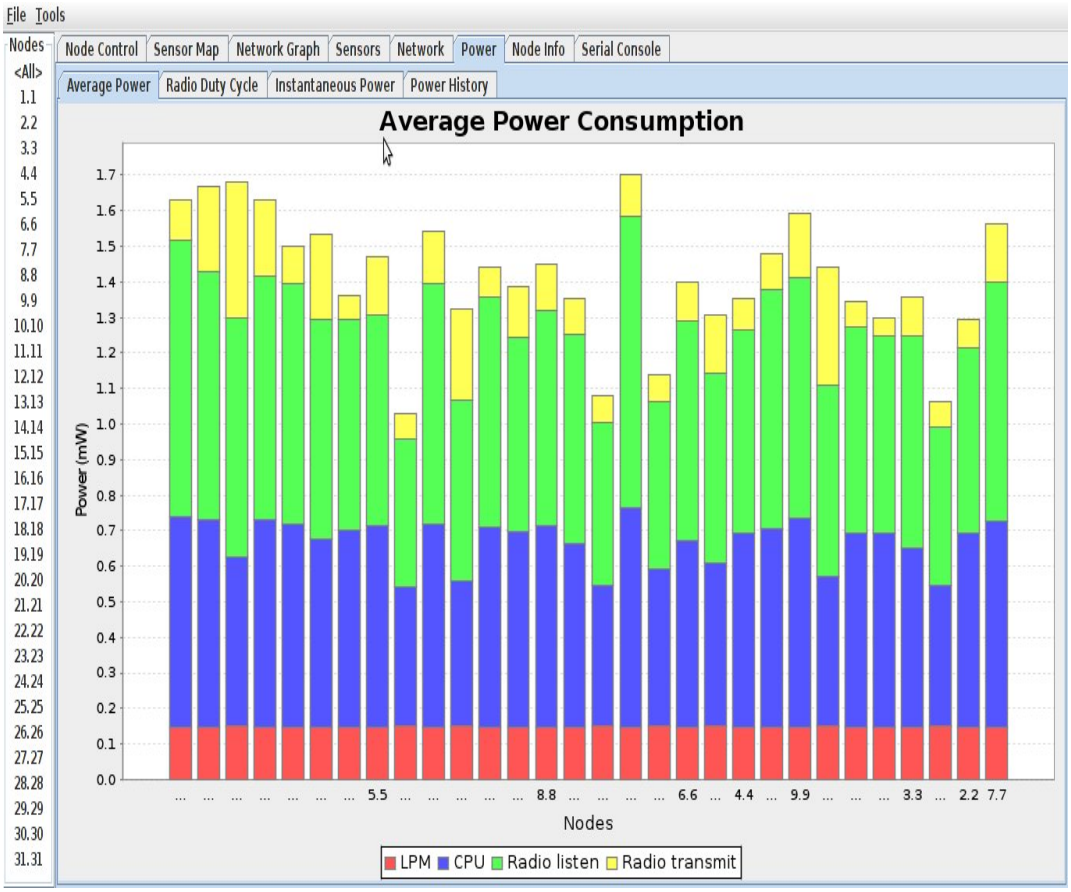

Figure 4.3. Average Power Usage at an RDC frequency of 16 Hz

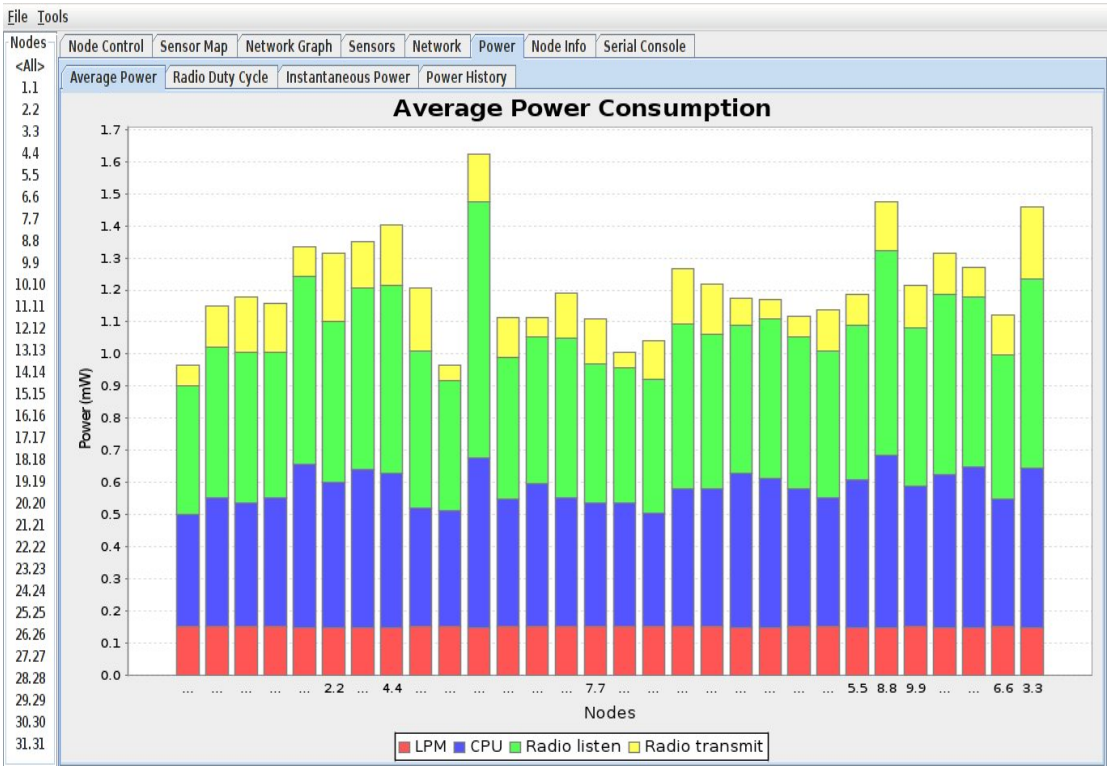

Figure 4.4. Average Power Usage at an RDC frequency of 8 Hz

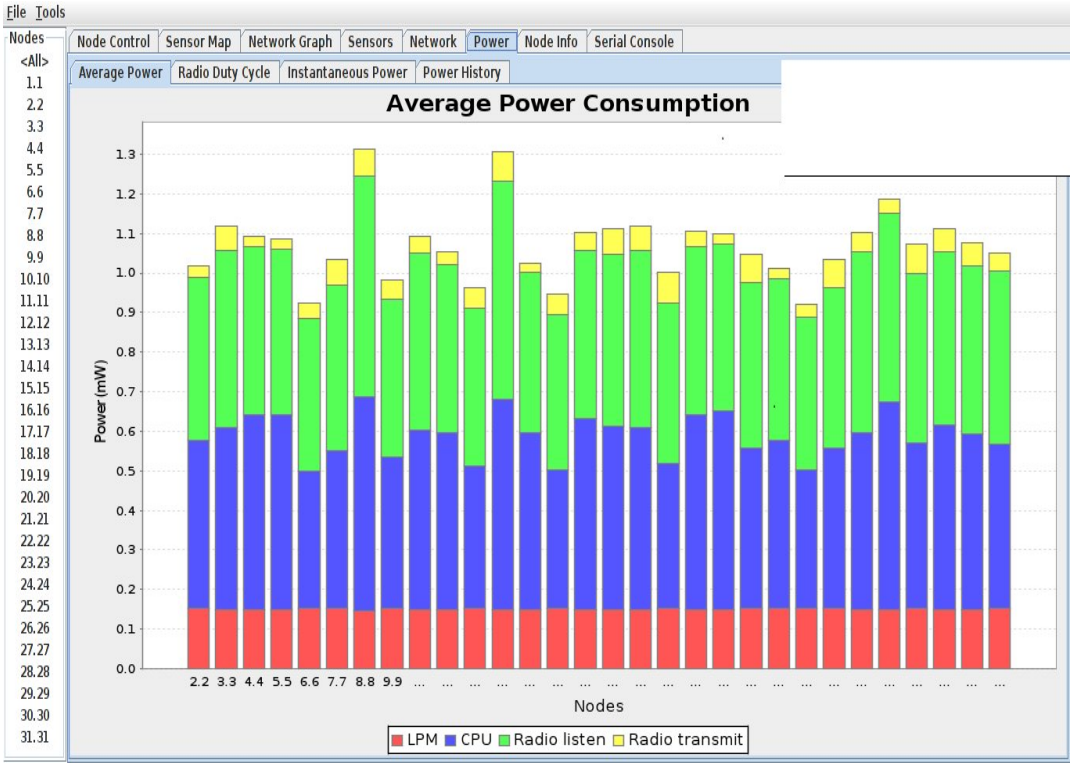

Figure 4.5. Average Power Usage at an RDC frequency of 4 Hz

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

In this project, the flexible volcano monitoring system was implemented using Internet of Things. By Contiki OS, a new mechanism based on a ContikiMAC radio duty cycle is used to reduce the power consumption of Tmote sky sensor node by a simple timing scheme to allow its wake-up mechanism. The radio listening and transmitting power of Tmote sky sensor node's transceiver (CC2420) is reduced from 60mW to less than 2.5mW. Thus collected sensor data from wireless sensor nodes are forwarded to the border router and it is viewed in any suitable web browser by the client.

5.2 FUTURE WORK

This project can be seen as an important step towards the design of Volcano monitoring system. However, it is necessary to build a secure website in future.

REFERENCES

[1] Roman Lara Cueva, etal.'Towards a New Volcano Monitoring System Using Wireless Sensor Networks'ESPE, Quito, 171-5-231B,2006.

[2]Dunkels,A,'TheContikiMAC Radio Duty Cycling Protocol,'Technical Report T2011:13 Swedish Institute of Computer Science.(2011).

[3]Sandra Sendra, et al.'Power saving and energy optimization techniques for Wireless Sensor Networks' Journal of Communications,Vol. 6, No.6, Sep-2011.

[4]Geoffrey Werner-Allen, et al.'Monitoring Volcanic Eruptions with a Wireless Sensor Network',IEEE-2005, page 108-120.

[5]Leila Ben Saad, et al.' Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: two cases studies',International Conference on Sensor Technologies and Applications,IARIA, 2011.

[6] Geoff Werner-Allen, et al.' Fidelity and yield in a volcano monitoring sensor network', Proceedings of the 7th symposium on Operating systems design and implementation ,2006,Pages 381-396.

[7] Fredrik O¨ sterlind, et al.' Cross-Level Sensor Network Simulation with COOJA',IEEE-2006, page 641-648.

[8]Chang, Kai-Di. 'A Survey Of Trust Management In Wsns, Internet Of Things And Future Internet'. KSII Transactions on Internet and Information Systems ,2012.

[9] Miao Wu. etal.'Research on the Architecture of Internet of Things'.3rd International Conference on Advanced Computer Theory and Engineering,2010,page 484-487.

[10] Mikhail, M., Maria D.’Emerging of new service-oriented approach based on the Internet of Services and Internet of Things’.IEEE 10th International Conference on e-Business Engineering,2013, page 429-434.

[11]Varaprasad, G. 'Wireless Sensor Network For Volcano Environments'. ACM SIGBED Review 6.3,2009.

[12]Werner-Allen, G. et al. 'Deploying A Wireless Sensor Network On An Active Volcano'.IEEE Internet Comput. 10.2,2006,page 18-25.

[13] Xu, Mingsen et al. 'Design Of Smart Sensing Components For Volcano Monitoring'. Pervasive and Mobile Computing 5.5, 2009, page 639-653.

[14]GitHub,.'Contiki-Os/Contiki'.N.p., 2014. 20 Mar. 2015.

[15] T. Torfs et al., “Low power wireless sensor network for building monitoring,” IEEE Sensor Journal, vol. 13, no. 3, March 2013, page 909-915.

[16] J.-H. Chang and L. Tassiulas, “Maximum lifetime routing in wireless sensor networks,” IEEE/ACM Transactions on Networking, vol. 12, no. 4, August 2004, page 609–619.

[17]V. Gallart, S. Felici-Castell, M. Delamo, A. Foster,J.J. Perez, “Evaluation of a real, low cost, urban WSN deployment for accurate environmental monitoring,” in IEEE MASS, Wuhan, China, 2011, pp. 634-639.

LIST OF PUBLICATIONS

- paper accepted on **using Internet of ThingsTo Monitor Volcanic Activity**in the international conference on **ICTTM '15**at **IndianInstitute of Technology-New Delhi.**
- Presented a paper on **Design of a volcano monitoring system using Internet of Things** in the international conference on Knowledge collaboration in Engineering ICKCE'15 at **Kathir College of Engineering.**