



# Energy Oriented Routing in Wireless Sensor Networks: Hardware Implementation

Dipanshu<sup>(✉)</sup> and A. Nagaraju<sup>(✉)</sup>

Department of Computer Science, Central University of Rajasthan, Ajmer 305801,  
Rajasthan, India

dipanshu.acad@gmail.com, nagaraju@curaj.ac.in

**Abstract.** In the recent Internet Technology world, sensor-oriented network domain has significant importance for real-world application to perform user's day to day activities. Energy efficiency in a sensor field is an extremely important characteristic for various on demand needs. The related literature on energy efficient path selection and message scheduling in a sensor-enabled wireless network is a predominant factor in the Internet of Things domain. The proposed work implements an energy-efficient routing scheme and provides design details for sensor nodes. The sensor nodes are divided into three types of nodes – the base station node, the cluster head node, and the normal sensor node. The base station node is responsible for setting up the network and the normal sensor node performs the various operations designed for the network while the cluster head node acts as a cluster head. A new simple application layer protocol is also presented. This application layer protocol provides a simple and powerful enough interface to the user. This scheme and architecture when employed produces a reliable and efficient network. The reliability and efficiency are measured and illustrated using real-time implementation. The real-time implementation is performed on custom-designed sensor nodes based on AVR Atmega8a and nRF24L01 transceiver chip. The Sensor Nodes run on a custom-designed firmware which provides basic and enough security. The custom-designed sensor nodes provide a reliable Wireless Sensor Network and can be deployed in real-world applications.

## 1 Introduction

The ubiquitous presence of Wireless Sensor Networks (WSN) demand new techniques for routing data reliably and efficiently, it is highly required for the ongoing demand of IoT applications. IoT and Wireless Sensor Networks (WSN) are among the most useful and essential technology in the evolution of smart city development. The implementation of these technologies is a very challenging task because of the discrepancies between real life hardware and implementation details on paper. The IoT infrastructure and WSNs are typically built on

numerous homogeneous small form factor computers. In case of IoT infrastructure, the individual computing devices are generally fully fledged computers running a typical operating system while in WSNs the individual computing unit is relatively a low power computationally limited device called a Wireless Sensor Node (SN). A Wireless Sensor Node (SN) is usually built using an 8-bit or 32-bit micro-controller, communications chip, power source, some sensors, and some external circuitry to provide extra features such as switching and actuation. Recent trends in miniaturization of chips and development and a healthy market of SoCs (System on Chips) will eventually lead to more sophisticated SNs running more sophisticated operating systems such as Linux.

The problem of providing power to SNs is one of the most researched topics in the research area of WSNs. The most straight forward way would be to provide power through wires but that would defeat the very purpose of WSNs. So the other way to provide power would be through batteries and this is what is usually done in the current trend. So the methods to provide power to SNs constitutes a very import design constraint. Instead of focusing on power sources it is relatively easy to focus on and improve and reduce power usage. An SN should usually consume around less than tens of mA of current.

The best way through which we can minimize the consumption of power is through improving the software and specifically improving the routing support. Good routing support in a particular deployment area would result in a better lifetime of the network. A routing protocol generally and in almost all cases uses the shortest path available to transmit the packets and doesn't care about the position, packet size and scheduling all in general. A routing algorithm proposed by Latif et al. [6] gives a good scenario and a routing protocol that takes into account the size of packets, position of the nodes and use a scheduling approach. The protocol consists of two algorithms. The first algorithm is named Shortest Path and Less number of Links (SPLL) and the second one is named Longer Hop (LH).

The first algorithm uses the criteria of the shortest distance in a hop and uses it to forward or route the packets in a network. The second algorithm uses the position of the nodes and schedules or sorts the packets according to their priority which in turn is based on the energy loss equation. The sensor nodes are generally divided into clusters using some specific techniques, like using the k-means algorithm on the location information of the sensor nodes. A cluster head (CH) is appointed in each cluster to collect the data from non-CH nodes. This ensures that the path on which the data travels is short as possible since only CH can communicate with the base station (BS). Both algorithms are combined to create a complete system for routing in wireless sensor networks.

To implement any theoretical work, the designer needs to have a sensor node which can be easily designed and modified according to the needs. Our aim is to ultimately propose a hardware design of sensor node and methods which can be used by researchers to test their ideas and further improve them.

## 1.1 Motivation

The motivation for the proposed work is from smart city development point of view in which the sensors are communicating to control the environment. So an efficient data transmission technique is very essential for better service delivery. The proposed work made an attempt to emulate the SPLL and LH routing mechanism using a custom-designed sensor node powered by AVR ATmega8a Microcontroller and Nordic semiconductor's nRF24L01+ transceiver. The hardware is chosen such that to maintain the system with high reliability, high efficiency, and high battery life which are the most obligatory and indispensable features for any network system.

## 1.2 Contributions of the Proposed Work

The main contributions of the proposed paper as follows.

- The proposed work is a real-time implementation of sensors for use in commercial and critical conditions.
- New open hardware design for wireless sensor nodes.
- Implementation of the routing algorithm in a small form factor sensor devices such as our custom designed sensor node.

Further, in this paper, we will showcase a new simple and straight forward application layer protocol for wireless sensor nodes.

## 2 Literature Survey

A **Wireless Sensor Network** is a network of devices called **sensor nodes** that are connected through some wireless means. The wireless connection is typically accomplished by using radio waves. Various proprietary and open radio communication protocols are present in the market. The networks typically work in the Industrial Scientific and Medical (ISM) bands and various lower frequency bands. Our implementation works in the 2.4–2.5 GHz range. A Sensor Node replies to requests and transmits, from and to a controlling device, called a **Base Station**. A Base Station is responsible for controlling the network, managing the data, management of Sensor Nodes and data requests. Base Station is typically a powerful computing device.

**Routing Protocols** are techniques, methods, algorithms that are employed in a network that guarantees and helps in transferring a data packet to its destination. They have acquired a prime interest in researches related to WSNs. Routing Protocols are generally categorized into two types, Network structure and Protocol Operation based. Network structure based protocols are employed in the lower level of the network while the Operation based protocols are designed for a higher level of the network. Routing protocols is a heavily researched topic but the implementation is never talked about.

**Routing Table** is a data structure that stores information about the address of destination and address of next hop. A routing protocol implementation uses a routing table to figure the next hop if the destination address is given. To populate a routing table is to generate data of addresses according to a certain algorithm. In generating the data is the only step in the process where any routing algorithm will be used. After the generation of address data, the program just needs to query the routing table.

A **Printed Circuit Board** (PCB) is a thin board onto which electrical circuits are put on and various components are soldered on. They are typically manufactured in a sophisticated environment in a computer controlled machinery. The computer reads a PCB describing file called a **Schematic**. The Schematic describes all the features, components position and dimensions, drill locations and various other things about the board. They are typically generated and designed using computer-aided design and drafting (CAD) software.

Farhan et al. [6] has proposed an energy efficient path selection and message scheduling mechanism in wireless sensor networks to improve the energy consumption in transmitting the data packets from source node to destination node. In the proposed work authors has suggested the shortest path with less number of links algorithm. The location of the message sender and the number of utilized processors are decided to improve the energy efficiency in the IoT. A system model is developed by authors who use the M/M/1 queuing analysis to showcase the effective cooperation of Shortest Path and Less Number of Links (SPLL) and LH towards energy efficiency.

Chang et al. [5] has proposed a novel tree-based power saving scheme is proposed to reduce the energy consumption and to extend the lifetime of the network, throughput, and transmission overhead. In the proposed work authors considered a mobile sink and some fixed sensor nodes. In the proposed model nodes are considered as non-CH Nodes, hop nodes, and CH nodes. Based on the location of the sensor nodes and distance between the sensor nodes and residual energy the structure will be defined and proposed scheme define routing structure based on the above information. Accounting for the sensing area, the number of sensor nodes and the movement of the sink node, a tree structure will be defined and multi-hop concept is used to create energy efficient architecture.

Mukherjee et al. [9] has proposed a scheme for integrating MANETs with WSN to improve the connectivity between ongoing number of objects which are going to share the information through the internet to perform various Internet of Things (IoT) applications. The most challenging part in the integration is various kind of devices, platforms and heterogeneous nodes grouped together to share information through wireless devices.

The description of hardware modules is necessary to move forward and discuss the proposed implementation.

A microcontroller is a small computation device that houses RAM, ROM and various communication protocols on the chip. The microcontroller market has various leading manufacturers which are Atmel, Microchip and STMicroelectronics. The choice of a microcontroller is entirely application dependent.

A microcontroller typically has an 8-bit or 32-bit architecture. The 8-bit architecture is well suited for various applications while the 32-bit architecture is suitable for tasks which require high computation power such as cryptography, data storage, data processing, and implementation of high-level network layers.

Atmel developed the AVR ATmega8, a cheap and powerful microcontroller that has been part of various projects throughout its lifetime. It supports free and open source programming tools and compilers. It is a low power device and can work on up to 16 MHz clock speed. The ATmega8 microcontroller is an Advanced RISC Architecture device with instruction executed in a single cycle. It has a decent memory profile. It has 8 KB of In-System Self-programmable Flash Program, 512 Bytes of ROM (EEPROM) and 1 KB internal SRAM. Its data retention is of about 100 years at room temperature. Various communication protocols such as USART, SPI, I2C, Watchdog timer and two 8-bit timers and one 16-bit timer are present. It has 23 programmable I/O pins out of which 8 are 10-bit ADC channels. It operates in the range 2.7–5.5 V and consumes about 3.6 mA and 0.5 A in active mode and power down mode respectively.

The radio transceiver chip, the nRF24L01+ by Nordic Semiconductor is a popular, powerful, easy to use and programmable radio wave communications chip. It is an ultra-low-power chip that works in 2.4 GHz ISM band. The device can be programmed through its SPI interface. The 29 programmable registers control the entire functioning of the chip. It supports automatic packet acknowledgment by using its Enhanced ShockBurst technology. It can receive data on six different address that makes it easy to implement the broadcasting feature of any Wireless Sensor Network scheme since we don't need to change the receiver address every time after and before broadcasting. It supports 250 kbps, 1 Mbps, and 2 Mbps on-air data rates. The operation of the device is typically done at 3.3V.

Lewis [8] and Al-Karaki et al. [3] gave an overview of wireless sensor networks in general and routing in wireless sensor networks. These two sources and the literature detailed here and in reference [1, 2, 7, 10] guided us to design a good entry level implementation.

### 3 Proposed System Model and Architecture

Sensor nodes used for implementation are custom designed, housing Atmel AVR ATmega8a microcontroller [4] as the main processing unit and Nordic Semiconductor's nRF24L01+ [11] as the communication unit. Both devices are powerful enough for the implementation and use very low power and even lower on certain modes. The main firmware running on the microcontroller uses very minimal features and have very straightforward capabilities such as sending, receiving and routing the data, and sensing, actuating and processing is also included.

The sensor node can be run in three modes normal, cluster head and base station. In normal mode the sensor node behaves normally in the sense that it can receive, send, route, sense, actuate and process. In cluster head mode the sensor node behaves normally and also acts as a sensor node where each node

of its cluster sends the data to. In base station mode the sensor node acts as a communication unit for a much powerful system. For example, in base station mode the sensor node together with a general purpose desktop computer can generate routing table and distribute it, ask for data, send commands to cluster heads or directly to other non-cluster head sensor nodes. In base station mode the sensor node requires a more powerful computation device because alone it cannot work.

The transceiver chip used can at maximum transmit at 0 dBm. The range of the device is environment and antenna depended. The custom designed hardware uses the on-chip meander line antenna. External antenna such dipole antenna can also be used. With the on-chip antenna, at 0 dBm, the indoor environment range advertised is around 100 m, the outdoor range is around 250 m. The outdoor range is significantly increased, up to 1 KM, when PA+LNA version (with dipole antenna) is used.

### 3.1 Hardware Specifications

The sensor node is custom designed and contains two main components, Atmel AVR ATmega8a, and Nordic nRF24L01+ RF Transceiver.

The modules are of two types, Type-1, and Type -2, and different on the basis of the nRF24L01+ module type. Type- 1 has SMD (Surface Mount Device) nRF24L01+ module while the Type-2 has generic through hole module. Their schematics are shown in Figs. 1 and 2.

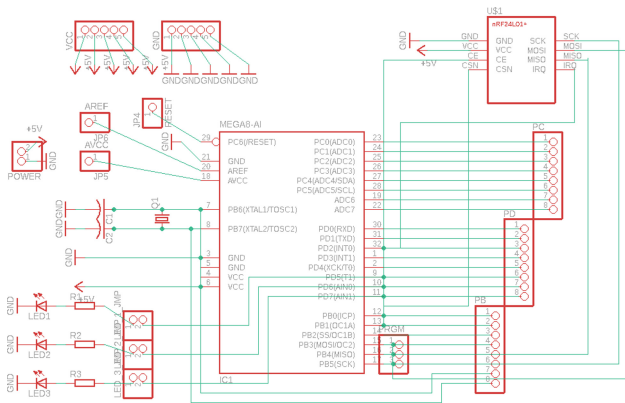


Fig. 1. Type 1 schematic.

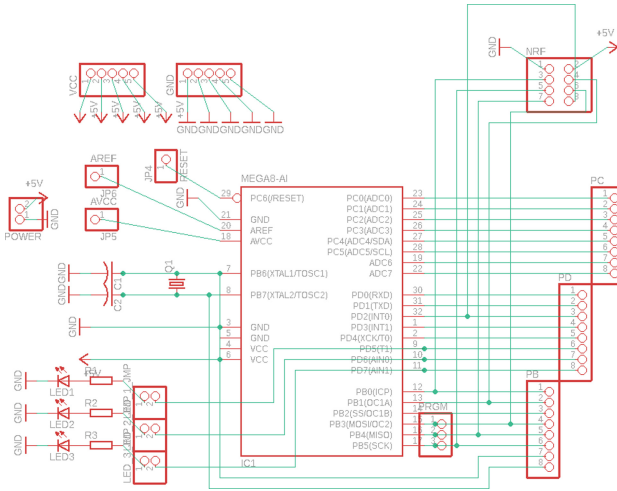


Fig. 2. Type 2 schematic.

### 3.2 Software Modules Developed

The software for the sensor node and base station are different. The sensor node software is written in C and the basic proposed architecture of the proposed model is given in Fig. 3.

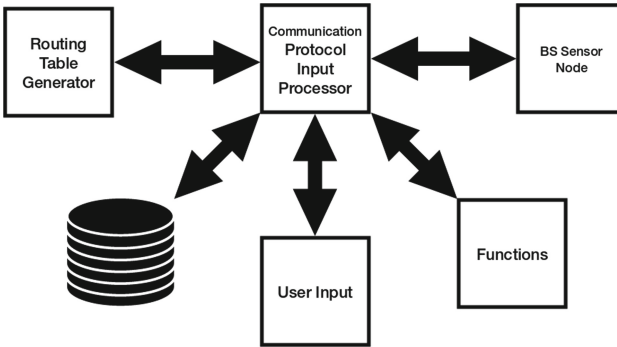


Fig. 3. Software architecture.

### 3.2.1 Routing Table

The routing table stores routing information related only to that particular node. Thus only require three columns and it is visualized in the Routing Table Generator section below.

### 3.2.2 Protocol Processor

The Protocol Processor (PP) queries the Routing Table (RT). It can also update, insert and delete table entries. By default, the RT can accommodate a maximum of 25 entries. The maximum entries size can be changed before compiling the firmware.

The PP is responsible for decoding the data packets and accordingly generating the reply packets. The PP can be modified by adding more packet types i.e. user can add more packet types that the Sensor Node can understand.

### 3.2.3 Communication Processor

The Communication Processor (CP) directly interacts with the communication chip (nRF24L01+). The CP is responsible for sending and receiving data packets. The CP can reject data packets if the checksum does not match. The CP, after successfully receiving data packets, transfers the data packet to the PP.

In sensor node software the incoming data is handled by the communication processor. The communication process further calls the protocol processor. The protocol processor identifies and decodes the packet and accordingly generates a reply packet using the routing table and calls CP with the reply packet as the argument. The communication processor then sends the packet to its destination.

### 3.2.4 Communication/Protocol/Input (CPI) Processor

In base station software either the data can come in through Base Station Sensor Node or User Input and then the Communication/Protocol/Input (CPI) Processor decodes the data and takes action accordingly. It can use Functions for various algorithms, Database for database functionality and Routing Table Generator for routing table generation. Each of these modules replies with appropriate data when they are called. The CPI processor then outputs appropriate information to the user or uses the Base Station Sensor Node to reply accordingly to the node.

### 3.2.5 Routing Table Generator

The Routing Table Generator or RTG is, as of now, capable of generating the routing table using the SPLL and LH routing scheme described in the Literature Survey Section of the paper. The routing table is visualized in the Fig. 4.



source	next_hop	destination
0xA000000006	0xA000000004	0xA000000003
0xA000000006	0xA000000004	0xA000000004
0xA000000007	0xA000000002	0xA000000005
0xA000000007	0xA000000004	0x0000000008

**Fig. 4.** An example of routing table generated by the Routing Table Generator after the application of the routing scheme.

### 3.2.6 Database

The database will store all the data coming from the network. The database of choice is SQLite and the choice is flexible and is user dependent provided that a proper plugin is developed so that the database can work with Base Station Software seamlessly.

### 3.2.7 User Input

This module interacts directly with the user and is responsible for data or command input/output. The way the user interacts with UIO, as of now, is through text commands i.e. it has Command Line Interface (CLI).

### 3.2.8 Functions

This module contains various predefined or user-defined program functions that can be used by the CPI P. The functions such as table traversal, lookup, etc. reside here.

### 3.2.9 Base Station Sensor Node

The Sensor Node connected to the High Computation Device (HCD) acts as a gateway between the sensor network and the HCD. This Sensor Node is not specially designed or has different firmware. In this implementation, it is connected to the HCD through RS-232 Serial Communication.

## 3.3 Application Layer Protocol

The application layer protocol is simple and straight forward. The packet is 32 bytes long. The example in the next paragraph explains the packet given in the Fig. 5.

5 Bytes source	5 Bytes destination	1 Bytes packet type	2 Bytes command
2 Bytes argument	15 Bytes data		2 Bytes checksum

**Fig. 5.** Application layer packet format.

ABCDE VWXYZRSN0100000000000000+0x73A1 ‘ABCDE’ is *source*, ‘VWXYZ’ is *destination*, *R* is packet type, *SN* is command, *01* is argument, *000...000* is data and *0x73A1* is the checksum.

We found the packet size and packet parts to be sufficient for complete data transmission in WSN.

### 3.4 Network

As of now, the network supports only one request at a time in the network. The reason for this is that the transport layer is not implemented in this version of the implementation. Certain TDMA techniques can be used to get data from all the sensor nodes.

#### Phases of Operation

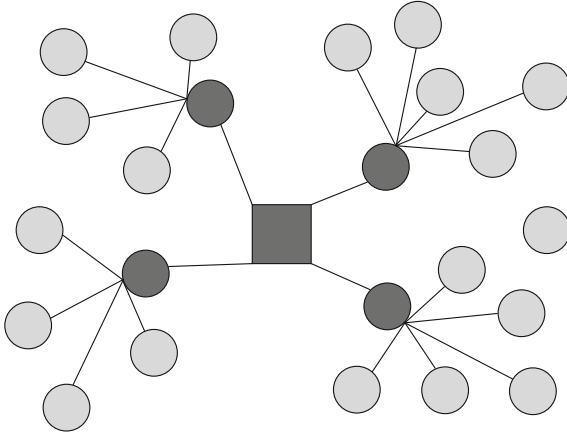
1. Initial Setup

If the position of the nodes is static and known the data can be directly used as an input to the Base Station Program (BSP) and the Routing Table Generator (RTG) will generate the routing table and call the Communication/Protocol/Input Processor (CPI P) to populate the routing table in network.

2. Normal operation

The user can get the data from any position or node by interacting with the User Input/Output (UIO).

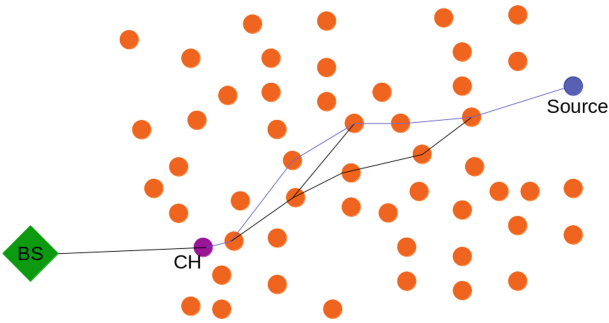
The entire software suite is written in C and Python, and its architecture is given in the Fig. 3. The entire software suite can be found on <https://github.com/c3rl/snfw> (Fig. 6).



**Fig. 6.** An example cluster head based wireless sensor network.



**Fig. 7.** LH scheduling.



**Fig. 8.** SPLL scenario.

## 4 Methodology

Figure 1 depicts the typical working of the SPL algorithm. This algorithm first calculates the shortest path available. If there are multiple shortest path available then it selects the one which has nodes which have the least number of links to other nodes. This idea enables the network to have the least number of paths which have common nodes thus reducing congestion and packet delay.

The LH algorithm has to do with the nodes situated at the border of the network. The LH algorithm is a scheduling algorithm. It schedules according to the hop distance from the Cluster Head. If two packets have the same distance then it takes into account the physical distance. If the physical distance is also same then, in our implementation, we just use the first in first out method. The packet which came first will be transmitted first. This algorithm is depicted nicely by the figure in Fig. 2.

## 5 Experimental Results

In the proposed work we have considered the network composed of our custom designed sensor devices, which can be of type 1 and type 2 and both of them can act either in normal mode or base station mode. The software modules are integrated with the sensor devices. The range of the data transmitted, speed and quality of received data and power consumption between type 1 and type 2 were the parameters chosen to be tested.

The reliability checks of the implemented hardware were performed using two nodes. Using only two nodes provides us with the range of data transmission, speed, quality of received data and power usage between the two nodes. The routing scheme was tested on a five nodes network.

Figures 7, 8 and 9 shows the packet drop in percentage with increasing distance between the two nodes at three different speeds - 250 Kbps, 1 Mbps, and

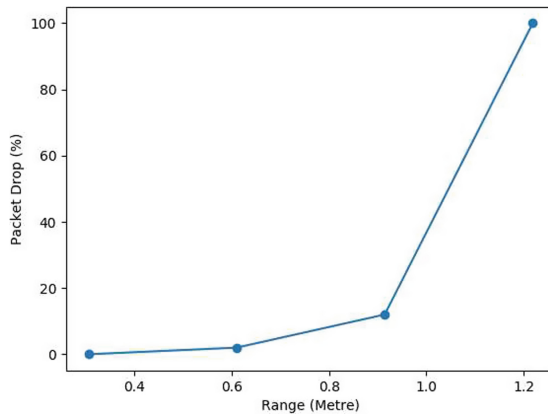
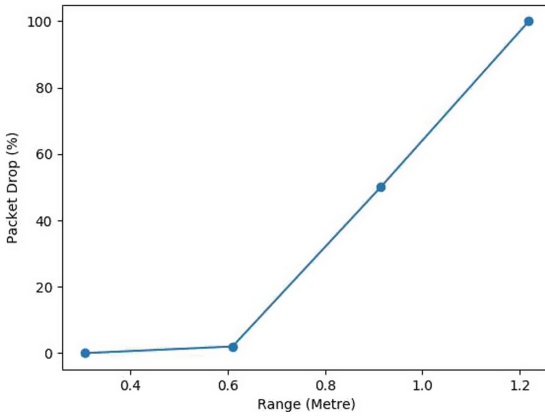
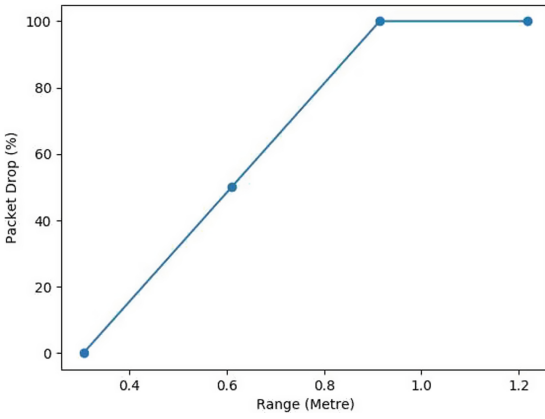


Fig. 9. Range vs packet drop at 250 Kbps

2 Mbps. Table 1 shows the power usage for model Type 1 and Type 2. While the power usage depends on the model the packet loss does not (Figs. 10 and 11).



**Fig. 10.** Range vs packet drop at 1 Mbps



**Fig. 11.** Range vs packet drop at 2 Mbps

**Table 1.** Power usage

	Type 1 model	Type 2 model
Normal mode	20 mA	10 mA
During transmission	+10–15 mA	+10–15 mA

## 6 Conclusions and Future Scope of the Proposed Work

The proposed research work is a preliminary base framework for a practical scenario and tested on a five nodes network. In the future, there will be the scope of implementation for hardware and software point of view.

Due to the nature of the used radio transceiver, we got a very low and unusable range. This issue can be easily resolved using better radio chip.

The next version will have the following things implemented

- Hardware
  - Embedded Sensors
  - More power and efficient micro-controller and Transceiver
  - Onboard non-volatile memory
- Software
  - Transport Layer
  - More Routing Protocols
  - Automated Data Collection
  - More Energy Efficient Code
  - More Modularity

## References

1. Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks. *Ad Hoc Netw.* **3**(3), 325–349 (2005)
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. *Comput. Netw.* **38**(4), 393–422 (2002)
3. Al-Karaki, J.N., Kamal, A.E.: Routing techniques in wireless sensor networks: a survey. *IEEE Wirel. Commun.* **11**(6), 6–28 (2004)
4. Atmel. 8-bit avr microcontroller atmega8a datasheet complete
5. Chang, J.-Y., Shen, T.-H.: An efficient tree-based power saving scheme for wireless sensor networks with mobile sink. *IEEE Sens. J.* **16**(20), 7545–7557 (2016)
6. Farhan, L., Kharel, R., Kaiwartya, O., Hammoudeh, M., Adebisi, B.: Towards green computing for internet of things: energy oriented path and message scheduling approach. *Sustain. Cities Soc.* **38**, 195–204 (2018)
7. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: attacks and countermeasures. In: *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003, pp. 113–127. IEEE (2003)
8. Lewis, F.L.: Wireless sensor networks. *Smart environments: technologies, protocols, and applications*. pp. 11–46 (2004)
9. Mukherjee, S., Biswas, G.P.: Networking for iot and applications using existing communication technology. *Egypt. Inf. J.* **19**(2), 107–127 (2018)
10. Perrig, A., Stankovic, J., Wagner, D.: Security in wireless sensor networks. *Commun. ACM* **47**(6), 53–57 (2004)
11. Nordic Semiconductor.: nrf2401+ single chip 2.4ghz transceiver preliminary product specification v1.0 (2008)