Wireless sensor network for campus monitoring

Nirmala P, Kishore R, Florintina C, Lakshmi S.VE, Mahalakshmi M
Electronics & Communication Engineering Department
SSN College of Engineering
Chennai, Tamilnadu, India

nirmalavp.ece@gmail.com, kishorer@ssn.edu.in, florintinachaarlas@gmail.com, virushajana@gmail.com,
mahaalakshmi28@gmail.com

ABSTRACT

Wireless sensor network is one of the emerging technologies which can be used for a variety of monitoring applications. With a focus on campus monitoring, this work proves the capability of wireless sensor network for complete automated HVAC system and street light monitoring system. A complete functional system consisting of ten wireless sensor nodes with temperature, humidity and light sensors were deployed across the campus for monitoring. The deployed system has collected vast amount of data for the past 6 months which can be used for continuous monitoring and better understanding of the scenario. The objective of the paper is to introduce to the research community, a report on unexpected challenges that were faced in the field and the possible solutions to overcome those challenges. The paper focuses on how to configure the wireless sensor nodes into networks using telosb motes with the help of Contiki OS.

Keywords

Contiki OS, Telosb motes, HVAC, WSN

1. INTRODUCTION

Wireless sensor network is a decentralized ad hoc network which is built from nodes, capable of measuring the various environmental parameters such as light, temperature, humidity, etc. The evolutionary development that we witness in smart system has the necessity for information sensed in real time. The key element required to automate a system is the real time sensory data. Wireless sensor networks find their applications in a wide range of domains. Environmental control in office buildings, Interactive museums, Detecting and monitoring car thefts, managing inventory control are only a few examples among them. Following are the most common applications of sensor networks: Military applications of sensor networks mainly focus on monitoring forces, damage assessment and attack detection. Further, battlefield surveillance and reconnaissance of opposing forces and terrains might also be an additional application. Environmental applications of wireless sensor networks include forest fire and flood detection, precision agriculture, monitoring the levels of air and water pollution and a newly added application which is the bio-complexity mapping of environment. Healthcare applications greatly benefit from sensor networks through vital sign monitoring and accident recognition. Tracking and monitoring of doctors and patients and drug administration in hospitals also prove to be of great use. Apart from applications that survey and control large area, small scale applications cater to the needs of home and office environments such as smart gardening system for home, office automation system to track employees and office monitoring systems to control the lighting and water systems. Industries also depend on sensor networks to reduce wiring effects in order to detect faults in machinery parts, measurements in chambers and rotating parts and technical parts inspection in remote places.

In this paper, wireless sensor network was employed to monitor an automated HVAC and automated illumination system. A set of ten nodes were deployed in the region under observation taking into account the range of each node and making sure that each node has at least one other node within its range to communicate to. To configure the nodes into the desired network and program them into their entitled work on open source software called the contiki was used. Satisfying the necessary conditions, a network was built with nine senders and one sink to collect the values sent by those senders, aggregate them and transmit the aggregate to a personal computer. The PC used a tool called the collect-view option of contiki os to represent the monitored data in a graphical form.

Valverde J et al [9], explained the monitoring of environmental parameters such as quality of sewage water and gas emissions in a coffee factory and the impact was evaluated. The paper laid emphasis on the need to maintain equilibrium between human development and a healthy environment. The need to make measurements in dangerous areas and the need for a more reliable way to do it made WSN all the more essential. A WSN platform called the Cookies was used. Certain nodes were programmed to sense the quality of water in the sewage compartment, some other nodes to measure the level of toxicity in the emissions and other nodes to act as routers to direct all the values to a sink node acting as a gateway and transmits them to a PC. The PC uses a tool called the Life Cycle Assessment (LCA) to store the data, retrieve them and represent them graphically. The paper had successfully determined a procedure to identify environment unfriendly processes in a more reliable and safe way. The complication of high power consumption due to frequent sensing was solved by connecting the motes to a direct power line. Though the temperature of the emissions would be way more than 100°C, the range of the temperature sensor is only -30°C to 50°C. Hence the experimentation time could not be long as the system to cool down the gas was not used.

Korkalainen et al [8], explained the common network simulation tools used to design a network for various applications. This paper gave a comparison about five common
network simulation tools based on their performance and data reliability. Tools NS-2, TOSSIM, OPNET, OMNET++, Prowler were selected based on the popularity of the tools among research community, active maintenance and support available. Among the compared tools, NS-2 and OMNET++ needed extensions and modifications for more accuracy. Those custom extensions can be written or downloaded from internet in all the above tools. Very high performance real-time simulation was near to impossible in all these tools without extensions. Commercial simulator OPNET offered better support and maintenance. In terms of scalability, NS-2 lags behind the other reviewed tools but leads in the C/C++ source code interchange when required interfaces and libraries are implemented on a target platform like TelosB. All presented tools lack proper environmental model and none of them is capable of 3D indoor simulation without modification.

Jong Chern Lim and Chris Bleakley [3] provided an efficient schedule for data gathering to maximize network lifetime while meeting user's requirements. This algorithm differed from previous works by adapting dynamically to schedule based on inter-node data correlation as well as the temporal correlation exhibited by the WSN measurements of data such as temperature, light, wind speed, humidity. This knowledge was utilized to reduce the number of measurements from a particular node at a specific period of time to meet application-specific sensing accuracy requirements. The algorithm incorporated novel round robin subset allocation methods for two-tier and multihop algorithm. Multihop method showed an increase in performance and data accuracy up to 55.3% and 30% when compared to the former algorithm. Rescheduling can maintain the performance of the system over a long duration of time but at the expense of increase in cost in terms of transmitted packets and load balancing can be used to lengthen the time the first node dies and hence retraining can be done.

Hwang Soo Lee et al [6] presented a WSN architecture to design the sensor network for military usage in remote, large scale environments. This architecture was based on the cluster-tree based multihop model with optimized cluster head election and the particular node design method to cater tactical requirements. Self organization of the nodes being deployed was ensured to develop energy-efficient model. QoS parameters such as emergency data, monitoring and tracking data and periodic simple data were introduced. Information flow in this WSN architecture only used one-way communication to gather and transfer information which cannot be employed in large scale environments where some nodes were out of range to the sink node. Security factors were introduced but not addressed properly in this architecture hence secured data transmission became an issue.

Constantinos Marios Angelopoulos et al [4], explained WSN based fully adaptive smart home irrigation system that catered to the water needs optimally. System architecture consisted of sensor nodes, externally connected soil moisture sensor along with the valve used to control the water flow towards the plants and a java application running in PC that stored the sensor information obtained from the soil moisture sensor. This architecture adapted itself to the current environmental conditions during the irrigation process so as to maintain the same level of soil humidity. Energy consumption became an issue when this system was deployed for a prolonged period of time and it required regular maintenance.

Basma M. Mohammad El-Baisoni et al [2], explained the proposed smart home automation system that provided authentication for home entrance and enhances home security. This system named as the Wireless Biometric Smart Home (WB-SH) system, provided seven services related to the automation systems namely entering system, burglar detection system, burglar deception system, monitoring and controlling home components system, monitoring health system, home plant system and internet access system. WB-SH had its own limitations due to the fact that it assumed a fixed BS and the fixed node, having to transmit the data frequently had to be awakened periodically for communication and that consumed more battery power and the lifetime of the system became an issue. The paper also described an improved SH design with a deep integration of wireless sensor networks and biometrics.

Iqbal Singh and Meenakshi Bansal [7] have monitored the water level in agriculture using specific algorithm in WSN. According to the algorithm, server node calculated its distance from the all other nodes and the shortest path was selected for relaying the data to the next node. The minimum threshold was calculated with the help of distance and minimum angle. If the minimum threshold was equal to zero then that particular node was chosen as the possible node and the packet information was relayed. Otherwise the connection was set to infinity. The values from the sender nodes were stored in the sink node. The sink node sent the stored data to the base station. This algorithm offered a maximum opportunity for delivery of data to the base station. The overall computation built a robust mechanism for delivery of information to the base station, thus reducing the packet loss. But the limitation was the lack of optimization technique to choose right cost path.

Santhosh simon [12] discussed the development and actual field deployment of wireless sensor network system. The architecture consisted of four sensor node for collecting the water level data. The sensor nodes were in a mesh structure and 50m apart. These sensors sensed the level of water and communicated with each other. Lower level wireless sensor nodes were attached to the sensor column and this collected the data from the electromechanical sensor and data packets were sent to the upper layers. The immediate upper layer consisted of cluster heads and the layer above this was the sink node. The sink node transmitted the data to the Field Control Centre (FCC). The sink node was the gateway and it was connected to an end system through a GPRS connection. Automated system monitored the water level and regulated it by using sophisticated systems and sending messages to the farmer. The disadvantage here was the difficulty to achieve the support from local farmers.

A.A.Nippunkumar [10] examined the intelligent lighting system using wireless sensor network. It consisted of an array of light sensor nodes which could communicate with the master node, providing the information about light condition at each sensor node. Based on the feedback information, master node decided as to which light sources were to be controlled. Once it was decided, master node transmitted the data frame to a particular light control node to control the light. It was easy to install and manage but it consumed high power. Recharging and replacing the batteries proved to be very difficult.

Reza mohammadoust’s [11] designed a system called Light Automatic Control System (LACS) which contained the centralized or distributed architecture determined by the
application requirement and space usage. Decision algorithm made use of constructed lighting effect dependency table which contained the calculated levels of lighting on each work plane. A decision process was developed which minimised the communication and computing resources required to moderate light intensity. When an external lighting did not have an effect on interior lighting, this system was capable of adjusting the light intensity of the room without using sensors. This paper also described the method to reduce the number of sensors, effect sensor power management and improved uniform distribution of illuminance on a workplace’s surface. It was showed that LACS design had superior integrated functions and had advantages over other designs. But it had low accuracy and calculation complexity.

The rest of the paper is organised as follows: Section 2 gives a brief overview of the hardware and software tools utilised in this paper. Section 3 illustrates the step by step procedure to build a basic two node network. Section 4 gives a detailed insight into the networking done in this paper. Section 5 provides the results obtained and their evaluation. Section 6 puts forth the challenges faced with possible solutions and the future work proposed.

2. HARDWARE AND SOFTWARE TOOLS USED

2.1 TelosB mote

TPR2420 ("TelosB") mote is an open-source experimental platform developed by the University of California, Berkeley. The platform provides integrated TI MSP430 microcontroller with 10k RAM, PCB radio, 2.4 GHz radio and sensor interfaces. TelosB mote is IEEE 802.15.4 compliant and hence it can be used for communication with other IEEE 802.15.4 devices provided that the configurable software protocols remain similar. These devices can be used as a platform for low power experimentation and research development. The TPR2420 is same as TPR2400 except that the former has inbuilt sensor suite provided.TPR2420 provides users with the possibility of using additional devices with the help of two expansion connectors to connect external sensors, LCD displays, digital peripherals. The device is powered by two AA batteries when operated independently and through host computer when connected through USB port

2.2 Contiki OS

Contiki OS is a pioneering open source operating system. Contiki was named after Thor Heyerdahl’s famous Kon-Tiki raft which travelled around the Pacific ocean with minimum resources. The ability of Contiki to function efficiently in a resource constraint environment earned it its name. It is built around an event driven kernel and possesses the advantage of multi-threading. Contiki makes loading and unloading of individual programs feasible even during the run time. The fact that it is implemented in the C language makes it more user friendly. Being an open source os, future development in Contiki is unhindered. Contiki has two communication stacks – uIP which uses TCP or UDP connections and rime which is a light weight layered communication stack. An application may use one of the two stacks.

Table 1: Important specifications of TelosB mote

<table>
<thead>
<tr>
<th>Specifications</th>
<th>TPR2420CA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor Performance</td>
<td>16-bit RISC</td>
</tr>
<tr>
<td>Program Flash Memory</td>
<td>48Kbytes</td>
</tr>
<tr>
<td>RAM</td>
<td>10k bytes</td>
</tr>
<tr>
<td>Configuration EEPROM</td>
<td>16k bytes</td>
</tr>
<tr>
<td>Serial communication</td>
<td>UART</td>
</tr>
<tr>
<td>frequency band</td>
<td>2.4GHz – 2.483GHz</td>
</tr>
<tr>
<td>Transmit(TX) data rate</td>
<td>250kbps</td>
</tr>
<tr>
<td>Outdoor range</td>
<td>70 m -100 m</td>
</tr>
<tr>
<td>Visible light sensor range</td>
<td>320nm to 730nm</td>
</tr>
<tr>
<td>Temperature sensor range</td>
<td>-40°C to 128.3°C</td>
</tr>
<tr>
<td>User interface</td>
<td>USB</td>
</tr>
<tr>
<td>Other interfaces</td>
<td>DigitalI/O,I2C,SPI</td>
</tr>
</tbody>
</table>

3. BUILDING A BASIC TWO NODE NETWORK

To demonstrate how the wireless sensor network used in this paper works, the programming and communication between a single sender and node is explained in detail initially. The following figure will enumerate the steps involved in programming the nodes.

![Figure 1: Steps involved to build a two node network](image)

3.1 Including the header files

The Contiki system is partitioned into core and loaded programs. While programming the node to act as a sender or receiver the various inbuilt programs in Contiki can be taken advantage of. To do so the required files must be included in the program. This includes the header files for the various sensors used such as light and temperature sensors that the sender uses to sense the environmental parameters. This inclusion process is followed by defining the static variables that the program uses.

3.2 Defining the process

An autostart process is defined with the functions essential for the sink to establish itself, for the sender to identify itself to the sink and obtain a global address for itself, for both the sender and receiver to transmit data and to handle any data it receives. Though Contiki provides us with an option of multi-threading, a single thread process would suffice this program.

3.3 Sensing the parameters
In order to configure the sensors and to manipulate them the respective header files had to be included. To sense any parameter, the corresponding sensors had to be activated initially and then de-activated. Once the sensor was activated, the desired parameter was obtained as an analog information. The inbuilt analog to digital convertor was employed to obtain the digital value. To retrieve that value, the inbuilt sensor.value() function was used. Once the value was obtained the sensor was de-activated. That value was then manipulated arithmetically to be expressed in the desired units. The following depicts how the temperature value was read from the in-built SHT11 temperature sensor of TelosB mote.

Header files required:

#include "lib/sensors.h"
#include "dev/sht11.h"
#include "dev/sht11-sensor.h"

Reading temperature value:

uint16_t temperature, temperature_in_celsius;
SENSORS_ACTIVATE(sht11_sensor);
temperature = sht11_sensor.value(SHT11_SENSOR_TEMP);
temperature_in_celsius (uint16_t)((-39.6+0.01*(float)temp));
SENSORS_DEACTIVATE(sht11_sensor);

3.4 Radioing the sensed value

Once the sensed value was obtained, it had to be transmitted to the sink. This task was made effortless by the TelosB motes with the aid of their cc2420 radio. As this experiment was conducted indoors, it was possible for the sender to transmit the data wirelessly to the sink that was within 20-30m range from it. Since it was a one to one communication, unicast transmission was employed, though broadcasting the data could also have worked. Initially a unicast connection was opened. The data was loaded into the buffer as a string and the address of the sink was set as the destination address. The address of the sink could be manually set to any desired value (For instance in this paper, the sink was given the address of 1.0). The data was then radioed to the sink by employing the cc2420 radio. The algorithm which was shown in figure 2 depicts the above mentioned process.

3.5 Receiving the radioed data and visualizing it on a PC

The sink was equipped with the function to receive the data transmitted to it by the sender. The data retrieved from the unicast connection was stored into the buffer. For future processing, the data was extracted from the buffer and transmitted through the serial port to a personal computer. The received value was then made known to the user by printing it on the terminal window. The required algorithm was shown in figure 2.

Step 1: Radioing the sensed value

Packet_buffer=(temperature_in_celsius, length_of_the_payload);
Unicast_address = address of sink;
Unicast_send(Unicast_connection, Unicast_address);

Step 2: Receiving the radioed data and visualizing it on a PC

Received_temperature = Packet_buffer;
Print the received temperature value on the terminal window;

3.6 Graphical representation of the sensed value

Contiki OS provides a graphical tool called as the collect-view which is capable of plotting the sensed values of the various in-built sensors against the time at which the value was sensed. In addition to the sensed values, the tool also represents graphically the battery power consumption, the wireless link between sender and sink, the data transmitted, the interval between which they are transmitted, the signal strength and other networking details. This tool was used to represent the various environmental parameters measured in one day graphically. The parameters were altered and the corresponding changes in the sensed value were duly noted. For instance, the mote was placed in a dark cabinet and then shifted to an illuminated area and the sharp increase in light intensity measured was portrayed by use of the collect-view. Similarly, the other sensors were verified.

4. NETWORKING THE NODES

The main experiment was done with a set of nine senders each of which sensed the required environmental parameters and routed it to a sink, either directly or with the help of neighbouring nodes. Both of the communication stacks, uIP and rime that Contiki offers, were employed to perform the same task.
4.1 uIP communication stack

When uIP stack was used, both sender and sink nodes were given specific port numbers and the sender nodes were programmed such that if the sink was within the range of that sender, it would transmit it directly to the sink and if the sink was out of the sender’s range, the sender was made to choose one of the other sender nodes that was within its range. Thus, each sender node had to be programmed to act as a router as well, capable of receiving the data transmitted to it and redirecting it to the sink. When the process began at the sender node, it was made to configure itself with the sink and obtain for itself a global address, unique within that particular network. Once configured and recognised as a member of the network, the sender nodes were programmed to activate the desired sensors, measure the corresponding parameters and deactivate the sensors. The sensed value after suitable arithmetic modulations so as to be expressed in desired units, was loaded into a buffer and transmitted to the sink by means of a uIP-udp connection.

To activate SHT11 temperature sensor:

\[\text{Temp} = \text{sensed value};\]
\[\text{Buffer} = \text{temperature in } ^\circ C;\]
\[\text{uip}_\text{udp}_\text{packet_sendto(}\text{client_uIP}_\text{udp}_\text{conn, Buffer, length of buffer, IP_address_of_sink, Port_address_of_sink});\]

To compare the new value with a predetermined threshold value:

\[\text{Value} = \text{Uip_appdata};\]
\[\text{If IP_address_of_sender == 1}\]
\[\text{Value}_1 = \text{Value};\]
\[\text{Aggregate} = \text{average of sensed values from the nine sender nodes};\]

Once the sink received the data packet, the sensed value was retrieved from the buffer and the values from all of the sender nodes were averaged. The aggregation process served as a means to obtain fairly correct data even when one or two of the nodes had gone rogue or transmitted incorrect values due to unstable conditions. A comparison with the sensed values from a known familiar environmental was helpful in obtaining an idea about the current situation. Algorithm for the sender and sink were given in the figure 6.

5. RESULTS, FURTHER DISCUSSIONS AND ANALYSIS

Data obtained by activating in-built temperature and light sensors were gathered and plotted over a period of time utilizing the collect-view tool integrated in the cooja simulator. Collect-view tool can also be used to plot various other parameters such as battery voltage, power consumption, neighbour nodes in the network, average temperature recorded in a network and relative humidity surrounding the nodes. Node Ids along with the number of packet received from that particular node and the interval between the arrival of successive packets can also be observed.

The temperature sensed by a single sensor was plotted with respect to time as shown in the figure 7. When the temperature of the environment was artificially increased using
an external source, a spike indicating sudden increase and decrease in temperature was observed.

Figure 8 shows the plot of light intensity values against time. When the surrounding’s light intensity was increased by using an external light source, corresponding variation was observed in the graph which was shown in figure 8.

<table>
<thead>
<tr>
<th>Figure 6: Algorithm for the sender and sink</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sender:</strong></td>
</tr>
<tr>
<td>Initialise neighbour table;</td>
</tr>
<tr>
<td>Equip the table with the identified neighbours;</td>
</tr>
<tr>
<td>Open a multihop connection on Rime channel;</td>
</tr>
<tr>
<td>If packet is received</td>
</tr>
<tr>
<td>Forward to a chosen node in the neighbour table list;</td>
</tr>
<tr>
<td>Activate the SHT11 temperature sensor;</td>
</tr>
<tr>
<td>Temp = sensed value;</td>
</tr>
<tr>
<td>Buffer = temperature in °C;</td>
</tr>
<tr>
<td>Packetbuf_copyfrom(buffer, length_of_buffer);</td>
</tr>
<tr>
<td>Multihop_send(Multihop_connection, address_of_the_sink);</td>
</tr>
</tbody>
</table>

|**Sink:**|
|If new data arrives:|
|Value = Packet_buffer;|
|If Rime_address_of_sender == 1|
|Value_1 = Value;|
|Aggregate = average of sensed values from the nine sender nodes;|
|Compare (Current_scenario, Familiar_environment);|

Group of 5 nodes, which consists of 4 client nodes and 1 server node, was deployed to illustrate the network formation as shown in figure 9. Client nodes activate their appropriate sensors periodically to monitor the field in which it is being deployed and relays those values to the server node which calculates aggregate with obtained values.

Terminal window which was shown in figure 10 gives the log of temperature collected from all the clients deployed in the network.
environment controlling application, the general purpose input/output pins that are available in the TelosB motes could be utilized.

To automate a lighting system, wireless sensor network could be utilized to sense the light intensity, compare it with a predetermined threshold to learn when the lights are needed and the corresponding digital value could be transmitted by the general purpose input/output pins to a relay circuit which could control the lights. A similar procedure could be employed to automate a HVAC system.

ACKNOWLEDGMENT

We thank SSN Trust, SSN College of Engineering for the financial support towards procuring equipments and components related to this work.

REFERENCES


