Design of an IoT Enabled Local Network Based Home Monitoring System with a Priority Scheme

Varun Tiwari
Department of Electronics and Communication Engineering,
Visvesvaraya National Institute of Technology, Nagpur, India
varun.etrx@gmail.com

Avinash Keskar
Department of Electronics and Communication Engineering,
Visvesvaraya National Institute of Technology, Nagpur, India
agkeskar@ece.vnit.ac.in

N. C. Shivaprakash
Department of Instrumentation and Applied Physics
Indian Institute of Science
Bengaluru, India
shiv@iap.iisc.ernet.in

*Abstract***—Internet of things (IoT) is a concept where all devices get connected to the internet to exchange information. IoT can be utilized in order to make a home environment interactive and smart. The required connectivity can be provided by either making all the devices Wi-Fi capable, which is costly, or by setting a local wireless sensor network (WSN) which later connects to internet. In this work various devices and sensors are designed and connected in a wireless sensor network within a home environment which generates a sensor dependent data packet. These data packets are then sent to the cloud through a master node. The master sends the data using a proposed priority scheme so that the critical information can be sent early. Simulations were performed to show that under the proposed scheme the loss of critical sensors information is reduced by around 75%.**

Keywords-internet of things; sensor network; wireless communication; home monitoring; priority scheme

I. INTRODUCTION

In order to improve human machine interaction all the machines need to have some information. Generally, these machines have sensors to provide some information but still a lot of information remains unknown. So in order to overcome this problem the concept of internet of things (IoT) came into existence. IoT is a concept under which every device is connected to internet. Hence, a lot of information is available to the device and can be processed when required. A smart home monitoring system can be a very suitable application of this IoT based interaction technology, as it involves a variety of devices and parameters to be connected. Connecting these devices requires the design of a network, through which all the information related to monitoring and control can be accessed over the internet. A lot of work has been done in the field of smart homes to increase human-system interaction [1]. Authors in [2] discussed an agent based middleware for smart home environment. A TV set top box with a special gateway to access home appliances was used in [3]. Authors in [4] used power line communication with a local wireless sensor network. Designing a smart home based on IoT requires smart units (smart light, smart sensors) which can be connected to a Wi-Fi network [5]. Enabling every device with Wi-Fi capability is a costly solution. Instead a local network can be set up which can collect all the data and send it to cloud. Such a local network based system was implemented in [6-8].

Proper monitoring of a home environment using local network based system requires a lot of sensor and control nodes. These nodes can generate a lot of data that can flood the network. So in order to overcome this problem, a local network based system is designed with a priority scheme, so that the most relevant data is sent to the cloud. A sensor depended data packet scheme is also used to decrease the data traffic [9]. The complete system is designed using open source platforms.

II. ARCHITECTURE

Figure 1 shows a typical arrangement of sensor network in a home environment. Figure 2 shows the general architecture of such a system. Different sensor nodes are connected to the master of the network, which is connected to internet to send/ receive data to/from cloud. If the number of sensors is more and the area to be covered is large then these sensor nodes can be connected through repeaters using different network topologies. The block diagram shows six sensors connected to master node through repeaters.

In general architecture the sensors can sense (temperature, humidity, intruder, etc.) and send the data to the master as well as can receive the data to perform any control operation (switching on/off fan, T.V, ACs, Alarm, etc.). Now depending upon the number of sensor nodes and local network topology there can be multiple hops before the data is actually received at the cloud. As the sensor node has to perform both sensing and controlling operation the network delay can be large, so a modified architecture can be designed in which the sensor nodes and control nodes can be formed to perform the separate operation. Also for nodes performing critical control/sense operation can be directly connected to cloud. Figure 2(b) shows the modified architecture. In this architecture the control and

The work was financially supported by the Department of Electronics & Information Technology under Ministry of communications and IT, Government of India and Center of Excellence on Combedded systems, Visvesvaraya National Institute of Technology, Nagpur, India.

sensor node performing normal operation is connected to cloud through a local network and nodes performing critical operation are provided with dedicated Wi-Fi capability to connect directly to the cloud.

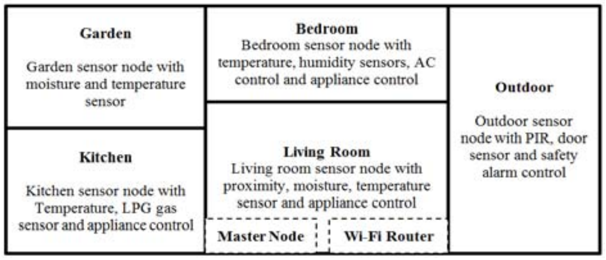

Garden
Garden sensor node with moisture and temperature sensor
Bedroom
Bedroom sensor node with temperature, humidity sensors, AC control and appliance control
Outdoor
Outdoor sensor node with PIR, door sensor and safety alarm control
Kitchen
Kitchen sensor node with Temperature, LPG gas sensor and appliance control
Living Room
Living room sensor node with proximity, moisture, temperature sensor and appliance control
Master Node
Wi-Fi Router

Fig. 1. Arrangement of sensors in a home environment.

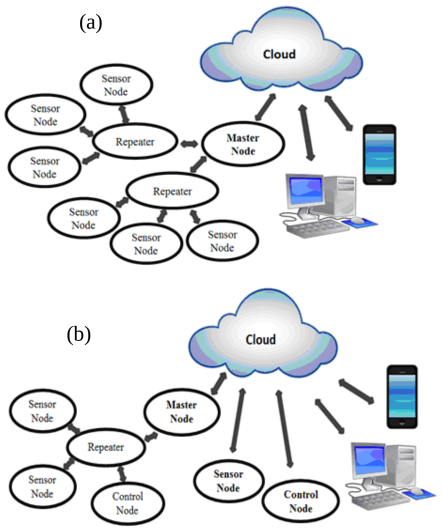


Fig. 2. Sensor network architecture: (a) General architecture with only master connected to cloud; (b) Modified architecture with master and critical nodes connected to cloud.

The addition of such nodes brings parallel data transfer capability to the system. It also minimizes the network delay. The separate control nodes minimize the sensor node's computational requirement. All the design components of the system are discussed in the next section.

III. System design

The implementation of this system requires design of various sensor nodes, repeaters, control nodes, master nodes along with development of networking algorithm, cloud network and mobile application development. All these designs are discussed below.

A. Sensor Node

The sensor nodes can be of two types, one that has a dedicated Wi-Fi module to directly connect to the cloud to send data and the other that forms a local network and then send the data through the master node. The local network based sensor nodes can be developed using Arduino Nano [10] as the controller and ZigBee/NRF24L01 as the transceivers. The nodes can be configured in star as well tree topology. A sensor node can send as well as receive data wirelessly. Arduino Nano V3 contains Atmega328 microcontroller and has a 8-channel 10bit analog to digital converter (ADC), which can be used to interface eight different analog sensors. It has 23 general multiplexed input-outputs (IOs) which can be used for relay/device control as per requirement [11]. Figure 3 shows the basic block diagram of a sensor node. Different sensor nodes can be designed using different sensors. The sensors can be interfaced as analog sensor or digital sensor. The analog sensor provides analog output in relation with the measuring parameter. Analog sensors can be interfaced by using ADC with proper conversion relation. The digital sensors can be interfaced to digital input of the controller. The general algorithm of an analog sensor node is as follow:

Algorithm 1:

1. Initialize the transceiver module with the address of the repeater/master node to send or receive data.
2. Get the analog value of the sensor and apply analog to digital conversion to get the 10-bit digital equivalent.
3. Apply the standard conversion formula to the obtained digital value to convert it to actual sensor value in proper units.
4. Send this converted value to the repeater/master node.
5. Wait for some time for other nodes to send data.
6. Repeat from step (2).

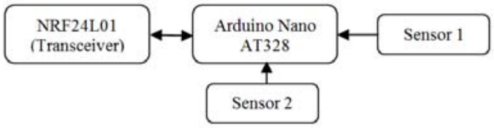


Fig. 3. Sensor node block diagram.

1) Sensor node with temperature sensor

LM35 can be used for temperature sensing. It provides the temperature information in terms of analog output [12]. Pin 2 of the LM35 provides the output it is connected to one of the ADC channel of the microcontroller. The ADC provides the 10-bit value of the analog data. This data needs to be converted to find the actual temperature using the conversion formula given in Table I. By using the conversion formula the temperature can be measured from -50°C to 150°C (upper limit of LM35). This range of value can be represented using 8-bit. The sensor node will send this 8-bit temperature data to the master node. Now if we consider a typical range of temperature in a home environment, we can limit the temperature range from -10°C to 53°C which can be represented using 6-bits only. This will reduce the data transmission load by 25%. Figure 4(a) shows a sensor node with a temperature sensor.

2) Sensor node with humidity sensor

DHT11 can be used for humidity sensing. It also calculates the temperature. It uses 1-wire protocol for data transfer. It gives the relative humidity in terms of percentage and the temperature in Celsius [13]. When the microcontroller sends a start pulse, DHT11 moves from low-power-consumption mode to running-mode and waits for the microcontroller to complete the start pulse. Once it is completed, DHT11 sends a response signal of 40-bit data that includes the 8-bit integer relative humidity, the 8-bit decimal relative humidity, the 8-bit integer temperature, the 8-bit decimal temperature and the 8-bit checksum. Once the data is received, DHT11 will change to the low power consumption mode until it receives a start signal from the microcontroller again. DHT11 has a measuring range of 20-90% relative humidity and 0-50°C temperature. The measuring range of temperature is in the typical range of temperature in home environments and can be send using 6-bits considering only the integer part after rounding. The measuring range of relative humidity is also in the typical range of home environments and can be sent using 8-bits only, again considering only the integer part after rounding. Now both the data can be sent using 14-bits with an accuracy of around ±0.5°C for temperature and 0.5% for relative humidity. This will reduce the data transmission load by around 43.75%. Figure 4(b) shows the image of one such sensor node.

3) Sensor node with soil moisture sensor

Soil moisture sensors are designed to estimate the water content of the soil based on dielectric constant of the soil [14]. The dielectric constant of the soil increases as the water content increases. The dielectric constant of water is much larger than the soil or air, so the increase of water content is soil increases its dielectric constants value. The sensor is interfaced as an analog sensor. For calibration purpose first the ADC value is measured when the sensor is in air and then again ADC value is measured when the sensor is in water. The in between values represent the different level of water content in soil. Table I shows the conversion. The ADC provides 10-bit value which can be classified as four different states of soil’s water content. Thus the sensor node will have to send only 2-bit of data to master node. Figure 4(c) shows the image of the sensor node.

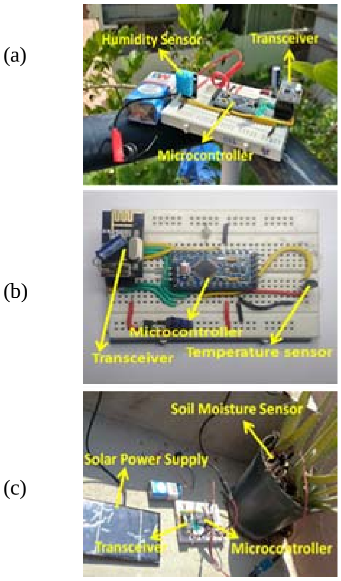


Fig. 4. Sensor nodes; (a) Sensor node with temperature sensor; (b) Sensor node with humidity sensor; (c) Sensor node with soil moisture sensor

TABLE I. SENSOR INTERFACING AND CONVERSION TABLE.

Sensor	Interface and Conversion method
Temperature (LM35)	Analog Interface [((ADC value) *3300 / 1024) - 500] / 10 Temperature in Celsius
Humidity DHT11	Sensor sends actual values through 1-wire protocol Relative humidity in percentage, Temperature in Celsius
Soil Moisture	ADC value < 200 => Dry soil ADC value ∈ [200,500) => Limited water ADC value ∈ [500,800] => Sufficient water ADC value > 800 => Excess water soil
Magnetic Relay (NO)	Digital value: High – Door open, Low – Door closed
PIR (HC-SR501)	Digital value: High – Change detected, Low – Settled no change
LPG gas (MQ6)	C = 1036.5*((1024/s - 1))^-2.392*(-0.0120*(T-20) * -0.0076*(H-65) + 1) Where, C=concentration in ppm s=sensor value in range of [0,1023] T=temperature in Celsius H=Relative humidity in percentage

4) Sensor node with magnetic relay

Magnetic relays [15] can be used as a digital sensor to monitor opening and closing of a door. It can be of normally open or normally closed type. Depending upon the type of relay it changes its state (open to close / close to open) with the effect of magnetic field. To use it for a door sensor a small magnet can be mounted to the door. The node will send only 1-bit to the master. Multiple sensors can be placed on critical doors. Figure 5(a) shows the image of the sensor node.

5) Sensor node with passive infra-red sensor

The passive infra-red sensors can be used for motion detection or light intensity change. HC-SR501 can be used as the sensor for motion detection [16]. It provides digital output and can be interfaced easily with any digital pin of the microcontroller. The node will also send only 1-bit to the master. Figure 5(b) shows the image of the sensor node.

6) Sensor node with LPG sensor

MQ-6 can be used as a LPG gas sensor. It is highly sensitive to LPG, iso-butane and propane with small sensitivity to alcohol and smoke. It can detect propane in the range of 300 ~ 10000 ppm [17]. It can be interfaced with the microcontroller as analog sensor. The 10-bit ADC value can be converted in ppm by using the conversion equation given in Table I. Now considering the lower explosive limit (LEL) of the propane and butane as specified by the ‘Centers for Disease Control and Prevention’ [18], the ppm concentration can be further classified as no leakage (<650ppm), leakage (650-950ppm) and danger (>950ppm). Leakage is triggered at about 35% of LEL (1900ppm) and danger is triggered at about 50% of the LEL. Now the node needs to send any of the three states which requires only 2-bit. Figure 5(c) shows the sensor node.

B. Control Node

A control node is designed to control any appliance over internet. A control node can be of two types, one that is connected to the cloud through master node and the other that has dedicated Wi-Fi module for cloud connectivity and a relay connected to one of the I/O of the controller. These control nodes can be used at critical locations where immediate control operation is required when triggered (such as: theft alarm system). It continuously fetches the data from the cloud to maintain the state of the relay as required. Figures 6(a) and 6(b) show the block diagram of control nodes. Figures 6(c) and 6(d) show the general and dedicated control nodes.

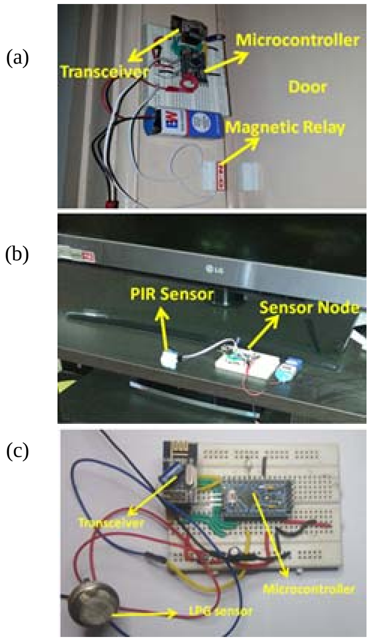


Fig. 5. Sensor nodes; (a) Sensor node with magnetic relay; (b) Sensor node with PIR sensor; (c) Sensor node with LPG sensor.

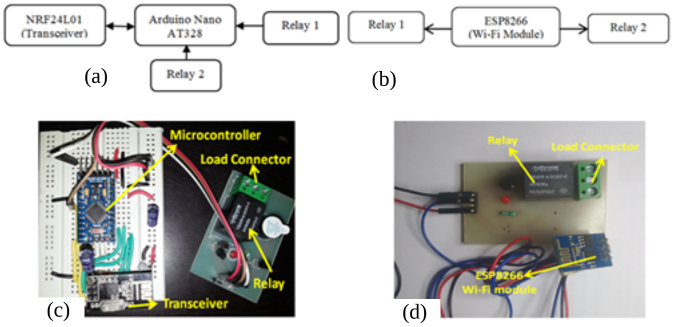


Fig. 6. Control node: (a) General control node block diagram with transceiver; (b) Dedicated control node block diagram with Wi-Fi module for direct connectivity to the cloud; (c) A sample image of general control node with relay; (d) A sample image of dedicated control node with relay and Wi-Fi module.

The dedicated control node algorithm is as below:

Algorithm 2 (Dedicated control node):

1. Initialize the Wi-Fi module to connect to a specified Wi-Fi network.
2. Connect to the cloud using API key.
3. Get the values of switches from the cloud.
4. Assign the switch status to the I/O pins of the controller.
5. Repeat from step (2).

C. *Repeater*

In order to increase the range of the wireless network repeaters can be used. A repeater consists of Arduino Nano as the controller and ZigBee/NRF24L01 as the transceivers. A repeater is configured to receive data from different child nodes and send it to the other repeater or master node. Figure 7(a) shows the block diagram of the repeater and Figure 7(c) shows a sample repeater. The interfacing algorithm is described below.

Algorithm 3 (Repeater node):

1. Initialize the transceiver module to receive data from child nodes.
2. Initialize the transceiver module with the address of the repeater/master node to send data.
3. Wait to receive data from any child node.
4. Send the received data to other repeater/master node.
5. Repeat from step (3).

D. *Master Node*

The master node also uses the Arduino Nano as the controller along with ZigBee/NRF24L01 as transceiver. A master node has Wi-Fi module for internet connectivity with the help of which it can send all the collected data from the nodes to the cloud. A master node can send as well receive data. Figure 7(b) and Figure 7(d) shows the block diagram and an image of master node.

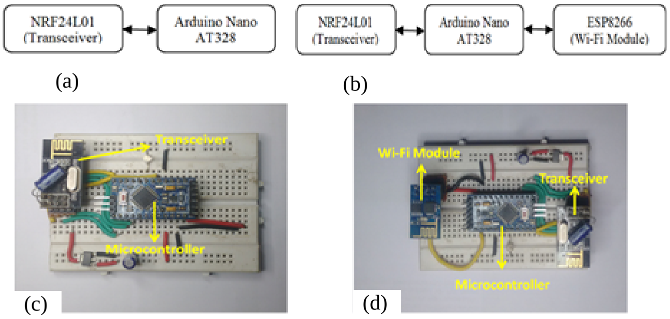


Fig. 7. Repeater and Master nodes: (a) Repeater node block diagram with transceiver; (b) Master node block diagram with Wi-Fi module and transceiver; (c) A sample image of repeater; (d) A sample image of master node.

E. *Cloud*

All the data collected by the master is continuously send to the cloud. A user can have his own private account so that nobody else can view the data. The private account is provided with a security key which is required by any other application programming interface (API) that wants to get the information. The information is kept and stored on the cloud and can be viewed by the user when required. Different channels can be created to store data. A single channel can have multiple fields which can store information from various sensors. A single sensor is linked to a particular field of a channel. The information can be accessed in different ways such as: HTTP request can be send to cloud along with API keys to get the data from any field of a channel; small programs can be created to react on any conditions raised by any sensor value; Talkbacks or commands can be created to handle switches or other HTTP requests. Figure 8 shows the architecture of the cloud [19].

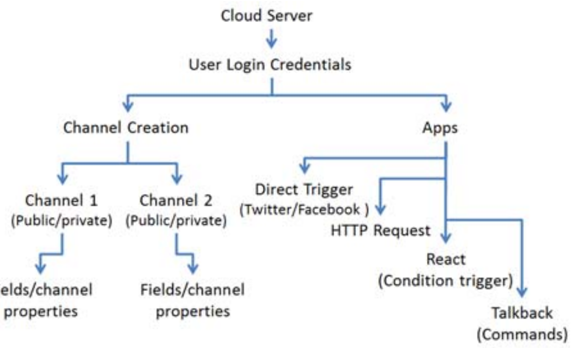
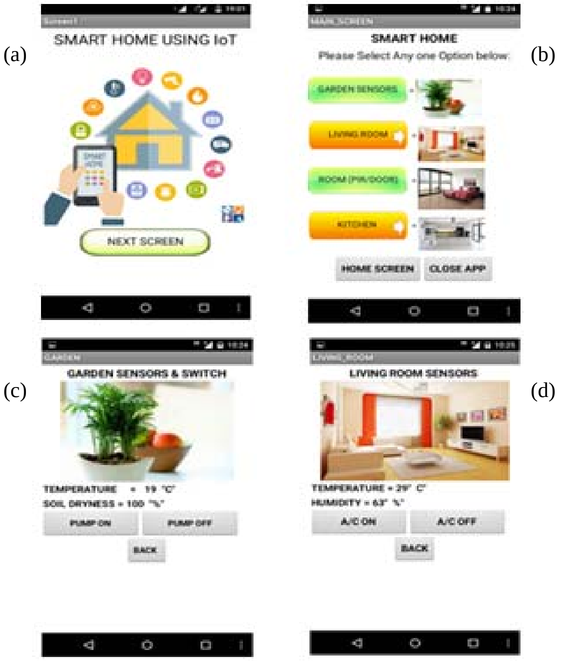

Cloud Server
User Login Credentials
Channel Creation
Apps
Channel 1 (Public/private)
Channel 2 (Public/private)
Direct Trigger (Twitter/Facebook)
HTTP Request
React (Condition trigger)
Talkback (Commands)
Fields/channel properties
Fields/channel properties

Fig. 8. Cloud architecture for IoT based device support.

F. Mobile Application

A mobile application can be developed for the system using MIT's App Inverter [20] which is an open source platform. It provides a block based technique for creating android applications. This application is also secured and specific to a particular user as it uses the secured key that the user has on the cloud account. The application has the capability to access the data and to send any control signal to home from anywhere around the world. Figure 9 shows the different screens of the application.



TEMPERATURE = 19 "C"
SOIL DRYNESS = 100 "%"

TEMPERATURE = 29" C"
HUMIDITY = 63" %"

Fig. 9. Mobile application for cloud based monitoring and control: (a) Applications home screen; (b) Main screen; (c) Garden sensor control/monitor screen; (d) Living room sensor control/monitor screen.

The complete system has been designed with 8 sensor nodes 2 repeaters 2 dedicated control nodes and 1 master node. Four sensor nodes are configured to send information to one repeater and the other four to the other repeater. The configuration requires updating transceivers with the address to which it has to send or receive data. The repeaters are configured to then send the data to the master node. Now the master node sends the data to the cloud with a priority scheme which is discussed in the next section.

IV. PROPOSED PRIORITY SCHEME

A sensor node will always sense and generate data. These sensor nodes if continuously send the data then the network will get flooded. So in order to avoid that, a periodic scheme can be adopted, where the sensor node will generate data depending upon its period. This sensing and sending period depends upon the priority of the sensor at the sensor node. Table II shows one such scheme where the intruder detection sensor has got the highest priority and the soil moisture sensor has got the lowest. A more critical parameter is measured more frequently. This periodic sensing reduces the energy consumption and the data transmission load of the network.

TABLE II. SENSOR PRIORITY SCHEME.

Priority List	**Traffic rate**	**No. of bits after conversion**	**No. of bits for transmission**
Intruder Detection sensor node	Every 1secs	1	1
LPG Gas sensor node	Every 5secs	10	2
Temperature sensor node	Every 1 minute	10	6
Humidity sensor node	Every 1 minute	8	8
Soil Moisture sensor	Every 15 minute	10	2

All the data received by the master needs to be sent to the cloud. But, many times it is possible that we end up sending same information all the time as no parameter is changing. So a priority scheme can be adopted to reduce this data transmission load. In this proposed scheme the sensor node sends the sensor data along with an additional priority bit. The priority bit is set to '1' whenever there is a change in the sensor value from the previous recorded value. So the master gets the data along with the information that the data is new or repeated. Now, according to the priority bit of the sensor data, master prepares a priority table to send the data. This priority table has the sensor data whose values have been changed from the previous recorded value. Again as different type of sensors has different priority level as shown in Table II this priority table is again re-arranged so that the most critical data can be send with highest priority. Also to reduce loss of information, redundancy in data transmission is used. A high priority data is given more redundancy compared to low priority data. Figure 10 shows the flow chart for the master node with the priority scheme.

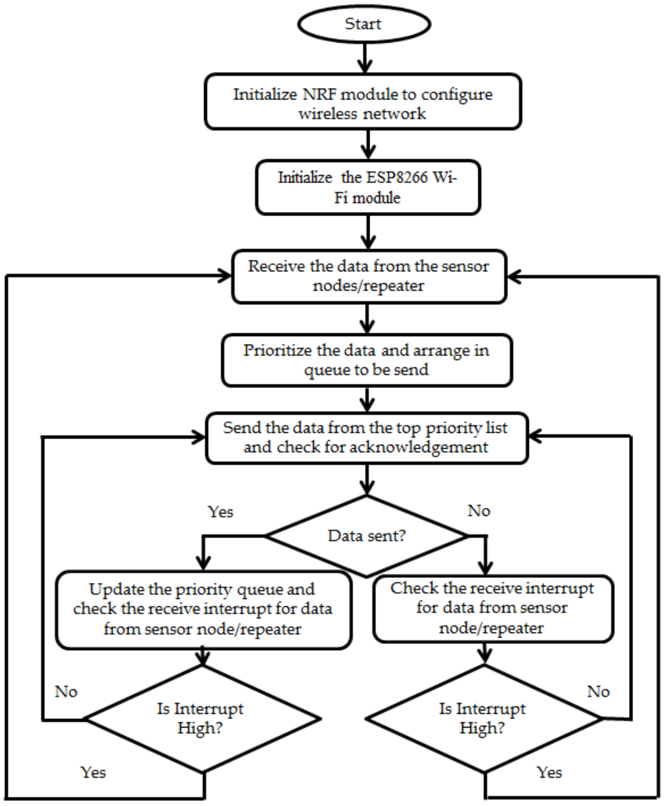

Start
Initialize NRF module to configure wireless network
Initialize the ESP8266 Wi-Fi module
Receive the data from the sensor nodes/repeater
Prioritize the data and arrange in queue to be send
Send the data from the top priority list and check for acknowledgement
Data sent?
Yes
No
Update the priority queue and check the receive interrupt for data from sensor node/repeater
Check the receive interrupt for data from sensor node/repeater
Is Interrupt High?
No
Yes
Is Interrupt High?
No
Yes

Fig. 10. Master node's flow chart with priority scheme.

V. SIMULATION

To investigate the benefit of the proposed priority scheme simulations were performed. Table III shows the simulation parameters.

TABLE III. SIMULATION PARAMETERS.

Sensor	Interface and Conversion method
Total number of sensors (S)	100
Priorities	P1>P2>P3>P4
Transmission Frequency	P1-every 1 second P2-every 2 second P3-every 3 second P4-every 4 second
No. of sensors with different priorities	25 each
Probability of change in sensor reading	Randomly distributed for 100 sensors
Probability of loss of data packet from master to cloud	P_{loss}
Simulation time	6000 seconds
Transfer schemes	1. Without redundancy in data 2. With redundancy 3. With proposed priority based redundancy

In this simulation comparison is done in between the data transfer schemes from the master to the cloud. We consider a system where we have 100 sensors with four equally distributed different priorities. Sensors with priority P1 sends data every 1 second, sensors with priority P2 sends data every 2 seconds and similarly with P3 and P4 sends at 3 & 4 seconds. As discussed previously, the master sends the data of the sensor for which there is change in the value. Every sensor can have different probability of change in the sensor value depending upon the measuring parameter and position. To model this condition we randomly provide these 100 sensors a probability of change in sensor value. This probability of change is considered a uniform distribution so that wide variety practical sensors can be modeled during simulation. At any given time the master will have to send the data that is generated due to change in sensor value from different sensors. The changed sensor value is considered as a single data packet that the master will send to the cloud. If at a given time 'n' sensors have observed changes in value then there will be 'n' number of data packets to be sent to the cloud. These 'n' sensors can be of different priorities. Figure 11 shows the plot of total number of sensors with change in sensor value with respect to increase in time. The plot shows the count of different priority sensors as well as the total count. Due to more frequency of the priority P1 sensors more data packets are generated by these sensors.

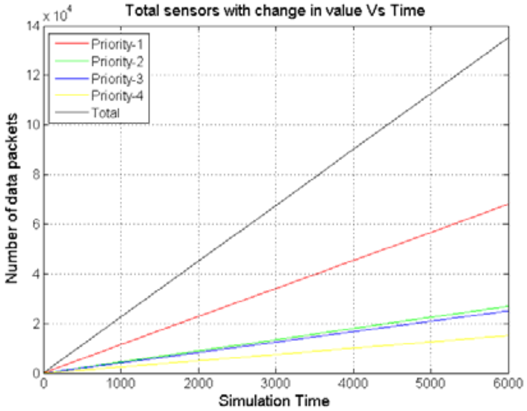
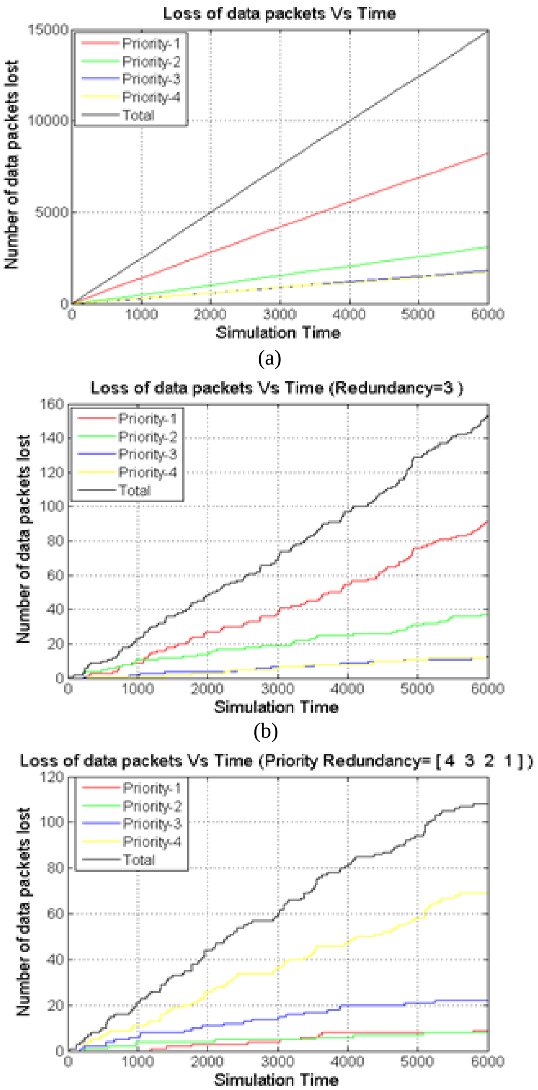


Fig. 11. Plot of number of sensors with change in sensor reading value versus time.

While transmitting the data to the cloud there can be some information loss. This loss can be because of network congestion or limited number of retransmission from the transmission control protocol under packet loss. To model this condition P_{loss} is defined as the probability of loss of data packet while transmission. Figure 12(a) shows the plot of loss of data packets with time. Here P_{loss} is taken as 0.1 to model the effect in short time. The loss of data packet is proportional to the generated data packets by the sensors. If we want to reduce this packet loss then a redundancy based data transmission can be applied where multiple copies of each data is transmitted. This method will reduce the packet loss but will increase the transmission load. Figure 12(b) shows the plot of the redundancy based transmission scheme. The overall data packet loss reduces with this scheme but the packet loss is again proportional to the generated packets. As we are more interested in high priority sensors data the proposed priority based scheme can be applied to the system. Under this priority based scheme the master arranges the changed sensor values according to the priority of the sensors and then more amount of data redundancy is applied to the high priority sensor compared to lower one. Figure 12(c) shows the plot of the loss of data packets with this priority based scheme. It can be observed that the proposed scheme reduces the packet loss for the high priority sensors and most of the packet loss is due to the low priority sensors only. Here the amount of redundancy applied to the sensor is four for P1 sensors, three for P3 sensors and two for P2 sensors. The packet loss for P1 sensors is reduced by about 88% and for P2 about 75%.



(a)

(b)

(c)

Fig. 12. Loss of data packets: (a) without redundancy in transmission; (b) with redundancy=3 in transmission; (c) with priority based redundancy of 4, 3 and 2 for sensors P1, P2 and P3.

Figure 13 shows the plot of amount of data packets generated by the different priority sensors for the three discussed schemes. For sensors with P1 priority four times the actual data is generated. For P2 the generated data is three times which is equal to the priority-less redundancy based scheme. For P3 sensors less data is generated as it is of low priority in the priority based scheme. For P1 no additional data is generated under the priority based scheme to keep the transmission load less. Figure 14 shows the plot of average total number of data packets generated under the three schemes after 10 iterations. It can be observed that there is a small increase in the data packets for the proposed scheme but with huge reduction in loss of data packets for high priority sensors. A minimum of 75% reduction in loss of data packets was observed for high priority sensors P1 and P2.

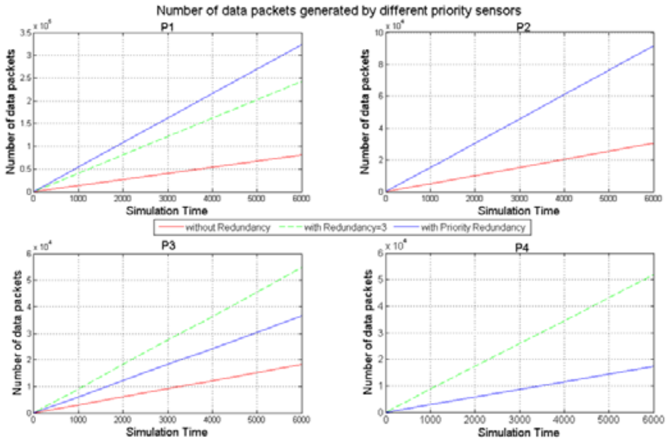


Fig. 13. Amount of data packets generated under different schemes for different priority sensors.

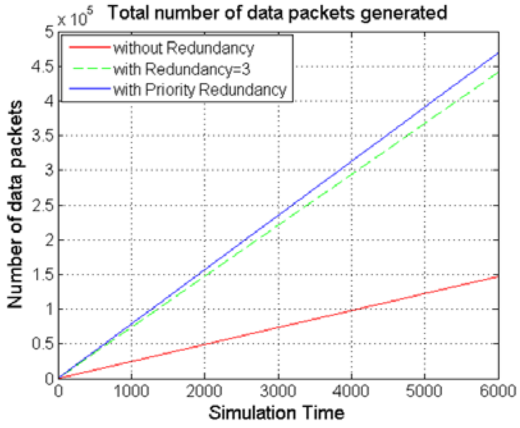


Fig. 14. Total data packets generated under different schemes.

VI. Reliability and Fault Detection

Sending most critical data at a given time makes the system reliable. Any critical event (fire, intruder, LPG leakage, etc.) can be notified with priority. Along with this the system can be further enhanced by maintaining redundancy in the number of same sensors with critical operations. A fire detection node will have multiple such sensors to detect fire, so that if one node fails the other can send the information. The system can also have a fault node detection scheme where the master can detect the faulty node. The network configuration is known to the master node. The network is connected using star and tree topology. So the repeaters are important in keeping the network connected. The master first sends an acknowledgement request to the repeaters one by one. If the acknowledgement is received within time Tw (Tw is the time for which master waits to receive the acknowledgement signal from the receiver) then it means the repeater is working, if not then the repeater is marked as a fault node. If all the repeaters are working properly then the master sends the acknowledgement request to the sensor nodes one by one. Again if the acknowledgement is received within time R*Tw (the factor R depends on the number of hops required from the master to the sensor through repeater) then the node is working properly otherwise is a fault node. Once a node is found to be a faulty node, the master sends the information to the cloud so that it can be reflected on all the connected devices (mobile phones, cloud accounts of the user).

VII. Result and Comparison

The system was designed using ultra low power ICs such as NRF24L01/ZigBee as transceiver modules and ESP8266/ CC300 as Wi-Fi modules. The complete system was designed and tested using 8 sensor nodes, 2 repeaters 1 master and 2 dedicated control nodes. Four sensor nodes were configured to send the data to master through one repeater and the other four through the other repeater. The dedicated control nodes were directly controlled through cloud. Once the system is live the user can monitor and control the system from anywhere using cloud. Figure 15 shows the temperature on one of the field of the cloud channel. Figure 9(c) and Figure 9(d) shows the mobile application screen with live data from the sensors. Table IV shows the power consumption of the ICs used [11-13, 16-17, 22, and 24]. Table V shows the comparison between two network modules ZigBee and NRF24L01 [21-22]. Table VI shows the comparison between the Wi-Fi modules [23-24].

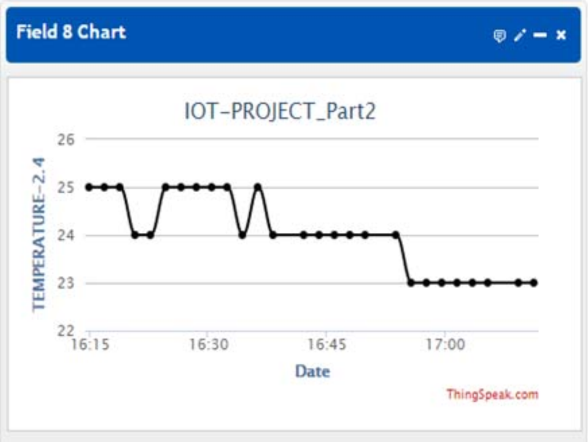


Fig. 15. Temperature plot on the field of cloud channel.

The system was designed and tested using both NRF24L01 and ZigBee modules. The NRF24L01 module provided all the necessary features at a very low cost. The system developed is secured and reliable because:

- The user has a login credentials for his cloud account to access or control live data.
- The account can be made private which provides a special security key for access over internet.
- An API key is must for any access along with the information of channel ID with private accounts.
- All the API keys can be hard coded in the hardware.

TABLE IV. POWER CONSUMPTION OF NETWORK MODULES.

Modules	Standby	Power Down	Active
NRF24L01	26uA	900nA	<13.5mA
ESP8266	--	10uA	0.3mA (average)
ATmega 328	0.5mA @ 1MHz, 3.3V	0.2uA @ 1MHz, 3.3V	>1mA
DHT11	100uA	100uA	0.2mA (average)
MQ6	--	--	Less than 750mW heating consumption
HC-SR501	--	--	<65mA
LM35	--	--	Less than 158uA @ 5V, 125°C

TABLE V. COMPARISON TABLE OF NETWORK TRANSCEIVER MODULES.

Parameter	XBee	NRF24L01+
Range	30m	50m
Power	Ultra-Low power device (1uA at standby)	Ultra-Low power device (1uA at standby)
Max Data rate	Upto-250Kbps	Upto-2 Mbps
Topology	Star, Tree & Mesh	Star & tree
Hardware Interface	TTL	SPI
Configuration	AT commands	Firmware update
Price (approx.)	15$	3$

TABLE VI. COMPARISON TABLE OF NETWORK WI-FI MODULES.

Parameter	CC3000	ESP8266
Packets	TCP and UDP	TCP and UDP
Modes	Client & Server	Client & Server
Interface	SPI	TTL Serial
Encryption	Up to WPA2-PSK	Up to WPA2-PSK
Digital Pins	0	9
Analog Pins	0	1
Programmable Microcontroller	No	Yes
Price (approx.)	35$	10$

VIII. Conclusion

The system is developed on complete open source platforms using low cost and ultra-low power ICs. The periodic data sending scheme helps in keeping the data traffic rate of the system low with low consumption of energy. Sending the typical range of sensors also reduces the data transmission load. The proposed priority scheme helps in sending the most relevant data at any given time to cloud. Sensor redundancy in the system along with the fault node detection scheme makes the system more reliable. The system provides the monitoring and control to the user over the internet which can be linked to many other applications as required.

ACKNOWLEDGMENT

The authors would like to thank Dr. Farukh Hasmi for his suggestions to improve the quality of the paper and Saurabh Chakole for his suggestions in improving the hardware design.

REFERENCES

[1] C. L. Wu, L. C. Fu, "Design and realization of a framework for human–system interaction in smart homes", IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol. 42, No. 1, pp. 15-31, 2012

[2] H. Takahashi, T. Ito, K. Sasai, G. Kitagata, T. Kinoshita, "Design and implementation of an agent-based smart home system", 4th IEEE International Conference on Awareness Science and Technology, pp. 227-232, 2012

[3] J. Kim, E. S. Jung, Y. T. Lee, W. Ryu, "Home appliance control framework based on smart TV set-top box", IEEE Transactions on Consumer Electronics, Vol. 61, No. 3, pp. 279-285, 2015

[4] M. Li, H. J. Lin, "Design and implementation of smart home control systems based on wireless sensor networks and power line communications", IEEE Transactions on Industrial Electronics, Vol. 62, No. 7, pp. 4430-4442, 2015

[5] Y. Wenbo, W. Quanyu, G. Zhenwei, "Smart home implementation based on Internet and WiFi technology", IEEE 34th Chinese Control Conference (CCC), China, July 28-30, 2015

[6] C. Osiegbu, S. B. Amsalu, F. Afghah, D. Limbrick, A. Homaifar, "Design and Implementation of an Autonomous Wireless Sensor-based Smart Home", IEEE 24th International Conference on Computer Communication and Networks (ICCCN), pp. 1-7, Las Vegas, USA, August3-6, 2015

[7] H. Zheng, H. Zhang, L. Pan, "Modeling and Analysis of ZigBee Based Smart Home System", IEEE 5th International Conference on Digital Home (ICDH), China, November 28-30, 2014

[8] S. Zhang, P. Xiao, J. Zhu, C. Wang, X. Li, "Design of smart home control system based on Cortex-A8 and ZigBee", IEEE 2014 5th International Conference on Software Engineering and Service Science (ICSESS), pp. 675-678, China, June 27-29, 2014

[9] A. Zanella, N. Nicola, A. Castellani, L. Vangelista, M. Zorzi, "Internet of things for smart cities", IEEE Internet of Things Journal, Vol. 1, No. 1, pp. 22-32, 2014

[10] Arduino, https://www.arduino.cc/en/Main/ArduinoBoardNano

[11] Atmel, "Atmel 8-bit microcontroller with 4/8/16/32kbytes In-system programmable flash", Datasheet, 2014

[12] Texas Instrument, "LM35 Precision Centigrade Temperature Sensors", Datasheet, January 2016

[13] D-Robotics, "DHT11 humidity & temperature sensor", Datasheet, July 2010

[14] Vernier, "Soil Moisture Sensor", Datasheet, June 2012

[15] Standex Meder Electronis, "MK04 series reed sensors", Datasheet, January 2015

[16] "HC-SR501 PIR motion detector", Datasheet

[17] Hanwei, "Technical data MQ-6 gas sensor", Datasheet

[18] "CDC - Immediately Dangerous to Life or Health Concentrations (IDLH): L.P.G. - NIOSH Publications and Products", http://www.cdc.gov/niosh/idlh/68476857.html, accessed: 25 September 2016

[19] "Internet Of Things – ThingSpeak", http://www.thingspeak.com

[20] "MIT App Inventor | Explore MIT App Inventor", http://appinventor.mit.edu/explore/

[21] Digi International, "XBee®/XBee‐PRO® RF Modules -802.15.4 ‐ v1.xEx", Datasheet, September 2009

[22] Nordic Semiconductor, "nRF24L01+ Single chip2.4GHz Transceiver", Datasheet, September 2008

[23] Texas Instrument, "TI SimpleLink™ CC3000 Module – Wi-Fi 802.11b/g Network Processor", Datasheet, November 2012

[24] Espressif Systems, "Espressif smart connectivity platform: ESP8266", Datasheet, October 2013