Designing and Fine Tuning a Fire Safety Monitoring System for Smart Buildings using RPL Protocol

T.Anusha and Dr.M.Pushpalatha

(aa5293@srmist.edu.in) (pushpalm@srmist.edu.in)
Department of Computer Science Engineering,SRM Institute of Science and Technology.

Abstract. Internet of Things is a well established technology for a wide variety of applications like Smart Homes, Smart Buildings, Healthcare, Agriculture, Aviation, AMI etc. Fire tragedy is one of the most damaging risk that could occur for an industrial, commercial or residential building. IoT can be successfully employed in large buildings to safeguard the occupants and also its infrastructure. To this end, a smart building is equipped with large number of economic sensors that forms an LLN (low power lossy network) with border router for Internet connectivity. RPL is a routing protocol for LLNs that ensures that tiny battery operated sensor devices could form a network with automated address configuration, spontaneous topology formation, IPv6 connectivity and fail-safe operation. This paper proposes an enhanced application of RPL protocol which increases the longevity of the portable nodes and assures quick response during a fire. This paper explains the required IoT infrastructure along with an application that provides a live web page with color codes to monitor the temperature of the different regions of a smart building.

Keywords: IoT, Smart Building, RPL, Fire Safety, Web Monitoring

1 Introduction

Smart Buildings have a plethora of functions like automated doors, windows, access control, temperature / lighting control, surveillance, fire safety etc. Fire safety of a building is a very important aspect as it assures the safety of the people and protects the material value of the property. Though there are building safety rules [1, 2] that are framed by the government for precautions, active monitoring of fire safety is an implication of the individual building owner. Mechanical devices like sprinklers with water storage whose plugs melt in fire, provide first line of defense in fire fighting. But they incur huge costs in procurement, installation, additional plumbing works and requires maintenance. Additionally, it may not be feasible to retrofit them in an older building.

In the event of a fire, there will be huge losses or critical damages even if it is put off at the earlier stage. But an intelligent system can automatically predict a fire even before its occurrence[3]. Such a system will monitor temperature of various units of the building and can deploy prevention tactics like switching off gas and power. It can also pin point the location of the problem and the extent of it. So it is highly advantageous to install an automated fire safety monitoring system (FSMS) to detect changes in temperature to avert fire. An efficient FSMS should be swift, reliable, economic, easy to install and easier to maintain.

IoT connects all devices to the Internet so that exchange of information can happen seamlessly between devices to devices or between humans and devices. There are many international bodies that have developed various standards to make IoT a reality. IETF (Internet Engineering Task Force) is such a body consisting of network professionals, industries who manufacture communication systems and researchers to come up many protocols that are needed for IoT[3]. Some of their very important protocols include 6LoWPANs (Internet over Low Power Wireless Personal Area Network) [4], RPL (Routing protocol for LLNs) [5] and there are standards for their applicability in automation [6]. Many surveys list RPL [7, 8] as the standard for WPAN as they are highly efficient and proactive in forming routes.

ICASISET 2020, May 16-17, Chennai, India
Copyright © 2021 EAI
DOI 10.4108/eai.16-5-2020.2303946

This paper proposes a system which uses plug and play IoT devices that do not require any wired connections or post installation configuration. It is economic as it involves open source software component like Contiki [9] as operating system for IoT devices. Contiki in turn has the entire stack required for mac (medium access control), Ipv6 (Internet protocol version 6), RPL (routing protocol for LLNs), udp (user datagram protocol), Httpd (http daemon) etc.

2 RPL protocol

RPL protocol is the routing protocol that interconnects all the devices in a LLN and formulates a network topology. It provides support for different types of data traffic in a network namely, P2MP (point to multi-point), MP2P (multi-point to point) and P2P (point to point). MP2P is used for collecting data from multiple sensor nodes in a central sink, P2MP is used for spreading control messages from a central server to a sensor / actuator in the network and P2P is for within nodes communication. It is a proactive routing protocol where routes are formed as and when a node joins the network. The central node initiates the network by advertising itself as the root of the destination oriented directed acyclic graph (DODAG) by sending a destination information object (DIO). Nodes that hear the DIO, join the DODAG by selecting the root node as parent and replies by a destination advertisement object (DAO). The child nodes again send DIO's spreading the formation of DODAG further and further.

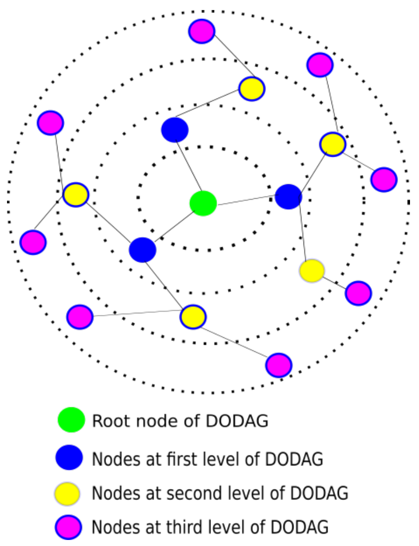

Root node of DODAG
Nodes at first level of DODAG
Nodes at second level of DODAG
Nodes at third level of DODAG

Fig. 2.1. Topology of a RPL network

Figure in [2.1] explains the general structure of a network topology created by RPL. The DODAG is like a tree shaped and oriented towards the destination which by default is the root node which initiates the network. There is an objective function which calculates the rank for each node depending on the rank of its parent and with the link quality to its parent. The link quality is measured as ETX (Estimated Transmission Time) that is directly proportional to the number of times a packet has to be transmitted before it reaches the parent. By each node choosing a parent that is closest to root, this protocol hops the data packets in an efficient manner to reach its destination, the root node. It also has provisions to store the routes in a table to support P2P routes.

3 FSMS System Design

Our proposed FSMS system has multiple layers with a sensing subsystem in the bottom layer, a routing subsystem with RPL in the networking layer, an application layer running udp and httpd server and a Internet connectivity layer running border router. On top, we have clients like browsers that can connect to the httpd server and get the data. All these layers come together to deliver the service to actively monitor temperature of a large building and help us to create fire detection capabilities. Figure [3.1] shows the system level view that clearly chalk out the flow of data from sensors to the client web browsers. All the subsystems other than the client web browser, are run on small battery operated IoT devices. These IoT devices can be active for years without any maintenance / change in battery as RPL that ties them all is an energy efficient protocol. There are three different type of nodes present in the FSMS system and are the border router, the udp/http server and all other nodes are udp client nodes.

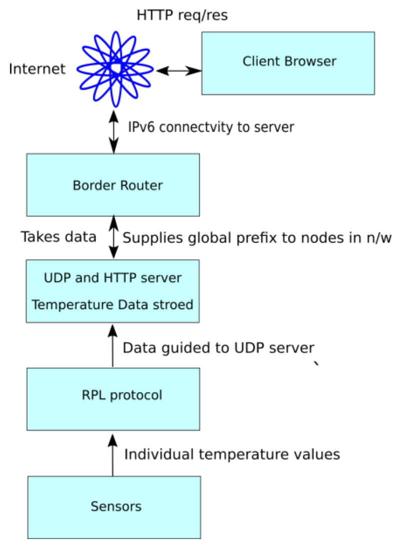

HTTP req/res
Internet
Client Browser
IPv6 connectvity to server
Border Router
Takes data
Supplies global prefix to nodes in n/w
UDP and HTTP server
Temperature Data stroed
Data guided to UDP server
RPL protocol
Individual temperature values
Sensors

Fig. 3.1.Layers of FSMS system

3.1 Sensor Nodes and Server Node

The client nodes are the small motes that have temperature sensors to sense the temperature of various regions in the building. They are designed to sense at an interval of 30 seconds. They can be deployed in an ad hoc manner depending on the specific building's monitoring requirement. Once in an hour, they send an radio packet to the udp server with the sensed data. The server node is also an mote which open an udp port and keeps listening to it for radio packets. The received radio packets provide the temperature data from different parts of the building and the same is stored for all client nodes. The server node also runs a http daemon that listens for tcp connection from web browsers.

3.2 Border Router

The border router is also a mote that creates an RPL DAG and acts as the root of the network. Thereby this node initiates the system and facilitates the connectivity between all client and server nodes. It supplies the global Ipv6 prefix to all the nodes in the network and in turn all the other nodes combines it with the link local address to arrive at its global Ipv6 address. This node also helps in keeping the sanity of the network by checking the version of the network and can restart a newer version in case of failures.

3.3 Remote Data Retrieval

A client browser can simply query the http server using the global Ipv6 address of the server node and the monitoring data is supplied through a html web page. The web page is designed to auto refresh every 30 seconds to get the most recent data from the building. Any personnel can monitor the live temperature values from anywhere in the world. The automation of any action like switching off power can be incorporated in the server node which can send an action command to any node in the network. It can be designed to add more features like sending alerts to fire stations, start an audio alarm, switch the sprinkler system if available etc.

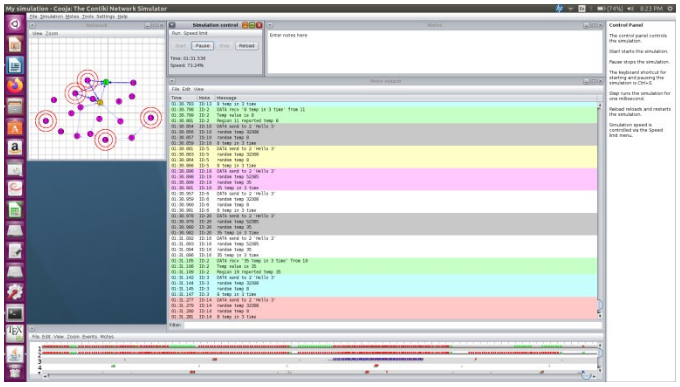
4 Simulation setup

The proposed system is implemented in Telosb mote having MSP430 processor [10] with an application that runs on top Contiki 3.0 [11]. The network is emulated in a simulator named cooja that is supplied with Contiki operating system. The table [Table 1] summarizes the simulation parameters supplied to each node. The client nodes have temperature sensors which sense the temperature/smoke values every 30 seconds. As the nodes are run in simulator, they are designed to report a random value between 0 to 50 degree Celsius as the temperature. If the value is well within the the safe temperature limit (configurable parameter), they send a radio packet to the server only once in 60 minutes. Even if the temperature rises above the safe value once, radio packets are sent as soon as they are sensed (every 30 seconds).

Table 1. Simulation parameters for FSMS

Radio Medium	UDGM
Mote	Telosb
Positioning	Random
Types of Motes	1 border router, 1udp/http server, 18 client nodes
Duration	60 minutes
Sensing Interval	30 seconds
Safe temp limit	25 celsius

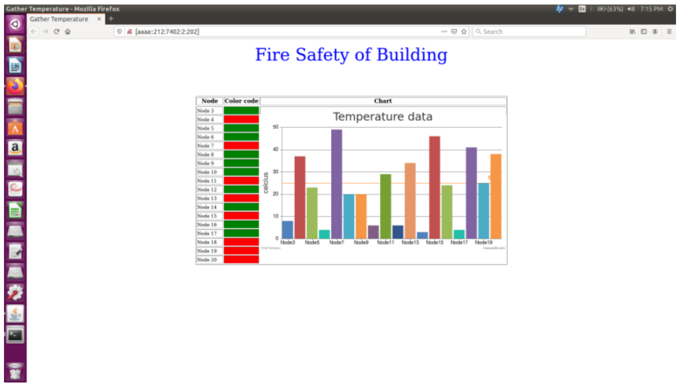
The below picture in Figure[4.1] provides the screen shot of the cooja simulation which shows the different types of nodes with varied colors and the radio traffic between nodes. The border router root node is shown in green, the server node in yellow and client nodes are in pink. The placement of nodes is in random fashion, allowing the real time placement to be ad-hoc. The web page from the client web browser is as shown Figure [4.2]. Once the simulation is started, a global prefix is supplied to the border router via serial interface. The border router then advertises the prefix in the RPL network and all the nodes get and set their global Ipv6 address. The server node can now be accessed from the web browser to get the monitoring data. At the start of the system, there are no temperature values from the sensor and all nodes are shown in green. Green code represents that the temperature value is below 25 degree Celsius (configuration parameter set by the building owner) where red represents higher than 25 degree Celsius. The page refreshes itself in every 30 seconds and keeps the monitoring live.



```
Time: 01:31.536
Speed: 73.24%
```

```
Time       Mote   Message
01:30.782  ID:13  8 temp in 3 time
01:30.798  ID:2   DATA recv '8 temp in 3 time' from 11
01:30.799  ID:2   Temp value is 8
01:30.801  ID:2   Region 11 reported temp 8
01:30.854  ID:10  DATA send to 2 'Hello 3'
01:30.856  ID:10  random temp 32308
01:30.857  ID:10  random temp 8
01:30.859  ID:10  8 temp in 3 time
01:30.861  ID:5   DATA send to 2 'Hello 3'
01:30.863  ID:5   random temp 32308
01:30.864  ID:5   random temp 8
01:30.866  ID:5   8 temp in 3 time
01:30.896  ID:19  DATA send to 2 'Hello 3'
01:30.898  ID:19  random temp 52385
01:30.899  ID:19  random temp 35
01:30.901  ID:19  35 temp in 3 time
01:30.957  ID:6   DATA send to 2 'Hello 3'
01:30.959  ID:6   random temp 32308
01:30.960  ID:6   random temp 8
01:30.961  ID:6   8 temp in 3 time
01:30.978  ID:20  DATA send to 2 'Hello 3'
01:30.979  ID:20  random temp 52385
01:30.980  ID:20  random temp 35
01:30.982  ID:20  35 temp in 3 time
01:31.092  ID:16  DATA send to 2 'Hello 3'
01:31.093  ID:16  random temp 52385
01:31.094  ID:16  random temp 35
01:31.096  ID:16  35 temp in 3 time
01:31.105  ID:2   DATA recv '35 temp in 3 time' from 19
01:31.106  ID:2   Temp value is 35
01:31.109  ID:2   Region 19 reported temp 35
01:31.142  ID:3   DATA send to 2 'Hello 3'
01:31.144  ID:3   random temp 32308
01:31.145  ID:3   random temp 8
01:31.147  ID:3   8 temp in 3 time
01:31.277  ID:14  DATA send to 2 'Hello 3'
01:31.279  ID:14  random temp 32308
01:31.280  ID:14  random temp 8
01:31.281  ID:14  8 temp in 3 time
```

Fig. 4.1.FSMS in Cooja Simulation



Fire Safety of Building

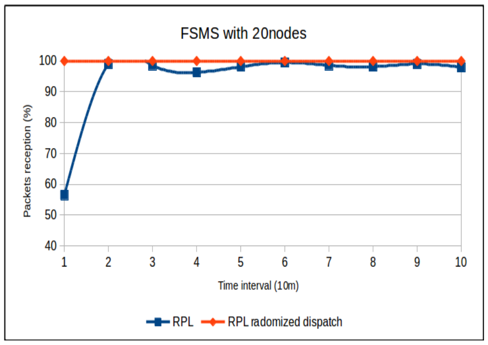
Node	Color code
Node 3	🟩
Node 4	🟥
Node 5	🟩
Node 6	🟩
Node 7	🟥
Node 8	🟩
Node 9	🟩
Node 10	🟩
Node 11	🟥
Node 12	🟩
Node 13	🟥
Node 14	🟩
Node 15	🟥
Node 16	🟩
Node 17	🟩
Node 18	🟥
Node 19	🟥
Node 20	🟥

Fig. 4.2. Live Web Monitoring

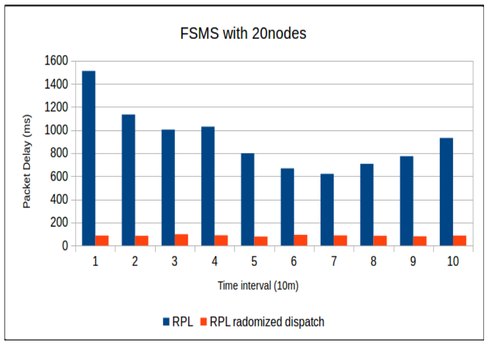
The graph in the web page gives the exact temperature value reported by a node in the building. Additional features like information on coordinates/floors of a node, historical data of a node, triggering an alert to mobile phone etc can be added to the system based on the requirement.

5 Measuring and Tuning Performance

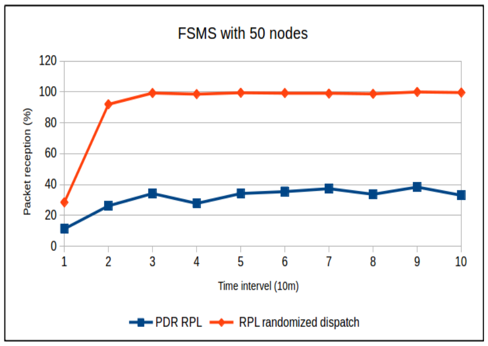
Initially, the system is designed with 20 client nodes that monitor and inform the server of the values. The performance of the FSMS system is studied with two parameters, one being the data reliability and the other being data delay. Data reliability shows the percentage of the sensed data that is successfully received in the server node to that of the packets that are sent. It is important to have highest reliability so that any data is not lost. When in distress, the data that is sensed is dispatched as soon as a new data becomes available. Figure [5.1a] shows the data reliability for this method with a blue line. Data delay or Latency is the measure of the average time taken for a data packet to reach the server from its origin. The simulation records the sent time and the reception time for each data packet. The difference in the time is calculated for all the packets and is averaged out with the total received packets to arrive at the average data delay. Figure [5.1b] shows the data delay metric for both the cases and the metric is measured and plotted over an interval of 10 minutes. As seen in the graph, a 100% reliability is not achieved.



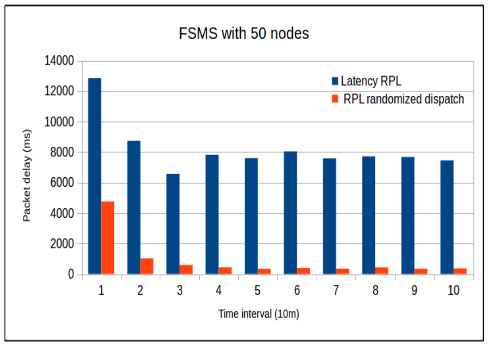
(a) Data Reliability for 20 nodes



(b) Data Delay for 20 nodes



(c) Data Reliability for 50 nodes



(d) Data Delay for 50 nodes

Fig. 5.1. Improved Reliability and reduced delay for randomized dispatch

The causes for not having 100% reliability are packet losses due to interference at the receiver, non available routes to the destination, dynamic nature of the wireless medium causing parent changes etc. After a maximum number of retries is reached, a packet is dropped. To avoid all packets being transmitted at same time, the system is upgraded to choose a random time between 30 seconds and transmit a radio packet. The red line in the graph shows that the reliability is improved to 100% for the system with 20 nodes. The delay is also significantly reduced as the number of retransmissions is reduced with this randomized dispatch. The first ten minutes duration of the graph has significant delay compared to the rest because of the time taken for the formation of routes.

As a next step, the system is upgraded to support larger buildings with 50 client nodes randomly deployed over a larger area. RPL is known to be scalable with support for hundreds of devices. The scaled up data reliability and data delay for both the packet dispatch approaches are shown in [5.1c] and [5.1d]. The red line shows the metric for the randomized packet dispatch approach. The initial dip in reliability is due to the increased number of hops as the network is scaled up. Increased number of hops takes more time for the route to be set up initially. Once the setup period is crossed, RPL shows up high reliability in packet delivery. Further configuration of RPL parameters can also yield good results as shown in [12][13][14].

Armed with these results, it could be said that a randomized data dispatch works well for a time critical application like Fire Safety Monitoring System. The results are also encouraging for a non guaranteed service like udp (user datagrams), could work well in a low power lossy environment.

6 Conclusion

A Fire Safety Monitoring System (FSMS) designed with RPL protocol shows promising capabilities for being a robust real time system. The proposed system is reliable, swift, allows ad-hoc node placement, wireless and consumes low power. FSMS also has the advantage of a fail safe system as RPL is self healing in nature. The local and global repair mechanisms help the node to reposition itself strategically in case of a prolonged parent node failure. As the number of packet retransmissions is reduced by using a randomized packet dispatch feature, the total energy consumption of the network is reduced. This assures longevity for the network and translates to less maintenance work for the system. By providing an Internet connectivity to the border router, the system can be effectively used to monitor a building from anywhere in the world. The proposed system can further be enhanced to include alert services and actuation services based on upon the availability of infrastructure.

References

[1] National Building Code of India, Fire and Life Safety.Bureau of Indian Standards.

[2] Framing of the TamilNadu Combined Development and Building Rules, 2019, Notification Issued. Refer Rule.

[3] Internet Engineering Task Force - a standards body that develops open standards for Internet. https://www.ietf.org

[4] N. Kushalnagar, G. Montenegro and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007.

[5] T. Winter, Ed., P. Thubert, Ed., A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012.

[6] A. Brandt, E. Baccelli, R. Cragie, and P. van der Stok, "Applicability Statement: The Use of the Routing Protocol for Low-Power and Lossy Networks (RPL) Protocol Suite in Home Automation and Building Control", RFC 7733, DOI 10.17487/RFC7733, February 2016.

[7] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, Operating systems for low-end devices in the Internet of Things: A survey, IEEE Internet Things Journal, Vol. 3, No. 5, pp. 720-734, October, 2016.

[8] Yousaf Bin Zikria, Muhammad Khalil Afzal, Farruh Ishmanov, Sung Won Kim and Heejung Yu, A survey on routing protocols supported by the Contiki Internet of things operating system, Future Generation Computer Systems, Vol. 82, 2018, pp. 200-219.

[9] The official git repository for Contiki, the open source OS for the Internet of Things https://github.com/contiki-os/contiki

[10] MSP430F1611: 16-bit Ultra-Low-Power MCU, 48kB Flash, 10240B RAM, 12-Bit ADC, Dual DAC, 2 USART, I2C, HW Mult, DMA.

[11] N. Tsiftes, J. Eriksson and A. Dunkels, Low-power wireless Ipv6 routing with ContikiRPL, Proceedings of the 9th ACM / IEEE International Conference on Information Processing in Sensor Networks (IPSN), Stockholm, Sweden, 2010, pp.406-407.

[12] X. Liu, Z. Sheng, C. Yin, F. Ali and D. Roggen, "Performance Analysis of Routing Protocol for Low Power and Lossy Networks (RPL) in Large Scale Networks," in IEEE Internet of Things Journal, vol. 4, no. 6, pp. 2172-2185, Dec. 2017.

[13] M. Anathi, K. Vijayakumar , "An intelligent approach for dynamic network traffic restriction using MAC address verification", Computer Communications,Elesvier,5 February 2020.

[14]K. Pradeep Mohan Kumar, M. Saravanan, M. Thenmozhi ,K. Vijayakumar, " Intrusion detection system based on GA-fuzzy classifier for detecting malicious attacks", wiley, Feb 2019.