



Decentralized weather monitoring system for smart cities

K. Sripath Roy¹, S. Sowmya^{2*}, M. Manasa², D. Alekhya², P. Abhinav²

¹ Asst Professor Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India-522502

² Student, Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India-522502

*Corresponding author E-mail: mails2sowmyas@gmail.com

Abstract

Internet of things involves monitoring as one of its major applications. It mostly includes condition-based decisions without human intervention. Monitoring systems continuously read the data from sensors, upload the data to a database and take appropriate decisions when they encounter an abnormal data. This provides a good basis for the IoT control systems which now are mostly common. This paper explains the solution to replace the single server system which means a centralized system with an individual server system which is a decentralized system. A decentralized system provides high security to the data and ensures data privacy which centralized systems fail to do. This decentralized system involves maintaining own server but ensuring the availability of data & communication without loss of privacy, ensuring minimization of security threats. This application involves building of a collective application on an android platform to present the data to the users based on the user hierarchy defined. The app provides high security through authentication based on the rights that user possesses. This app is designed to continuously monitor the air pollution in the areas across the city and give alerts or notifications whenever the pollution level reaches to a threshold. Additionally, the app monitors temperature and humidity level. Analytics are performed on the data from sensors and a graph is plotted as the value of a parameter across time. Separate graphs are plotted for each parameter. A user with administrator privilege can view the data of all the areas whereas a user from a single location or a user with general privilege gets the data of his own location.

Keywords: Internet of Things, Decentralized system, Android Application, Security, Privacy.

1. Introduction

Internet of Things is a highly developing area where in the physical layer, sensors keep collecting the data, in the network layer, this data is transmitted to a data center and decisions are taken. In the process layer, these decisions are implemented without human intervention. Monitoring systems which are a major part of IoT applications, involve collection and storage of large amounts of data. It is highly important to preserve the privacy of the data and providing security to the data but still ensuring availability and ways to update and retrieve the data. Decentralized systems replace the storage of entire data in a single central server. When entire data is stored in a single server, it creates multiple problems such as storage and retrieval as IoT applications need collection of large amount of data. It also brings in the issues of data availability as server failures make the data unavailable until they are fixed. Security threats and privacy breaches are the most important problems in centralized systems. A decentralized system is easy to install and maintain as each area has its own server. These systems reduce the overhead of data storage as the data to be stored belongs to only one area. When the data storage overhead decreases, system gives good performance for retrieval and search operations.

Android applications are being extensively used after good availability of smart mobiles in the industry. App development is one of the best IoT solutions to many real world problems.

Android applications are easy to install and maintain and they also provide ease to understand and use. Any changes in the application after installation such as addition of sensors or modules can be easily added in the form of updates.

2. Modules

2.1 DHT11:

DHT11 is a sensor used to measure temperature and humidity in any environment. It is the best sensor as per cost effectiveness as it costs very less and is also easy to install in any equipment. It is highly reliable, and long-lasting. The range of this sensor is upto 20 meters. It is a digital sensor used for acquisition of data digitally for specific applications and recognizes the temperature in terms of degree Celsius. It requires a 3.3V power supply and a ground connection.

2.2 MQ2:

This sensor is used to detect gas leakages in home and industry. It is also useful to detect other gases like H₂, LPG, CH₄, CO, PROPANE and ALCOHOL. It is an analog sensor and it requires 5V power supply and a ground connection. This sensor senses the values accurately, and sends those values with faster speeds. This sensor is highly stable and has long life time.

2.3 NODE MCU:

This is used to transmit the data from sensors to Server. Node MCU has one analog pin and 10 digital pins. So it supports number of sensors at any given point of time. Sensors can be added dynamically even after installation of the entire equipment. Embedded C language code is deployed on the node MCU module and it is done only once. After dumping the code once on node MCU, only power supply is required for it to function.

2.4 MQTT:

Mqtt (Message Queuing Telemetry Transport) is a well known publish-subscribe based message transfer protocol used to establish connection between nodes and message transfer. It gives good QoS as it assures us with at most once delivery that is once fire and then forget. Mosquitto is one such application using Mqtt developed by eclipse organization which supports C and Python languages. We use mosquitto as our server to receive the data from nodes with sensors. The node which is connected to the wifi network is coded with the ip address of server running mosquitto. Now the mosquitto server receives the data in the terminal and processes it to send into a database.

2.5 MYSQL DATABASE:

MYSQL is a relational database management system supporting all the requirements. It stores the data in the form of tables, and mainly used for web based applications. It is simple, secure and easy to learn. To use this database is the cost effective idea as memory management is done very effectively. MYSQL supports JDBC, ODBC development interfaces, and runs on various operating systems like windows, linux etc.

2.6 ANDROID APP:

Android app is most widely used open source platform now-a-days. Being open source, it doesn't charge any money for license purpose, so development cost is low. It provides features which are easy to use, develop and modify. This android application is designed to intake login credentials and also provides the sign up option. It asks for the location of the user while signing up and stores the credentials in an encrypted format along with data in MYSQL database. This app continuously reads the data from the database in the server and plots the graph with specified parameters. When the value exceeds the threshold value set by the user alert messages are sent.

3. Implementation

Temperature, humidity and Smoke levels are the three chosen parameters to define the atmosphere in any given area. The sensors used to measure these parameters are DHT 11 which monitors humidity and temperature and MQ2 which monitors the level of smoke in the area. These sensors are less costly with wide coverage area. DHT 11 is a digital sensor and MQ2 is an analog sensor. These sensors are connected to ESP8266, a wifi module. ESP 8266 is capable of reading both digital and analog data. It connects to the wifi network specified and sends the data to the server. The protocol used to send the data over the wifi network is MQTT. Node MCU acts as an MQTT client and it sends the data to the MQTT server. As DHT 11 is a digital sensor, libraries need to be imported to deal with the digital data.

The data collected through the sensors is sent over the MQTT broker is redirected to a MySQL database in the server. It is a continuous process of sending the data, so the database must support concurrent access of read and write without synchronization problems. MySQL is one such database management system ensuring concurrent access to data. The database also consists of another table with four columns maintaining the data of username, password, type of the user and location of the user. This data is used by the app for authentication purpose.

A java server page (JSP) in the server reads the data from the database upon request. It displays the data of last ten (user specified. Default is ten) values in the database. It calls the graph displaying module with this data as parameters. A graph is generated in the same JSP. This process occurs with the call for JSP with the credentials of location, parameter to be displayed or monitored.

The application when initialized, intakes the login credentials of the user. It searches for the match with the data in the database. It then fetches the data of location and type of the user on successful login. If the user is of administrator type, it displays the locations the user has access to, along with the parameters to be monitored. Upon selection of the location and parameter, a request is sent to the server to call a JSP. The JSP fetches the data of location and parameter from the request URL. Based upon the location and parameter, it fetches the recent data from database and generates a graph, which is sent as a response to the app. If the user is of general type, he is shown the parameters that he can monitor and the data or the graph he gets in response belongs to his own registered location. A user with general privileges doesn't have access to the data of locations other than his registered location.

A threshold value is set for each parameter, which is the highest acceptable value, upon exceeding which, measures have to be taken to control. A java program continuously monitors the data in the database and whenever it encounters a value more than threshold, it sends a request to the firebase (A module in android application development) with the arguments as name of the parameter and data of the parameter. The firebase module sends notification to the user that a particular parameter has exceeded threshold. User can also set the gap between each notification, which reduces the number of notifications to be sent, as the value once exceeds the threshold, it keeps exceeding until some measure is taken to control it. If this time isn't set, the app continuously sends the notifications, disturbing the user.

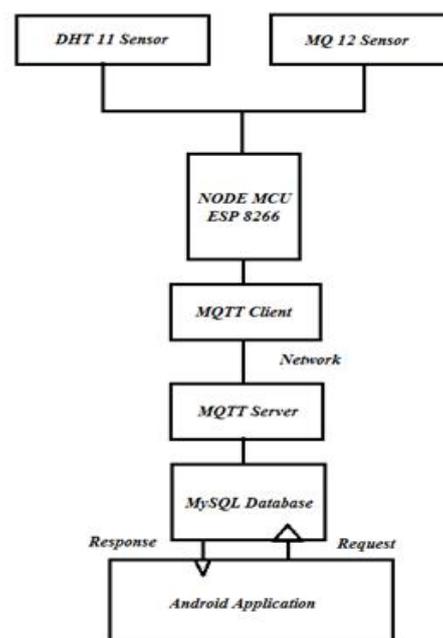


Fig 1. System Architecture

4. Results

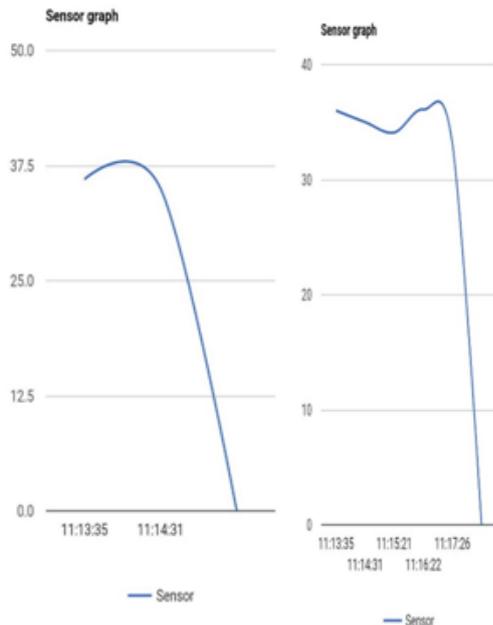


Fig 2. Graphs showing data of Temperature and Smoke

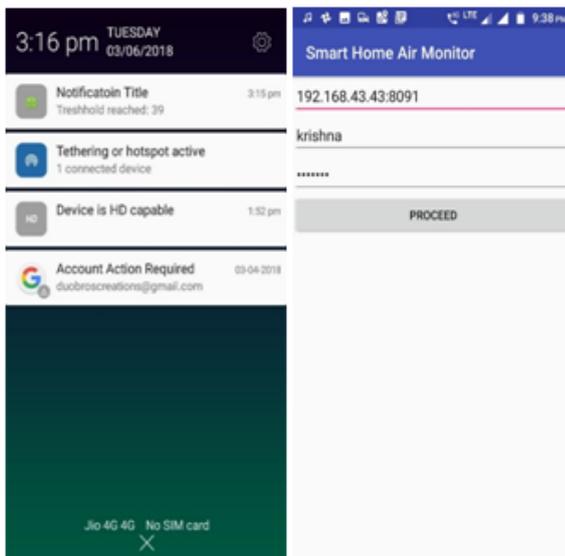


Fig 3. Application Interface

5. Conclusion

The application that we built for Decentralized weather monitoring system is used for monitoring temperature, humidity and smoke levels in atmosphere. The data collected from the sensors is sent over the MQTT broker is redirected to a MySQL database in the server. App uses that data, and displays graph for recent values, and notifications are received to the app when particular parameter has exceeded its threshold value. Decentralized weather monitoring system is designed mainly to focus on continuous monitoring, processing and storage of data. This Decentralized system reduces the overhead of data storage, which enhances the performance for different database operations. Data security is provided and privacy is preserved by proper authentication of the user.

6. References

- [1] "A low cost self-deployable localised weather update and monitoring system in remote areas using open hardware", International Journal of Pure and Applied Mathematics, Volume 115, Issue 8 Special Issue, 2017, Pages 511-516.
- [2] O. A. Postolache, J. M. D. Pereira, P. M. B. S. Girao, "Smart sensors network for air quality monitoring applications", IEEE Trans. Instrum. Meas., vol. 58, no. 9, pp. 3253-3262, Sep. 2009.
- [3] S.-C. Hu, Y.-C. Wang, C.-Y. Huang, Y.-C. Tseng, "Measuring air quality in city areas by vehicular wireless sensor networks", J. Syst. Softw., vol. 84, no. 11, pp. 2005-2012, 2011.
- [4] "Monitoring ambient air quality for health impact assessment", 1999.
- [5] Global Environmental Change, 2005.
- [6] D. D. Lee, D. S. Lee, "Environmental gas sensors", IEEE Sensors J., vol. 1, no. 3, pp. 214-215, Oct. 2001.
- [7] A guide to air quality and your health Air Quality Index A Cross-Agency U.S. Government WebSite., 24th Mar. 2005.
- [8] "Air Pollution Monitoring", U.S. Environmental Protection Agency.
- [9] "Smart City-Strategie Berlin", Senatsverwaltung für Stadtentwicklung und Umwelt - Berlin, 2015.
- [10] D. Hasenfratz, O. Saukh, S. Sturzenegger, L. Thiele, "Participatory air pollution monitoring using smartphones", In Proceedings of 2nd International Workshop on Mobile Sensing, 16 April 2012.
- [11] "Executive Environment Agency - Air Quality Monitoring".
- [12] FIGARO, Figaro Gas Sensors 2000 Series, Product Catalogue, 2006.
- [13] FIGARO, TGS 2600 - for the detection of Air Contaminants, Product Information, 2001.
- [14] Caetano, J., P., R. J. Jesus, P. A. Alves, V. F. Dias, EcoMonitor – A Continuous Air Quality Monitoring System, INESC, Lisbon, Portugal, National Instruments, User Solution, 1999.
- [15] Abdullah Kadri, Elias Yaacoub, Mohammed Mushtaha, And Adnan Abu-Dayya "Wireless Sensor Network for Real-Time Air Pollution Monitoring" IEEE Forum On Strategic Technology -2013.
- [16] K. G. Sutar, "Low Cost Wireless Weather Monitoring System", International Journal of Engineering Technology and Management Research, vol. 1, no. 1, pp. 35-39, 2015.
- [17] D Bri, M Fernandez-Diego, M Garcia, F Ramos, J Lloret, "How the weather impacts on the performance of an outdoor WLAN", IEEE Communications Letters, vol. 16, no. 8, pp. 1184-1187, 2012.
- [18] Kang, J. and Park S. "Integrated comfort sensing system on indoor climate" Sensors and Actuators. 2000. 302- 307.
- [19] Moghavvemi M. and Tan. S. "A reliable and economically feasible remote sensing system for temperature and relative humidity measurement". Sensors and Actuators. 2005. 181-185.
- [20] Guo X. & Song Y., "Design of Automatic Weather Station Based on GSM Module", Int. Conf. on Computer, Mechatronics, Control and Electronic Engineering.
- [21] Sparks L. & Sumner G., "Microcomputer Based Weather Station Monitoring System", Journal of Microcomputer Applications, 7, pp. 233-24, 1984