

Research Article

An IoT-Based Network for Smart Urbanization

Sabeeh Ahmad Saeed,¹ Farrukh Zeeshan Khan,² Zeshan Iqbal,² Roobaea Alroobaea ³,
Muneer Ahmad ⁴, Muhammad Talha,⁵ Muhammad Ahsan Raza,⁶ and Ihsan Ali ⁴

¹Department of Computer Science, COMSATS University Islamabad, WAH Campus, Pakistan

²Department of Computer Science, University of Engineering and Technology, Taxila, Pakistan

³Department of Computer Science, College of Computers and Information Technology, Taif University, P. O. Box 11099, Taif 21944, Saudi Arabia

⁴Faculty of Computer Science & Information Technology, Universiti Malaya, 50603 Kuala Lumpur, Malaysia

⁵Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia

⁶Department of Information Technology, Bahauddin Zakariya University Multan, Pakistan

Correspondence should be addressed to Ihsan Ali; ihsanalichd@siswa.um.edu.my

Received 4 February 2021; Revised 4 March 2021; Accepted 18 March 2021; Published 5 April 2021

Academic Editor: Habib Ullah Khan

Copyright © 2021 Sabeeh Ahmed Saeed et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Internet of Things (IoT) is considered one of the world's ruling technologies. Billions of IoT devices connected together through IoT forming smart cities. As the concept grows, it is very challenging to design an infrastructure that is capable of handling large number of devices and process data effectively in a smart city paradigm. This paper proposed a structure for smart cities. It is implemented using a lightweight easy to implement network design and a simpler data format for information exchange that is suitable for developing countries like Pakistan. Using MQTT as network protocol, different sensor nodes were deployed for collecting data from the environment. Environmental factors like temperature, moisture, humidity, and percentage of CO₂ and methane gas were recorded and transferred to sink node for information sharing over the IoT cloud using an MQTT broker that can be accessed any time using Mosquitto client. The experiment results provide the performance analysis of the proposed network at different QoS levels for the MQTT protocol for IoT-based smart cities. JSON structure is used to formulate the communication data structure for the proposed system.

1. Introduction

Internet of Things (IoT) is considered to be another important Information Communication Technology (ICT) wave after the invention of personal computers (PCs), Ethernet, Internet, and the cellular communication [1]. IoT has taken over the world since 2005 and became the very core of the future economic developments in the field of Internet, communication, and networking [2]. Many countries around the world have taken into account IoT as part of national strategy for sustainable development in their governmental and general public sectors by completing the logical and conceptual studies at service level. For example, Japan's broadband access is based on ubiquitous and people-oriented technology, providing services with an objective to help effi-

cient communication between people and people and things and between things as well [3]. The South Korean smart home automation systems help the residence to control many home appliances remotely and also enjoy bidirectional multimedia services [4]. Singapore is also second to none; her next-generation I-hub main objective is to provide secure next-generation "U"-type networks through ubiquitous networking [5]. All these and other such similar projects have laid the foundation of IoT firmly around the world.

IoT domains include healthcare, industry, transportation, education, and emergency response to any sort of natural or man-made disasters under stressed conditions. IoT enables the people to interact (see, hear, and think) with the sensors that help them to share information, make intelligent decisions, and respond to queries efficiently. In other words,

IoT helps to see, hear, and think an environment from an eye of technology (sensors). On the other hand, IoT also changes traditional devices into smart objects by changing the underlying technologies (ubiquitous, embedded systems, sensor networks, communication technologies, internet protocols, applications, and pervasive computing) of these systems. To better understand the concept of IoT, it is important to learn about the elements of IoT. The related examples and categories of each element are listed and discussed in [6–22]. The major elements of an IoT network are identification, sensing, communication, composition, services, and semantics.

The IoT's main objective is to make Internet, communication, and networking more interesting, impressive, and persuasive by providing easy interaction with a wide variety of devices. This paradigm is spread along a vast set of fields and covers almost all human domains of profession [23]. These complex scenarios develop the particular interest of smart urbanization in an IoT paradigm. Thus, the concept of "smart cities" is born [24]. Smart urbanization is capable of optimizing traditional public service processes by increasing the percentage benefit gained from many services like transportation, lightening, maintenance, surveillance of public areas, preservation of history and cultural heritage, parking, automation of industries, education, hospitals, garbage collection, and many more [1, 25]. According to Pike Research [23] conducted on smart cities, it was reported that by 2020, the estimated market share of smart city will be more than hundred billion dollars with an annual expenditure of just 16 billion. Smart city industry emerges from the interconnection between key industries and the service sectors and form several smart city sectors like smart governance, smart utilities, smart buildings, smart environment, and smart mobility [26]. There are a number of reasons that hinder the growth of smart cities to full capacity, including political, financial, and technical barriers. Authors in [27–30] have discussed several solutions related to these problems. The contribution of this research is to design a system for developing smart cities in underdeveloped countries using IoT technology. The system is easy to use and easy to implement as it uses simple technology, lightweight communication mechanism, and cheap technology. The proposed technology is a suitable small business organization with less budget to spend on their technological needs. Moreover, it is also feasible for governments having low budget to spend on technology adaption.

The rest of the paper is structured as follows: Section 2 provides the review of the already developed technologies; Section 3 provides the detail of the proposed system; Section 4 provides the detail on the working of the system; Section 5 provides the deployment of the system, data collection mechanism simulation implementation, and simulation results; and Section 6 concluded the overall research and provide future directions.

2. Literature Review

In this section, different smart cities around the world are discussed briefly. These projects also provide foundations

for the realization of smart cities in underdeveloped countries.

The SyMPHOnY project [31] was a smart city project designed using SIP protocol. A special hardware called the MTCG node was designed as a communication device using the SIP protocol for data transmission. The MTCG node receives data from many sensory devices, like home appliances, temperature sensors, and humidity sensors, using a wireless M-BUS interface and transmits it to a SIP server. The data can be accessed using a SIP client [31–33]. The MTCG node was a set of different packages joined together in a single package device called a core. Eco-U-CITY [34] was the first South Korean project for smart city implementation. The project was completed in 2008 to convert the cities of Hwaseong and Dengtan into smart cities. Eco-U-CITY is a project based on green technology, for better safety and comfort of the public. The major aim of this project was to use green technology to reduce the emission of carbon contents in the atmosphere. The project was implemented using a specially developed system called "An Integrated Service Management Platform" (ISMP). ISMP is a 3-layered model for smart cities presented in [35]. The layers present in an ISMP system are service layer, middle-ware layer, and the infrastructure layer.

Under the supervision of the New York City Mayor's Office, the city launched the New York Digital City Program sponsored by the Mayor's Office itself. The program was based on an IT-driven portal called the NYC.GOV portal, with an aim of combining all city's general public on a single platform, i.e., the portal. All citizens were able to access all the services, functions, and applications through their smart devices, mobile phones, and commercial social media. Barcelona Smart City Program implements a three-layer model for a wide variety of technical capabilities stated in [36]. The first layer constitutes sensors, the second layer of the model was based on a City Operating System (City OS), and the third layer was comprised of customer interface. It was using ICT throughput throughout the implementation and development of smart cities using smart city models presented in [37–39]. The city has started a series of projects supporting the concept of smart city, over a physical network covering more than 500 kilometers of area via fiber optics. The city project is aimed at integrating telecommunication and Internet together using twelve initiatives identified via used smart city models. The project has four stages and has successfully completed its three stages, and the fourth stage is under process [39]. Padova Smart City [26] was an implementation of urban IoT concept in the Padova city with the alliance of public and private bodies of the city. The major parties of the alliance were the municipality of Padova, the Department of Information Engineering University of Padova, and Patavina Technologies. The first provides the financial assistance required to aid the project; the second provides the background for the project to start and also give its feasibility report. Finally, the third party, which is a spin-off of the University of Padova and is specialized in the development of creative IoT solutions, provided design and implementations for the IoT nodes and the software required to control the network.

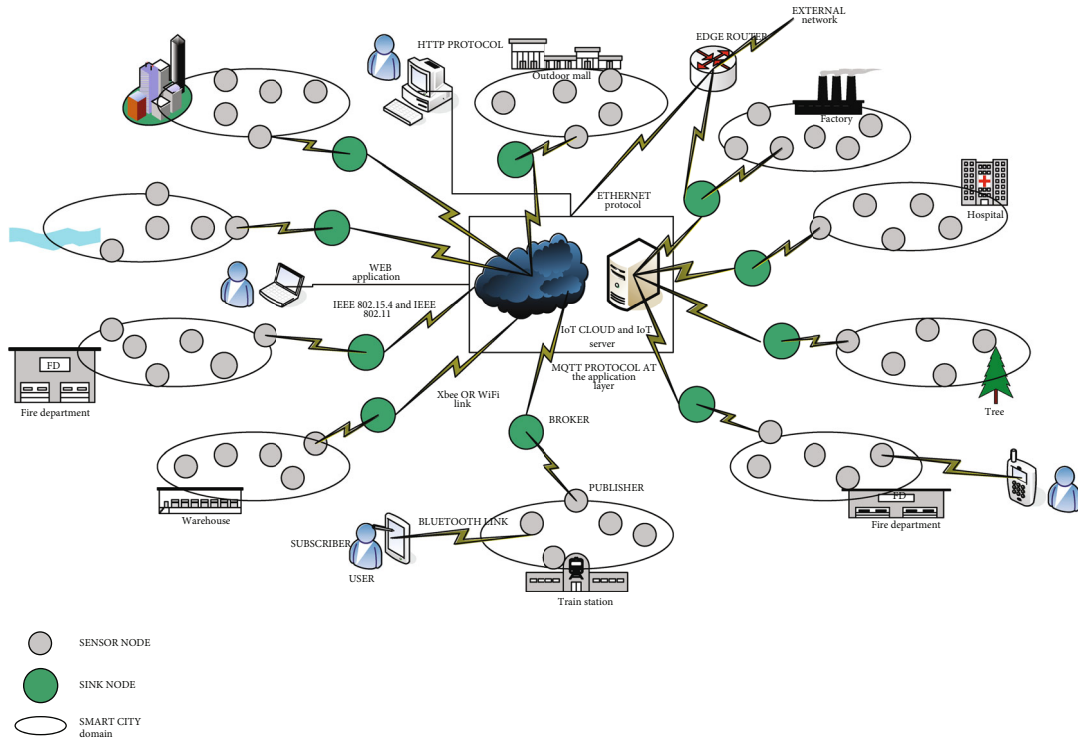


FIGURE 1: Proposed network architecture.

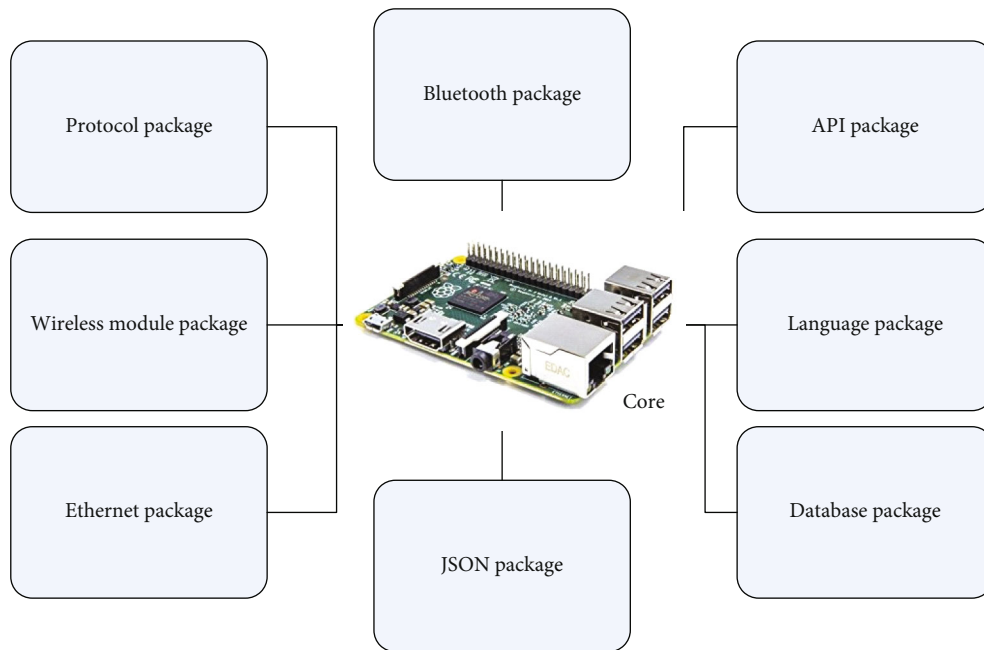


FIGURE 2: Proposed framework of the sink node.

3. Proposed IoT Network Design

Keeping in mind the study done in the literature review and different information provided in different research papers and research projects, the proposed network architecture used to carry out our research is given Figure 1. It is evident

from Figure 1 that the proposed network architecture consists of the following network entities.

The proposed network consists of several devices connected together. The devices are classified into sensor nodes, sink nodes, edge router, IoT cloud, and end user all discussed in the later paragraphs. The network is formed when several

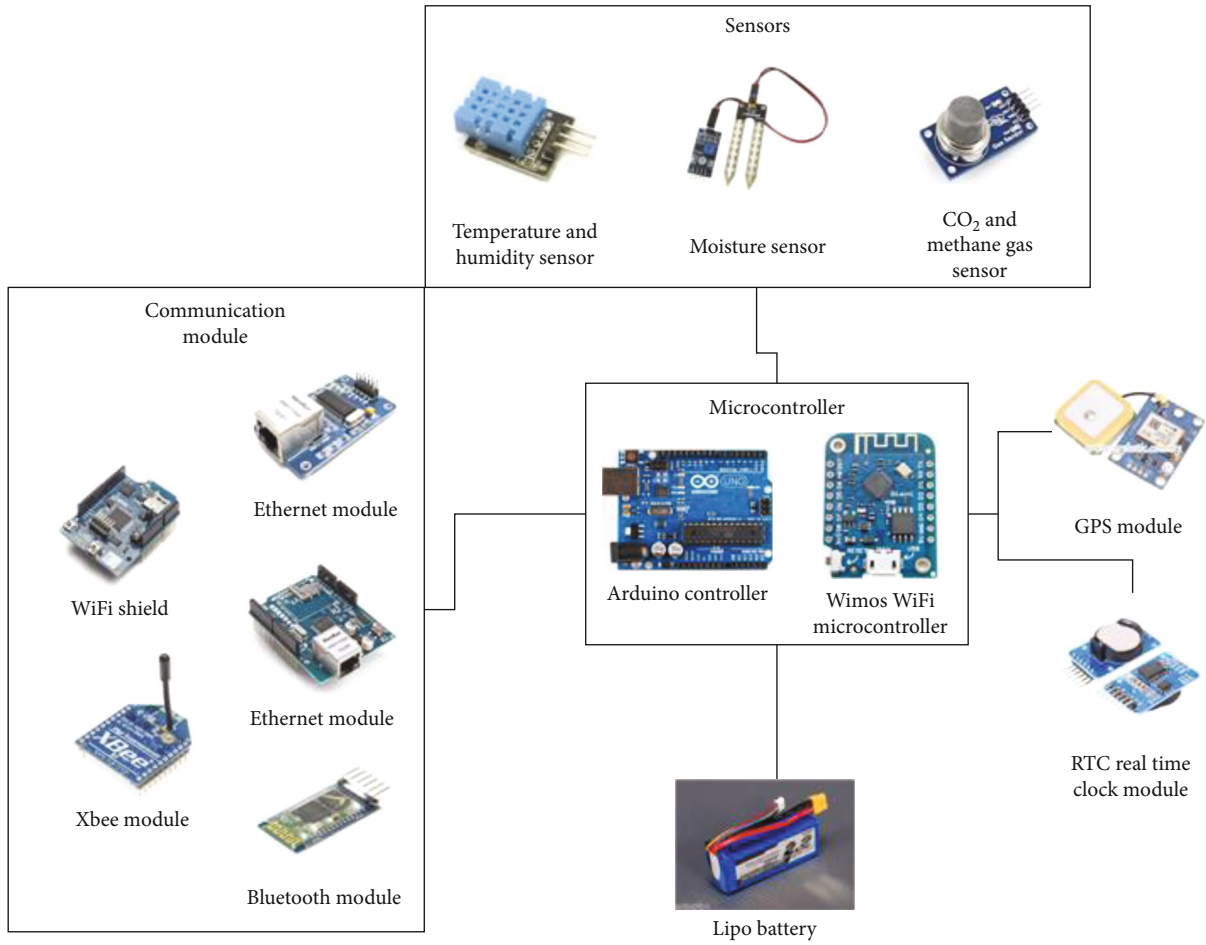


FIGURE 3: Block diagram of sensor node with main components.

```

Algorithm:
Structure Reading {
    "Sensor" :{
        "System":
            "Type": {
                "Name": "Sensor Name",
                "ID": "value"
            },
            "Placement of the Sensor": {
                "Latitude": "value",
                "Longitude": "value",
            },
        },
        "Read_Value": ["Reading Type": "Reading Name", "Value": "value", "units":
"value"],
        "Time": "Time Stamp",
        "Date": "Date Stamp",
        "Status": {
            "Name": "Device Status",
            "Value": {"NULL =0", "OK =1", "ERROR =2", "UNKNOWN =3"},
        }
    };
}
    
```

ALGORITHM 1: Proposed data structure format.

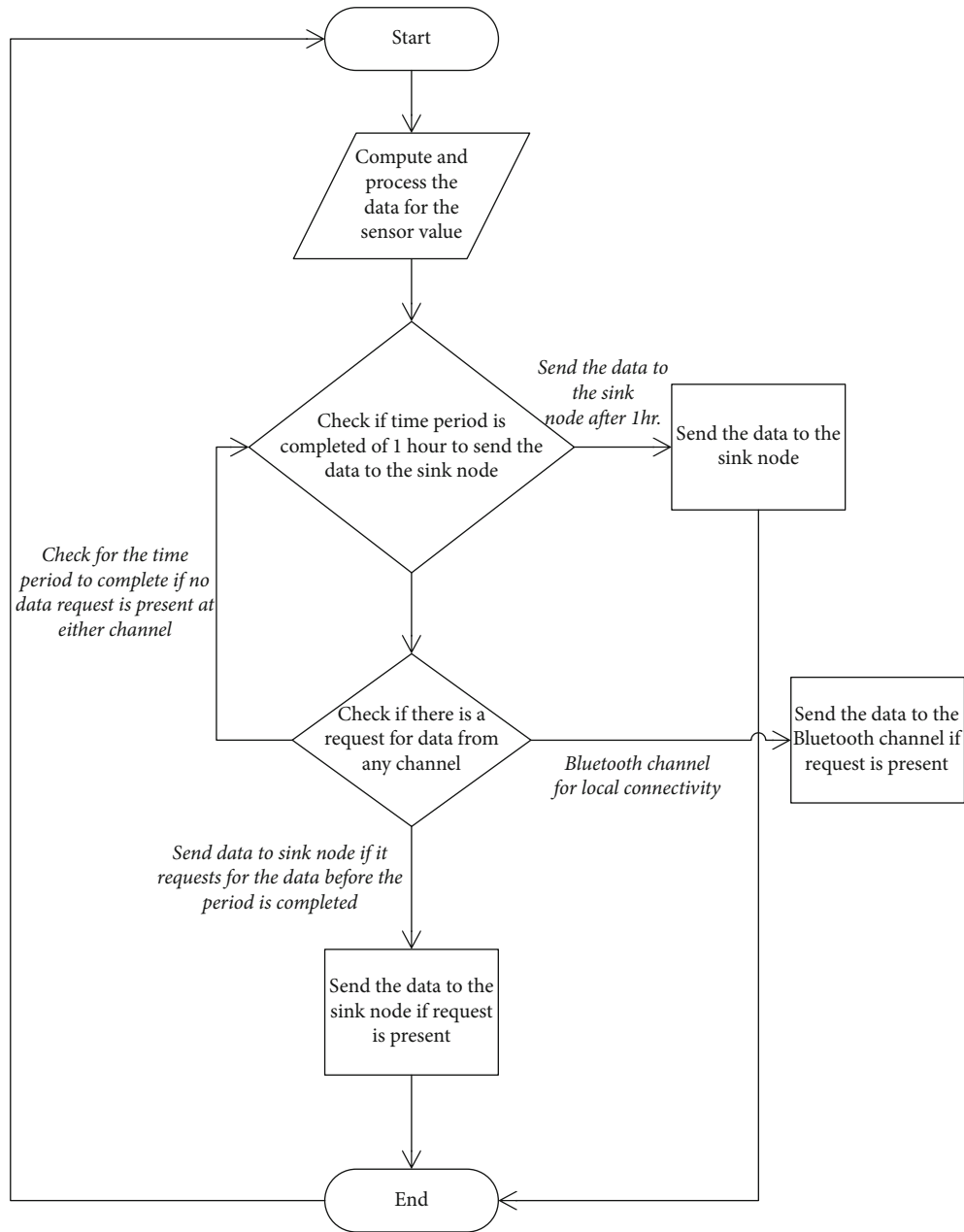


FIGURE 4: Sensor node activity.

sensors deployed connect to a common sink node that acts as a broker to collect and share information. Sensor nodes continuously observe the deployed vicinity for recording data related to several environmental attributes like temperature, humidity, moisture, and gaseous content percentage in the atmosphere; the sensor node then transfers the data to the sink node periodically that keeps data safe for uploading the information to the IoT cloud after a regular interval or whenever a demand for any particular information or reading is received. The IoT cloud provides user an interface to observe and keep track of the changes occurring in any location by keeping a periodic track of the information received from the sink node.

The *sink node* is implemented using a Raspberry Pi board 2 with a Windows 10 IoT core. The core supports the bundles

for using Java Programming Environment. Arduino IDE sketch is also installed on the system to support Arduino. The Mosquitto version 1.4.9 which is an open-source MQTT server was deployed on an Amazon Web Server. Specifications of the Amazon Web Server are as follows: AWS service: EC2, instance type: t2. Micro, OS: Windows server 2016 base1 virtual CPU, storage: 30 GB, and RAM: 1 GB. Node-RED is a visual tool for wiring IoT devices. Node-RED provides web interface, which can send commands to the MQTT server. Node-RED provides interaction between clients and server. Node-RED offers a browser-based flow editor to wire together flows by applying a broad range of nodes in the palette. Flows can be then implemented dynamically in a single click. Due to built-in library in node-RED, useful functions and flows can be saved for reuse. The Bluetooth module is

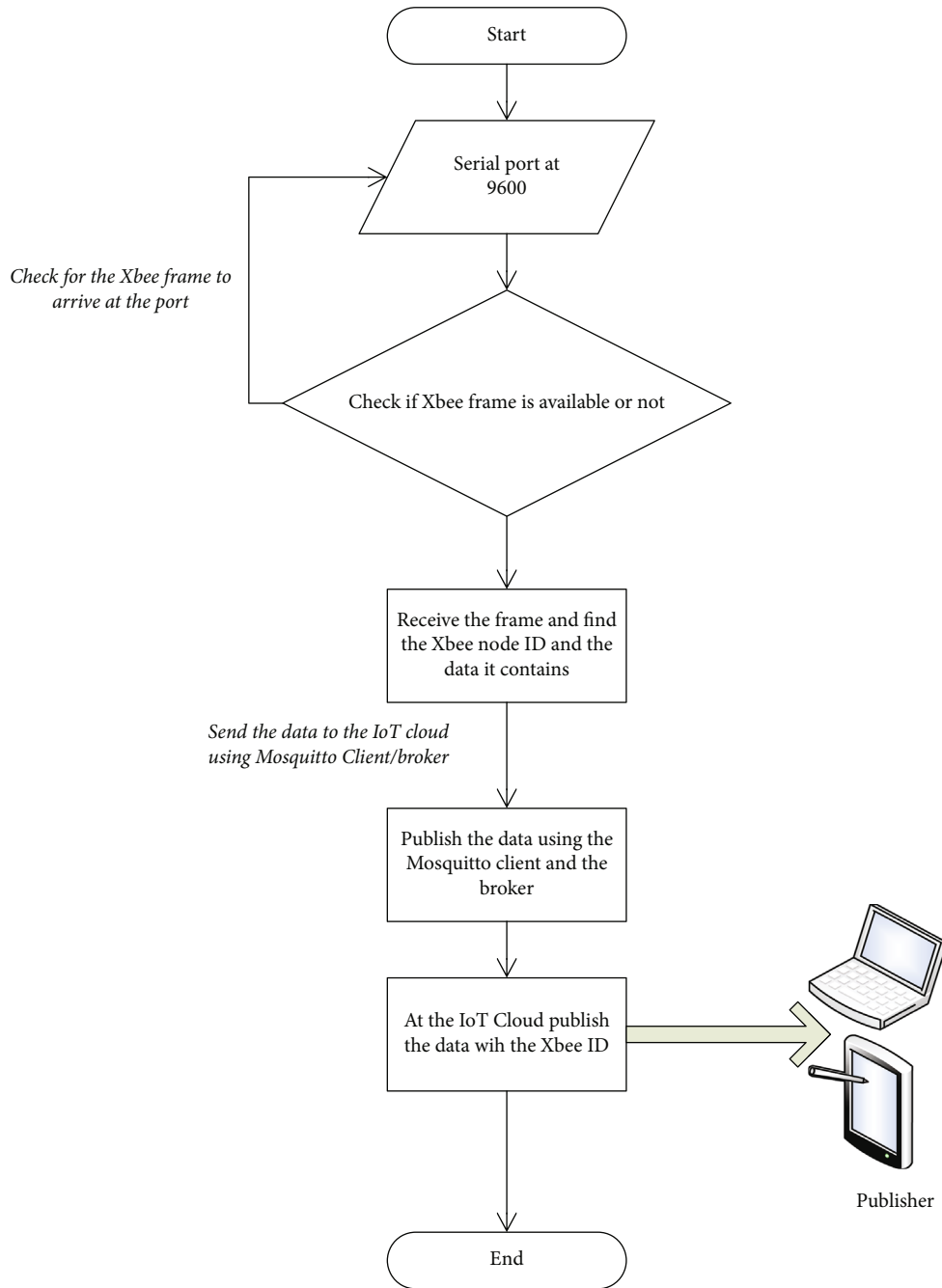


FIGURE 5: Publisher.

also attached to the sink node to connect any local user to receive any information using a mobile application data can be designed to access data from the Bluetooth module. The framework used for the proposed system is shown in Figure 2.

As shown in Figure 2, the framework consists a core package including a powerful board with some operating or the real-time operating system. The core is attached to several bundles depending on the needs of the system. The major bundles included in the framework are as follows: core bundle: the main package that controls the communication and other bundles present in the framework; protocol bundle:

the package containing the definition and implementation of the said protocols for the proposed network; wireless module bundle: the package containing the possible definition of any wireless module used for connecting the sink node to the other devices; Ethernet bundle: the package for installing and using Ethernet in the framework; JSON bundle: the package giving the information regarding the data structure format used for receiving and extracting the required information from the data received from the sensor nodes; database bundle: the package giving information about the used database in the proposed research; language bundle: the package giving information about the programming language used; and

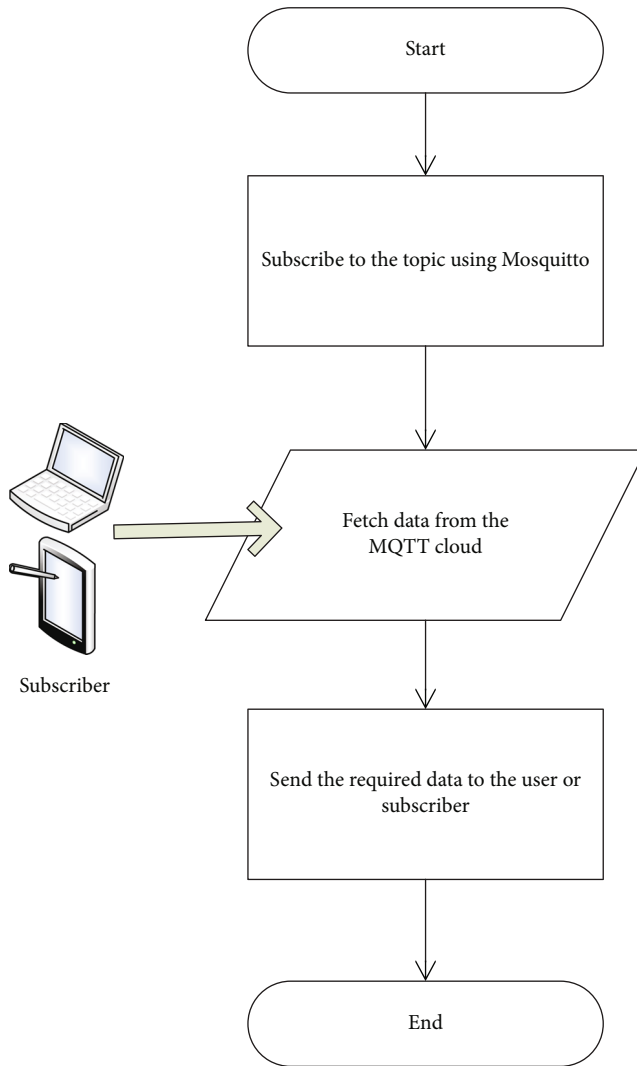


FIGURE 6: Publisher.

API bundle: the package containing the cloud API's and other API's used to provide communication ease and development ease.

The control system of the *sensor node* is implemented using an Arduino microcontroller (any of UNO, MEGA, Mini, and Micro). The sensors used for measuring different environment values are Arduino compatible temperature sensor DHT-11 temperature and humidity sensor; the sensor used to measure the CO₂ and methane level is the Arduino compatible MQ-135 gas sensor. The sensor used for moisture measurement is the Arduino compatible HL-69 soil moisture sensor. The communication module used is the Xbee module using IEEE 802.15.4 Standard and forms the core of the network. The other module used to form the core network is the WiFi shield for Arduino ESP-8266-WRL-13287. The HC-05 Bluetooth module is used to provide local connectivity with the sensor node. Arduino Ethernet shield Wiznet-W5100 or the ENC-28J60 Ethernet module can also be used to provide wired connectivity that is an optional part of the sensor node. Arduino-based RTC DS-3231 real-time clock and the Arduino-based Global Positioning System (GPS) module

NEO-6M is used to provide additional information related to the sensor regarding date, time, and location of the sensor. A 12-volt Lipo battery is attached to the sensor node to provide the power to the complete system. The complete hardware package is installed in a box for safe keeping and to preserve it from severe atmospheric effects. This system can also be achieved using a We MOs D1 Mini ESP8266 a wireless 802.11 (Wi-Fi) microcontroller development board. Its key features are as follows: micro-USB, compatible with Arduino, microprocessor: ESP-8266EXN_r, pin:/input/output 11 pin, one input pin, operating voltage is 3.3 V, frequency: 80 MHz/160 MHz, and of 4 Mb flash memory. The overall system is illustrated in Figure 3.

Edge router is a network layer device used to link the proposed network to the underlying external network. The edge router if deployed inside the sink node is called as inner edge router. Or when deployed outside the sink node is called as external edge router. The *IoT cloud* or the *IoT server* is responsible for joining all the sensor/sink networks at different environments at different areas. These entities provide the central point and are the core of the network architecture. A *user* is a person or a machine or an application that requires the data generated from the sensor nodes for some information or just for record keeping or for making any useful decision using the information generated through the proposed system.

Data structure format used for transmitting the data between the network devices is based on the information provided by the JSON structure as presented in [40–44]. The structure format is consisted of a structure having various variable strings, values, and arrays to represent several values received from the sensor node. The major values that are received from the sensor node include the information regarding the sensor: type of sensor, placement of the sensor, and the sensor ID. The second information that is received from the sensor node includes the reading that is generated at the sensor: numeric value and the unit. The third value received from the sensor node is the time and date stamp, and finally, the fourth reading received from the sensor node is the status of the node. The data structure for each of the values received from the sensor node is given below. The structure format for declaring strings, numeric variables, arrays, structures, and objects is the same as described by the JSON structure. The compiled form of the structure is called as the reading structure that contains four subparts. Each part has its own value depending on the data received from the sensor node. The structure is shown in Algorithm 1.

4. Working of the Proposed System

The working algorithm of the structure is divided into three stages shown in Figure 4, Figure 5, and Figure 6, respectively. The sensor waits for the change in the value that it is reading; upon successful reading, a signal is generated and a value is produced. This value is transmitted (published) to the broker using Xbee communication module. At the broker, the packet is received and the value is checked; if found correct, it is written on the file for record keeping. At the broker, if any user requests to subscribe to the data, the broker writes

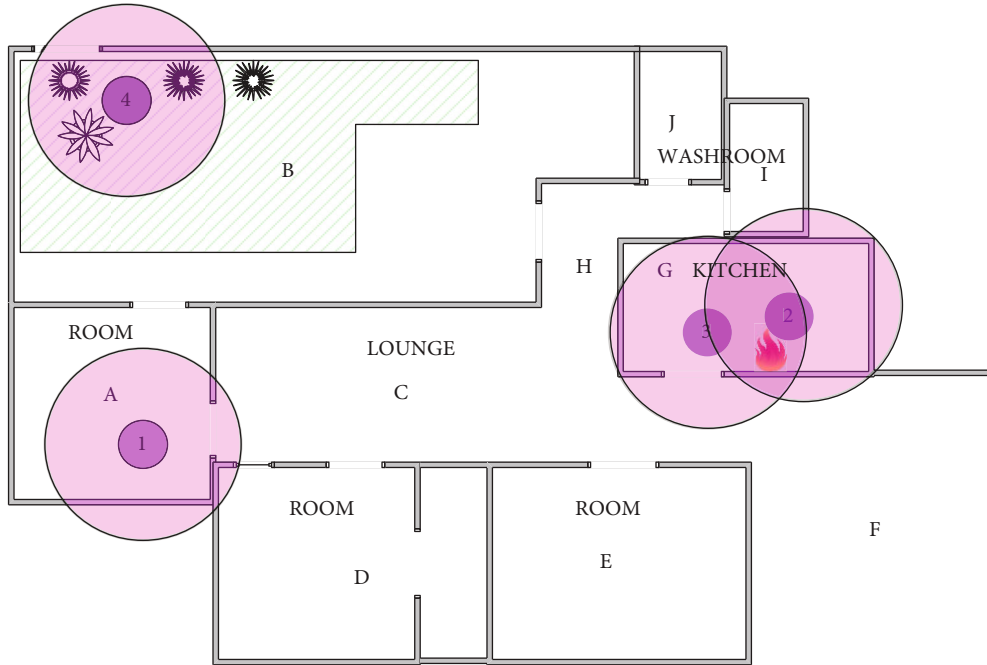


FIGURE 7: Area of implementation.

TABLE 1: Simulation parameters.

Protocol/layer	Parameter/option
	QoS-1, QoS-2, QoS-3
MQTT	Payload size
	Publisher's sending rate
	Number of MQTT clients
TCP	Timestamp option
	Maximum segment size (MSS)
MAC layer	IEEE 802.11 MAC
Physical layer	Bandwidth
	Propagation delay
	Error rate
	Error burst

the data to the user who is subscribing to the data. The communication is carried out in three stages that are as follows.

4.1. Stage 1. At the sensor node, the value of the reading under consideration is computed. The sensor node then waits for the time period to expire after 1 hour to send the recorded value to the sink node. If at any time, the sensor node receives a request from the data from the local Bluetooth channel or from the Xbee channel, then the sensor transmits the data to the channel for the request. The Bluetooth channel has priority higher than that of Xbee channel, and if the Xbee channel has a data request, then the time period is restarted. The overall process is shown in Figure 4.

4.2. Stage 2. When data reaches at the sink node, the sink node also waits for an hour before publishing the data at the cloud using the MQTT Mosquitto broker. If there is

already a subscription request present at the broker for the data, the sink node then immediately publishes the data at the cloud with the Xbee ID from which data is received at the sink. The process is illustrated in Figure 5.

4.3. Stage 3. Any published data can be subscribed from the IoT cloud using the MQTT Mosquitto client. The process is shown in Figure 6.

The placement of the modules in the experimental area is shown in Figure 7. To publish data at the broker, the sensor nodes are to take several readings from the surrounding and convert them into signals to form a measurable reading. This measurable reading is then sent to the broker using the Xbee communication module. At the broker, the program waits until an Xbee packet is received or not; when a packet is received, the program extracts the required value from the packet; and using file handling technique, the value along with some additional information is recorded into a file with an extension (.CSV or.TXT). These files are then copied to the MS-Excel sheet, and the graphs are plotted for different sensor values.

4.4. Network Simulation. For the performance evaluation of the proposed network, the proposed network is simulated in OMNET++ and results are analyzed using Wireshark that supports TCP for wireless network models and also supports MQTT protocol. The QoS-0 and QoS-1 described for the MQTT are used to evaluate the end-to-end delay and message delay in the network. To evaluate the performance of the MQTT server, 500 clients were dynamically created. All clients competed for the connection to the server. Once a client gets connected to the server, it sends request to the server and when it receives response from the server, it terminates its connection from the server and hence, a new client

TABLE 2: Simulation parameters values.

Protocol/layer	Parameter/option
MQTT	QoS-1, QoS-2, QoS-3
	Payload size = 100 bytes
	Publisher's sending rate = 250 Kbits/s
	Number of MQTT clients = 1000
TCP	Timestamp option = yes
MAC layer	Maximum Segment Size (MSS) = 536
Physical layer	IEEE 802.11 MAC
	Bandwidth = 2.4 GHz
	Propagation delay = $T_{air}(100) = (25 + 100) * 32 \mu s = 4.000 \text{ ms}$
	Error rate = 1%
	Error burst = 0.25

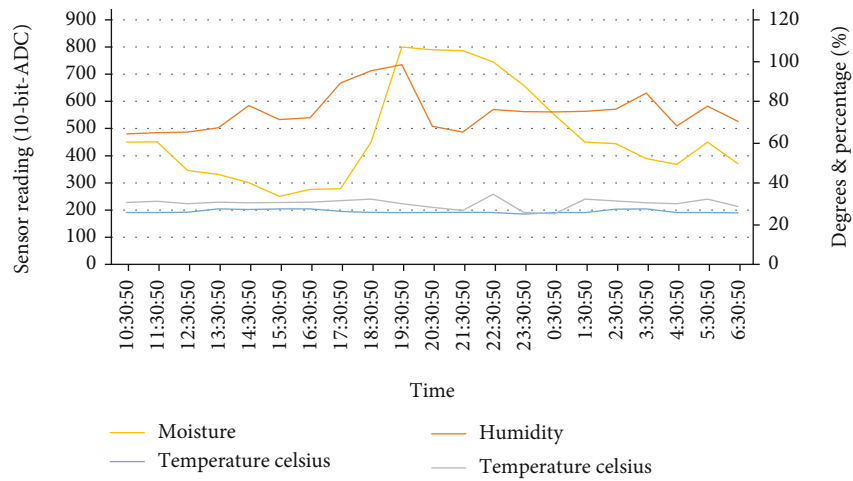


FIGURE 8: Data collected from sensor nodes.

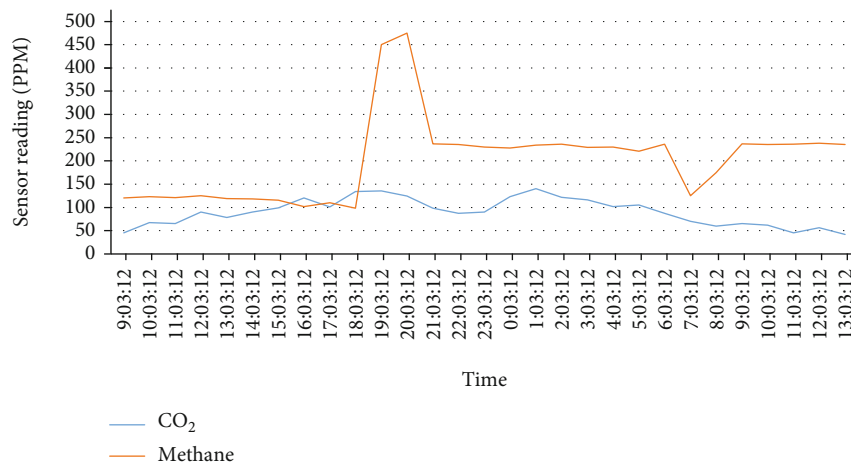


FIGURE 9: CO₂ and methane gas reading in ppm taken from sensor node 3.

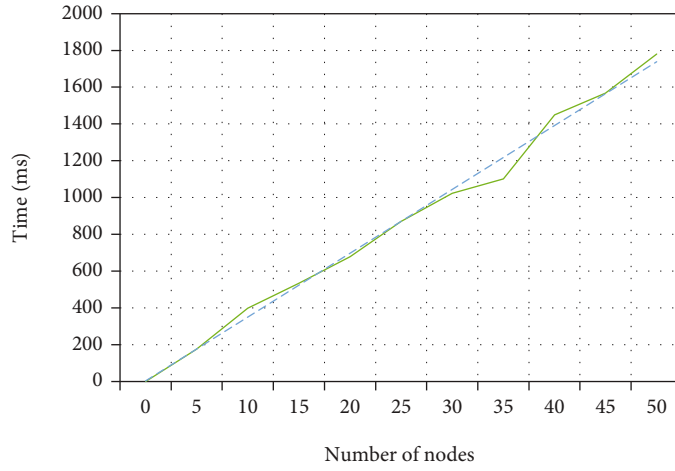


FIGURE 10: Mean connection time delay of network nodes.

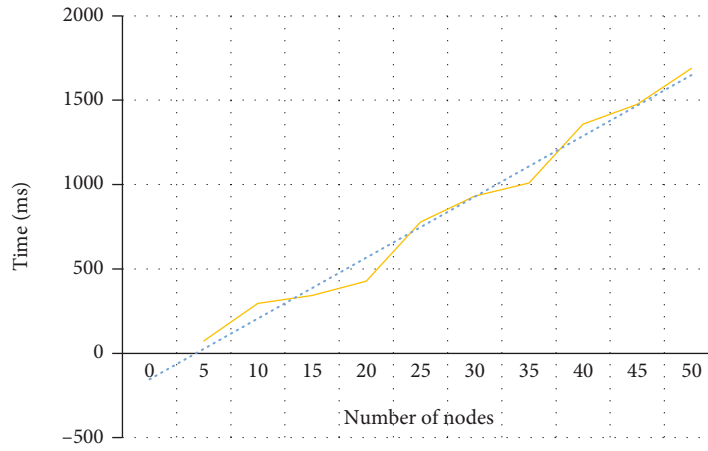


FIGURE 11: Mean response time delay of server.

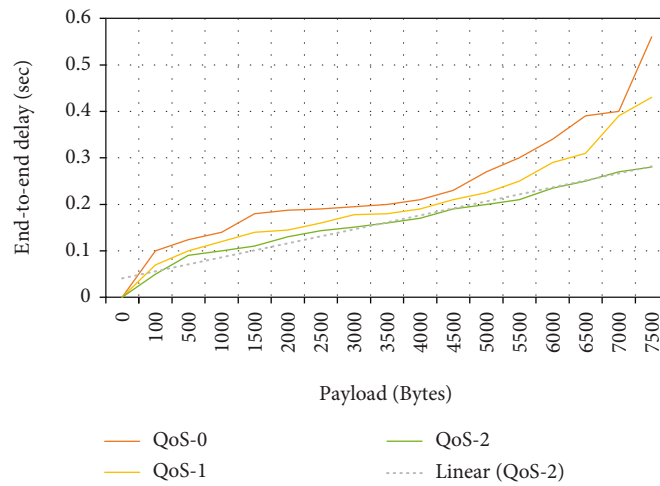


FIGURE 12: Mean end-to-end delay in the network under QoS.

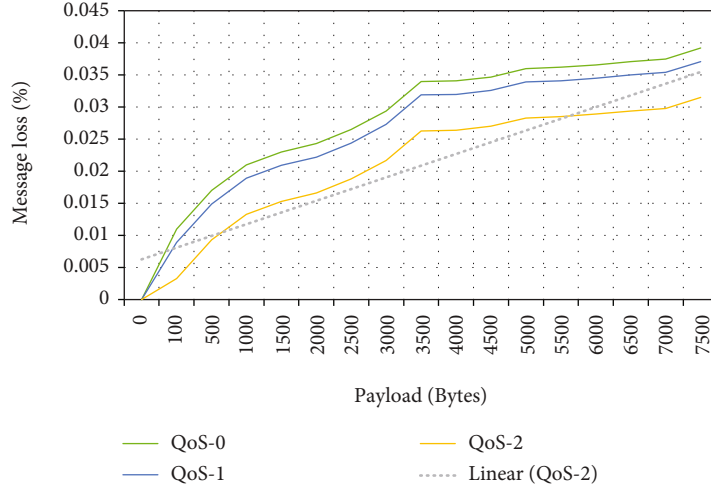


FIGURE 13: Mean message loss in the network under QoS.

competes for the connection to the server. Here, we observed delay as the difference of time when client sends request and when client receives response from the MQTT server as shown in Equation (1). We also considered delay, experienced in connection time of clients with the server. To get the promising results, experiments were performed in two different scenarios by varying the number of devices and the number of clients generated on these devices and by making changes in their connection times. Simulation parameters used to run the simulation and to calculate results are shown in Table 1.

$$D_{ete} = RT_r - ST_s, \quad (1)$$

where D_{ete} is end-to-end delay of the n^{th} client, RT_r time the response was received, and ST_s time the request was sent.

The experiment was performed for 1,000 clients on each device, and an average delay was calculated as described in

$$\overline{AD} = \sum_{n=1}^N D_{ete}, \quad (2)$$

where \overline{AD} is average delay experienced by 1,000 clients, D_{ete} end-to-end delay of the n^{th} client, and $N = 1,000$ clients.

The simulation is implemented for 1000 nodes, and the payload size is kept variable; the minimum payload size is 100 bytes while the maximum payload size used is 250 bytes. The nodes kept connecting and terminating the server after publishing the data on the server. For the ease of working and to make simulation more realistic, 25 nodes are connected to the sink node simultaneously and are allowed to publish the sensor data on the sink node which is then published to the MQTT server. All nodes are then disconnected from the sink node after data is published. The simulation parameter values are listed in Table 2.

5. Results and Discussion

To evaluate the system, the following use cases were used: sensor node with temperature and humidity sensor: the sensor node with temperature and humidity sensor module DHT-11 is deployed using Arduino and Xbee communication module, and several readings for the temperature and humidity are taken, and then, the readings are plotted in a graph; sensor node with CO₂ and methane gas sensor: the sensor node with CO₂ gas and methane gas detector, the MQ-135 gas sensor is deployed using Arduino and Xbee communication module, and several readings for the presence of CO₂ gas and methane gas are taken, and then, a graph was plotted; and sensor node with moisture sensor: the sensor node with soil moisture sensor HL-69 is deployed using Arduino and Xbee communication module to take several moisture readings, and a graph was plotted.

Readings were recorded from different sensor modules placed at different locations in the experimental area. The data is recorded after an interval of 1 hour, and the graphs were plotted. In Figure 8, the blue line shows the temperature reading taken from the sensor node 1 placed in room A of the experimental area. The temperature is taken and is measured in the Celsius scale. The maximum value recorded was 27.2 degrees while average temperature recorded was about 26.72 degrees. The silver line shows the temperature reading taken from sensor node 2 placed in room G near a gas stove. The maximum value recorded was 32.6 degrees while average temperature recorded was about 29.21 degrees. The orange line shows the humidity value recorded from sensor node 1 placed in room A. As the sensor supports for both temperature and humidity, the values can be retrieved from the sensor by sending the request for the specific value. In our case, the sensor retrieves the value of the temperature, if it receives a request containing "t" and sends the value for the humidity if the receiving request contains "h." The maximum value was 98% average humidity value recorded was 74.9%. The yellow line shows the moisture reading taken from the sensor node 4 placed in open area B. The moisture is taken and is

measured in the 10-bit ADC. The maximum value recorded was 800 while average moisture value recorded was about 479.63.

Figure 9 presents the readings related to methane and CO₂ gas recorded from the sensor node 3 placed in room G near the gas stove with gas supply on. The values of the methane gas and CO₂ gas were recorded in the parts per million (ppm) unit. The maximum values recorded for methane and CO₂ were 475 ppm and 140 ppm, respectively, while average readings recorded for the methane gas and CO₂ gas were 202.5 ppm and 90.29 ppm, respectively.

Following graphs show the result of simulation performed using OMNET++ simulator and Wireshark. Figure 10 presents the delay experienced in connection time with respect to the number of devices. As it is shown, average delay in connection increases with respect to the number of devices. By increasing the number of devices as well as the number of clients, load on server increases because of which every client experienced delay in its connection time. Similarly, minimum connection time increased linearly.

Figure 11 represents the average response time of the MQTT server. It is observed that average response time becomes largely linear with the number of devices. By making a small change in number of devices, clients experienced larger delay in response time.

Data shown in Figure 12 is the mean end-to-end delay recorded in seconds. The delay is higher for QoS-0 as it is the simpler level and no acknowledgement is delivered for any data either published or subscribed.

Similarly, Figure 13 shows the mean message loss in the network. It is seen clearly that the message loss in QoS-2 is less as compared to that in QoS-0 and QoS-1.

From the results, it is clear that the proposed system can be used to develop an IoT-based smart city and provide different facilities using the IoT services. Moreover, the MQTT protocol that is used for the development of the system is better than the SIP protocol because of the following reasons: (1) MQTT is a lightweight protocol than SIP; (2) MQTT provides a very light header of just 2 bytes but is also capable of providing a flexible header size of up to 256 bytes making it suitable for handling video transmission over the network using green MQTT; (3) the MQTT is a published/subscription-based network where SIP is a request/response-based network; hence, MQTT handles requests efficiently than SIP; (4) MQTT also supports message payload up to 1000 bytes and makes the packet size relatively easier to handle; and (5) the average end-to-end delay and message loss are relatively less in the QoS-2 level, and it is clear that more an optimized form of the network can support more number of devices with less failure.

The proposed model is also a cost-effective model in terms of sensor node design as it provides a low-cost sensor node. Also, using a sink node common for several sensor nodes (1 sink node for at least 150 sensor nodes) helps to reduce the cost of the network deployment. The proposed model is also good for implementation in the countries, like Pakistan, as it is suitable for the weather condition as that of Pakistan.

6. Conclusion

The model proposed in this research is cost-effective and adaptive by the addition of many other services and firm technological support. The proposed model is simple and easy to implement with simple technology. The working and data collection and sharing are easy as it uses a simple way of communication. Moreover, as data is sent periodically between the sensor node and sink and sink and IoT cloud, thus the unnecessary overhead on the cloud as well as on the sink is removed. The data can also be fetched using local Bluetooth connection so not every user needs to connect with the IoT cloud. Third, the data can be fetched on demand; hence, new values can also be available even if the periodic cycle is not complete. The network is capable of handling more than 2000 devices in a single scenario with minimum delay and acceptable performance and efficiency. It is because of the stage-wise communication of the system. This model, at its infant stage, can be implemented at many places in Pakistan, even at a small-scale level, i.e., house, offices, or in small industrial areas. Furthermore, these small-scale projects also help the concept of smart urbanization to get fame and acceptance by the people of Pakistan. As a future prospective of the proposed model, a smart mobile-based package can be presented that supports connectivity of smart mobile devices. Besides, the factors regarding security, flexibility, scalability, and mobility can be addressed. For reducing the connectivity of the devices with the network and to mitigate the linear affect in connectivity by increasing the device number, the proposed design can be modified to introduce the factor mobility in the sink node. Another modification can be done by increasing the number of sink nodes twice relative to the sensor nodes.

Data Availability

All the necessary data is available in this manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful to the Taif University Researchers Supporting Project (number TURSP-2020/36), Taif University, Taif, Saudi Arabia. The authors are also grateful to the Deanship of Scientific Research at King Saud University for funding this research. This research work was also partially supported by the Faculty of Computer Science and Information Technology, University of Malaya, under Postgraduate Research Grant (PG035-2016A).

References

- [1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [2] I. Ganchev, Z. Ji, and M. O'Droma, *A Generic IoT Architecture for Smart Cities*, 2014.

- [3] L. Srivastava, *Japan's Ubiquitous Mobile Information Society*, info, 2004.
- [4] S. Giroux and H. Pigot, *From Smart Homes to Smart Care: ICOST 2005, 3rd International Conference on Smart Homes and Health Telematics*, vol. 15, IOS Press, 2005.
- [5] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of Things," *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1101-1102, 2012.
- [6] M. A. Chaqfeh and N. Mohamed, "Challenges in middleware solutions for the internet of things," in *2012 International Conference on Collaboration Technologies and Systems (CTS)*, Denver, CO, USA, May 2012.
- [7] M. Gigli and S. G. Koo, "Internet of Things: services and applications categorization," *Advances in Internet of Things*, vol. 1, no. 2, pp. 27-31, 2011.
- [8] S. Jain, S. Mane, J. Lopez et al., "A low-cost custom HF RFID system for hand washing compliance monitoring," in *2009 IEEE 8th International Conference on ASIC*, Changsha, China, October 2009.
- [9] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112-121, 2014.
- [10] N. Koshizuka and K. Sakamura, "Ubiquitous ID: standards for ubiquitous computing and the Internet of Things," *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 98-101, 2010.
- [11] R. S. Kshetrimayum, "An introduction to UWB communication systems," *IEEE Potentials*, vol. 28, no. 2, pp. 9-13, 2009.
- [12] N. Kushalnagar, G. Montenegro, and C. Schumacher, *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*, 2007.
- [13] Y. Leng and L. Zhao, "Novel design of intelligent Internet-of-Vehicles management system based on cloud-computing and Internet-of-Things," in *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, Harbin, China, August 2011.
- [14] P. Levis, S. Madden, J. Polastre et al., "TinyOS: an operating system for sensor networks," in *Ambient intelligence*, pp. 115-148, Springer, 2005.
- [15] P. McDermott-Wells, "Bluetooth scatternet models," *IEEE Potentials*, vol. 23, no. 5, pp. 36-39, 2004.
- [16] G. Montenegro et al., "Transmission of IPv6 packets over IEEE 802.15. 4 networks," *Internet proposed standard RFC*, vol. 4944, p. 130, 2007.
- [17] U. Rushden, *Belkin Brings Your Home to Your Fingertips with WeMo Home Automation System*, Press Room Belkin, 2012.
- [18] I. Ungurean, N.-C. Gaitan, and V. G. Gaitan, "An IoT architecture for things from industrial environment," in *2014 10th International Conference on Communications (COMM)*, Bucharest, Romania, May 2014.
- [19] C. Wang, Z. Bi, and L. da Xu, "IoT and cloud computing in automation of assembly modeling systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1426-1434, 2014.
- [20] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25-33, 2006.
- [21] R. Want, "Near field communication," *IEEE Pervasive Computing*, vol. 10, no. 3, pp. 4-7, 2011.
- [22] X. Xiaojiang, W. Jianli, and L. Mingdong, "Services and key technologies of the Internet of Things," *ZTE Communications*, vol. 8, no. 2, pp. 26-29, 2020.
- [23] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of MANET and WSN in IoT urban scenarios," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3558-3567, 2013.
- [24] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, "Smart cities and the future internet: towards cooperation frameworks for open innovation," in *The future internet assembly*, Springer, Berlin, Heidelberg, 2011.
- [25] D. Evans, "The internet of things: how the next evolution of the internet is changing everything," *CISCO White Paper*, vol. 1, no. 2011, pp. 1-11, 2011.
- [26] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22-32, 2014.
- [27] I. Vilajosana, J. Llosa, B. Martinez, M. Domingo-Prieto, A. Angles, and X. Vilajosana, "Bootstrapping smart cities through a self-sustainable model based on big data flows," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 128-134, 2013.
- [28] J. M. Hernández-Muñoz, J. B. Vercher, L. Muñoz et al., "Smart cities at the forefront of the future internet," in *Future internet assembly*, 2011.
- [29] E. A. Mulligan and M. Olsson, "Architectural implications of smart city business models: an evolutionary perspective," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 80-85, 2013.
- [30] N. Walravens and P. Ballon, "Platform business models for smart cities: from control and value to governance and public value," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 72-79, 2013.
- [31] P. Masek, J. Hosek, K. Zeman et al., "Implementation of true IoT vision: survey on enabling protocols and hands-on experience," *International Journal of Distributed Sensor Networks*, vol. 12, no. 4, 2016.
- [32] P. Masek et al., "Use case study on embedded systems serving as smart home gateways," *Recent Advances in Circuits, Systems and Automatic Control*, pp. 310-315, 2013.
- [33] W. T. Muswera and A. Terzoli, *Development of an IMS Compliant, Cross Platform Client Using the JAIN SIP Applet Phone*, Development of an IMS Compliant, Cross Platform Client Using the JAIN SIP Applet Phone, 2010.
- [34] S. Zapolskytė and V. Palevičius, "Overview and analysis of smart cities," *Mokslas-Lietuvos ateitis/Science-Future of Lithuania*, vol. 10, 2018.
- [35] J. Lee, S. Baik, and C. Choonhwa Lee, "Building an integrated service management platform for ubiquitous cities," *Computer*, vol. 44, no. 6, pp. 56-63, 2011.
- [36] V. Buscher and L. Doody, "Global innovators: international case studies on smart cities," *BIS Research Paper*, vol. 135, 2013.
- [37] K. Ergazakis, K. Metaxiotis, and J. Psarras, "Towards knowledge cities: conceptual analysis and success stories," *Journal of Knowledge Management*, vol. 8, no. 5, pp. 5-15, 2004.
- [38] R. R. Harmon, E. G. Castro-Leon, and S. Bhide, "Smart cities and the Internet of Things," in *2015 Portland International Conference on Management of Engineering and Technology (PICMET)*, Portland, OR, USA, August 2015.
- [39] L. Laursen, *Barcelona's Smart City Ecosystem*, Cities Get Smarter, 2014.

- [40] L. Bassett, *Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON*, O'Reilly Media, Inc., 2015.
- [41] T. Bray, "The javascript object notation (json) data interchange format (No. RFC 8259)," Technical Report, 2017.
- [42] C. Ortega-Corral, L. E. Palafox, J. A. García-Macías, J. Sánchez-García, and L. Aguilar, "End-to-end message exchange in a deployable marine environment hierarchical wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 10, no. 1, Article ID 950973, 2014.
- [43] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, "Foundations of JSON schema," in *Proceedings of the 25th International Conference on World Wide Web*, Montréal, Québec, Canada, April 2016.
- [44] B. Smith, "Introducing JSON," in *Beginning JSON*, pp. 37–47, Springer, 2015.