iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** XI **Month of publication:** November 2017

DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

Design and Deployment of MQTT Based HeTNeT Using IEEE 802.15.4 and IEEE 802.11 for Internet of Things

Ayaskanta Mishra[1]
[1]*Assistant Professor, School of Electronics Engineering, KIIT University, Bhubaneswar, Odisha, Postal Zip-751024 India.*

Abstract: In the era of Internet of Things (IoT),a heterogeneous wireless networks (HetNet) is a viable approach for optimal use of design advantage of both IEEE 802.15.4 and IEEE 802.11 networks. The primary focus of this paper is to propose heterogeneous network architecture based on IEEE 802.15.4 (LR-WPAN) and IEEE 802.11 (WLAN). Deployment of proposed architecture is done by designing apython based gateway device using Raspberry Pi. The proposed HetNet architecture is designed for a generic IoT based application platform; suitable for most of the real world IoT applications with minor modification. The LR-WPAN segment of the network will be connected with low data-rate, power constrained motes equipped with sensors and actuators where as WLAN network would be utilized as a backbone for link aggregation of numerous LR-WPAN networks. Light weight MQTT is used as application layer protocol. XBEE® modules are used to deploy WPAN & WLAN adaptor is used to connect to cloud.
***Keywords**: HetNet, Heterogeneous Network, IoT, IEEE 802.15.4, Zigbee, LR-WPAN, IEEE 802.11, WLAN, XBEE®, Raspberry Pi*

I. INTRODUCTION

This paper is focused towards proposing a heterogeneity wireless network architecture, HetNet[1] specially keeping in mind the Internet of Things (IoT) applications. The heterogeneity of the network architecture is achieved by combining IEEE 802.15.4[9] and IEEE 802.11[10] standards. The main idea for combining a LR-WPAN and WLAN network is to optimize the network by making use of the fundamental advantage of each of the above two types of network architecture for implementing a IoT[2] based application network. The IEEE 802.11 WLAN network is having higher transmission power and higher data-rate making it a good solution for the backbone for the mentioned IoT deployment scenario. However the IEEE 802.15.4 (Zigbee) LR-WPAN is a considerably low power making it more suitable for reduced power or power constraint applications for remotely deployed, battery powered devices in a typical IoT application. The lower data-rate of 250 Kbps of such networks is practical for sensor networks especially for IoT applications. Many such applications like smart city, smart health care, assisted living, smart appliances/ home automation is not bandwidth hungry. Since these IoT applications merely require just a single or few sensors to give some parametric date and in addition to that might require a actuator or relay to be driven. These typical applications required just few bytes of control signals and sensor data to be sent as a payload over the network, may be under-utilizing the payload a power hungry high data-rate WLAN network. In any typical real world IoT deployments scenario tiny embedded devices/motes will be mostly battery powered, which will have a very significantly less data-rate and computational power requirements. Thus, a LR-WPAN network using IEEE 802.15.4 would suffice the need of the above. [9]the other hand the WLAN segment of the HetNet, has a specific role to play in this case. Needless to say IEEE 802.11 networks are significantly having higher data rate 11/54/72/144/288 Mbps even up to 600 Mbps using MIMO IEEE 802.11n standard. These WLAN can be used for interconnection of such LR-WPANs and can be utilized as the backbone network for link aggregation. These WLAN NICs will be placed only on the gateway devices which will act as a bridge between IEEE 802.15.4 and IEEE 802.11 networks. These gateway devices are preferably PAN Co-coordinators and static from the LR-WPAN prospective. Further another aspect to IEEE 802.11 WLAN network is the transmission power is typically 100 mW can go up to 1 W as per FCC guidelines. This makes the WLAN quite a good choice for longer range communication as compared to the IEEE 802.15.4 (WPAN) counterpart. [10]

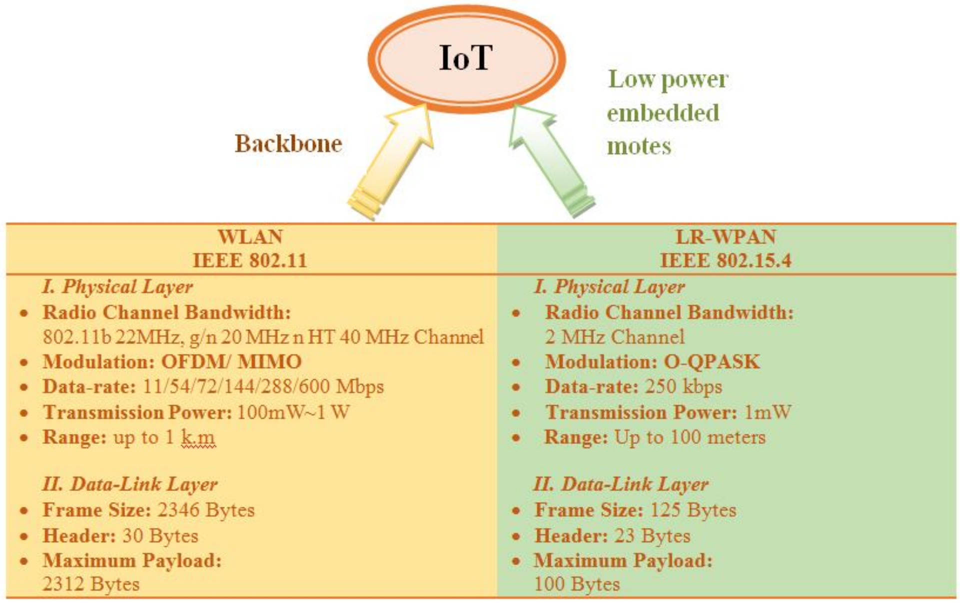
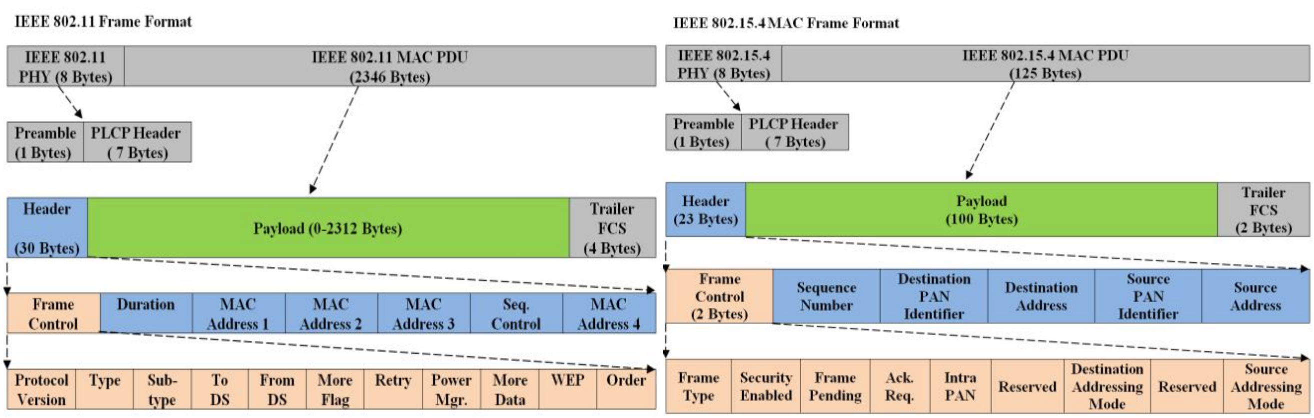


Fig.1Comparison between WLAN and LR-WPAN for Internet of Things Applications

Fig.1 shows a comparison between WLAN and LR-WPAN. The figure shows the comparative analysis of advantage of each of the two standards of wireless networks especially to cater different network requirements for an IoT application deployment or prototyping [3].

II. HETNET: HETEROGENEOUS WIRELESS NETWORKS FOR INTERNET OF THINGS

The proposed a HetNet: heterogeneous Wireless Networks architecture for Internet of Things (IoT) application is shown in Fig.3. Internet of Things is not a technology rather a vision to interconnect all the smart devices to seamlessly interact and exchange data [2]. IoT application requires a underlying platform comprises of LAYER-I: Physical and LAYER-II: Data-link technologies to transmission of data over IP based network or Internet [6].



(a) IEEE 802.11 Frame (b) IEEE 802.15.4 Frame
Fig.2 Frame Format IEEE 802.11 and 802.15.4 Comparison Size

Fig.2 shows a comparison between the frame structure of IEEE 802.11 and IEEE 802.15.4 [1]. It is evident from the figure that the prior one is having a maximum payload capacity of 2312 bytes which is considerably higher than the later having only 100 bytes of maximum payload per frame. This gives a conclusive remark about the use of IEEE 802.1.5.4 is limited to a low rate and low power network suitability for tiny sensor motes. The IEEE 802.11 is a more complex and performance savvy alternative, most suitable for high data-rate, high bandwidth, more payload, longer range wireless communication. All these additional benefits comes with a cost rather, the cost here is power. IEEE 802.11 is having significantly higher power requirements, which makes it suitable for interconnecting multiple WPANs by acting as a backbone network in this case. The gateway device [4] will accumulate all the data coming from numerous motes connected in LR-WPAN and transmit over IEEE 802.11 network. This Layer: I/II underlying technologies are not fixed for Internet of Things, rather this technology has to be judiciously chosen synchronous with the scenario and application for which the particular IoT network is deployed [5]. Especially two technologies are prominent for deployment of IoT network in different scenario specific requirements. Firstly the IEEE 802.11 WLAN network for deployment of IoT and secondly the IEEE 802.15.4 (LR-WPAN) Zigbee network. Both the IEEE standard wireless networks mentioned in this context are

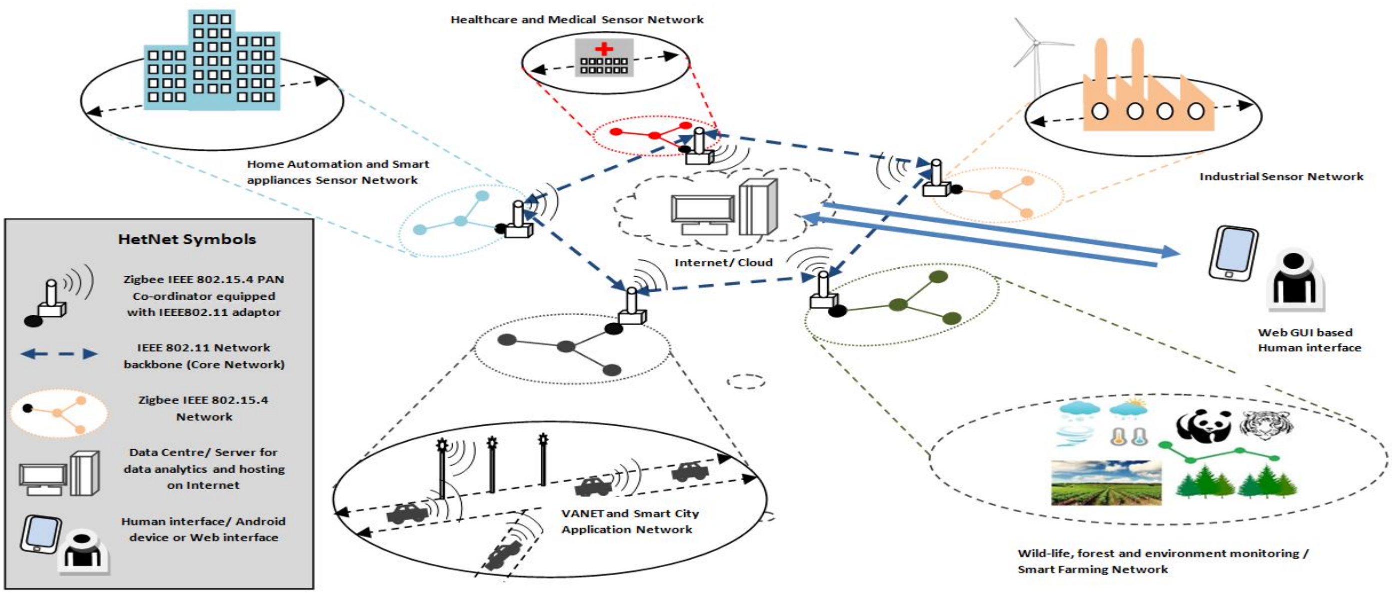


Fig. 3 Proposed IEEE 802.15.4 and IEEE 802.11 based Heterogeneous Wireless Network (HetNet) for Internet of Things (IoT) applications

deployed using 2.4 GHz ISM unlicensed band. To combine both the networks, a gateway device has to be designed for interoperability of these two networks; Since both the network are not compatible with each other as their design specification like radio channel, bandwidth, modulation scheme, frame format are different. Hence here a python based software module is designed and implemented using Raspberry Pi. The python based gateway module which is designed here is to act as a bridge between IEEE 802.15.4 and 802.11 networks. As per the proposed HetNet architecture the LR-WPAN Mesh network would be deployed using XBEE® modules [11] [12]. The devices can be configured a Zigbee Routers/RFD: Reduced Function Device based on their location from PAN coordinator. The PAN Coordinator will be connected directly to the Raspberry Pi through Serial Link/ USB. The XBEE® modules have Analog/Digital I/O pins which can be interfaced with Sensors and relays based on the IoT application. The Sensor data or actuator control signals can be sent or received by these XBEE® to the PAN Coordinator over the Wireless network. The XBEE® works in two modes (1) AT/ transparent Mode (2) API Mode. [11] This context the IEEE 802.15.4 Zigbee Frames would be crafted from so API mode is suggested in this proposed work. The API mode data would be sending in Zigbee Frames as data-link layer. The AT mode the commands are send by AT commands which is not relevant form out HetNet context.

The PAN Coordinator will identify the XBEE® module based on its ID or 64-BIT XBEE(R) ADDRESS. The datagram from all the motes in WPAN will be forwarded to PAN Coordinator, which is hardwired to the Raspberry Pi. The designed gateway module will access the UART of the Raspberry pi to read the datagram from all the motes in the PAN. From the Header the SA: Source 64-bit XBEE(R) Address can be found out which in later stage will be helpful for gateway to map the device to any MQTT topic as discussed later.

The gateway module can be configured to map the particular XBEE® module to be a particular MQTT topic. Case-(i): If the XBEE® is connected to a sensor then it will read data from the sensor and send the data to the cloud, to do this, the gateway device would be configured to MQTT_ publish command. In the gateway it ID of the device can be mapped to the designed MQTT topic and when the sensor data is received by the gateway from that particular XBEE® mote the sensor data can be published to the MQTT broker using above mention MQTT_ publish command programmed using python part of gateway algorithm. Case (ii): If the XBEE® is connected to a actuator / relay to drive or control any external device then it XBEE® DIO Pin would be used (Digital

Output). In this case the particular XBEE® mote would be subscribed to the designated MQTT topic using *MQTT_Subscribe* command programmed in gateway device.

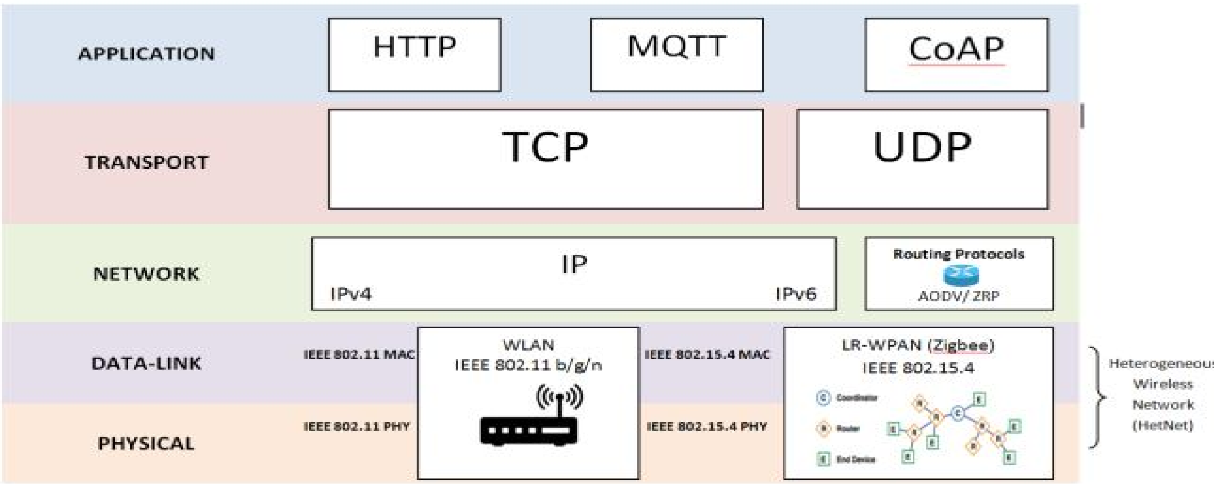


Fig.4 OSI Layer Architecture for IoT using HetNet

The gateway device will call the mosquito MQTT module to make the required data forwarding to the MQTT broker based on the mapping , which is pre configured. The data from the gateway device will now use the IEEE 802.11 network to reach the IP based network. Using the packet crafting or data accumulation and packet crafting can be implemented to send the data to the cloud. In this case Cloud MQTT is considered as the MQTT broker as it deployed over internet and accessible without any geographical or location constraint. The data will go to the cloud MQTT from the mosquito module using the designed gateway device. The data can also be retrieved for Cloud MQTT and given to WPAN XBEE® motes using MQTT_Subscribe.

III. MQTT ARCHITECTURE

Message Queue Telemetry and Transport (MQTT) is a light weight application layer protocol [7] most suitable for reduced power low- data Sensor networks. MQTT protocol works on two types on controls 1. Publish: The MQTT clients publish data (Sends data) to MQTT broker to a particular topic. 2. Subscribe: The MQTT client may subscribe to a topic. Whatever data is published to that topic the subscribed client will get the data immediately (receives data). There can be multiple devices (clients) subscribed to the same topic, which makes all such devices subscribed to the topic to get a copy of received from the MQTT broker.

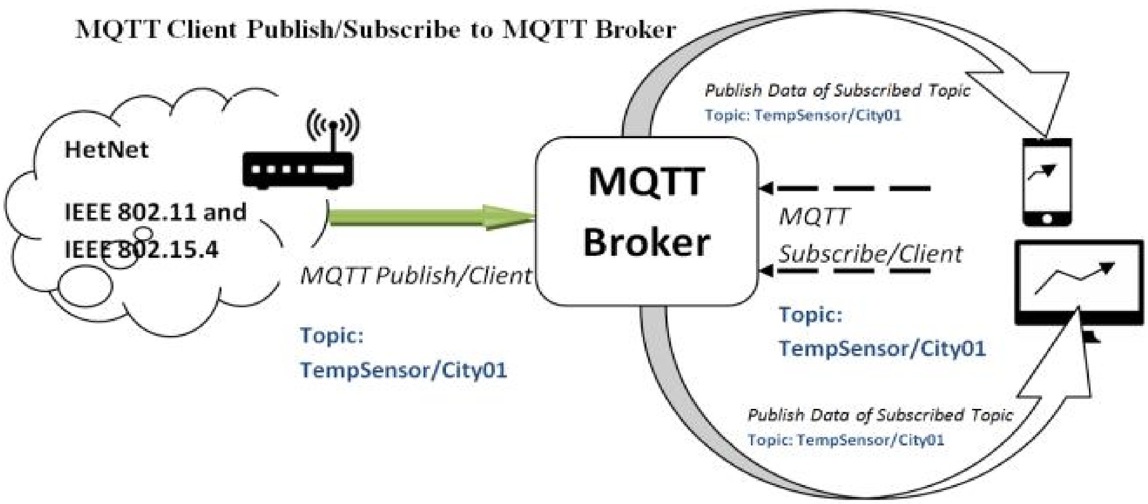


Fig.5 MQTT Protocol: MQTT_publish and MQTT_Subscribe

In MQTT terminology, Broker is the Server for MQTT protocol. All publish and subscribe control messages are sent to broker. Its broker's responsibility to get the data from publishing devices and forward the data to subscribed devices. Fig.5 shows the MQTT protocol working principle.

A.MQTT over Heterogeneous Wireless Network (HetNet)

From the prospective of this paper, MQTT protocol as discussed above is implemented using two network technologies. IEEE 802.11 being the backbone and the IEEE 802.15.4 being the LR-WPAN connecting all the Sensor/Actuators. Fig.6 shows the Encapsulation of data in the different PDUs (protocol data units). The XBEE® motes which are physically connected to the Sensors/ Actuators are encapsulating the data in IEEE 802.15.4 frames and all the data over whole WPAN is forwarded to PAN Coordinator, connected to Gateway device (Raspberry Pi). The Gateway software module in Raspberry Pi captures the XBEE® frames using Serial communication. Further it extracts the Sensor data from the payload. This sensor data is then used as the message to MQTT Publish message sent to MQTT broker.

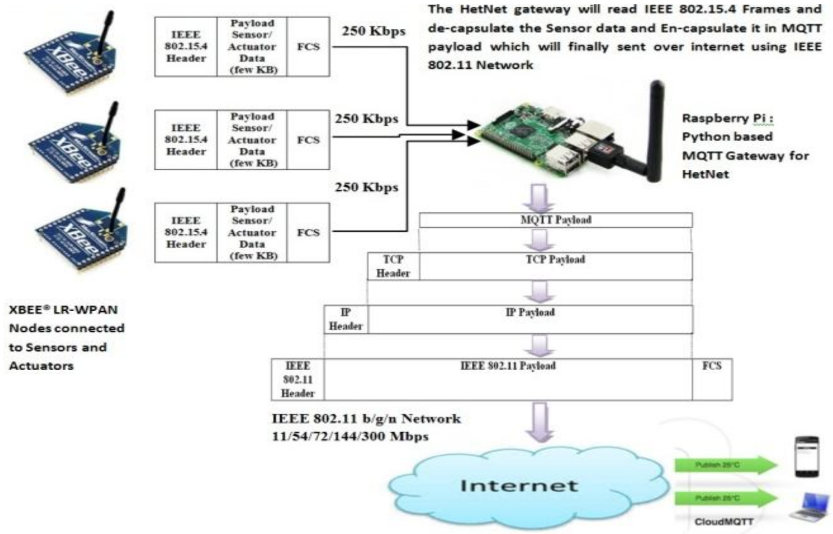


Fig.6 MQTT over TCP/IP Encapsulation (Packet crafting Encapsulation / Utilization of IEEE 802.11 payload)

Fig.6 further shows the MQTT application layer message is encapsulated in TCP Segment. TCP Segment being encapsulated in IP datagram, which again encapsulated in IEEE 802.11 frame and transmitted over internet and de-capsulated at the MQTT broker to receive the data.

IV. DEPLOYMENT

A MQTT based HetNet using XBEE® as IEEE 802.15.4 motes and a IEEE 802.11n adaptor is used in the Raspberry Pi. The Raspberry Pi 3 is used as a gateway device which works as a bridge between IEEE 802.15.4 network and IEEE 802.11 network. The prior LR-WPAN network is connected to sensors and relays for deploying a IoT application, the later one is connected to internet to push data using MQTT publish to Cloud MQTT broker. The MQTT publish command is issued using mosquito using python [8]. The PAN Coordinator of the IEEE 802.15.4(XBEE®) is connected to Raspberry Pi using a Serial-USB adaptor. The XBEE® in this case which is used to deploy IoT application is working in API Mode. The IEEE 802.15.4 data frames are captured by PAN coordinator over air interface and mirror that over UART connected to raspberry Pi. The python gateway module starts a serial communication between XBEE® PAN Coordinator o capture the frames which are transmitted over the Wireless PAN.

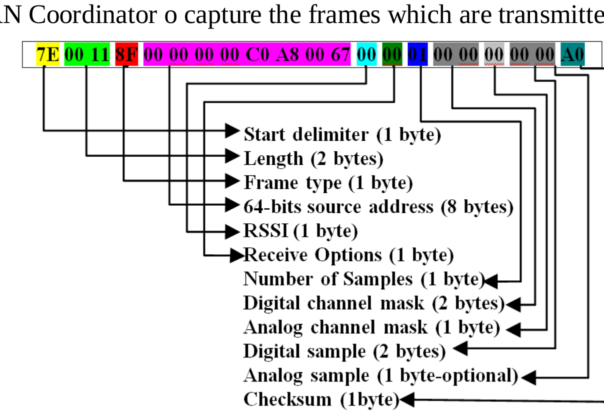


Fig.7 XBEE® Frame Format: Identifying fields extracting Sensor data from Analog/Digital Sample with the help of Channel mask

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

The Fig.7 shows the XBEE® frame structure. The captured frame at Raspberry Pi using the python based gateway software module is programmed to segregate the fields and find out the Analog/ Digital Sensor data coming from any particular XBEE® mote by knowing it 64-bits source address in the frame.

The gateway software program which is designed here would be pre-configured to map the 64-bit source address of the XBEE® to a particular MQTT topic to publish the Analog/Digital data coming from the sensors. The following command can be executed inside python using system call within the python script to publish data to the Cloud MQTT broker using mosquito. The algorithm of python based gateway module for publishing Sensor data is shown in Fig.8.

```
mosquitto_pub –h cloudmqtt.com –p 1883 –u pikumax –P as78xzj –m “$time $temprature”-t TempSensor/City01
```

A. Algorithm python based Gateway MQTT Publish Sensor data over HetNet:

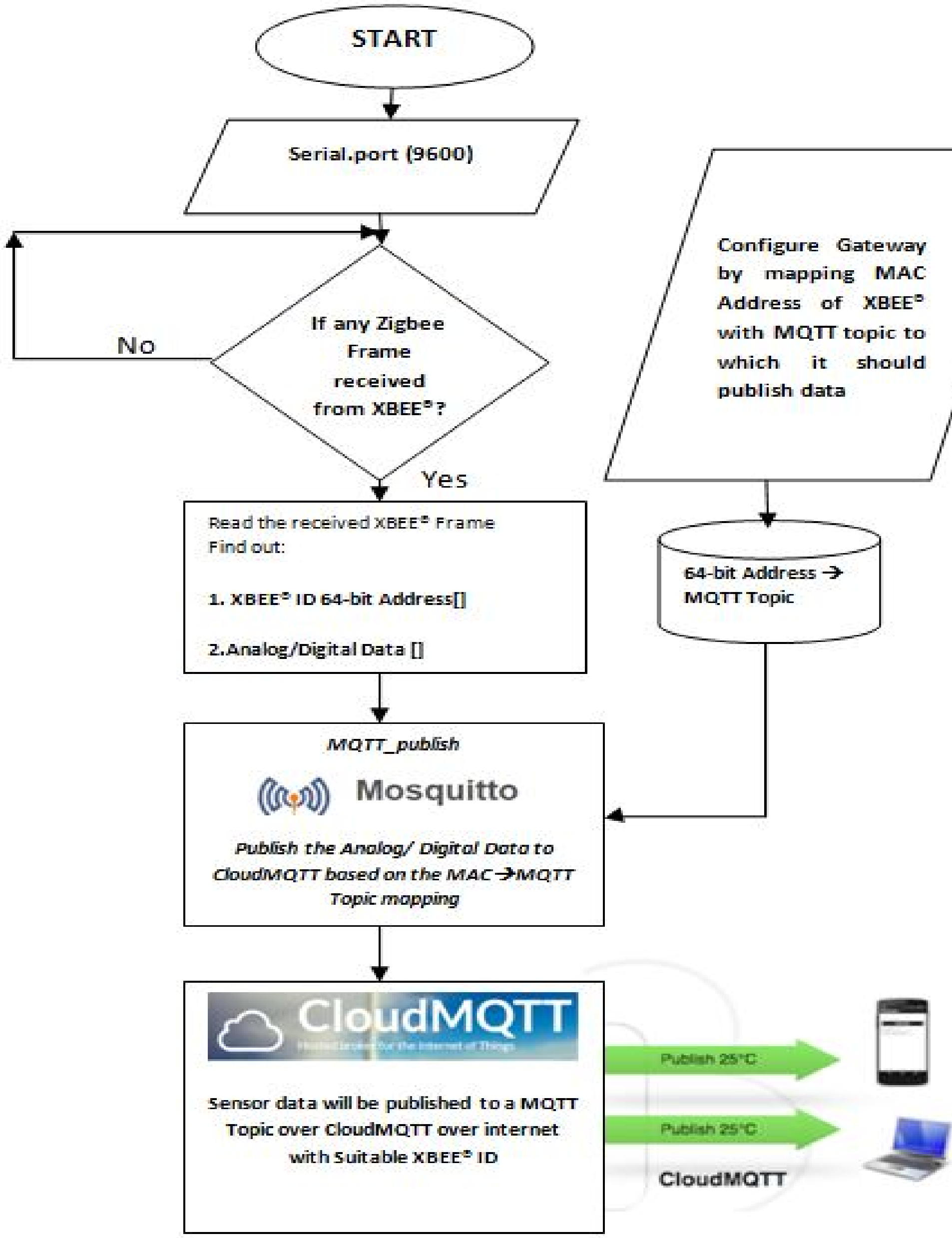


Fig.8Publishing Sensor data from XBEE® to Cloud MQTT through HetNet gateway

Considering another scenario, where a user require a particular hardware to be controlled by driving a reply connected to XBEE® in LR-WPAN network can also be implemented in the python based gateway in Raspberry Pi.

B. Algorithm python based Gateway MQTT Subscribe to drive actuators over HetNet

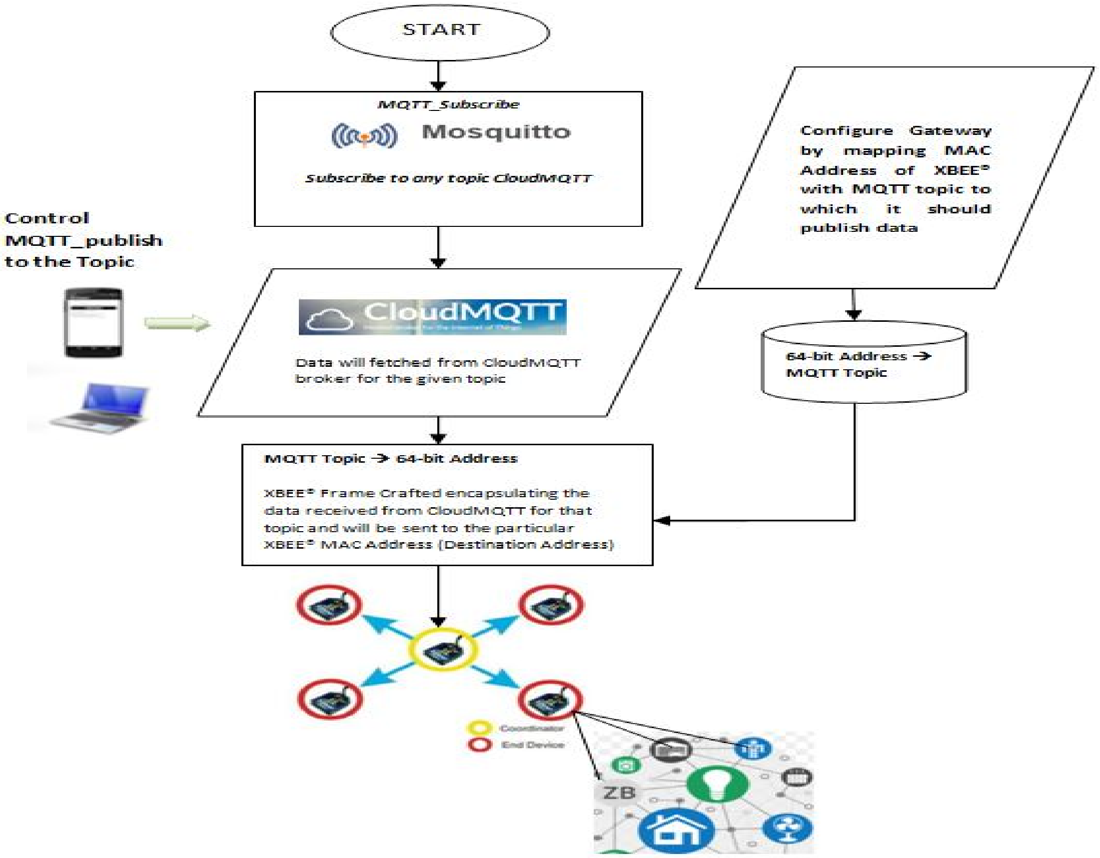


Fig.9 XBEE® Subscribe to Cloud MQTT through HetNet Gateway: Controlling Actuators/ Relays using MQTT_Subscribe

Fig.9 shows the algorithm for controlling relays. Firstly the GUI from web based application can take user response using button or any other interactive method to take user input to take MQTT publish message sent to broker to a particular topic like control/relay. The gateway module is preconfigured to subscribe to the same topic. Whenever the user sends a command to control any relay, it will be published to control/reply topic to which gateway device is subscribed. Thus the command will be fetched by gateway device. At the gateway the topic to 64-bit XBEE® address mapping look-up table would be stored in database. The gateway python software module would refer the database the control/reply topic information would be forwarded to the mapped XBEE® 64-bit address. This will change the digital pin data of the particular XBEE® hence control the relay physically connected to that XBEE® [11].

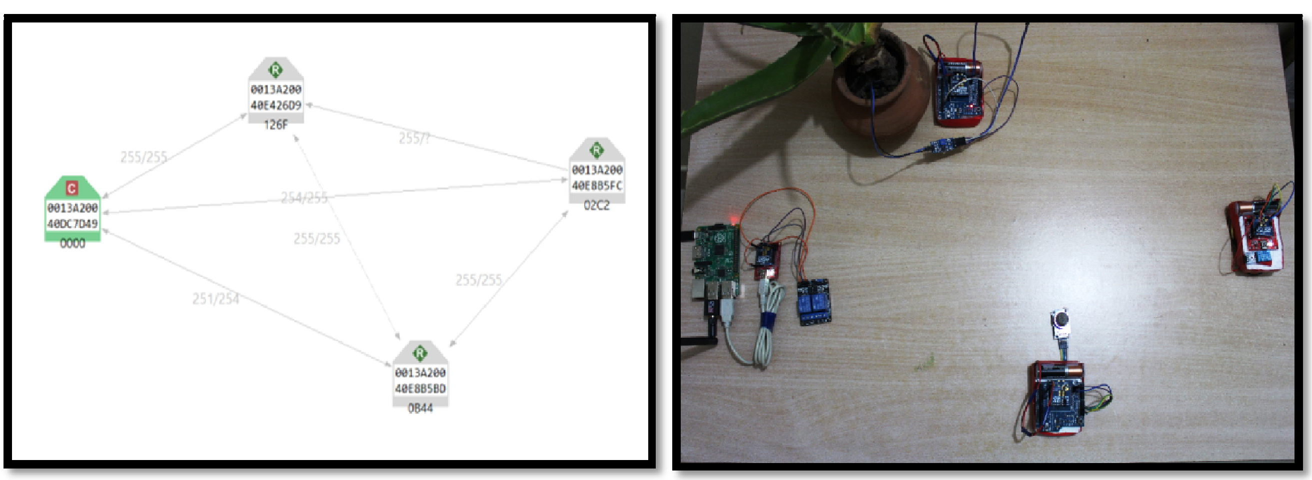


Fig.10 HetNet using IEEE 802.11 network in Raspberry Pi gateway device and IEEE 802.15.4 XBEE®

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

Fig.10 shows the WPAN using Digi-XCTU network topology and further shows the HetNet deployment using four XBEE® motes designed as per application requirements.

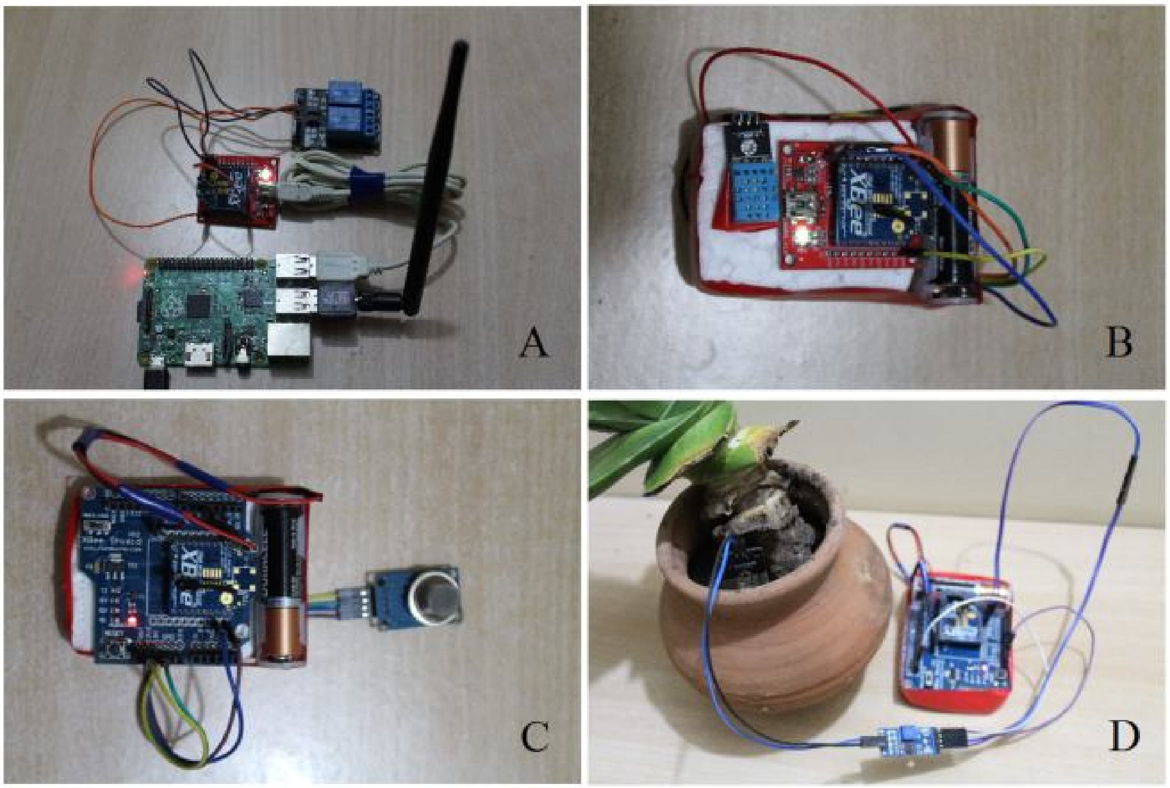


Fig.11 A. Raspberry Pi Gateway device for interconnecting XBEE® IEEE 802.15.4 (LR-WPAN) with IEEE 802.11 Network to connect to Cloud MQTT, B. DHT11 Sensor equipped XBEE® Mote, C. MQ135 Sensor equipped XBEE® Mote, D. HL-69 Sensor equipped XBEE® mote.

Fig.11 shows all the four XBEE® modules which make the WPAN. Out of these four, one is connected directly to Raspberry Pi works as the PAN Coordinator as also seen in Fig.10 XCTU [11] topology diagram. Other three XBEE® motes are dedicated to different types of Sensors as per application requirements as shown in Fig.11. All the motes are battery powered.

V. RESULTS:VISUALIZATION OF SENSOR DATA USING MQTT SUBSCRIBE

Visualization of data coming from XBEE® connected sensors MQTT subscribe is used. Sensor data coming from three individual Sensors namely DHT11, MQ135 and HL69 are publishing to their respective topic through python gateway module implemented in Raspberry Pi as discussed prior. Now the gateway module takes the data from XBEE® frame refer the look-up table for mapping of 64-bit source address of XBEE® to map it to a particular MQTT topic. Once the Sensor data is published to the respective topics to broker now it's the question of retrieve the data globally using internet. Two aspect of the data handling are discussed here: 1. Visualization 2. Storage of data/ data analytics.

The Visualization or storage of the Sensor data can be done by any internet connected device globally. The Device can issue a mosquito subscribe message to the Cloud MQTT broker to the topic in which it is required to fetch data as per the sample python code below [8].

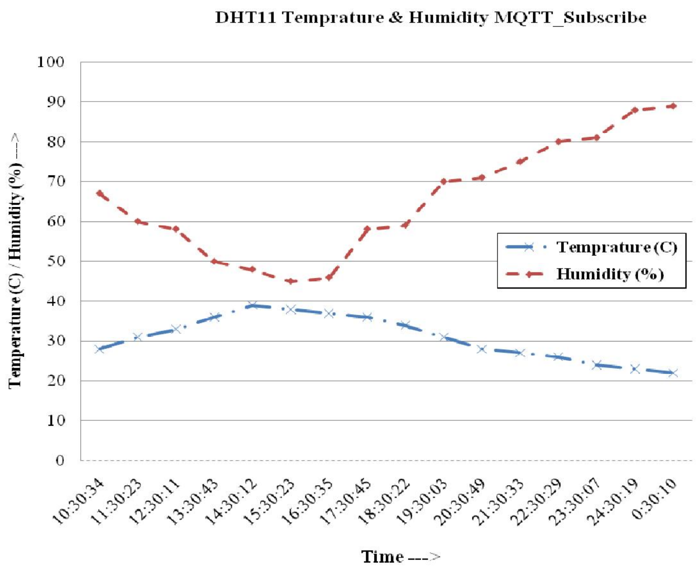
```
mosquitto_sub –h cloudmqtt.com –p 1883 –u pikumax –P as78xzj -t TempSensor/City01
```

Once the device is subscribed to a topic the broker would forward the Sensor data which is published in real-time. The python program can use file handling and store the received sensor data to a database. In this case a simple .CSV file. The .CSV file can now be used by any graph plotting tool like xgraph, MATLAB or MS-EXCEL or any graphical plotting tool for visualization of

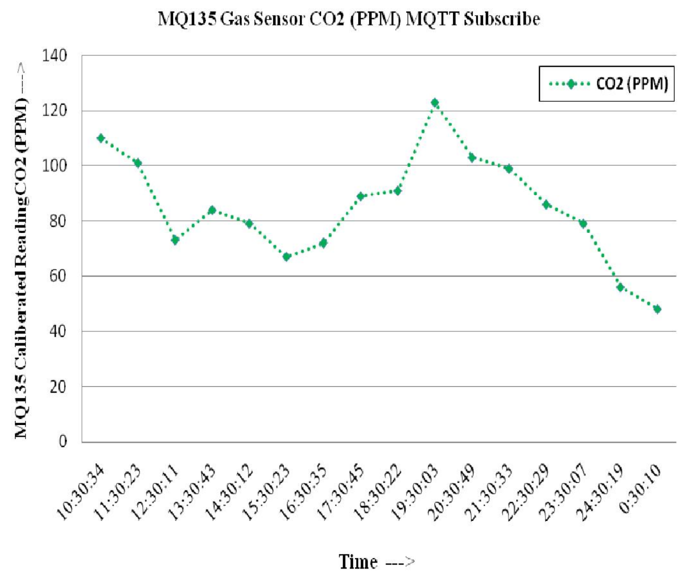
International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

Sensor data on any GUI and storage of data. If the complexity of Sensor data is more or more number of Sensors are deployed SQL can be a viable option for database or storage of sensor data.

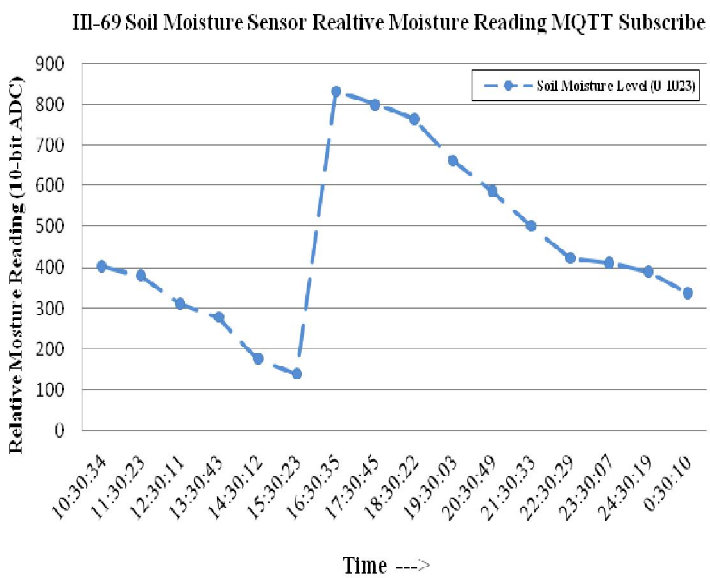
A. Sample Sensor data using MQTT Subscribe using msquitto_sub:



(A)



(B)



(C)

Fig.12 (A): Data Visualization from remote DHT11 temperature & humidity Sensor using Mosquitto MQTT Subscribe to Sensor/Temp_Humdity topic. Fig.12 (B): Data Visualization from remote MQ135 Gas Sensor calibrated CO2 concentration using Mosquito MQTT Subscribe to Sensor/Gas topic Fig.12 (C): Data Visualization from remote HL-69 Soil Moisture Sensor calibrated 10 bit ADC reading using Mosquito MQTT Subscribe to Sensor/Gas topic.

Fig.12 (A) shows the visualization of .CSV file which stored using MQTT subscribe to Sensor/DHT11 topic. The topic, to which XBEE® is publishing data from DHT11 Temperature& Humidity Sensor as shown in Fig.11. Fig.12 (B) shows the visualization of .CSV file which stored using MQTT subscribe to Sensor/MQ135 topic. The topic, to which XBEE® is publishing data from MQ135: Gas Sensor as shown in Fig.11. The Gas Sensor is capable of showing CO2 Concentration (PPM) as per figure. Fig.12 (C) shows the visualization of .CSV file which stored using MQTT subscribe to Sensor/DHT11 topic. The topic, to which XBEE® is publishing data from HL69 Soil moisture Sensor as shown in Fig.11.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

VI. CONCLUSION

This paper scope was to design and deploy a MQTT based gateway using python using heterogonous wireless network (HetNet). The merit of this is to use a LR-WPAN, which is optimum of battery power low data rate IoT Sensor/Actuator motes. Further MQTT is a light weight simple application layer protocol to suffice the need in this scenario. In addition to this the IEEE 802.11 network from Raspberry Pi Gateway device to the MQTT broker, Cloud MQTT which is deployed over internet is a added advantage to accumulate all the sensor/Actuator control signal together from hundreds of such tiny embedded motes using link-aggregation (IEEE 802.11 having higher data-rate and longer range IP based network; which can handle huge number of IEEE 802.15.4 data accumulated from a whole PAN through PAN Coordinator. The HetNet architecture proposed here is a generic model so any other IoT application apart from the one deployed in this paper can also be implemented with little bit of tweaking in the software.

VII. FUTURE WORK

The work can be extended by designing a android app based platform integrating MQTT publish and subscribe framework to develop a generic IoT platform to deploy any Sensor and actuator applications in the context of future Internet of Things.

REFERENCES

[1] H. Chen, "Heterogeneous network integration based on protocol conversion," 2016 35th Chinese Control Conference (CCC), Chengdu, 2016, pp. 6888-6893.doi: 10.1109/ChiCC.2016.755444

[2] S. Chen, H. Xu, D. Liu, B. Hu and H. Wang, "A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective," in *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 349-359, Aug. 2014.

[3] G. V. Vivek and M. P. Sunil, "Enabling IOT services using WIFI - ZigBee gateway for a home automation system," 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Kolkata, 2015, pp. 77-80

[4] D. C. Yacchirema Vargas and C. E. Palau Salvador, "Smart IoT Gateway For Heterogeneous Devices Interoperability," in *IEEE Latin America Transactions*, vol. 14, no. 8, pp. 3900-3906, Aug. 2016

[5] Ni, Y. , Miao, F. , Liu, J. and Chai, J. (2013) Implementation of Wireless Gateway for Smart Home. Communications and Network, 5, 16-20.

[6] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. Mccann and K. K. Leung, "A survey on the ietf protocol suite for the internet of things: standards, challenges, and opportunities," in IEEE Wireless Communications, vol. 20, no. 6, pp. 91-98, December 2013.

[7] S. Bandyopadhyay and A. Bhattacharyya, "Lightweight Internet protocols for web enablement of sensors using constrained gateway devices," 2013 International Conference on Computing, Networking and Communications (ICNC*)*, San Diego, CA, 2013, pp. 334-340.

[8] Ding Yi, Fan Binwen, Kong Xiaoming and Ma Qianqian, "Design and implementation of mobile health monitoring system based on MQTT protocol," *2016 IEEE* Advanced Information Management, Communicates, Electronic and Automation Control Conference *(IMCEC)*, Xi'an, 2016, pp. 1679-1682.

[9] IEEE Std 802.15.4™-2015 (Revision of IEEE Std 802.15.4-2011) IEEE Standard for Low-Rate Wireless Networks, IEEE Computer Society ISBN 978-1-5044-0846-2

[10] IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput," in IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009) , vol., no., pp.1-565, Oct. 29 2009

[11] D. International, "XBee User Manual," ed: Digi International, 2012, pp. 1-155.

[12] Piyare, Rajeev & Lee, Seong-Ro. "Performance Analysis of XBee ZB Module Based Wireless Sensor Networks" International Journal of Scientific & Engineering Research Volume 4, Issue 5, May -2013