EFFICIENT SENSOR DATA TRANSMISSION TO THE IOT APPLICATIONS USING MQTT-SN AND MQTT

Mr. M.OBULAREDDY,
Research scholar, Gitam (Deemed to be University), Vishakhapatnam
moreddy2003@gmail.com

Mr. M.SRINU,
Research Scholar, Osmania University, Hyderabad.
srinumtech277@gmail.com

Abstract: ***Internet of Things (IOT) Technology rapidly growing area and active research happening in different Layers of Internet Protocol Stacks. IOT stack have Physical layer& Data Link layer, Network Layer, Transport deposit and Application deposit Messaging protocols and Applications. Now IOT has many applications like Smart cities, Smart Homes, Environmental monitoring, Agricultural Application and Medical data transmission from remote places to expert doctors. Numbers of IOT Application etiquette (MQTT, MQTT-SN, COAP, XMPP, and HTPP) are available and implemented in rich resource Environments like good computing power and bandwidth. However, all Application etiquette is not appropriate under lossy wireless antenna complex environment. In IOT Environment consists of antenna Nodes, End Systems and related applications Latency and bandwidth problems exist between sensor node and End Systems in two tiers IOT Environment. To resolve this issue, we proposed Three Tier IOT Architecture; it consists of antenna Node, IOT Gateway /Fog Computing Node, End Systems and Applications. We contain discuss proposed flexible design, development and integration of IOT Gateway for different IOT applications using more efficient MQTT-SN and MQTT application messaging Protocols.***

KEYWORDS: IOT, MQTT, MQTT-SN, Wireshark, Node-Red, Ardunio

I. INTRODUCTION

Wireless antenna complex is a group of distributed embedded devices (nodes) with the capabilities of sensing, processing and transmission of data through wireless communications to the application. WSN has come out seeing that a major research area which provides the interaction between the substantial globe and humankind. There have been various applications developed using WSN, such as environment monitoring, Building monitoring, military systems, smart home etc. Traditional WSN complex is designed either as proprietary or non standard building monitoring, military Systems, smart home etc. Traditional WSN complex is designed either as proprietary or non standard application protocols. Current Wireless antenna complex application etiquette are not compatible to the wide spread Internet Application protocols.

Various Applications messaging protocols like HTPP, MQTT, COAP, XMPP, DDS etc are designed, developed and deployed in the distributed computing environment. But this etiquette is not appropriate to loss wireless antenna complex due to constrained computing power and bandwidth. So it is essential that light weight IOT application messaging protocol for transportation of real time and non real time data to the IOT Applications has to be designed and developed. As per [1] MQTT- SN specifications are released for academic, explore addition to engineering uses. MQTT-SN is communication line Telemetry convey etiquette designed for antenna Networks. In this paper, MQTT-SN etiquette is designed with adaptability of loss WSN networks and is as much as compatible with MQTT. As per [2], theoretical IOT application messaging protocols are compared within conditions of protocol message over head, throughput and Bandwidth. The same as for each [3],

MQTT-SN end to terminate excellence promise theoretical frame work is discussed. The declaration of the manuscript is seeing that tag along: Into Segment II IOT Architecture [Sensor Node, IOT Gateway, MQTT Server, and IOT Application] is explained, Section III discusses about IOT application etiquette stack and message sequence diagram, MQTT-SN messaging format, important MQTT-SN messages description, MQTT-SN QOS model and MQTT-SN Topic management, In Section IV Proposed work, new system is discussed. In Section V, Sensor node, Gateway in addition to IOT application software flowchart discussed. In 2 VI trace logs for all nodes (Sensor Device, IOT Gateway, MQTT Server, and IOT Application) are converse. Into Segment VIII results and conclusions are explained.

II. IOT ARCHITECTURE

The IOT architecture diagram is uncovered inside Fig.2.1. It consists of antenna Nodes and Actuators, IOT Gateway/Fog computing Node, MQTT Server and IOT Applications.

Sensor Node: Sensor Node consists of antenna information Collector, JSON Data formatter and MQTT-SN application Pub/Sub protocol stack. Inside these employments Node MCU/ESP82826 IOT hardware platform is used in the direction of talk through the MQTT-SN Gateway.

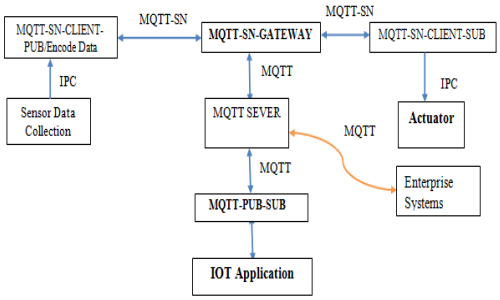


Fig.2.1: IOT Architetcure

Actuator Node:

Actuator Node subscribes to the required topic with Gateway and Server. Application drive applicable instructions to MQTT-Server, MQTT Server forwards to IOT Gateway and IOT Gateway forwards to Actuator device. After receiving commands from IOT Gateway, actuator processes the commands and takes appropriate action.

IOT Gateway:

IOT Gateway/Fog Computing Node is an important Node in the IOT Architecture. It consists of MQTT-SN protocol Encoder/ Decoders, Protocol state machine, Event Queue, MQTT protocol Encoder/Decoder, MQTT Protocol state machine. Basically it converts the MQTT-SN etiquette to MQTT protocol and communicates with the Sensor Node platform and MQTT Server. It is designed and developed on multithreaded environment under Linux in service arrangement.

MQTT Server:

MQTT is light weight protocol and implemented in favor of appliance in the direction of appliance Communication. In the IOT backend systems MQTT is deployed in number of IOT applications like Amazon Web services, IBM Watson IOT platform and Microsoft Azure platform.

IOT Application:

IOT Application has integer of request as per requirement. In this research work simple IOT Application is developed for hotness, dampness in addition to device parameters are monitored and stored in SQL Lite database for further use cases.

III. IOT PROTOCOL STACK AND MQTT-SN MESSAGE FLOWS

The IOT Protocol stack diagram is exposed inside Fig 3.1.

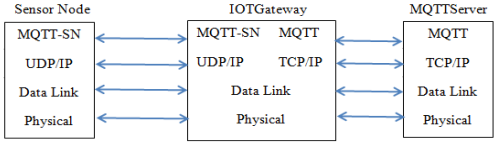


Fig. 3.1 : IOT Protocol stack diagram

MQTT-SN Architecture:

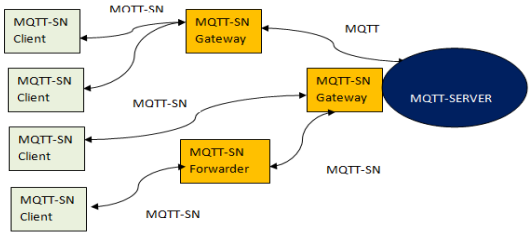


Fig. 3.2:MQTT-SN Architecture

MQTT-SN differences from MQTT:

- Connect message divided into three messages. One message is mandatory and reaming two are optional as will message requirement.
- Topic ids are worn inside position of long topic names.
- Small theme names
- Pre-defined topic names.
- Clients can find out the entryway
- Determination theme in addition to messages can be able to distorted throughout during the conference time
- Tracking of sleeping clients

MQTT-SN QOS-01 Messsage flow:

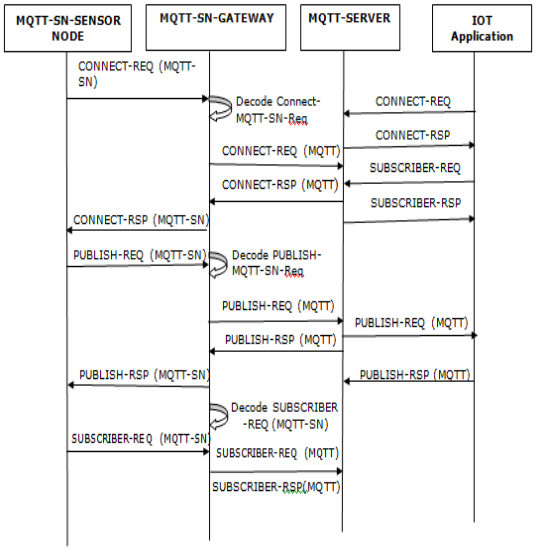


Fig .3.3 : MQTT-SN QOS-01 flow

MQTT-SN General Message Format:

Communication slogan (2 or 4 octets)	Communication changeable fraction (n octets)

MQTT-SN Message Header format:

Length (1 or 3 octets)	MsgType (1 octets)

MsgType Field Value	MsgType	MsgType Field Value	MsgType
0X00	ADVERTISE	0x0B	REGACK
0x02	GWINFO	0x0D	PUBACK
0x04	CONNECT	0x0F	PUBREC
0x06	WILLTOPICREQ	0x13	SUBACK
0x08	WILLMSGREQ	0x15	UNSUBACK
0x0A	REGISTER	0x17	PINGRESP
0x0C	PUBLISH	0x1B	WILLTOPICRESP
0x0E	PUBCOMP	0x1D	WILLMSGRESP
0x10	PUBREL	0xFE	Encapsulated message
0x12	SUBSCRIBE		
0x14	UNSUBSCRIBE		
0x16	PINGREQ		
0x18	DISCONNECT		
0x1A	WILLTOPICUPD		
0x1C	WILLMSGUPD		
0x01	SEARCHGW		
0x05	CONNACK		
0x07	WILLTOPIC		
0x09	WILLMSG		

Fig.3.4: MQTT-SN Message Table

MQTT-SN protocol Messages:

MQTT-SN protocol messages are shown in Fig-3.4

MQTT-SN QOS Table:

QOS	Description

QOS-0	Communications are transport at most once(may be messages are lost)
QOS-1	Communications are transport at least once(chances to receive duplicate messages)
QOS-2	messages are delivered exactly once
QOS-(-1or 3)	Initial connection need not be required, directly publish the message using predefined or short topic id's.

MQTT-SN Topic management:

Topic is a logical addressing entity in MQTT-SN and MQTT protocols. Subscriber can subscribe required topics for particular type data. Whenever MQTT Server received data on the topic, it publishes the data to the subscribed topics of subscribers. Topic name is any type of alpha numeric string in the MQTT.But it is not standardized, application can chose desired topic name as per requirement. For Ex: Topic Name:/Home/Room1/Temp/,Home/Room1 /Humidity. Wild card also supports for topics. For example: /Home/Room1/# means /Home/Room1/ all types of data received from /Home/Room1/.But in the lossy network and frequent transmission of data, long topics consume bandwidth. To resolve this issue MQTT-SN Client can send register message with long topic name to the gateway, gateway returns short topic id.MQTT-SN clients use the short topic id for subsequent publish messages.

IV. PROPOSED WORK AND EXPERIE- MENTAL SETUP

In this work the design, development and implementation of MQTT-SN based gateway is carried out. For practical verification of gateway, Sensor Node is implemented using node MCU IOT platform and MQTT-SN Publish/Subscribe tools.IOT Application is implemented in Python based patho eclipse MQTT framework. Mosquito MQTT Server also used for IOT Gateway integration works.

Hardware's and Software's are used in this research work.

DHT22: The DHT22 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin.

Node MCU: NodeMCU/ESP8286 is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Expressive Systems, and hardware which is based on the ESP-12 module. Node MCU&DHT22 Hardwares used for Sensor Node Implementation.

ArdinuoIDE: Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

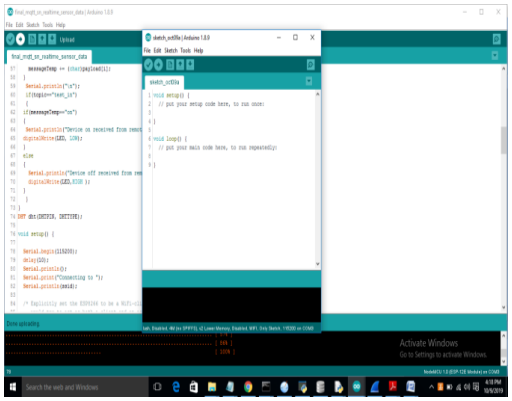


Fig **.4.1: Arduino IDE**

Rasberrypi-3:

The Raspberry Pi is a small computer that can do lot of work. Raspberry loaded with customized Linux Operating Systems. In this project Raspberry Pi used for IOT Gateway development, Integration and deployment.

Node-red:

Node-RED is a programming tool for wiring together hardware devices, APIs and online devices in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a sing le-click. In this project Node red used for publishing and subscribing the MQTT messages from MQTT Server.

Personal Computer computing Node:

In the Normal Intel PC (I3 Processor and 4GB Ram) Mosquito MQTT SERVER deployed and configured to connect IOT Gateway. Python IOT application developed to receive the real time sensor data on MQTT protocol from the MQTT-Server.

Programming Languages, Tools used in this work:

Linux, windows OS, C, C++, Python, Network programming software's, Linux multi threaded Programming and wire shark.

V.SENSOR NODE, MQTT-SN GATEWAY SOFTWARE ARCHITECTURE AND PYTHON APPLICATION DATA BASE SCHEMA

Sensor Node flow chart:

Sensor Node implemented on Node MCU Hardware platform, DHT22 Sensor (Humidity &Temperature). Arduino Prototype platform used for software code development and deployment.

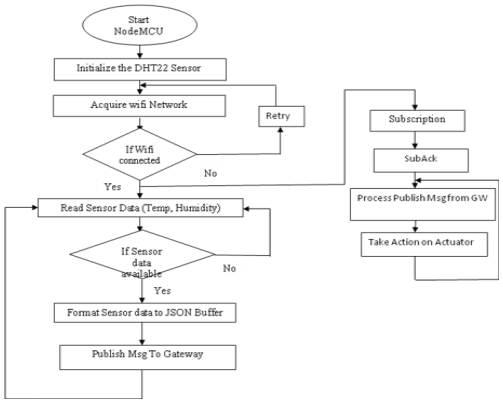


Fig.5.1: Sensor Node flow chart

IOT Gateway Node flow chart

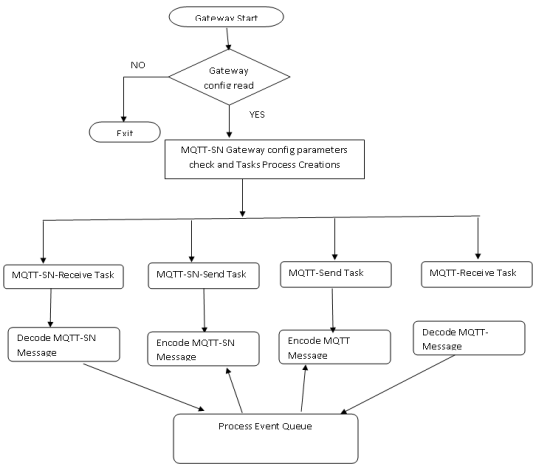


Fig.5.2: IOT Gateway flow chart

IOT Gateway development and deployment in Raspberry Pi-3 Model hardware board. Customized Linux Operating System is used for this work.

MQTT IOT APPLICATION:

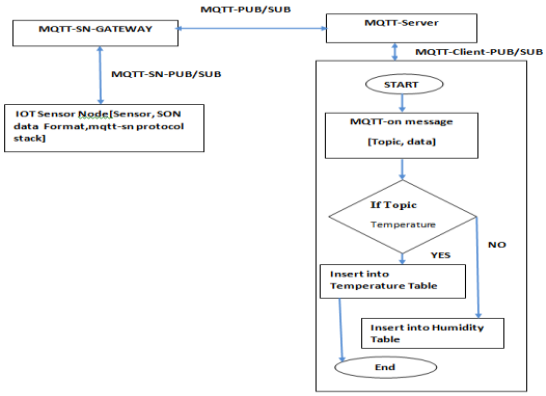


Fig.5.3 : MQTT IOT Application

Simple IOT application developed to receive Sensor data from MQTT Server using Python scripting Language. Sensor data stored in SQLlite Database for further uses cases.

IOT Application database schema (SQLite)

Create table DHT22_Temperature_Data

(Id integer primary key auto increment, Device text, Sensor Type text,Temperature_farin float, Temperature_celius float,Humidity float, Topic text ,Qos Integer, Date_n_Time text);

VI. SENSOR NODES, MQTT-SN GATEWAY, MQTT-SERVER, IOTAPPLICATIONLOGS

A. SENSOR NODE SAMPLE LOGS JSON DATA FORMAT:

06:48:24.367 -> Sending message to MQTTSN topic..

{"Device":"ESP32","SensorType":"Temperature_Humidity","Temperature_farin":79.16,"Temperature_celius":26.2,"Humidity":89,"time":88645}

06:48:24.367 -> MQTTSN_PUBLISH

07:25:16.332 -> MQTTSN_PUBLISH

07:25:16.332 -> Received - Topic: test_in Payload: off Lenght: 3 , 07:25:16.332 -> ON/OFF

B. MQTT-SN Gateway

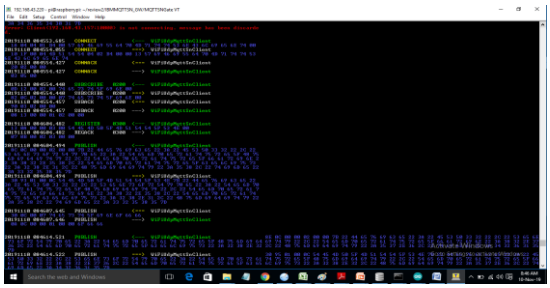


Fig.6.1:IOT Gateway Log window

Sequence of MQTT-SN and MQTT Messagesfor QOS-01:

S1: **MQTT**-SNconnect------→MQTT-SNGateway

S2: **MQTT CONNECT**----→**M**QTT-Server

S3: **MQTT-CONACK**<----MQTT-SNGateway

S4: **MQTT-SN CONACK**----→MQTT-SNSensor

S5: **MQTT-SNsubscribe**----→ MQTT-SN Gateway

S6: **MQTT SUBSCRIBE**----→MQTT-Server

S7: MQTT SUBACK <----MQTT-SN Gateway

S8: **MQTT-SN suback**----→ **MQTT-SN** Sensor Node

S9:MQTT-SNregister----→MQTT-SN Gateway

S10:**MQTT-SNREGACK**<----MQTT-SN Sensor

S11: MQTT-SN publish----→ MQTT-SNGW

S12: MQTT PUBLISH----→MQTT-Server

S13: MQTT **PUBACK----→**MQTT-SN Gateway

S14:MQTT-SNPUBACK----→ MQTT-SN Sensor Node

C. MQTT Server

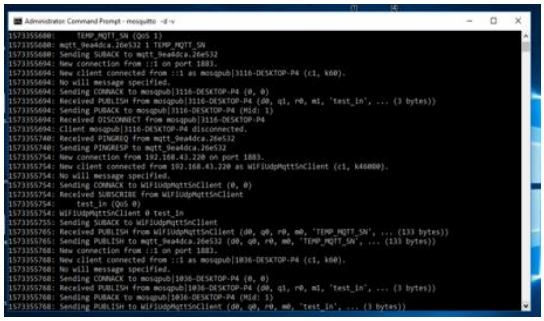


Fig.6.2: MQTT Server Log window

D.Node Red-Logs

{ topic: 'TEMP_MQTT_SN',

payload:

{ **Device: 'ESP32',**

SensorType: 'Temperature_Humidity',

Temperature_farin: 79.16,

Temperature_celius: 26.2,

Humidity: 89,

time: 32580 },

qos: 1,

retain: false,

_msgid: '1181609e.28732f' }

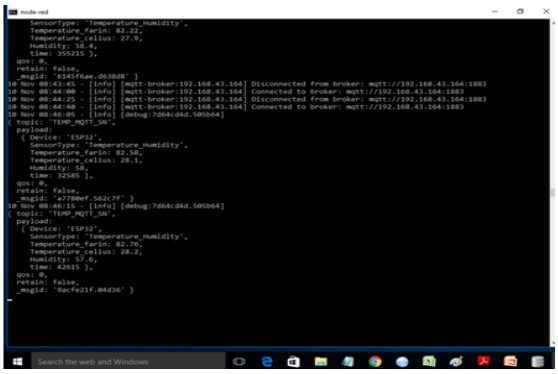


Fig.6.3: Node-Red Console window

Node-Red GUI (MQTT-SUB)

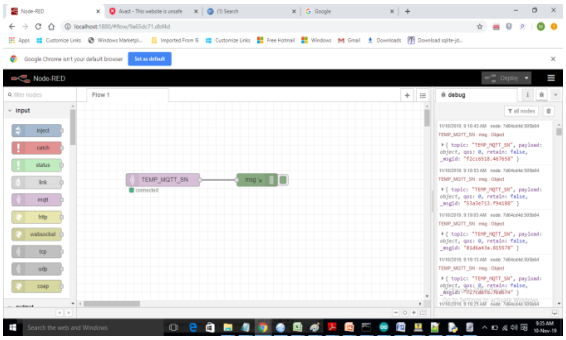


Fig.6.4:Node-Red MQTT-Sub

E.IOT Application Log:

{"Device":"ESP32","SensorType":"Temperature_Humidity","Temperature_farin":77.72,"Temperature_celius":25.4,"Humidity":91.8,"time":31455}

Enter DHT22_Temp_Data_Handler

SensorID :ESP32

SensorType :Temperature_Humidity

Temperature_farin : 77.72

Temperature_celius : 25.4

Humidity : 91.8

Data_and_Time: 21-Oct-2019 07:21:09:940631

Inserted Temperature Data into Database

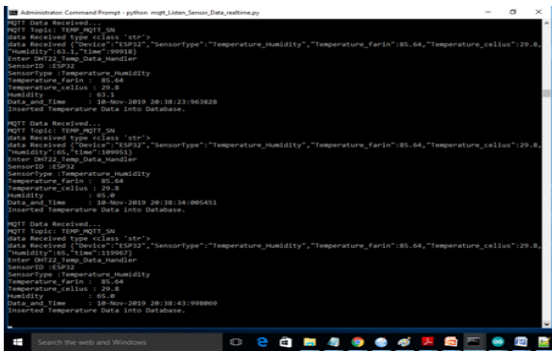


Fig.6.5: IOT Application Log

F.Actuator_on_off using MQTT-SN Pub client

1./mqtt-sn-pub-h192.168.43.220-p 10000 -t test_in -m "off" -q 1 -d

2.pi@raspberrypi:~/review2/mqtt-sn-tools-master $**./mqtt-sn-pub -h 192.168.43.220 -p 10000 –t test_in -m "on" -q 1 -d**

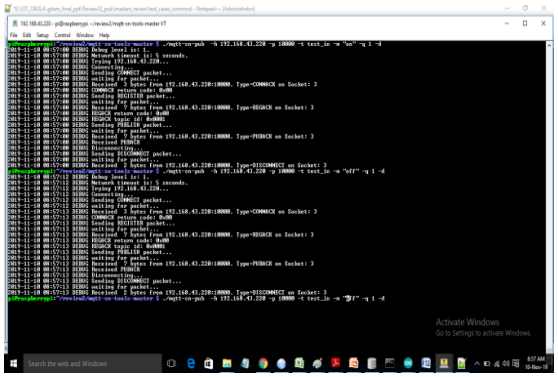


Fig.6.6:MQTT-SN Pub/Sub tools

G.Actuator_on_off using MQTT Pub client

C:\ProgramFiles\mosquitto\mosquitto_pub -t test_in -m "on" -d -q 1;

C:\Program Files\mosquitto>mosquitto_pub -t test_in -m "off" -d -q 1

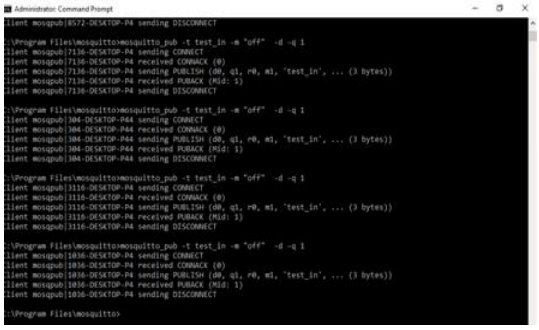


Fig.6.7:MQTT-Pub/Sub tools

H.**Sensor data values (Temp, Humidity & Device parameters)**

id	Device	SNType	Temp(F)	Temp©	Humidity	Date_n_Time
1	ESP32	TH	84.56	29.2	51.2	10-Nov-2019 09:19:03:308728
2	ESP32	TH	84.56	29.2	51.2	10-Nov-2019 09:19:13:316439
3	ESP32	TH	84.56	29.2	51.2	10-Nov-2019 09:19:23:416277
4	ESP32	TH	84.56	29.2	51.4	10-Nov-2019 09:19:33:460665
5	ESP32	TH	84.56	29.2	51.5	10-Nov-2019 09:19:43:458445
6	ESP32	TH	84.56	29.2	51.7	10-Nov-2019 09:19:53:467951
7	ESP32	TH	84.56	29.2	51.9	10-Nov-2019 09:20:03:453599
8	ESP32	TH	84.56	29.2	51.8	10-Nov-2019 09:20:13:470940
9	ESP32	TH	84.56	29.2	51.9	10-Nov-2019 09:20:23:493006
10	ESP32	TH	84.56	29.2	51.8	10-Nov-2019 09:20:33:522815
11	ESP32	TH	84.56	29.2	51.8	10-Nov-2019 09:20:43:519773
12	ESP32	TH	84.56	29.2	51.9	10-Nov-2019 09:20:53:537578
13	ESP32	TH	84.56	29.2	52	10-Nov-2019 09:21:03:546644
14	ESP32	TH	84.56	29.2	52	10-Nov-2019 09:21:13:597488
15	ESP32	TH	84.56	29.2	51.9	10-Nov-2019 09:21:23:584971
16	ESP32	TH	84.56	29.2	51.6	10-Nov-2019 09:21:33:602582
17	ESP32	TH	84.56	29.2	51.7	10-Nov-2019 09:21:43:631787
18	ESP32	TH	84.56	29.2	51.9	10-Nov-2019 09:21:53:619564

Fig.6.8: Sensor Data values and Device parameters

MQTT-Messages in Wire shark tool

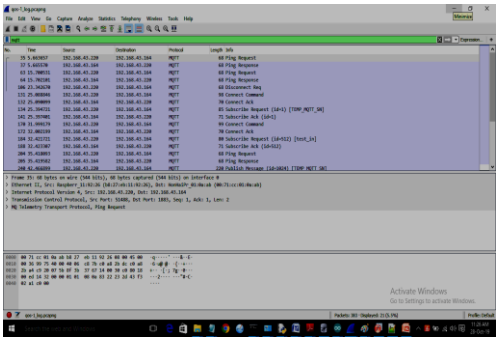


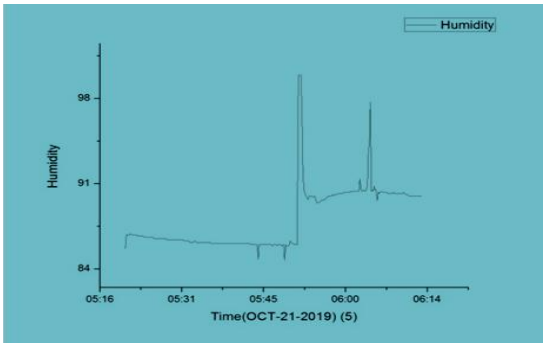
Fig.6.9 : Wire shark log

VII. RESULTS

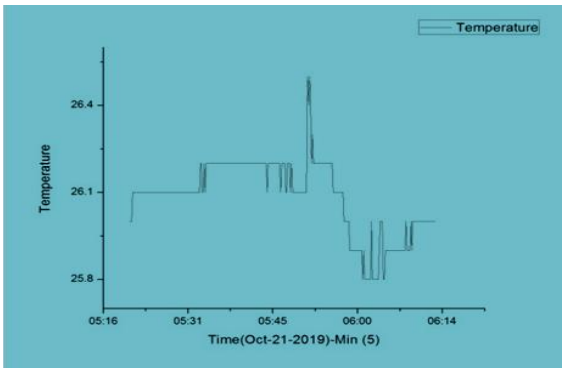
A. Temperature and Humidity Graphs:

In this research work, real-time sensor data is transmitted to the IOT application using light weight and more efficient MQTT-SN and MQTT Application messaging protocol. Sensor data stored in application database for further processing. Sample graphs are plotted for temperature and humidity for various time intervals.

Humidity data:



Temperature data:



VIII .CONCLUSION

In this paper, low cost MQTT-SN Gateway development and integration of MQTT Server for real time sensor data transmission to the IOT application has been discussed. In future, implementation of load balancing of MQTT-SN Gateway and integration of real-time IOT application with MQTT-SN IOT gateway shall be executed.

REFERENCES

[1] Urs Hunkeler, IHong Linh Truong, Andy Stanford-Clark "MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks"- 3rd International Conferences on Communication Systems Software and Middleware and workshops, 2008.

[2] Andre Gloria, Francisco Cercas, NunoSouto -"Design and implementation of an IoT gateway to create smart environments", in 8th International Conference on Ambient Systems, Networks and Technologies, Elsevier, pp.568–575, 2017.

[3] Guha Roy, D., Mahato, B., De, D., &Buyya, R. "Application-aware end-to-end delay and message loss estimation in Internet of Things (IoT) — MQTT-SN protocols". Future Generation Computer Systems, Vol 89, pp. 300–316, 2018

[4] Tantitharanukul, N., Osathanunkul, K., Hantrakul, K., Pramokchon, P.&Khoenkaw, P, "MQTT-Topics Management System for sharing of Open Data." in International Conference on Digital Arts, Media and Technology ", 2017.

[5] Naik.N, "Choice of effective messaging protocols for IoT systems:MQTT, CoAP, AMQP and HTTP", in 2017 IEEE InternationalSystemsEngineering Symposium, 2017.

[6] Zanella, A., Bui, N., Castellani, A., Vangelista, L., &Zorzi, M. "Internet of Things for Smart Cities.", IEEE Internet of Things Journal, Vol 1,pp. 22–32,2014.

[7] ESP8266 Arduino Core Documentation Release 2.4.0

[8] https://nodered.org/

[9]http://www.mqtt.org/new/wpcontent/uploads/2009/06/MQTT-SN_spec_v1.2.pdf

[10]http://mqtt.org/documentation/MQTT v3.1.1.pdf

[11]F.Jerald, M.Anand, N.Deepika," Design of an Industrial IOT Architecture Based on MQTT Protocol for End Device to Cloud Communication", in International Journal of Recent Technology and Engineering, Vol-7, ISSN: 2277-3878, 2019.

[12]Gopi Krishna, P., Sreenivasa Ravi, K., Hari Kishore, K., KrishnaVeni, K., N. Siva Rao, K., & D. Prasad, R. "Design and development of bi-directional IoT gateway using ZigBee and Wi-Fi technologies with MQTT protocol.", International Journal of Engineering & Technology, Vol-7,pp. 125-129,2018.

[13]Govindan, K., & Azad, A. P. "End-to-end service assurance in IoT MQTT-SN.", 12th Annual IEEE ConsumerCommunications and Networking Conference, 2015

[14]Amaran, M. H. Noh, N. A. M., Rohmad, M. S., &Hashim, H. "A comparison of Lightweight Communication Protocols in Robotic Applications.", Procedia Computer Science, Vol 76,pp. 400–405, 2015

[15] Guoqiang, S., Yanming,C.,Chao,Z.,&Yanxu, Z. "Design and Implementation of a Smart IoT Gateway." IEEE International Conferenceon Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, 2013

[16] https://www.sqlite.org

[17]https://www.python.org

18]http://www.steves-internet-guide.com /mqtt-sn

[19]https://www.eclipse.org/paho/components/mqtt-sn-transparent-gateway/

AUTHORS PROFILE

Mr. M.Obula Reddy *received his B.Tech degree in Electronics and Communication from JNTU University, Hyderbad, A.P. M.Tech from NIT, Calicut, Kerala. He has 11 Years of Industry experience in Telecom and Data communication protocol Engg.7 Year of Teaching experience in Engineering for UG&PG courses.*

Mr. M.Srinu *received his B.Tech degree in Electronics and Communication from JNTU University, Hyderbad, A.P. M.Tech from NIT, Calicut, Kerala. He has 10 Years of Teaching experience in Engineering for UG&PG courses.*