International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Real Time Sensor Data transmission to the IoT Applications using MQTT-SN and MQTT

M Obula Reddy, J B Seventline

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 layer and Application layer 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 Hospital. Numbers of IOT Application Protocols (MQTT, MQTT-SN, COAP, XMPP, and HTPP) are available and implemented in rich resource Environments like good computing power and bandwidth. However all Application protocols are not suitable under lossy wireless sensor network environment. In IOT Environment consists of Sensor Nodes, End Systems and related applications. Latency and bandwidth problems exist between sensor node and End Systems in two tier IOT Environment. To resolve this issue, we proposed Three Tier IOT Architecture; it consists of Sensor Node, IOT Gateway /Fog Computing Node, End Systems and Applications. In this paper we proposed flexible design, development and integration of IOT gateway for different IOT applications using MQTT and MQTT-SN Application Messaging Protocols.

Keywords* : *IoT, MQTT, MQTT-SN and WIRESHARK.

I. INTRODUCTION

Wireless sensor network (WSN) is a group of distributed embedded devices (nodes) with the capabilities of sensing, processing and transmission of data through wireless communications to the application. WSNs have emerged as a major research area which provides the interaction between the physical world and humankind. There have been developed various applications using WSNs, such as habitat monitoring, building monitoring, military systems, smart home etc. Traditional WSN networks are designed either proprietary or non standard Application protocols. Current Wireless Sensor network application protocols 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 these protocols are not suitable to lossy wireless sensor network due to constrained computing power and bandwidth. Need to Design, Development of light weight IOT application messaging protocol essential for transportation of real time and non real time data to IOT Applications. As per [1] MQTT-SN specifications are released for academic, research and industrial uses. MQTT-SN is Message Queue Telemetry Transport protocol for Sensor Networks. MQTT-SN protocol designed with adaptability of lossy WSN networks and as much as compatible with MQTT. As per [2], theoretical IOT application messaging protocols are compared in terms of protocol message over head, throughput and Bandwidth. As per [3], MQTT-SN end to end Quality Assurance theoretical frame work discussed, author suggested verification needed using eclipse patho embedded IOT gateway. In Section II IOT Architecture [Sensor Node, IOT Gateway, MQTT Server, and IOT Application], Section III IOT application protocol stack and Message sequence diagram, MQTT-SN messaging format and Important MQTT-SN messages description, MQTT-SN Qos models, MQTT-SN Topic management, Section IV Related Work, Experimental Setup, Trace logs for all Nodes (Sensor Device, IOT Gateway, MQTT Server, IOT Application).In Section V Results and Conclusions are discussed.

Manuscript published on November 30, 2019.
* Correspondence Author
M Obula Reddy*, Research Scholar, Gitam University, Vishakhapatnam, India. Email: moreddy2003@gmail.com
J B Seventline, Professor, Gitam University, Vishakhapatnam, India. Email: seventline.joseph@gmail.com

II. IOT ARCHITECTURE

IoT architecture diagram shown in Fig-01 and consists of Sensor Node, Actuator, IOT Gateway, MQTT Server and IoT Applications.

Sensor Node: Sensor Node consists of Sensor data collector, JSON Data formatter, MQTT-SN application Pub/Sub protocol stack. In this project Node MCU/ESP82826 hardware platform used to communicate with the MQTT-SN Gateway.

Actuator Node: Actuator Node subscribe to the required topic with Gateway and Server. Application send relevant commands 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 relevant action.

IOT Gateway: IOT gateway/Fog Computing Node consist of MQTT-SN protocol Encoder/Decoders, Protocol state machine, Event Queue ,MQTT protocol Encoder/Decoder, Protocol state machine. Basically it converts the MQTT-SN protocol to MQTT protocol and communicates with the Sensor Node platform and MQTT Server.

It is designed and developed on multithreaded environment under Linux Operating Systems.

MQTT Server: MQTT is light weight protocol and implemented for Machine to Machine Communication and in IOT backend systems deployed in number of Applications. For example Amazon Web services, IBM Watson IOT platform, Microsoft Azure platform.

IoT Application: IoT Application is any application as per requirement. In this Project Temperature, Humidity monitoring and stored in SQL Lite database.

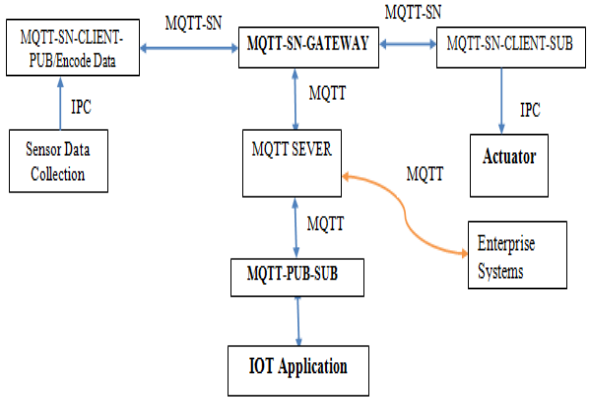

Fig.1 IoT Architecture

III. IoT PROTOCOL STACK AND MQTT-SN MESSAGE FLOWS

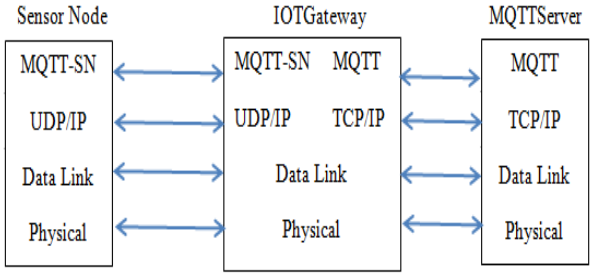

Fig.2 IoT Protocol Stack Diagram

MQTT-SN Architecture:

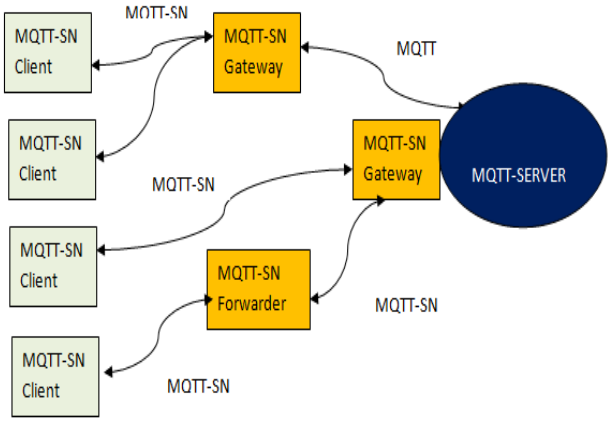

Fig.3 MQTT-SN Architecture

MQTT-SN Architetcure consists of MQTT-SN Clients, MQTT-SN Forwarder, MQTT-SNGateway.

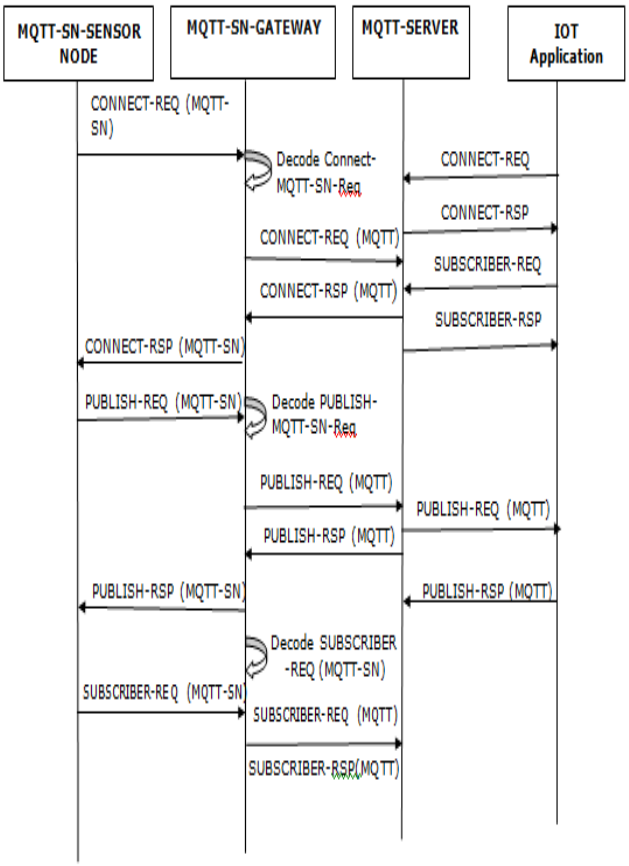

Fig.4 MQTT-SN QOS-01 Message Flow

MQTT-SN differences from MQTT: Connect message split into three messages two are optional and are used for the will message

- Topic id's used in place of topic names.
- Short Topic names
- Pre-defined topics.
- Discovery process to let clients discover the Gateway
- Will Topic and messages can be changes during the session
- Offline keep alive procedure for sleeping clients

MQTT-SN General Message Format:

Message Header (2 or 4 octets)	Message Variable Part (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.5 MQTT-SN Message Table

MQTT-SN Qos Model:

QOS	Description
QOS-0	messages are delivered at most once(may be messages are lost)
QOS-1	messages are delivered 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 publishes the data to the subscribed topics of subscribers. Topic name is any time of string in the MQTT. But it is not standardized, Application can chose desired topic name as per requirement. For ex Topic Name :/Home/Room1/Temperature :/Home/Room1/Humidity Wild card topic also supports For example: /Home/Room1/# from the /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, gate returns the short topic id. MQTT-SN clients use the short topic id for subsequent publish messages

IV. EXPERIMENTAL SETUP

Hardware's and Software are used in this project.

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 (no analog input pins needed).

Fig.6 DHT22

Node MCU: Node MCU 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.

Fig.7 Node MCU

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.

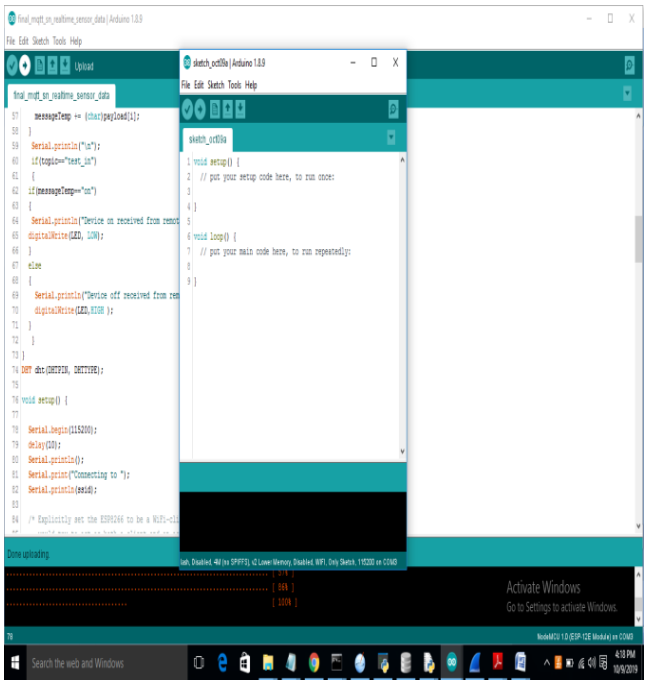

Fig.8 Arduino IDE

Rasberrypi-3:

Fig.9 Rasberrypi-3

The Raspberry Pi is a small computer that can do lot of work. In this project Raspberry Pi used for IOT gate way 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.

Programming Languages &Tools used in this project: 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, ALGORITHM AND PYTHON APPLICATION DATA BASE SCHEMA

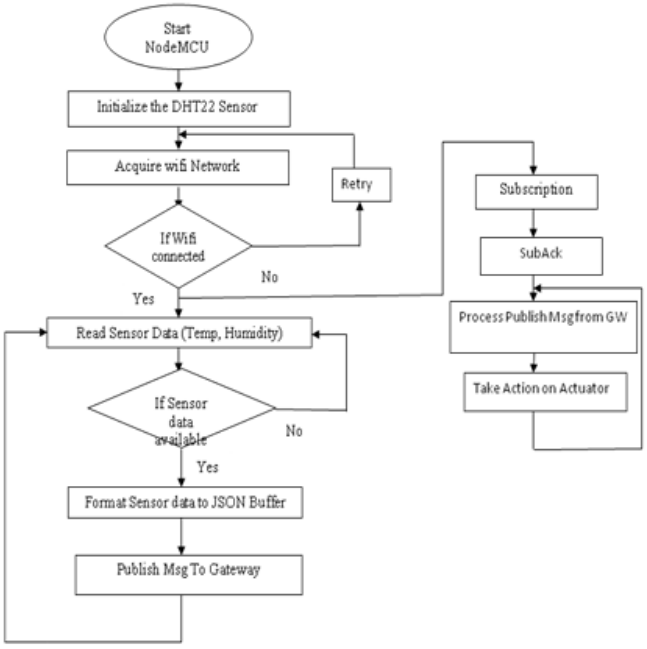

Fig.10 Sensor Node Flow Chart

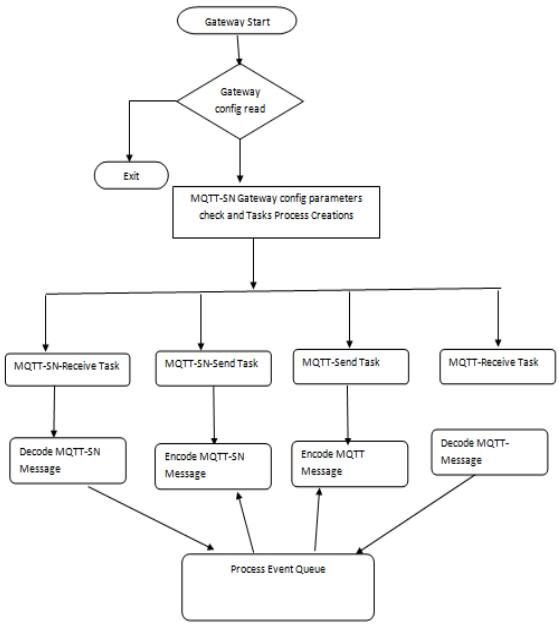

Fig.11 Gateway Flow Chart

MQTT IoT Application

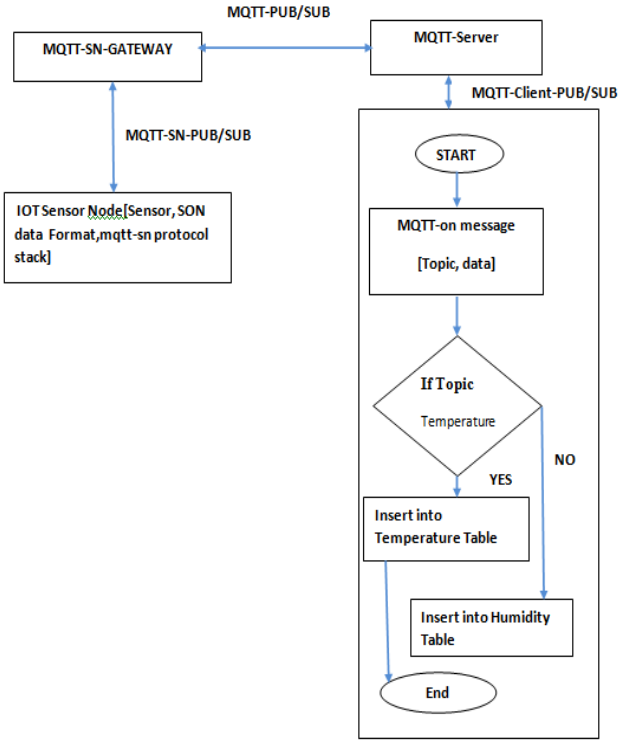

Fig.12 MQTT IoT Application

IOT Application database schema defined as follows:

DHT22_Temperature_Humidity_Data;
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, IoTAPPLICATION LOGS

A. SENSOR NODE LOGS

06:47:15.969 -> Connecting to Obula(Wifi SSID)
06:47:15.969 -> .....
06:47:17.937 -> WiFi connected
06:47:17.937 -> IP address:
06:47:17.937 -> 192.168.43.157
06:47:17.937 -> Starting MqttSnClient - DHTxx test!
06:47:17.984 -> ready!
06:47:17.984 -> MQTT-SN Gateway device_address: 192, 168, 43, 220, 39, 16Could not connect MQTT-SN Client.
06:47:58.951 -> MQTT-SN Gateway device_address: 192, 168, 43, 220, 39, 16MQTT-SN Client connected.
06:48:24.367 -> Sending message to MQTTSN topic..
06:48:24.367 -> **{"Device":"ESP32","SensorType":"Temperature_Humidity","Temperature_farin":79.16,"Temperature_celius":26.2,"Humidity":89,"time":88645}**
06:48:24.367 -> MQTTSN_PUBLISH
07:24:52.369 -> Device on received from remote web
07:25:16.332 -> MQTTSN_PUBLISH
07:25:16.332 -> Received - Topic: test_in Payload: off Length: 3
07:25:16.332 -> off
07:25:16.332 ->
07:25:16.332 -> Device off received from remote web
07:25:31.886 -> MQTTSN_PUBLISH
07:25:31.932 -> Received - Topic: test_in Payload: on Length: 2
07:25:31.932 -> on

B. MQTT-SN Gateway Logs:

MQTT-SN –QOS -01:

./Build/MQTT-SNGateway gateway.conf
20191021 063918.876 PahoGateway-01 has been started.
ConfigFile: ./gateway.conf
PreDefFile: ./predefinedTopic.conf
SensorN/W: UDP Multicast 225.1.1.1:1883 Gateway Port 10000
Broker: 192.168.43.164 : 1883, 8883
RootCApath: (null)
RootCAfile: (null)
CertKey: (null)
PrivateKey: (null)

20191021 064811.625 **CONNECT** <--- WiFiUdpMqttSnClient -→**Sensor node to IOTGW**
20191021 064811.629 **CONNECT** ===> WiFiUdpMqttSnClient →**IOT GW to MQTTServer** 6E 43 6C 69 65 6E 74
20191021 064811.931 **CONNACK** <=== WiFiUdpMqttSnClient →**MQTT Server to IOTGW**
20191021 064811.932 **CONNACK** ---> **WiFiUdpMqttSnClient** -→**IOTGW to Sensor node**
20191021 064811.992 **SUBSCRIBE** 0200 <--- WiFiUdpMqttSnClient →**Sensor node to IOTGW**
20191021 064811.993 **SUBSCRIBE** 0200 ===> WiFiUdpMqttSnClient →**IOTGW to MQTTServer** 82 0C 02 00 00 07 74 65 73 74 5F 69 6E 01
20191021 064812.009 **SUBACK** 0200 <=== WiFiUdpMqttSnClient -→**MQTTServer to IOTGW** 90 03 02 00 01
20191021 064812.009 **SUBACK** 0200 ---> WiFiUdpMqttSnClient →**IOTGW to SensorNode** 08 13 20 00 01 02 00 00
20191021 064822.035 **REGISTER** 0300 <--- WiFiUdpMqttSnClient -→**Sensor Node to IOTGW** 13 0A 00 00 03 00 54 45 4D 50 5F 4D 51 54 54 5F 53 4E 00
20191021 064822.035 **REGACK** 0300 ---> WiFiUdpMqttSnClient -→**IOTGW to Sensor Node**
20191021 064822.042 **PUBLISH** 0400 <--- WiFiUdpMqttSnClient -→**sensor node to IOTGW**
20191021 064822.043 **PUBLISH** 0400 ===> WiFiUdpMqttSnClient →**IOTGW to MQTTServer**
20191021 064822.049 **PUBACK** 0400 <=== WiFiUdpMqttSnClient --→**MQTTServer to IOTGW**
20191021 064822.050 PUBACK 0400 ---> WiFiUdpMqttSnClient -→**IOTGW to Sensornode**

C. MQTT Server Logs

C:\Program Files\mosquitto>mosquitto -d -v
1571619629: Warning: Can't start in daemon mode in Windows.
1571619629: mosquitto version 1.5.8 starting
1571619629: Using default config.
1571619629: Opening ipv6 listen socket on port 1883.
1571619629: Opening ipv4 listen socket on port 1883.
1571619685: New connection from 192.168.43.164 on port 1883.
1571619685: New client connected from 192.168.43.164 as `TVTL@wnWhhPpOhZ[N0j]w[4nj1`kX4:LuDD;iuj1@3EC:K\txSo3MsGV3uUs_sw (c1, k45).
1571619685: No will message specified.
1571622192: New client connected from 192.168.43.220 as WiFiUdpMqttSnClient (c1, k46080).
1571622192: No will message specified.
1571622192: Sending CONNACK to WiFiUdpMqttSnClient (0, 0)
1571622192: Received SUBSCRIBE from WiFiUdpMqttSnClient
1571622192: test_in (QoS 1)
1571622192: WiFiUdpMqttSnClient 1 test_in
1571622192: Sending SUBACK to WiFiUdpMqttSnClient
1571622202: Received PUBLISH from WiFiUdpMqttSnClient (d0, q1, r0, m1024, 'TEMP_MQTT_SN', ... (135 bytes))
1571622202: Sending PUBACK to WiFiUdpMqttSnClient (Mid: 1024)

D.Node Red-Logs:

21 Oct 06:44:52 - [info] [debug:7d64cd4d.505b64]
{ 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' }
21 Oct 06:48:24 - [info] [debug:7d64cd4d.505b64]
{ topic: 'TEMP_MQTT_SN',
payload:
{ Device: 'ESP32',
SensorType: 'Temperature_Humidity',
Temperature_farin: 79.16,
Temperature_celius: 26.2,
Humidity: 89,
time: 88645 },
qos: 1,
retain: false,
_msgid: 'c6c7e154.37447' }

E.IoT Application Log:

Connected with result code 0
Subscribing topic
MQTT Data Received...
MQTT Topic: TEMP_MQTT_SN
data Received type <class 'str'>
data Received {"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

F.Mqtt-sn-tools _on_off:

./mqtt-sn-pub -h 192.168.43.220 -p 10000 -t test_in -m "off" -q 1 -d

2019-10-21 07:25:13 DEBUG Debug level is: 1.
2019-10-21 07:25:13 DEBUG Network timeout is: 5 seconds.
2019-10-21 07:25:13 DEBUG Trying 192.168.43.220...
2019-10-21 07:25:13 DEBUG Connecting...
2019-10-21 07:25:13 DEBUG Sending CONNECT packet...
2019-10-21 07:25:13 DEBUG waiting for packet...
2019-10-21 07:25:13 DEBUG Received 3 bytes from 192.168.43.220:10000. Type=CONNACK on Socket: 3
2019-10-21 07:25:13 DEBUG CONNACK return code: 0x00
2019-10-21 07:25:13 DEBUG Sending REGISTER packet...
2019-10-21 07:25:13 DEBUG waiting for packet...
2019-10-21 07:25:13 DEBUG Received 7 bytes from 192.168.43.220:10000. Type=REGACK on Socket: 3
2019-10-21 07:25:13 DEBUG REGACK return code: 0x00
2019-10-21 07:25:13 DEBUG REGACK topic id: 0x0001
2019-10-21 07:25:13 DEBUG Sending PUBLISH packet...
2019-10-21 07:25:13 DEBUG waiting for packet...
2019-10-21 07:25:13 DEBUG Received 7 bytes from 192.168.43.220:10000. Type=PUBACK on Socket: 3
2019-10-21 07:25:13 DEBUG Received PUBACK
2019-10-21 07:25:13 DEBUG Disconnecting...
2019-10-21 07:25:13 DEBUG Sending DISCONNECT packet...
2019-10-21 07:25:13 DEBUG waiting for packet...
2019-10-21 07:25:13 DEBUG Received 2 bytes from 192.168.43.220:10000. Type=DISCONNECT on Socket: 3
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

2019-10-21 07:25:29 DEBUG Debug level is: 1.
2019-10-21 07:25:29 DEBUG Network timeout is: 5 seconds.
2019-10-21 07:25:29 DEBUG Trying 192.168.43.220...
2019-10-21 07:25:29 DEBUG Connecting...
2019-10-21 07:25:29 DEBUG Sending CONNECT packet...
2019-10-21 07:25:29 DEBUG waiting for packet...
2019-10-21 07:25:29 DEBUG Received 3 bytes from 192.168.43.220:10000. Type=CONNACK on Socket: 3
2019-10-21 07:25:29 DEBUG CONNACK return code: 0x00
2019-10-21 07:25:29 DEBUG Sending REGISTER packet...
2019-10-21 07:25:29 DEBUG waiting for packet...
2019-10-21 07:25:29 DEBUG Received 7 bytes from 192.168.43.220:10000. Type=REGACK on Socket: 3
2019-10-21 07:25:29 DEBUG REGACK return code: 0x00
2019-10-21 07:25:29 DEBUG REGACK topic id: 0x0001
2019-10-21 07:25:29 DEBUG Sending PUBLISH packet...
2019-10-21 07:25:29 DEBUG waiting for packet...
2019-10-21 07:25:29 DEBUG Received 7 bytes from 192.168.43.220:10000. Type=PUBACK on Socket: 3
2019-10-21 07:25:29 DEBUG Received PUBACK
2019-10-21 07:25:29 DEBUG Disconnecting...
2019-10-21 07:25:29 DEBUG Sending DISCONNECT packet...
2019-10-21 07:25:29 DEBUG waiting for packet...

2019-10-21 07:25:29 DEBUG Received 2 bytes from 192.168.43.220:10000. Type=DISC ONNECT on Socket: 3

20191021 064822.050 PUBACK 0400 ---> WiFiUdpMqttSnClient ->IOTGW to Sensornode

G. Actuator_on_off using MQTT Pub client

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

Client mosqpub|6172-DESKTOP-P4 sending CONNECT

Client mosqpub|6172-DESKTOP-P4 received CONNACK (0)

Client mosqpub|6172-DESKTOP-P4 sending PUBLISH (d0, q1, r0, m1, 'test_in', ... (2 bytes))

Client mosqpub|6172-DESKTOP-P4 received PUBACK (Mid: 1)

Client mosqpub|6172-DESKTOP-P4 sending DISCONNECT

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

Client mosqpub|9244-DESKTOP-P4 sending CONNECT

Client mosqpub|9244-DESKTOP-P4 received CONNACK (0)

Client mosqpub|9244-DESKTOP-P4 sending PUBLISH (d0, q1, r0, m1, 'test_in', ... (3 bytes))

Clientmosqpub|9244-DESKTOP-P4 received PUBACK (Mid: 1)

Client mosqpub|9244-DESKTOP-P4 sending DISCONNECT

MQTT-Messages in Wire shark tool:

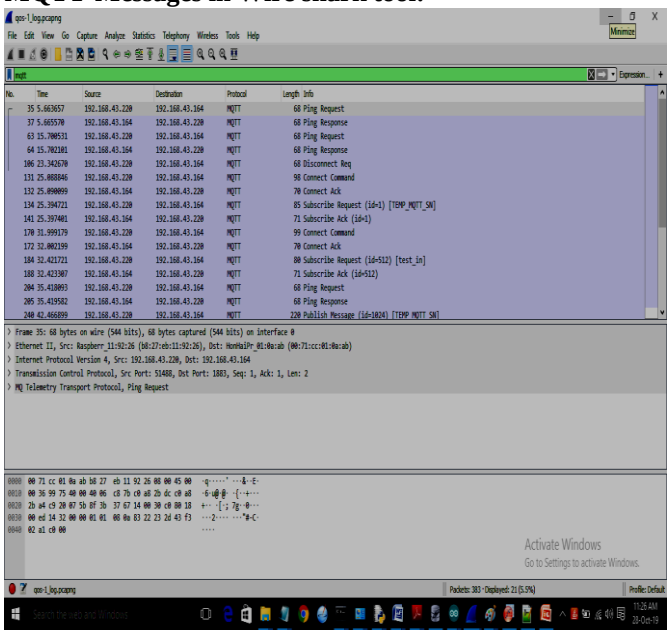

Fig.13 Wire shark log

VII. RESULT AND DISCUSSION

In this project, real-time sensor data transmitted to IOT application and stored in application database. Sample graphs are plotted for temperature and humidity.

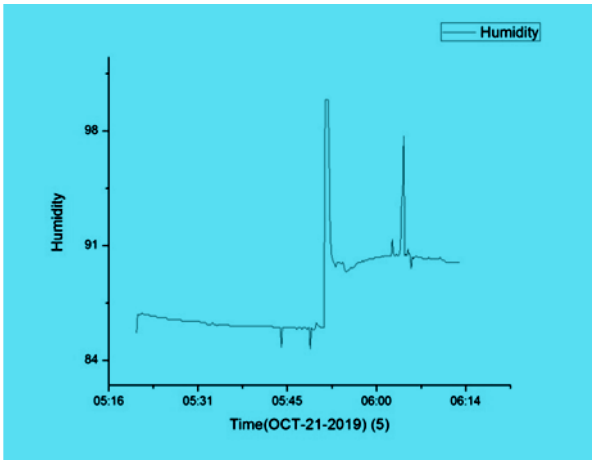

Fig.14 Humidity Data

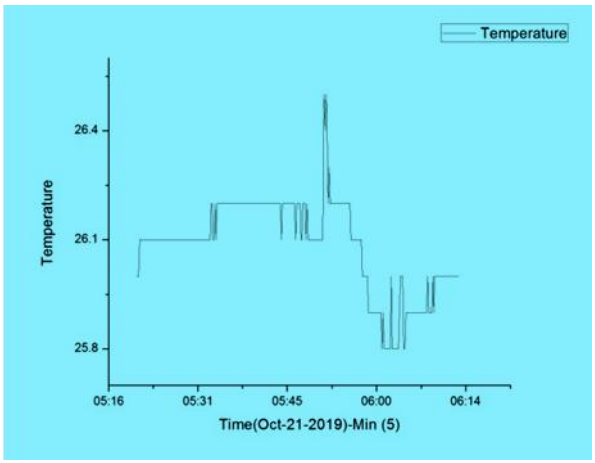

Fig.15 Temperature Data

VIII. CONCLUSION

In this project, real time sensor data transmitted to the IOT Application using MQTT, MQTT-SN Application protocols. In future we will implement Load balancing of MQTT-SN Gateway and integration of real-time IOT application with IOT gateway.

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, Nuno Souto -"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 International Systems Engineering 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. 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.
8. 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.
9. Govindan, K., & Azad, A. P. "End-to-end service assurance in IoT MQTT-SN.", 12thAnnual IEEE Consumer Communications and Networking Conference, 2015.
10. Amaran, M. H. Noh, N. A. M., Rohmad, M. S., &Hashim, H. "A comparison of Light weight Communication Protocols in Robotic Applications.", Procedia Computer Science, Vol 76, pp. 400–405, 2015.
11. Guoqiang, S., Yanming,C.,Chao,Z.,& Yanxu , Z. "Design and Implementation of a Smart IoT Gateway." IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, 2013.

AUTHORS PROFILE

Mr. M.Obula Reddy received his B.Tech degree in Electronics and Communication from JNTU University,Hyderabad,A.P. M.Tech fromNIT, 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.

J.B.Seventline received B.Tech degree in ECE from Bharathiar University, Coimbatore. TN.ME from Madurai kamaraj University, Madurai,TN and Ph.D in Radar Signal Processing from Andhra University, Visakhapatnam. She has 26 years of teaching experience and presently working as Professor of ECE, GITAM deemed to be University, Visakhapatnam. She has almost 50 technical papers published in reputed journals and conferences. Her research interests include Radar Signal Processing, Image Processing, VLSI Signal Processing and Internet of things application messaging protocols.