International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-6, April 2020

Communication Protocol in Internet of Things

R. Chawngsangpuii, Prodipto Das

Abstract*: Constrained devices are commonly used in the Internet of Things systems. Since these devices have limited communication and computation resources, communication protocols which are lightweight are needed. A lightweight protocol called Message Queue Telemetry Transport, which is a publish/subscribe messaging protocol, is utilized with the constrained devices. Hence, this paper is aimed at monitoring data by using machine-to-machine communication protocol with the help of an IoT device, Raspberry Pi.

***Keywords*: Communication protocols, Internet of Things, Message Queue Telemetry Transport, MQTT.**

I. INTRODUCTION

In most recent developments in the fields such as Internet protocols, Radio-Frequency Identification(RFID) technology, smart sensors, and communication technologies, the Internet of Things has found its extensive use. By linking the different technologies by means of joining the physical objects collectively, IoT in the near future is expected to bring many innovative application which support smart decision making.

IoT takes advantage of the primary technologies of the physical objects to renovate it from traditional devices to more intelligent devices. Therefore, the IoT should contribute enormously to the growth of the world economy and to the improvement of the value of life.

The IoT has the capability to transmit data across a network without the need to interact with other objects, such as humans or other computers. Kevin Ashton first coined this concept in 1999. Internet of Objects is another name for IoT and it consists of self-configuration network and wireless communication between the different objects. It offers a platform for smart devices to electronically communicate with the rest of the world. IoT has become a description of the numerous research disciplines and technologies that enable the Internet to communicate with physical objects in the real world [1, 2].

"Message Queuing Telemetry Transport (MQTT)" is presented in this document, a largely used protocol. It is used by eminent companies like Facebook and Amazon in recent times.

Revised Manuscript Received on April 30, 2020.
***** Correspondence Author
R. Chawngsangpuii*, Department of Information Technology, Mizoram University, Aizawl, India. Email: sangpuii_77g@hotmail.com
Dr. Prodipto Das, Department of Computer Science, Assam University, Silchar, India. Email: prodiptodas@gmail.com

It is exceedingly simple and extremely lightweight publication / subscription protocol, which is designed for devices with restrictions and low bandwidth networks, at high latency or unreliable. The design principles are to minimize device resource requirements and network bandwidth while trying to guarantee reliability and a certain degree of security of delivery. These principles earn the protocol perfect solution for the rising "machine to machine" (M2M) or "Internet of things" of connected-devices and also for mobile applications in which battery power and bandwidth are very important [3].

In 1999, Dr Andy Stanford-Clark of IBM, and Arlen Nipper of Arcom invented MQTT. Hence, it is a "standardized network protocol" wherein transmission of short messages/commands is possible.

II. RELATED WORK

Thangavel, Dinesh, et al in [4] designed and implemented a common "extensible middleware" to sustain upcoming protocols for instance CoAP and MQTT. Experiments for examining the performance of CoAP and MQTT with the common middleware was conducted by them, in which end–to–end delay and bandwidth consumption are observed. The outcome from their experiment illustrates that MQTT messages contain higher delay with higher loss rate than CoAP messages. Furthermore, the outcome also provides evidence that when comparing with the CoAP messages, the MQTT proves to have lower-delay in the lower-packet.

The authors in [5] employed two application-tests for obtaining a complete study about the performance of IoT Web application. The first application utilized various message encoding and communication protocols to measure graphics rendering performance and latencies while doing performance comparison to implement diverse web-platforms. Latency of the "message throughput rate" and "sensor data message delivery" were calculated in the latter application. This was performed so as to facilitate in comparing the performance of the various messaging protocols in IoT like DDS, XMPP, AMQP and MQTT. The outcome from the test illustrated that HTML5 platform had "higher capability" for running IoT Web-applications during real-time and that the "optimum messaging protocol" is MQTT on behalf of wide-range of applications of IoT mWeb.

The authors in [6] evaluated performance of Constrained Application Protocol(CoAP), WebSocket, and for Message Queuing Telemetry Transport(MQTT). Their comparison implemented the three protocols appropriately for IoT-applications in the same hardware-platform of lower complexity and cost.

The efficiency of the protocol depended on the average RoundTrip Time (RTT) and also overhead performance. For all IoT device scenarios,

the server is being "transmitted" with data and replies are awaited.

HARDWARE DESCRIPTION

The Raspberry Pi 3 Model B consists of a Broadcom BCM2837 64bit ARMv7 QuadCore Processor powered SingleBoard Computer running at 1.2GHz(Fig. 1).

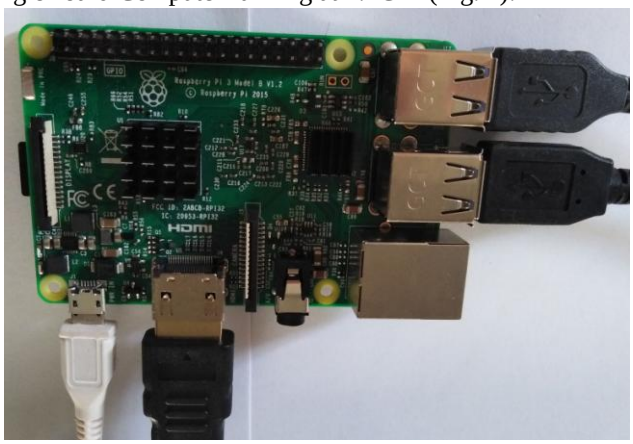

Fig. 1 The Raspberry Pi 3 Model B board

It has 40 pins extended GPIO, four USB2.0 ports, audio jack, 1 GB of RAM memory, one 100 Mbps Ethernet port and one full size HDMI port. As the Raspberry Pi is not equipped with internal storage device, MicroSD Card is required for storing the program files and operating system.

A. Raspberry Pi

It is a very low-cost computer card/board that is the size of a credit card that connects to a computer or television monitor, able to do everything a desktop computer can do and uses a standard mouse and keyboard. Raspberry Pi Foundation in the United Kingdom developed the board with the aim of promoting basic PC education in schools [7]. It is manufactured in two board configurations.

The most recent product comes in the form of Raspberry Pi 3 Model B+ range. This model has a quad-core processor of 1.4 GHz and 64-bit, dual-band wireless LAN, faster Ethernet, Bluetooth 4.2 / BLE, and Ethernet power support (with independent HAT PoE) [8].

Table- I: Hardware Specification of Raspberry Pi 3

System on Chip	Broadcom BCM2837B0
CPU Type/Speed	ARM Cortex-A53 1.4GHz
RAM size	1GB SRAM
Integrated Wi–Fi	2.4GHz and 5GHz
Ethernet speed	300Mbps
PoE	Yes
Bluetooth	4.2

III. SOFTWARE TOOLS

To implement machine-to-machine communication with the raspberry pi, various software tools are used such as Rasbian, Putty, VNC Viewer, Python and paho-mqtt.

A. Rasbian

It is based on Debian, a free operating system enhanced for Raspberry Pi hardware. Raspberry Pi needed operating system to make its system work. Raspbian is much more than a just an operating system as it accommodates packages more than 35,000. These packages are software used for easy installation of the Raspberry Pi [9].

B. PuTTY

PuTTY is used to access the Raspberry Pi in wireless mode by typing in the IP address of Raspberry Pi. It is a telnet and SSH, originally developed for the Windows platform by Simon Tatham. A group of volunteers developed and supported PuTTy. It is open-source software accessible with the source code. [10].

C. VNC Viewer

It is a free remote connection application presented under remote desktop software and is available from RealVNC Limited for Windows. It is licensed as freeware for the Windows operating system/platform (32 bit and 64 bit) from remote desktop software without restrictions [11].

D. Python Language

Python is a high-level interpreted language for generic programming that is also used for scripting. It was created by Guido van Rossum and launched for the first time in 1991. It was recommended by Raspberry Pi Foundation to use Python as a language for students. Also Pi in Raspberry Pi comes from the name of the Python programming language [12].

E. Paho MQTT

The MQTT-SN messaging protocols and open-source client implementations of MQTT are provided by Eclipse Paho project. The project is intended for the various applications of the Internet of Things(IoT) [13].

IV. RESULT AND DISCUSSION

Reducing the amount of data transferred between smart devices is very important with Internet of Things. There are more and more objects getting connected to our local networks every day. These objects getting connected are often having little resources, hence it is vital to bound the data getting transferred to the bare minimum. However, this limitation finds the MQTT protocol very useful. With MQTT protocol, one device is used as a “broker”, which acts like a master in the network and can send /route the messages to the intended /correct devices. All the other devices may “publish and/or subscribe” to “topics”, some types of message. In a publish and subscribe(pub-sub) system, any device can “publish” a message on a topic, or can “subscribe” to a particular topic in order to receive messages If a device publishes any message on a particular “topic”, the broker can transport /transfer the message to all those devices that subscribed to the “topic”. The use of this architecture greatly reduced resources needed on the devices as well as the amount of data being transferred.

We update the signing key by downloading the key before installing Mosquitto via the apt-get command:

```
sudo wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
```

We add the repository since Mosquitto package is not present in basic Debian repositories. Going to the folder which has the repository lists is done and the repository file for Mosquitto should be downloaded:

```
sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list
```

Installing the package mosquitto, MQTT broker, and mosquitto-clients is done which will be used for testing the installation.

```
sudo apt-get install mosquitto mosquitto-clients
```

It is required to install "Paho MQTT", which will replace the old "Mosquitto Python" module, if MQTT is used in a Python project. Hence, we install python-pip in our Raspberry Pi if it is not done earlier:

```
sudo apt-get install python-pip
sudo pip install paho-mqtt
```

A. Configuring Mosquitto

Path for the configuration file is found in the path /etc/mosquitto/mosquitto.conf. Hence, we make a backup of this file before making any changes to it:

```
cd /etc/mosquitto
sudo cp mosquitto.conf mosquitto.conf.original
```

B. Testing Mosquitto

We can use two terminals where the first terminal can be used for subscribing to a topic "hello/world". The second terminal can be used for publishing message to the said topic. We have successful test when the publisher's message can get recorded on the other terminal (i.e., subscriber's terminal).

In the first terminal, we execute the mosquitto_sub command in order to start the subscriber. The option -d enables the debug mode, and the option -t specifies the topic:

```
mosquitto_sub -d -t hello/world
```

```
pi@raspberrypi:~ $ mosquitto_sub -d -t hello/world
Client mosqsub/2904-raspberryp sending CONNECT
Client mosqsub/2904-raspberryp received CONNACK
Client mosqsub/2904-raspberryp sending SUBSCRIBE (Mid: 1, Topic: hello/world, Qo
S: 0)
Client mosqsub/2904-raspberryp received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/2904-raspberryp received PUBLISH (d0, q0, r0, m0, 'hello/world',
... (29 bytes))
Hello from Terminal window 2!
```

We run the mosquitto_pub command with option -d for enabling debug mode, and option -t for specifying the topic. This command is used for publishing a message in which message is to be preceded by the option –m:

```
mosquitto_pub -d -t hello/world -m "Hello from Terminal window 2!"
```

```
pi@raspberrypi:~ $ mosquitto_pub -d -t hello/world -m "Hello from Terminal windo
w 2!"
Client mosqpub/2905-raspberryp sending CONNECT
Client mosqpub/2905-raspberryp received CONNACK
Client mosqpub/2905-raspberryp sending PUBLISH (d0, q0, r0, m1, 'hello/world',
.. (29 bytes))
Client mosqpub/2905-raspberryp sending DISCONNECT
pi@raspberrypi:~ $
```

At this time, the test message "Hello from Terminal window 2!" appears in the first terminal if everything works fine.

C. Testing data exchange between Raspberry Pi MQTT and Python

For this purpose, the receiver is started which is followed by creation of new file with the content given as under:

```
sudo nano mqtt_subscriber.py
```

```
import paho.mqtt.client as mqtt
MQTT_SERVER = "localhost"
MQTT_PATH = "test_channel"

# The callback for when the client receives a CONNACK
#response from the server.
def on_connect(client, userdata, flags, rc):
 print("Connected with result code "+str(rc))

 # Subscribing in on_connect() means that if we lose the
 # connection and reconnect then subscriptions will be
# renewed.
 client.subscribe(MQTT_PATH)

# The callback for when a PUBLISH message is received #
from the server.
def on_message(client, userdata, msg):
 print(msg.topic+" "+str(msg.payload))
 # more callbacks, etc

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

client.connect(MQTT_SERVER, 1883, 60)

# Blocking call that processes network traffic, dispatches #
callbacks and handles reconnecting.
# Other loop*() functions are available that give a
# threaded interface and a manual interface.
client.loop_forever()
```

The code for the publisher is written in Python as below:

```
import paho.mqtt.publish as publish

MQTT_SERVER = "192.168.1.5"
MQTT_PATH = "test_channel"

publish.single(MQTT_PATH, "Hello World!",
hostname=MQTT_SERVER)
```

V. CONCLUSION

Raspberry Pi offers many capabilities in recent times as the system uses advanced sensor technology and automatic control technology. The data collected in the field can be sent through the existing network which improves the applicability of the system. "Message Queuing Telemetry Transport(MQTT)" protocol utilized for Raspberry Pi is presented in this paper. MQTT represents the M2M communication protocol and follows publish/subscribe communication pattern.

Reliable delivery of data between sensors and devices is focussed in MQTT with the help of messaging protocol. Due to the reason that MQTT focuses in high-latency and low-bandwidth environments, it becomes a fitting protocol for “machine-to-machine” communication.

REFERENCES

1. Al-Fuqaha, Ala, et al. "Toward better horizontal integration among IoT services." IEEE Communications Magazine 53.9 (2015),pp. 72-79.
2. Aljawarneh, Shadi, and Muneer Bani Yassein. "A resource-efficient encryption algorithm for multimedia big data." Multimedia Tools and Applications (2017),pp. 1-22.
3. "MQTT", Mqtt.org, 2018. [Online]. Available: http://mqtt.org/. [Accessed: 11- Nov- 2019].
4. Thangavel, Dinesh, et al. "Performance evaluation of MQTT and CoAP via a common middleware." Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on. IEEE, 2014.
5. Babovic, Zoran, Jelica Protic, and Veljko Milutinovic. "Web Performance Evaluation for Internet of Things Applications." IEEE Access..
6. Mijovic, Stefan, Erion Shehu, and Chiara Buratti. "Comparing application layer protocols for the Internet of Things via experimentation." Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum on IEEE, 2016.
7. En.wikipedia.org, ‘Raspberry Pi’, 2012. [Online]. Available:https://en.wikipedia.org/wiki/Raspberry_Pi. [Accessed 5 Nov.2019].
8. Raspberry Pi, ‘Raspberry Pi 3 Model B+ - Raspberry Pi’, 2016. [Online] Available:https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/ [Accessed 7 Nov. 2018].
9. "FrontPage - Raspbian", Raspbian.org, 2016. [Online]. Available: http://www.raspbian.org/. [Accessed: 12- Nov- 2019].
10. "Download PuTTY - a free SSH and telnet client for Windows", Putty.org, 2016. [Online]. Available: https://putty.org/. [Accessed: 13- Nov- 2019].
11. "VNC Viewer", LO4D.com, 2016. [Online]. Available: https://vnc-viewer.en.lo4d.com. [Accessed: 14- Nov- 2019].
12. Raspberrypi.org, ‘FAQs - Raspberry Pi Documentation’, 2012. [Online].Available:https://www.raspberrypi.org/help/faqs/#softwareLanguages. [Accessed 4 Nov. 2018].
13. "Eclipse Paho - MQTT and MQTT-SN software", Eclipse.org, 2016. [Online]. Available: http://www.eclipse.org/paho/. [Accessed: 14- Nov- 2019].

AUTHORS PROFILE

R. Chawngsangpuii received her M.Sc degree in Computer Science from Bharathiar University, Coimbatore, India. She is working in Mizoram University since 2007 as Assistant Professor in the Department of Information Technology. She started the Engineering Department with B.Tech(IT) course in August, 2007. Since then the Department has grown manifolds with many alumni members joining prestigious companies all over India. She had been conferred with The Best Citizen of India Award in 2014 by The International Publishing House for her work. She is pursuing research in the field of Internet of Things in Assam University. Her research work involves implementation in the security aspects of the Internet of Things using Raspberry Pi 3. She has published and presented many papers on the various issues in the Internet of Things on the research findings in both National and International Journals and Conferences. She has reviewed Research Papers for Journals in Proceedings of International Conference on Computational Intelligence and Data Engineering 2018 (ICCIDE-2018). She had been Resource Person in UGC-Sponsored Refresher Courses and Orientation Courses. She had delivered many Invited Talks in National and International Workshops/Conferences such as Short Term Course on MOOCs organized by UGC-HRDC Mizoram University. Her other interests for area of research work are in Blockchain and Robotics.

Dr. Prodipto Das received his M.Sc. (Computer Science) from Assam University Silchar in 2003, PhD from Assam University, Silchar in 2011, Post-Doctoral Fellow at University of Plymouth, United Kingdom in 2012. He had many research scholars under his guidance and many had been awarded with PhD degree. He is member of many scientific/professional bodies such as IEEE, International Association of Computer Sciences & Information Technology(IACSIT), Singapore, Association of Computing Machineries (ACM), USA, etc. He has been awarded with Best Paper Award in 2007 at National Conference NCSCA-2007, Bhubaneswar, Odisha. He has been appointed as Session Chair for International Conferences and delivered many Invited Talks in National and International Workshops/Conferences. He also had organised many Conferences and Workshops such as 3-day ISI (ACMU)-AUS Winter School on Graph Algorithms with special focus on Applications to Network and VLSI in 2016, ISEA Workshop on Cyber Crimes and Cyber Hazards in Assam University(collaboration with CDAC Kolkata) in 2012. He had more than 30 publications in National/International Journals and had published books “Computer Graphics: Principles and Practices,” P. Das , New Delhi Publishers, New Delhi, India, 2016 and “Advanced Computing Applications, Databases and Networks,” S. A. Begum and P. Das (Eds.), Narosa Publishing House Ltd., New Delhi, India, 2011.