

M2M Communication in Ad-Hoc WSNs for Industrial Application Using MQTT Protocol

Suma Manuvinakurike Narasimhasastry, Pramodh Basavaraju, Kirthana Bhaskar, Ranganatha Horigowda, Bhimalinga Kore

Department of Electronics and Communication, BMS College of Engineering, Bangalore, India

Email address:

suma.ece@bmsce.ac.in (S. M. Narasimhasastry), pramodhb05@gmail.com (P. Basavaraju)

To cite this article:

Suma Manuvinakurike Narasimhasastry, Pramodh Basavaraju, Kirthana Bhaskar, Ranganatha Horigowda, Bhimalinga Kore. M2M Communication in Ad-Hoc WSNs for Industrial Application Using MQTT Protocol. *Advances in Wireless Communications and Networks*. Vol. 3, No. 4, 2017, pp. 39-44. doi: 10.11648/j.awcn.20170304.12

Received: May 8, 2017; **Accepted:** June 6, 2017; **Published:** July 25, 2017

Abstract: M2M communication is based on data transfer data between devices to a remote central location for effective monitoring and control. M2M enables networked devices to share information and perform actions, without the manual assistance or not much of human intervention. In this paper, M2M communication connected in an Adhoc using the MQTT protocol is presented and demonstrated. The open cloud is used for the database storage and the results are shown in jsonformat. Experimental studies suggest Adhoc M2M communications established with MQTT protocol has provided effective communications for industrial applications with minimum packet loss and delays.

Keywords: M2M, Adhoc Network, Wireless Sensor Network, MQTT

1. Introduction

Communications between the machines started as wired communication which was not very cost effective and flexible. Development of wireless communication standards 2G, 3G, 4G (LTE) provided flexibility for remote communicate anywhere and specifically for industrial application it is need of the hour. In future machines will be autonomously programmed to communicate with each other without human intervention.

The current technology in M2M communication is based on is INDUSTRY4.0. In this technology all the machines will be interconnected to each other wirelessly and are connected to the internet. The basic functions of M2M communications are

- Data collection: Here the data is collected from the machines. Ex: sensor value, temperature value, etc.
- Data Transmission: The data collected are transmitted through the wireless technology to the server.
- Data Storage: the CLOUD can be open cloud or service provided.
- Response: Data from the storage device area accessed further Communications and actions are performed.

The author explains the M2M technology and its practical

applications [1]. The growth of technological capabilities of mobile devices, the evolution of wireless communication technologies, M2M refers to data communication between machines without human intervention. M2M is the convergence of mobile agent connected by wired network or by wireless network, with an information system capable of making decisions. The grand schemes of a model to be used in an agricultural decision support System is being presented. It tells how the multi agents programming and constraint programming are applied to hybrid system. There are also some constraints for distributed architecture and all this is done on the java platform [2].

Using sensors and raspberry piB+ model are used for creating as network gateway and by implementing this sensor data is transferred and received and these data are being monitored on the web page [3]. Here it describes how to use the cayenne cloud platform and also provides a clear view of the raspberry pi or Arduino as connected to the platform [4]. In this a web-based Internet of things solution aimed for monitoring, tracking and analysing data in agriculture area is proposed, all this are done using the MQTT protocol. A real time data is achieved using broker-based publishing/subscribing MQTT. Collected data from sensors is shown and stored in web part of the information system. This type of implementation is developed by the

manufacturers need for monitoring and tracking data. [5]. Cloud computing is an emerging area that affects IT infrastructure, network services, and applications. In this article it introduces various aspects of cloud computing, including the rationale, underlying models, and infrastructures and provides more details about some of the specific technologies and scenarios [6-9].

The manual process of controlling machines offsite industry is laborious and time consuming. As there are many industries where automation is needed for group of machines performing different tasks, it is challenging to control these machines work in synchronized pattern with proper co-ordination. In this direction there is a need to work for efficient protocol for creating adhoc M2M communication with less packet loss and seamless connectivity. In this paper we present use of MQTT protocol for Adhoc network for

industrial M2M communications. Paper is organised with section 2 on proposed solution and in section 3, implementation steps are discussed. Section 4 we present results and discussion.

2. Proposed Solution

To resolve the issues in present M2M for industrial applications we prototype ad-hoc network scenario using devices with MQTT protocol for communication. A centralized monitoring system will collect the data from the network available in cloud and manager can review the collected information to control the entire operation of the plant. The scenario with block description is shown in figure 1.

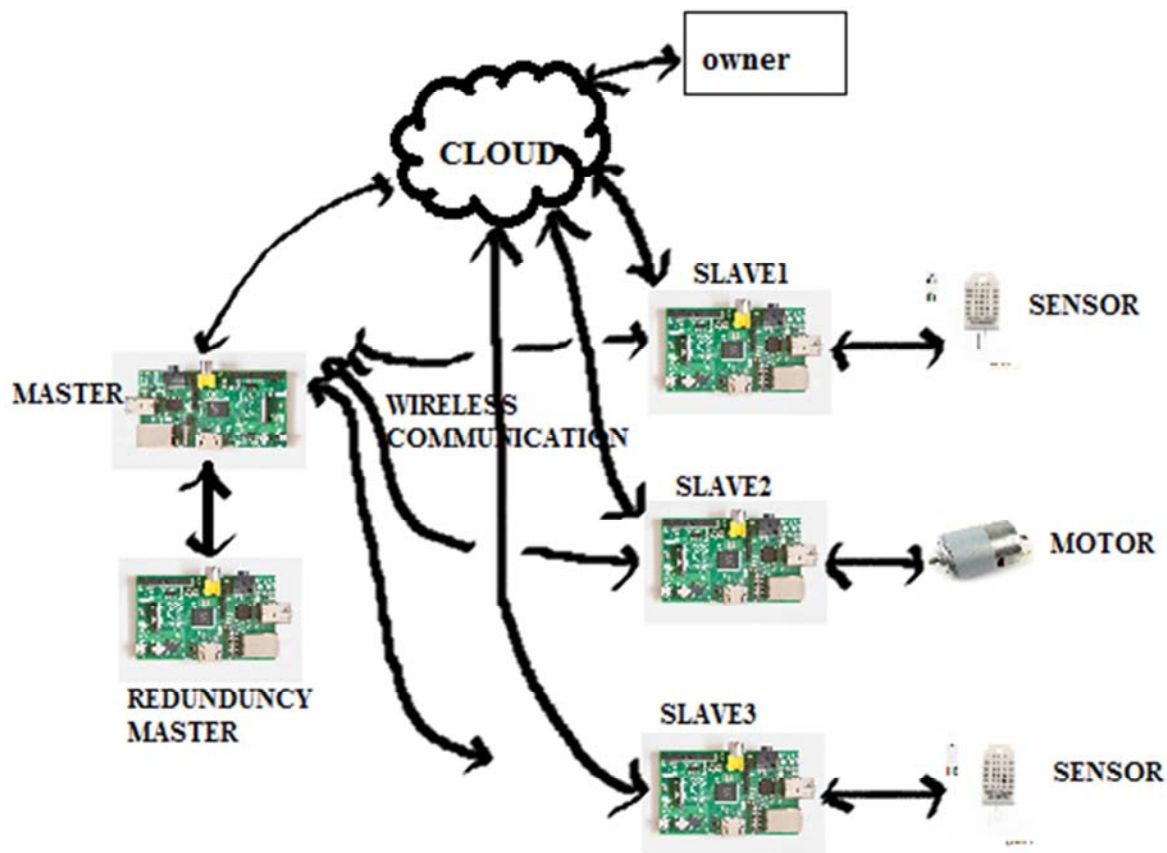


Figure 1. Block description of M2M ad-hoc network.

We have employed hardware which constitutes Raspberry pi3 boards that have the ability to interact with the external world. All the nodes are networked using RS485 network along with RP kit. RP kit acts as master and will collect the information from all the nodes. The collected data is displayed on the LCD monitor for the Manager and other to see the status. One of the Raspberry Pi3 is considered as master and the other three are slaves. For data transfer we have temperature and the humidity sensor to send the data to the master.

The required Raspberry Pi packages are downloaded and installed and created an ad-hoc network by suitable

commands in LINUX platform.

PUTTY is a terminal emulator application that is used as a client for the SSH, Telnet, rlogin, and raw TCP computing protocols. Xming is an X11 display server for Microsoft Windows operating systems, including Windows XP or later.

MQTT (Message Queuing Telemetry Transport): "The MQTT protocol enables a publish/subscribe messaging model in a flexible manner. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium. MQTT-SN (MQTT for Sensor Networks) is open and lightweight publish/subscribe protocol designed specifically for machine-

to-machine and mobile applications. Required MQTT packages are downloaded and installed and accordingly the publisher/subscriber will take place between the master and slave. Basic communication between Master and slaves in ad-hoc within range is initially tested.

In the next case, when one of the slave moves out of range from the master network, communication is not broken, instead the adjacent slave to which it can connect will transfer information to master. This communication is done using MQTT protocol and they use the terminology called publisher/subscriber. Here the master will publish its presence to which the slaves need to be subscribed. Once

done, then slaves will receive the information from master.

When the data is to be sent the delay is measured by using the timestamp command and this will indicate the delay taken by the master to slave and vice versa to communicate with each other.

All the machines are connected to the cloud called cayenne cloud, an open platform. In this cloud dashboard the machine ID will be displayed and can take the data value and store in the database. Further any disconnection of the slave or if the slave goes down, the buzzer will beep to indicate that the slave is in offline.

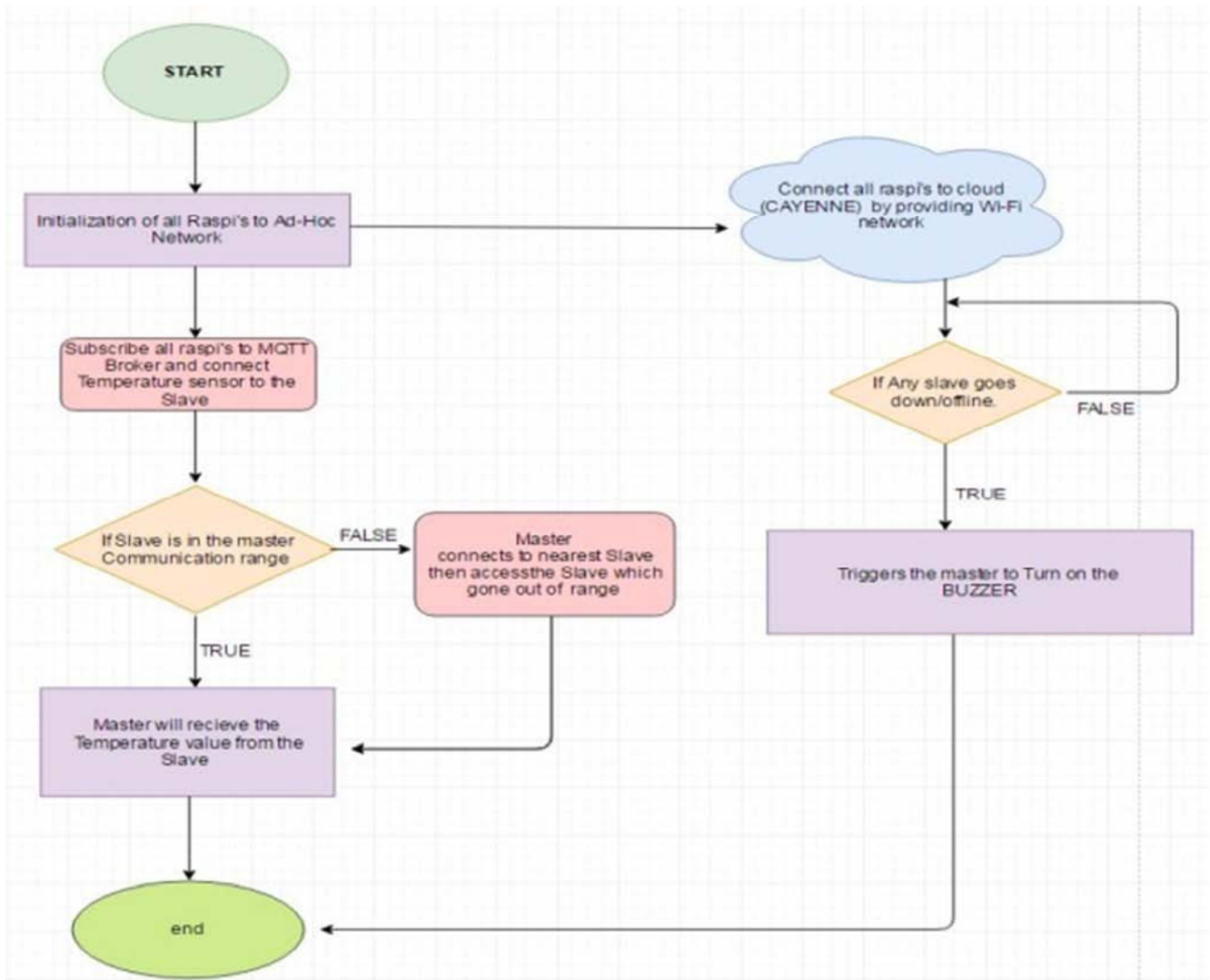


Figure 2. Flow Chart for M2M Ad-Hoc Network.

3. Implementation and Experimental Steps

The initial hardware connection set up is shown in figure 3.

The flow of complete communications is provided in steps below.

Step 1: Connecting the raspberry pi as master as shown below in figure 3.

Step 2: login to master is shown in figure 4



Figure 3. Raspberry Pi Initial Setup.

```

pi@Master:~$ ssh 192.168.1.2
ssh: connect to host 192.168.1.2 port 22: No route to host
pi@Master:~$ ssh 192.168.1.4
pi@192.168.1.4's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 26 13:18:38 2017 from 192.168.1.7
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.
pi@Slave3:~$ ssh 192.168.1.2
pi@192.168.1.2's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 26 10:18:22 2017
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.
pi@Slave1:~$

```

Figure 4. MasterTerminal Window.

Step 3: connect the temperature sensor (DHT 22) as shown in figure 5.

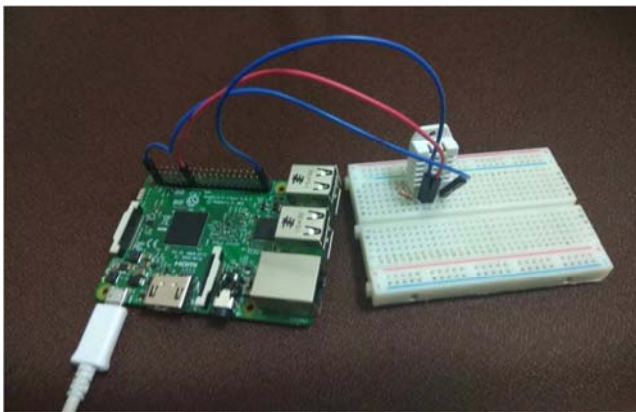


Figure 5. Temperature Sensor Connection.

Step 4: Initially slave 1 connects to master and communication is established. For this secure shell command are used as shown in figure 6.

```

pi@Master:~$ ssh 192.168.1.2
pi@192.168.1.2's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 26 10:29:08 2017 from 192.168.1.7
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.
pi@Slave1:~$

```

Figure 6. Login from master to slave 1 using secure shell login.

The communication starts and data related to temperature sensor are transferred to master as shown in figure 7

```

pi@Master:~/Desktop/Master1 $ python Pub.py
enter your command:'GET S1'
Message Published...
Subscribed to MQTT TopicBlock1/Temp
i am connected to u
enter your command:'GET Temp S1'
Message Published...
Subscribed to MQTT Topic
Temperature is: -999.00C
Temperature is: -999.00C
Temperature is: 31.60C
Temperature is: 31.60C
Temperature is: 31.60C
Temperature is: 31.50C
Temperature is: 31.50C

```

Figure 7. Slave 1 sends the temperature value to the master.

4. Results and Discussion

Observation is made on the proposed scenario which has given seamless communication between devices using MQTT protocol as in figure 9 Python is used to program commands for Raspberry pi devices. When slave 1 and 2 are connected to the master, within range of network i.e. the slave is within the (15m), the temperature and humidity sensor connected to the slave 1 and 2 are transferred to master through cloud. Then slave 1 is moved out of network where in its login shows error.

The control is transferred to the slave-3 that act as a mediator between slave 1 and the master for reading the temperature value. The slave 1 is kept away from the master pi at a distance of 25m.

The master will publish and ask for the temperature value from slave 1 by running the python code for it. The slave 2 which connected to the temperature sensor when not in range. i.e at a distance of 25m from the master. The master can login to nearer slave 3 and

Can communicate with slave 1 using the secure shell login. (Shown in figure 10)


```

pi@Master:~/Desktop/Master1 $ python Pub.py
enter your command:'GET S1'
Message Published...
Subscribed to MQTT Topic
i am connected to 192.168.1.4
enter your command:'GET Temp S1'
Message Published...
Subscribed to MQTT Topic
Temperature is: -999.00C
Temperature is: -999.00C
Temperature is: 29.80C
Temperature is: 29.70C
Temperature is: 29.70C
Temperature is: 29.70C
Temperature is: 29.70C
Temperature is: 29.70C

```

Figure 9. The communication with slave 1 using MQTT.

```

pi@Master:~$ ssh 192.168.1.2
ssh: connect to host 192.168.1.2 port 22: No route to host
pi@Master:~$ ssh 192.168.1.4
pi@192.168.1.4's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 26 13:18:38 2017 from 192.168.1.7
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.
pi@Slave3:~$ ssh 192.168.1.2
pi@192.168.1.2's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 26 10:18:22 2017
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.
pi@Slave1:~$

```

Figure 10. Master slave 1 through Slave 3 connected through secure shell command.

The master gets the temperature value from slave 1 through slave 3 by MQTT publisher code. The slave 1 acknowledges and sends the required data. (As in figure 11)

```

pi@Slave1:~/Desktop/r/Slave1 $ sudo pigpiod
pi@Slave1:~/Desktop/r/Slave1 $ python Subscriber.py
Message Published...
Subscribed to MQTT Topic
GET Temp S1
Humidity is: -999.00%
Temperature is: -999.00C
Message Published...
Humidity is: -999.00%
Temperature is: -999.00C
Message Published...
Humidity is: 34.20%
Temperature is: 29.80C
Message Published...
Humidity is: 34.10%
Temperature is: 29.70C
Message Published...
Humidity is: 34.10%
Temperature is: 29.70C
Message Published...
Humidity is: 34.10%
Temperature is: 29.70C
Message Published...

```

Figure 11. Slave 1 sends e Temperature data to the Master through slave 3.

```

pi@Master:~/Desktop/Master1 $ python Pub.py
enter your command:'GET S1'
Message Published...
2017-05-03 15:02:31.047287
Subscribed to MQTT TopicBlock1/Temp
i am connected to u
2017-05-03 15:02:33.247011
enter your command:'GET Temp S1'
Message Published...
2017-05-03 15:02:43.768335
Subscribed to MQTT Topic
Temperature is: -999.00C
2017-05-03 15:02:49.387664
Temperature is: -999.00C
2017-05-03 15:02:52.359242
Temperature is: 31.00C
2017-05-03 15:02:55.401060
Temperature is: 31.00C
2017-05-03 15:02:58.407668
Temperature is: 31.00C
2017-05-03 15:03:01.470822
Temperature is: 31.00C
2017-05-03 15:03:04.489082

```

Figure 12. Time for communication of packets.

We find out delay by taking difference between sent and received message time stamp as shown in figure 12

```

pi@Master:~/Desktop/Master1 $ python Pub.py
enter your command:'GET S1'
Message Published...
2017-05-03 15:02:31.047287
Subscribed to MQTT TopicBlock1/Temp
i am connected to u
2017-05-03 15:02:33.247011
enter your command:'GET Temp S1'
Message Published...
2017-05-03 15:02:43.768335
Subscribed to MQTT Topic
Temperature is: -999.00C
2017-05-03 15:02:49.387664
Temperature is: -999.00C
2017-05-03 15:02:52.359242
Temperature is: 31.00C
2017-05-03 15:02:55.401060
Temperature is: 31.00C
2017-05-03 15:02:58.407668
Temperature is: 31.00C
2017-05-03 15:03:01.470822
Temperature is: 31.00C
2017-05-03 15:03:04.489082

```

Figure 13. Time for slave 1 in range.

The observed packet loss is zero, placing Slave 8 meters apart from the master and found faster data transfer which is indicated as time stamps. This is shown in figure 13

```

pi@Master:~$ sudo ping 192.168.1.4
PING 192.168.1.4 (192.168.1.4) 56(84) bytes of data.
64 bytes from 192.168.1.4: icmp_seq=1 ttl=64 time=17.6 ms
64 bytes from 192.168.1.4: icmp_seq=2 ttl=64 time=9.65 ms
64 bytes from 192.168.1.4: icmp_seq=3 ttl=64 time=9.72 ms
64 bytes from 192.168.1.4: icmp_seq=4 ttl=64 time=9.85 ms
64 bytes from 192.168.1.4: icmp_seq=5 ttl=64 time=8.05 ms
64 bytes from 192.168.1.4: icmp_seq=6 ttl=64 time=10.3 ms
64 bytes from 192.168.1.4: icmp_seq=7 ttl=64 time=9.79 ms
64 bytes from 192.168.1.4: icmp_seq=8 ttl=64 time=8.82 ms
64 bytes from 192.168.1.4: icmp_seq=9 ttl=64 time=7.93 ms
64 bytes from 192.168.1.4: icmp_seq=10 ttl=64 time=8.60 ms
64 bytes from 192.168.1.4: icmp_seq=11 ttl=64 time=8.16 ms
64 bytes from 192.168.1.4: icmp_seq=12 ttl=64 time=47.7 ms
^C
--- 192.168.1.4 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11015ms
rtt min/avg/max/mdev = 7.938/13.025/47.712/10.745 ms

```

Figure 14. Packet Transfer rate test (Slave 3 is at a distance of 8 meter from Master).

Also, we observed the packet loss and data transfer rate by placing Slave 15 meter apart from the Master, we found packet transfer is 100% successful but there is a slightly higher delay while transferring because of longer distance as shown in figure 14 and figure 15 for distances of 8mts and 15 mts for slave from master.

```

pi@Master:~$ sudo ping 192.168.1.4
PING 192.168.1.4 (192.168.1.4) 56(84) bytes of data.
64 bytes from 192.168.1.4: icmp_seq=1 ttl=64 time=7.80 ms
64 bytes from 192.168.1.4: icmp_seq=2 ttl=64 time=33.0 ms
64 bytes from 192.168.1.4: icmp_seq=3 ttl=64 time=32.4 ms
64 bytes from 192.168.1.4: icmp_seq=4 ttl=64 time=124 ms
64 bytes from 192.168.1.4: icmp_seq=5 ttl=64 time=12.0 ms
64 bytes from 192.168.1.4: icmp_seq=6 ttl=64 time=828 ms
64 bytes from 192.168.1.4: icmp_seq=7 ttl=64 time=9.99 ms
64 bytes from 192.168.1.4: icmp_seq=8 ttl=64 time=8.89 ms
64 bytes from 192.168.1.4: icmp_seq=9 ttl=64 time=8.28 ms
64 bytes from 192.168.1.4: icmp_seq=10 ttl=64 time=8.81 ms
64 bytes from 192.168.1.4: icmp_seq=11 ttl=64 time=34.4 ms
64 bytes from 192.168.1.4: icmp_seq=12 ttl=64 time=13.3 ms
64 bytes from 192.168.1.4: icmp_seq=13 ttl=64 time=8.30 ms
64 bytes from 192.168.1.4: icmp_seq=14 ttl=64 time=14.7 ms
^C
--- 192.168.1.4 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13012ms
rtt min/avg/max/mdev = 7.804/81.779/828.588/209.193 ms

```

Figure 15. Packet Transfer rate test -Slave 3 is at a distance 15meter.

5. Conclusion

M2M is the present technology employed in the industries for connecting machines in the ad-hoc network. After survey among the communication protocols, COAP and MQTT are efficient for M2M industrial IOT applications. MQTT being open source is preferred for M2M implementation as it has provided faster data transmission and less time delay. Further open cloud platform allows all the data to be stored, reviewed by the manager whenever needed. MQTT protocol employed configures devices for very less power consumption and the header required for it is very less compared to http protocol. In the scenario with adhoc network we have studied the effective communication between M2M with negligible

delay and this prototype is being extended forward for communication in real industrial environment to control motors and other equipment through cloud.

In addition to effective communication, power budget was less as consumption depends upon the number of raspberry Pi boards connected. The power budget suggested tentatively is around 3.3W with all slaves are simultaneously communicating with master.

This suggests that M2M Adhoc network for industrial IOT applications with MQTT protocol serves as good and flexible platform for connecting devices.

Acknowledgement

We thank Technical Education Quality Improvement Programme (TEQIP)-phase-II for the support in carrying out this project work.

References

- [1] https://www.iot-now.com/wp-content/uploads/2011/05/M2MTechnology_Challenges-and-Opportunities.pdf
- [2] Generic model based on constraint programming and multi-agent system for M2M services and agricultural decision support Kamal Moummadi; Rachida Abidar; Hicham Medromi, 2011 International Conference on Multimedia Computing and Systems.
- [3] <http://ieeexplore.ieee.org/document/7570945/> "Mqtt based home automation system" published in Colossal Data Analysis and Networking (CDAN), Symposium
- [4] <https://mydevices.com/cayenne/docs/>
- [5] A web-based IoT solution for monitoring data using MQTT protocol Krešimir Grgić; Ivan Špeh; Ivan Hedi 2016 International Conference on Smart Systems and Technologies (SST).
- [6] T. Sridhar, "Cloud Computing - A Primer," The Internet Protocol Journal, Volume 12, No. 3, September 2009.
- [7] <http://www.oneM2M.org> One M2M Technical Specifications, 30, Aug, 2016.
- [8] Ullas B, Anush, Roopa J, Govinda Raju, "Machine to Machine Communication for Smart Systems using MQTT", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Issue 3, March 2014.
- [9] Ad-hoc & sensor networking for M2M Communications January 2017 European Union Agency for Network and Information Security (ENISA).