



Traffic Prioritization In Message Queue Telemetry Transport For Sensor Networks (MQTT-SN) Gateway

A Dissertation Report Submitted

By

Tabinda

(11506405)

To

Department of Computer Science & Engineering

In partial fulfillment of the Requirement for the

Award of the Degree of

Master of Technology in Computer Science and Engineering

Under the guidance of

Ms. Nahita Pathania

(June 2017)



TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering

Program : P172::M.Tech. (Computer Science and Engineering) [Full Time]

COURSE CODE : CSE546 **REGULAR/BACKLOG :** Regular **GROUP NUMBER :** CSERGD0246

Supervisor Name : Nahita Pathania **UID :** 19372 **Designation :** Assistant Professor

Qualification : _____ **Research Experience :** _____

SR.NO.	NAME OF STUDENT	REGISTRATION NO	BATCH	SECTION	CONTACT NUMBER
1	Tabinda	11506405	2015	K1518	9797086746

SPECIALIZATION AREA : Networking and Security **Supervisor Signature:** _____

PROPOSED TOPIC : Comparative analysis of IoT Protocols

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	Parameter	Rating (out of 10)
1	Project Novelty: Potential of the project to create new knowledge	6.40
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.00
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	6.40
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	6.80
5	Social Applicability: Project work intends to solve a practical problem.	6.60
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	6.60

PAC Committee Members		
PAC Member 1 Name: Prateek Agrawal	UID: 13714	Recommended (Y/N): Yes
PAC Member 2 Name: Pushpendra Kumar Pateriya	UID: 14623	Recommended (Y/N): Yes
PAC Member 3 Name: Deepak Prashar	UID: 13897	Recommended (Y/N): Yes
PAC Member 4 Name: Kewal Krishan	UID: 11179	Recommended (Y/N): Yes
PAC Member 5 Name: Anupinder Singh	UID: 19385	Recommended (Y/N): NA
DAA Nominee Name: Kanwar Preet Singh	UID: 15367	Recommended (Y/N): Yes

Final Topic Approved by PAC: Comparative analysis of IoT Protocols(Which protocols of IoT)

Overall Remarks: Approved (with major changes)

PAC CHAIRPERSON Name: 11011::Dr. Rajeev Sobti **Approval Date:** 28 Oct 2016

ABSTRACT

It has not been much time since the Internet of Things (IoT) came into existence. It is a fresh concept that is always evolving. Ubiquitous computing, wireless technologies, sensing technologies, Internet Protocol (IP) and devices are mingled together in order to devise a system where the virtual or abstract world meets the real world meet and they interact continuously with each other. The very basic building block of Internet of Things is the “smart object”, which can be produced by putting perception and intelligence into the normal or day- to- day objects. These smart objects are capable of collecting information from the environment as well as being connected to each other through the network to exchange data and information. IoT has a large number of protocols. In this report we discuss one of the Application Layer Protocols namely Message Queue Telemetry Transport Protocol and also devise a mechanism to prioritize data in a Message Queue Telemetry Transport Gateway in order to mitigate the delay of data packets which is necessary for time critical applications. Data needs to be prioritised at an MQTT-SN Gateway so that high priority sensor data such as data from a sensor monitoring a patient’s heart rate gets better Quality of Service (QoS) as compared to low priority data such as data from a sensor monitoring temperature changes. Two scheduling algorithms namely First in First out Scheduling Algorithm and Round Robin Scheduling Algorithm have been used in order to prioritise the sensor data that arrives at the MQTT-SN Gateway.

Apart from prioritising sensor data with the help of the above mentioned scheduling algorithms, we also compared these two scheduling algorithms on the basis of some parameters so as to determine which one is better suited for this purpose.

Keywords: Internet of Things (IoT), Wireless Sensor Networks (WSN), Message Queue Telemetry Transport Protocol for Sensor Networks (MQTT-SN), Gateways, Traffic Prioritization.

DECLARATION

I hereby declare that the research work reported in the dissertation entitled “**TRAFFIC PRIORITIZATION IN MESSAGE QUEUE TELEMETRY TRANSPORT FOR SENSOR NETWORKS (MQTT-SN) GATEWAY**” in partial fulfilment of the requirement for the award of Degree for Master of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Ms Nahita Pathania. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University’s Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Date:

Name: Tabinda

Registration No: 11506405

CERTIFICATE

This is to certify that the work reported in the M.Tech Dissertation entitled “**TRAFFIC PRIORITIZATION IN MESSAGE QUEUE TELEMETRY TRANSPORT FOR SENSOR NETWORKS (MQTT-SN) GATEWAY**”, submitted by **Tabinda** at **Lovely Professional University, Phagwara, India** is a bonafide record of her original work carried out under my supervision. This work has not been submitted elsewhere for any other degree.

Signature of Supervisor

Date:

Name:

Counter Signed by:

1) Concerned HOD:

HoD's Signature: _____

HoD Name: _____

Date: _____

2) Neutral Examiners:

External Examiner

Signature: _____

Name: _____

Affiliation: _____

Date: _____

Internal Examiner

Signature: _____

Name: _____

Date: _____

ACKNOWLEDGEMENT

I owe a debt of deepest gratitude to my thesis supervisor, **Ms. Nahita Pathania**, Department of Computer Science And Engineering, for her guidance, support, motivation and encouragement throughout the period this work was carried out. Her readiness for consultation at all times, educative comments, concern and assistance even with practical things have been invaluable.

I would like to thank the Almighty and my parents who were always there as a source of support throughout the period of this work.

TABLE OF CONTENTS

Abstract	i
Declaration	ii
Certificate	iii
Acknowledgment	iv
Table of Contents	v
List of Figures	vii
List of Tables	x
 CHAPTER PLAN	
Chapter 1 Introduction	1-17
1.1 Architecture of Internet of Things	2-3
1.2 Technologies Used in Internet of Things	3-5
1.3 Protocol Overview	5-15
1.4 Applications	16-17
1.5 Challenges Faced by Internet of Things	17
 Chapter 2 Review of Literature	 18-27
 Chapter 3 Present work	 28-35
3.1 Problem Formulation	28-33
3.2 Objectives of the Study	33
3.3 Research Methodology	33-35
3.3.1 Simulation Tool	33-34
3.3.2 Algorithm Steps	34-35
 Chapter 4 Results and Discussions	 36-56
4.1 Assumptions	36

4.2	Simulation Parameters	36
4.3	Simulation	36-45
4.4	Comparison of Results	45-51
4.5	Hardware Implementation	51-56
Chapter 5 Conclusion and Future Scope		57-58
5.1	Conclusion	57
5.2	Future Scope	57-58
Chapter 6 References		59-62
Chapter 7 Appendix		63-64

LIST OF FIGURES

Fig. 1 Internet of Things	2
Fig. 2 MQTT Protocol	7
Fig. 3 MQTT Control Packet Structure	9
Fig. 4 Fixed Header Format	10
Fig. 5 Publish-Subscribe in MQTT	11
Fig. 6 Connection between MQTT Client and Broker	12
Fig. 7 MQTT Message Format	12
Fig. 8 General Message Format	14
Fig. 9 Message Header Format	14
Fig.10 Types of MQTT Gateways	29
Fig.11 Buffers	32
Fig.12 Dataflow Diagram of proposed Method	35
Fig.13 High Priority Tasks Completed	37
Fig.14 Medium Priority Tasks Completed	37
Fig.15 Low Priority Tasks Completed	38
Fig.16 Priority Value for Tasks	38
Fig.17 Packets Departed from the Server	39
Fig.18 Server Utilization	39
Fig.19 Average Waiting Time of Packets	40

Fig.20 Waiting Time of Class 1 Packets	41
Fig.21 Waiting Time of Class 2 Packets	41
Fig.22 Waiting Time of Class 3 Packets	42
Fig.23 Server Utilization	42
Fig.24 Packets Departed from the Server	43
Fig.25 Average Waiting Time of Packets	43
Fig.26 Packets Departed from the Server	44
Fig.27 Server Utilization	44
Fig.28 Average Waiting Time of Packets	45
Fig.29 Packets Departed from the Server in Priority Based Scheduling Algorithm	46
Fig.30 Packets Departed from the Server in Round Robin Scheduling Algorithm	46
Fig.31 Packets Departed from the Server in an Unprioritized MQTT-SN Gateway Model	47
Fig.32 Server Utilization in Priority Based Scheduling Algorithm	47
Fig.33 Server Utilization in Round Robin Scheduling Algorithm	48
Fig.34 Server Utilization in an Unprioritized MQTT-SN Gateway Model	48
Fig.35 Average Waiting Time of Packets in Priority Based Scheduling Algorithm	49
Fig.36 Average Waiting Time of Packets in Round Robin Scheduling Algorithm	50
Fig.37 Average Waiting Time of Packets in an Unprioritized MQTT-SN Gateway Model	50
Fig.38 Sensor Data of Priority 1 in First in First out Scheduling Algorithm	52

Fig.39 Sensor Data of Priority 2 in First in First out Scheduling Algorithm	53
Fig.40 Sensor Data of Priority 3 in First in First out Scheduling Algorithm	53
Fig.41 Arrival of Sensor Data in First in First out Scheduling Algorithm	54
Fig.42 Sensor Data of Priority 1 in Round Robin Scheduling Algorithm	54
Fig.43 Sensor Data of Priority 2 in Round Robin Scheduling Algorithm	55
Fig.44 Sensor Data of Priority 3 in Round Robin Scheduling Algorithm	55
Fig.45 Arrival of Sensor Data in Round Robin Scheduling Algorithm	56

LIST OF TABLES

Table 1: Simulation Parameters

36

CHAPTER 1

INTRODUCTION

The man behind Internet of Things is Kevin Ashton, who was the co-founder of MIT's (Massachusetts Institute of Technology) Auto ID lab. In fact it was he who coined the term Internet of Things (IoT).

Internet of Things is a concept where different kinds of devices, humans or objects are provided with identifiers that are unique and can transmit data over a network without the requirement of any human to human or human to machine intervention. In the internet of Things a “Thing” can be anything ranging from- an automobile that alerts the driver (due to in-built sensors) when the pressure of any tyre is low to a person with a wearable body glucose monitor, etc.

IoT can also be defined as an extensive and open network of objects which are intelligent and have the capability to arranging, sharing data and information, acting and reacting according to situation and variations in their environment. IoT has become quite important in terms of industries, enterprises, medical field and engineering circles, etc. Some new products of Iot like home automation devices, energy management devices, and various internet enabled appliances lead one to the concept of an application called smart home that gives more efficiency as well as security. Some wearable health monitoring devices and which are network enabled are taking health care services being provided to new heights. IoT has one more important application - in smart cities where actuators, sensors, etc that are embedded in bridges, roads and intelligent traffic systems will be used.

Radio frequency identification (RFID) is a very important technology which is used in Internet of Things due to its capability of tracking down a huge number of objects (that are uniquely identifiable) by using Electronic Product Code (EPC), which gives a unique identity to a specific physical object. Some other technologies such as-2D codes, barcodes, etc can also be used for the purpose.

The idea of combination of computers and networks so as to monitor and manage devices or things has been in the minds of innovators and scientists for a long time. The first internet enabled device was an IP enabled toaster and it was created in the year 1990. The operations of this smart toaster could be controlled over the network. Soon after other things like soda vending

machine and coffee pot were IP enabled. This was just the beginning which eventually paved the way for the present day's Internet of Things.



Fig. 1. Internet of Things (Khan, Rafiullah et al. 2012)

1.1 ARCHITECTURE OF INTERNET OF THINGS

Internet of Things has different architectural models which have been deduced from different angles. There exists the basic three layer architecture (Jing, Qi et al. 2014), six layer architecture (Zang, Minghui et al.2012), etc.

Implementation of Internet of Things is based on an architecture that consists of several layers. This kind of architecture is to be designed in such a way that it can meet the different requirements of various enterprises, industries, government, etc. Khan, Rafiullah, et al. proposed a five layer architecture of IoT (Khan, Rafiullah, et al. 2012).

A five layered architecture is given below-

The various layers are as follows:

i) Perception Layer: This layer is also known as the device layer consists of physical devices or objects and sensors. Here the sensors present can be barcode sensors, RFID sensors, etc. This layer is concerned with identifying and then collecting information about objects with the help of sensors. The collected information can be regarding temperature, location, motion, chemical

changes etc depending upon the kind of sensors used. The information that has been collected is sent to the network layer so that it is securely transmitted towards the information processing system.

ii) Network Layer: Also known as transmission layer, this layer deals with the transmission of information collected, from sensor devices towards the information processing centre in a secure manner. The medium of transmission can be WiFi, Bluetooth, UMTS, etc.

This layer transfers the collected information from to middleware perception layer.

iii) Middleware Layer: All the objects in Internet of Things implement services of different kinds. Each and every device connects and later communicates only with those devices that implement the similar types of services. Middleware layer gets the information from the transmission or network layer and afterwards it stores the information in the database. This layer processes the information, does the computation and eventually automatically takes decisions on the basis of computed results.

iv) Application Layer: This layer is concerned with the management of applications on the basis of processed information that is present inside the middleware layer. Smart city, smart home, smart healthcare, etc are some of the applications implemented by Internet of Things.

v) Business Layer: This layer deals with managing the Internet of Things System along with its applications as well as services. This layer builds graphs, models, flowcharts, etc on the basis of data that has been obtained from the application layer.

1.2 TECHNOLOGIES USED IN INTERNET OF THINGS

The development of a pervasive computing system, where objects can be identified uniquely and can be capable of thinking as well as interacting with the other objects in order to collect data based on which automated actions will be taken, needs a combination of effective and new technologies that is only possible by integrating different technologies that can identify objects which in turn can communicate with one another.

1. Radio Frequency Identification: RFID uses tags that are fixed to the objects that are supposed to be identified. It has radio transmitter-receiver also known as interrogators that send a signal to the label or tag so as to know its response. RFID tags can be either active or passive (has a small sized battery attached to it and gets activated an RFID reader is present). RFID is more reliable, inexpensive, accurate and secure. It has large number of wireless applications such as monitoring patients, in military field, etc.

2. Wireless Sensor Networks (WSN): It is a bidirectional sensor network that is connected wirelessly and is built from various nodes that are present in a sensor field. Each and every sensor node is connected to one or more than one sensors that can collect data like humidity, location, temperature, etc and then this information is passed from sensor to sensor till it reaches processing equipment. Each sensor is equipped with an antenna, a micro-controller, and an interfacing circuit that act as communication, actuation and sensing unit respectively along with a power source. Data collected by the sensors is shared amongst the sensor nodes and then it is sent to a centralized or a distributed system for the purpose of analytics (Whitmore, Andrew et al 2015).

3. Near Field Communication (NFC): It is a relatively new technology that is based on RFID standards (Alur, Rajeev, et al. 2016). NFC is a short range standard for communication in which devices can involve in radio communication with each other on being touched or brought into each other's close proximity. Each tag of NFC has a Unique Identification which is associated with the label. This technology is usually embedded in smart phones so as to make them capable of exchanging data with each other when brought close to each other.

4. 2D Barcodes: These are labels that are machine readable and are attached to items in order to note the information that is related to those items. In this method letters or numbers are represented by using a combination of bars and spaces of varying widths. The other name for 2D barcodes is Quick Response (QR) codes. They are named so, because they allow fast data access. These QR codes are usually used together with a smart phone. The user simply scans the code with the help of his/her smart phone, having a bar code reader installed. The barcode reader then translates the encoded URL and finally forwards the browser to the relevant or suitable information present on the website.

5. Wireless Fidelity (WiFi): WiFi is a wireless technology which enables computers to communicate over a wireless signal to other devices. It is widely used in schools, businesses, agencies, homes, etc as an alternative to wired local area network (LAN). The integration of Wireless Fidelity into mobile phones, notebooks, handhelds, etc has tremendously increased the WiFi adoption to a point where it has nearly become a default in these devices.

6. Bluetooth: It is a short range radio technology that is quite cheap and also terminates the need for cabling between devices such as handheld PC's, notebook PC's, cameras, etc in the effective range of 10-100 meters. WiFi enables devices to communicate at a speed that is less than 1

Mbps. It is used for creating PANs (Personal Area Networks). A collection of Bluetooth devices sharing a common communication channel is known as Piconet that is capable of connecting 2 to 8 devices at a time for the purpose of sharing data which may be in the form of text, picture, audio and video. Ericson Mobile Company was the first to start a project named Bluetooth in 1994.

7. Zigbee: This protocol was developed for the enhancement of features of wireless sensor networks. Zigbee Alliance which was founded in 2001 created Zigbee technology. It is a low cost and relatively short transmission protocol. It is reliable as well as scalable. Its range is around 100 meters and its bandwidth is about 250 kbps. It is widely used in industries, medical field, home automation, etc.

8. Actuators: It is a device that is used to convert energy signal into motion. They are capable of producing linear, oscillatory or rotatory motion. They communicate at less than 1 Mbps and can cover distances ranging to 30 feet. They have their applications in industries and manufacturing units.

1.3 PROTOCOL OVERVIEW

The different protocols used in case of IoT are as follows:

- 1. Constraint Application Protocol (CoAP):** This is a request/response protocol of the application layer. It was designed using a subset of Hypertext Transfer Protocol (HTTP) methods to make it interoperable with HTTP. CoAP runs over User Datagram Protocol (UDP) for the sake of keeping the implementation light weight. UDP is used since it reduces bandwidth requirements. Since it runs over UDP which is unreliable, CoAP provides its own mechanisms to achieve reliability. The header of each packet has two bits that state the type of message and the Quality of Service (QoS) level or message reliability in other words, that is required. It has no in-built security features.

There are four message types:

- i) Confirmable:** An acknowledgement (ACK) is required by this kind of message request. Response can be sent synchronously (i.e. within acknowledgement) or if needs more time, it can be sent with a separate message asynchronously.
- ii) Non-Confirmable:** this type of message does not need an acknowledged.
- iii) Acknowledgement:** This type of message confirms the reception of a confirmable message.

iv) **Reset:** It just confirms the reception of a message that could not be processed.

2. Advanced Message Queuing Protocol: AMQP is based on queues. It sends transactional messages between servers. Thousands of reliable queued transactions can be processed by it. Its main focus is on not losing messages. It runs over TCP, which strictly provides point to point reliable connection. Further, the acceptance of each message must be acknowledged by the end points. AMQP was originally developed for banking industry so its middleware lays focus on tracking of all the messages and it also ensures that every message is delivered according to intention, even in presence of failures or reboots.

3. Websocket Protocol: It is neither a request/response nor a publisher/subscriber protocol. In this case handshake is initialized by the client with the server in order to setup a websocket session. This handshake is almost identical to HTTP so that the web servers are capable of handling websocket sessions as well as Hypertext Transfer Protocol connections through the same port. Clients and servers exchange their messages in a full duplex connection asynchronously during a session. The session is terminated when it is no longer needed either by the client or by the server. It runs over TCP and does not require any reliability mechanisms of its own.

4. Extensible Messaging and Presence Protocol (XMPP): It was formerly known as “Jabber” and was developed for the purpose of instant messaging in order to connect different people by means of text messages. It derives its strength from addressing scheme which helps to connect huge number of people across internet. In context of IoT, XMPP provides a simple and easy way to locate a device. It is useful when data is travelling between distant, usually unrelated points. This protocol is used in a number of applications for instance, to connect the lights of a person’s home to a web server so he/she can access them from his/her smart phone. It is ideal for consumer-oriented applications because of its strengths in security, addressing and scalability.

5. Data Distribution Service (DDS): Its main focus is on devices that directly use device data. It deals with the distribution of data to other devices. The main purpose of DDS is to connect different devices. This protocol can deliver a large number of messages to a number of simultaneous receivers per second. The way in which devices demand data varies considerably from the way in which IT infrastructure demands data. DDS is the

sole technology that delivers the flexibility, the necessary speed and reliability in order to build complex real time applications (Karagiannis, Vasileios, et al. 2015). Military systems, hospitals, wind farms, etc are some of the applications of DDS. DDS has transport priority policy of Quality of Service which permits different applications to handle the importance of a particular topic as a result of which more important data can be prioritized as compared to the less important data (Corsaro, Angelo et al. 2012).

- 6. Message Queue Telemetry Transport (MQTT):** MQTT is designed for the transport of telemetry data (sensor and actuator data). It is very lightweight and therefore suitable for Wireless Sensor Networks, Mobile to Mobile and ultimately Internet of Things scenarios where sensors and actuators communicate with different applications by means of an MQTT message broker. MQTT's primary job is the collection of data that is generated by different devices. As suggested by its name, the main purpose of MQTT is telemetry (remote monitoring). It is a publish/subscribe messaging transport which simply lets the receivers (subscribers) let the publisher (sender) know the fact that they are interested and the receiver (publisher) in turn stores the addresses of these receivers so as to know which message is to be sent where.

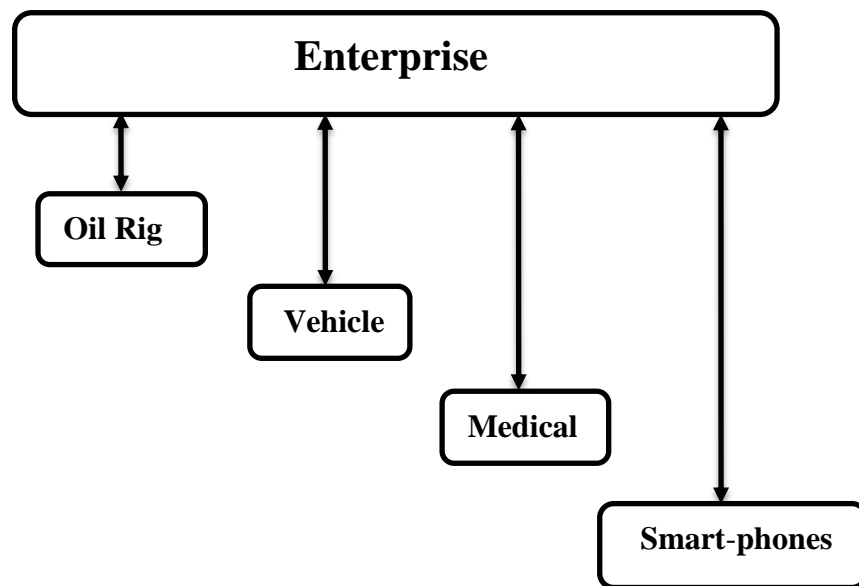


Fig .2. Message Queue Telemetry Transport Protocol.

Message Queue Telemetry Transport protocol is a very light weight and therefore is used for the purpose of connecting small devices to constrained networks. It collects data from devices and

then the same is sent to IT infrastructure. Applications like Facebook Messenger, etc use this protocol. Since losing data is not desirable, so this protocol runs over TCP, which guarantees reliability. MQTT finds its use in various kinds of applications, for instance it is used to monitor a huge oil pipeline in order to check leaks or any kind of vandalism. It is also used in power usage monitoring, intelligent gardening, lighting control, etc.

MQTT ensures reliability by providing three levels of QoS:

- i) **Fire and forget:** The message in this case is only sent once and no acknowledgement is required.
- ii) **Delivered at least once:** Here the message is sent at least one time but can be sent more than once. It also requires an acknowledgement. When the QoS level equals one, this protocol ensures that the server receives the message at least once (Luzuriaga, Jorge, et al. 2015).
- iii) **Delivered exactly once:** It uses four way handshake mechanism in order to make sure that the message is delivered exactly once.

MQTT (formerly known as the MQ Telemetry Transport) is light weight protocol that was actually designed for the purpose of connecting devices having power constraints over networks with low-bandwidth. Andy Stanford-Clark and Arlen Nipper originally designed this protocol. They were assigned the task of inventing a protocol that could be used to connect oil pipelines over networks that were not considered reliable. There are not adequate security features in MQTT protocol. It employs a simple and basic user-password authentication for security sake. Moreover, it does not employ any authorization mechanism (Niruntasukrat, Aimaschana, et al. 2016).

Concept

MQTT employees the publish/subscribe mechanism so as to connect parties that show interest in communicating with each other. A message is sent by the publisher (sender) to a particular topic for which different subscribers (receivers) are waiting so as to receive the message. Both senders as well as the receivers are autonomous and therefore do not need to be aware of each other's presence.

Components of MQTT

Client – Any publisher or subscriber that has the ability to connect itself to a broker rather over a network is known as a client. MQTT makes use of both servers and clients. Clients are of two types: persistent and transient. A persistent client maintains its session with the broker while the broker cannot track the transient clients.

Broker – It is the software that receives all the messages from the publisher side and forwards them towards the subscriber side. The problem with a broker is that it can result in a single point of failure (bottleneck), so it is clustered for the purpose of scalability and reliability. Broker is the entity that makes sure that the messages/data from publisher side reaches the receivers side (Hunkeler, Urs, et al. 2008).

Topic – Topics are endpoints to which the different clients connect. Topics are considered to be the central concept of this protocol (Lesjak, Christian, et al. 2015). Topics are simple strings that are constructed in a hierarchical manner. Case sensitivity is another characteristic of topics.

Topics have two levels:

- i. Single level: apartment +/- humidity
 - * apartment /kitchen/ humidity
- ii. Multiple level (only at the end): apartment / room/ #
 - * apartment /lobby/wall/ temperature.

MQTT Control Packet format

The MQTT protocol works by means of exchanging a number of MQTT Control Packets.

An MQTT Control Packet is made up of the following parts:

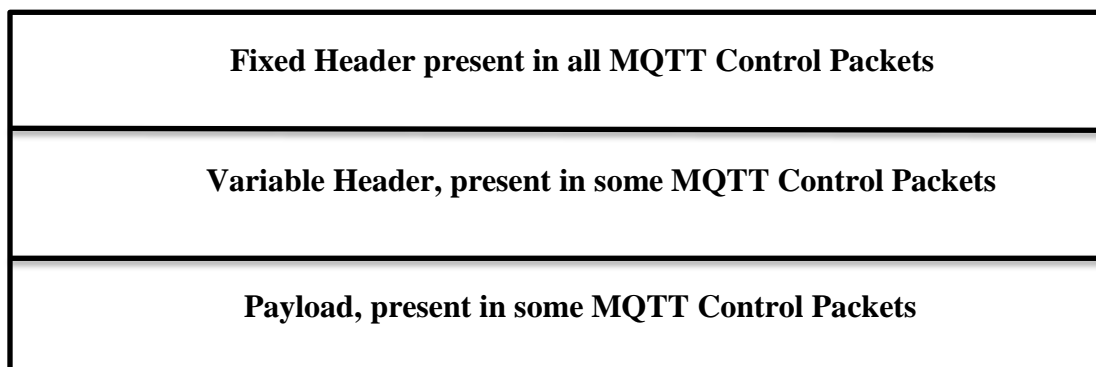


Fig.3. MQTT Control Packet Structure

Each MQTT Control Packet contains a fixed header. Figure illustrates the fixed header format.

Bit	7	6	5	4	3	2	1	0
Byte 1	MQTT Control Packet Type				Flags specific to each MQTT Control Type			
Byte 2	Remaining Length							

Fig.4. Fixed Header Format

Some of the control packets of MQTT header are:

Connect Packet: A Connect Packet consists of the following fields:

1. **Client Id:** This is a unique Id between a client and a broker.
2. **Clean Session:** It has a flag to indicate whether the session must be a persistent one or not. If a session is persistent, then it's clean session is equal to false which means that the broker will store all the subscriptions and missed messages for the client provided the QoS is either 1 or 2. If the status of the flag is set to true then the broker will not store anything for the client.
3. **Username/Password:** These things are simply sent in plain text.
4. **Will Message:** Its purpose is to notify other clients when a particular client disconnects ungracefully. The broker sends this message on behalf of the client.
5. **Keep Alive:** Time-period in seconds in which the client is committed to send a Ping to the broker so that each of them knows whether the other end is alive and reachable.

Publish Packet: The publish packet contains the following fields:

1. **Topic Name:** It is in the form of a string which contains forward slashes as delimiters. For instance, "apartment/ room_number / humidity."
2. **Quality of Service (QoS):** The possible values of Qos are:
 - i. **Level 0-** Atmost once. Zero is the minimal level and it guarantees a best effort delivery. In this case a message won't be acknowledged by the receiver or stored and retransmitted by the sender. This is often called "fire and forget" and provides the same guarantee as the underlying TCP protocol.

- ii. **Level 1-** At least once. When using this level of Quality of Service, it is ensured that a message will be at least delivered one time to the receiver. But it can be also delivered more than once.

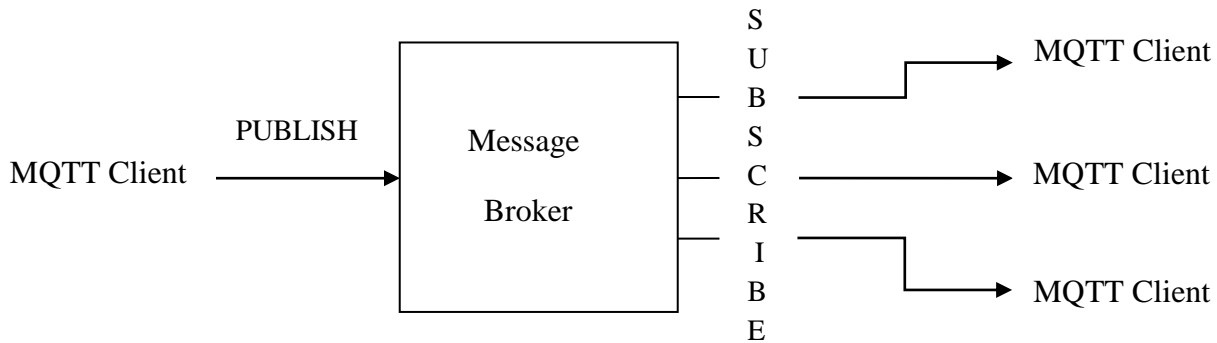


Fig.5. Publish-Subscribe in MQTT

- iii. **Level 2-** Exactly once. The highest QoS is 2, here it is guaranteed that each message is received only once by the counterpart. It is the safest but also the slowest level of quality of service.

3. **Retain Flag:** This message indicates whether the flag saves the latest message for a particular specified topic which as the last known good value. When new clients will subscribe to that topic they will receive the last retained message on that topic immediately after subscribing.
4. **Payload:** It is in the binary form.
5. **Packet Identifier:** It contains a unique identifier that exists between the client and the broker in order to identify a particular message provided the QoS level is either 1 or 2.
6. **Duplicate Flag:** This flag indicates that message is a duplicate copy and has been resent because no acknowledgement was received for it. This is relevant for QoS 0 only.

Subscribe Packet: This packet includes:

1. **Packet Identifier:** This is required only for Quality of Service level greater than zero.
2. **List of Subscriptions:** A SUBSCRIBE message contains a random number of subscriptions for a particular client. Any arbitrary number of messages are valid for a SUBSCRIBE message. Each subscription consists of a topic and a subscription level.

Unsubscribe Packet: This packet includes:

1. **Packet Identifier:** The acknowledgement for an UNSUBSCRIBE packet will have the same packet ID.
2. **List of Topics:** This is the list of topics to unsubscribe from. Here only the topic is specified and not the QoS.

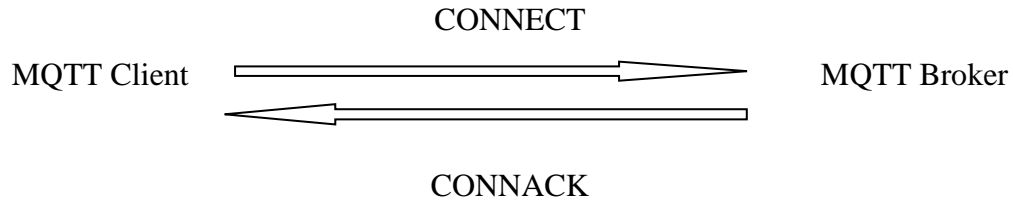


Fig.6. Connection between MQTT client and broker

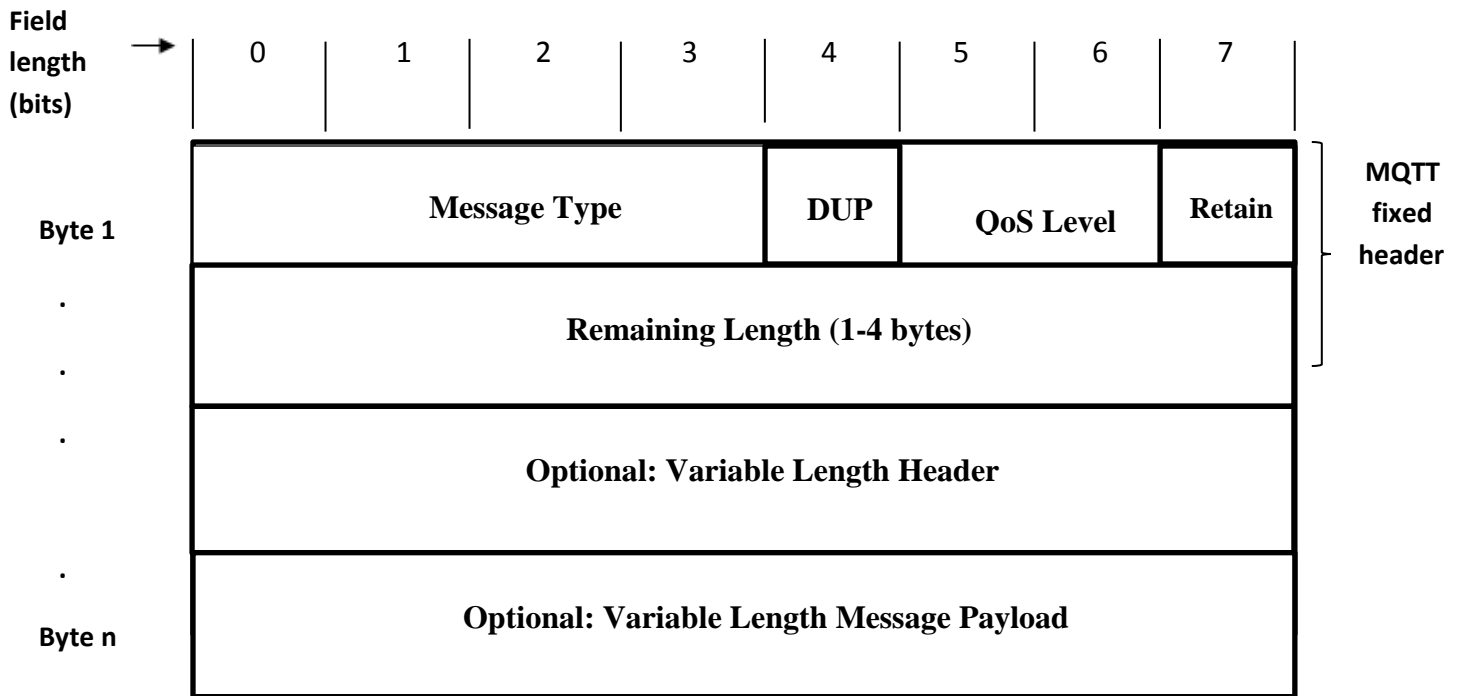


Fig.7. MQTT Message format

MQTT Message format

MQTT messages include:

- a fixed-length header that is mandatory, the length being 2 bytes and

- optional message-specific variable length header and message payload.

MQTT is optimized for unreliable and bandwidth constrained networks (typically wireless networks), therefore optional fields are used to reduce data transmissions as much as possible.

7. **MQTT-SN:** Message Queue Telemetry Transport for Sensor Networks (MQTT-SN) is an extension of the Message Queue Telemetry Transport (MQTT) protocol, which is an open publish/subscribe protocol and was developed to be used on the top of TCP/IP protocol. Formerly it was known as MQTT-S, where S was commonly confused with Security. The main usage of this protocol is to provide a scalable yet simple communication means, at the same time allow an ideal integration of the Wireless Sensor Networks into the conventional/traditional networks. The devices as well as the running applications in an MQTT-SN system can act both as Publisher and Subscribers. The topics are constructed in a hierarchical scheme, e.g. building /kitchen/ humidity. Three levels of Quality of Service (QoS) are supported by MQTT-SN. The devices/clients present inside the WSN communicate with the traditional network by means of a Broker. This protocol supports more than one running gateway, it hence provides more robustness.

MQTT SN has two types of components:

MQTT-S clients and MQTT-S gateways (GWs).

MQTT-S clients are on the WSN side and they enable the Sensor-Actuator devices to access the publish/subscribe services of an MQTT broker that happens to be located on the traditional network. Clients use MQTT-S protocol to connect to the gateway, and the gateway in turn connects to the broker. The gateway's main function is to act as a translator between the MQTT and MQTT-S protocols. An MQTT-S gateway may or may not be integrated with the broker. In the case of stand-alone operation, i.e., the gateway is not integrated into the broker; the gateway uses the MQTT protocol to communicate with the broker.

General Message Format of MQTT-SN

A MQTT-SN message has two parts:

- A header that is 2 or 4 octets in length.
- A variable part (optional).

The message's header part is always present and it has the same fields but its variable part is determined by the type/kind of the considered message.

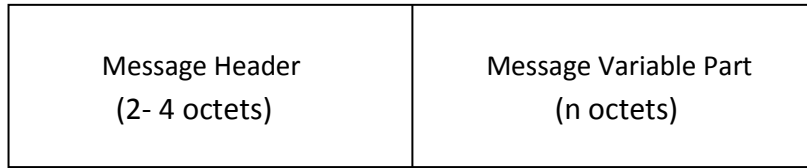


Fig.8. General Message Format

Message Header

Its format is given below:

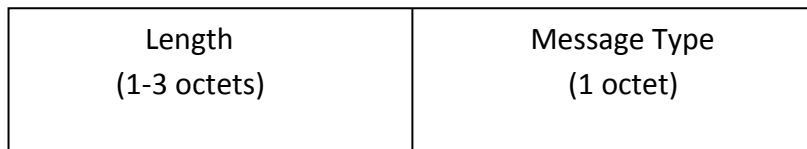


Fig.9. Message Header Format

1. **Length:** This field is either 1 or 3 octets in length and it gives the total number of octets present in the message.
2. **Message Type:** The length of Message Type field is 1-octet and specifies the message type. Some of the message types are:
 - CONNECT: The client sends this message so as to establish a connection.
 - CONNACK: Server sends this message in response to the request for connection from the client side.
 - PUBLISH: Both clients as well as gateways publish data for a particular /certain topic by using PUBLISH message.
 - PUBACK: A client or a gateway sends this message so as to acknowledge the receipt and processing of a PUBLISH message.
 - SUBSCRIBE: Clients use this message so that they can subscribe to a particular topic name.
 - SUBACK: A gateway uses the SUBACK message to acknowledge the receipt and processing of a SUBSCRIBE message. This message is sent to the client.
3. **Message Variable Part:** The type of the message decides the content of the message variable part. Message variable part is made up of the following fields:
 - ClientId: This field holds a string (1-23 character long) that is required so as to uniquely identify a client to the server. The length of this field is variable.
 - Data: The Data field is similar to an MQTT PUBLISH message payload. This field has the application data that needs to be published. The length of this field is variable.
 - Duration: The length of the field named Duration is 2-octets. It determines the duration of a time period in seconds.

- Flags: This field is 1-octet long and it contains the flags given below:
 - a. DUP: If a message is sent for the first time, then this flag is set to “0”. But if the message is retransmitted then this flag is set to 1.
 - b. QoS: Three QoS levels:
 1. Level 0- At most once. Zero is the lowest level and it gives a guaranty of a best effort delivery. Here neither the receiver acknowledges a message nor does the sender retransmit it. This is also called “fire and forget” and provides the same guarantee as is given by the TCP protocol underlying it.
 2. Level 1- While this level of Quality of Service is used, it is ensured that the sender will deliver a message at least once to the receiver. But the sender can deliver the message more than once.
 3. Level 2- Exactly once. The highest QoS is 2, here it is guaranteed that each message is received only once by the counterpart. It is the safest but also the slowest level of quality of service.
 - c. Retain: Same as in MQTT i.e. the Retain message indicates whether or not the latest message for a particular specified topic which as the last known good value is saved by the flag. When a particular topic is subscribed to by new clients, they will receive the last retained message on that topic immediately after subscribing.
 - d. Will: When set, this flag indicates that a Will topic is being asked for by a client.
 - e. CleanSession: Same as in case of MQTT
 - f. TopicIdType: This field determines whether the field TopicId that is present in this message has a short topic name, normal topic id, etc.
- GwAdd: It contains the address of a GW (Gateway) and has a variable length.
- GwId: Its function is to identify a gateway uniquely. The length of this field is 1-octet.
- MsgId: Here the sender is allowed to match a particular message with the acknowledgment corresponding to it.
- ProtocolId: This field is present in a CONNECT message only and is a 1-octet long field. It is identical to the MQTT ‘protocol version’ and ‘protocol name’.
- Radius: This field gives the value of the broadcast radius. The length of this field is 1-octet.
- TopicId: The value of the topic id is present in the TopicId field and the length of this field is 2-octets.
- TopicName: The TopicName field contains the topic name and has a variable length.
- WillMsg: This field contains the Will message. It also has a variable length.
- WillTopic: This field contains the Will topic name and has a variable length.

1.4 APPLICATIONS

IoT has got huge amount of potential for creating and evolving new intelligent applications in almost every field (Borgia, Eleonora. 2014). The applications of Internet of Things aim at making an individual's life quite easy. Some of the applications of Internet of Things are as follows:

1. Healthcare: This is one of the most important applications of IoT. It can help in taking better care of patients by recording and monitoring the records of different patients and reducing infection risks because of non-intervention of humans (Pande, Prajakta, et al. 2014). Doctors get the benefit of practice knowledge from a database of hundreds of thousands of guidelines that can lead to better patient care. An Internet of Healthcare Things (IoHT) can bring about a revolution in healthcare, medicine and consumer health (Atzori, Luigi, et al 2010).

2. Smart Homes: Actuators and sensors are used in homes to make life more comfortable. For instance room heating and cooling can be adapted according to weather and our preferences, lights of rooms can change according to time. It leads to efficient use of energy and hence reduced bills.

3. Assisted Driving: Different vehicles along roads and rails provided with sensors, actuators, etc may give information to the driver which is quite important in order to allow better navigation as well as safety. Government departments will also benefit from this as they would get accurate information about various things like pattern of traffic for purpose of planning.

4. Mobile Ticketing: Different banners, panels or posters, etc which provide information such as cost, schedule, etc about transportation services can be provided with a Near Field Communication (NFC) tag and a unique identifier. The user can then obtain information regarding the transportation service from web by means of hovering a smart phone over the NFC tag. The phone automatically gets information such as stations, number of passengers, cost, etc from associated web services and hence lets the user to purchase tickets accordingly.

5. Wearables: These devices are embedded with RFID tags and sensors so as to track the activities of a person. Wearables also help in monitoring health, especially that of elderly people. Wearables like smart watches, bluetooth handsets, etc allow people to access data hands free from Wi-Fi networks.

6. Smart Pills: Smart pills are sensors that are ingestible and can be swallowed. They are used to record different physiological measures. They can also be used for confirmation of the fact that

whether or not a patient has taken the medicines that have been prescribed and these smart pills can also record the effects of the medicines on the patient (Al-Fuqaha, Ala, et al. 2015).

7. Smart Retailing and Supply-chain Management: Internet of Things with RFID provides large number of benefits to the retailers. By using RFID equipped products, retailers are easily able to keep a track of the stocks of things and even detect shoplifting in their shops. Moreover the retailer can also create charts as well as graphs in order to make effective strategies (Schneider, et al. 2013).

8. Waste Management: Levels of rubbish in containers can be detected in order to optimize the collection of trash (Farooq, M. U., et al. 2015).

1.5 CHALLENGES FACED BY INTERNET OF THINGS

- 1. Security Concerns** –IoT aims to interconnect hundreds and thousands devices, providing security to so many devices is not an easy task. If these devices are not strongly secured, they can be used by attackers to harm other devices in the network. All this will result in scenarios such as loss of personal information and this will in turn lead to loss of entire trust and faith between internet connected devices and people who use them. Therefore to evade such problems, it is extremely critical to ensure the security and reliability of the internet applications so as to promote IoT.
- 2. Privacy issues** - Internet of Things has made it possible to track people. Government and private agencies can easily track people with the help of IoT devices. These IoT devices collect data about users without their permission and then analyse it for their own purposes.
- 3. Standardization and Inter-operability issues** – IoT has the potential to interconnect large number of devices over the internet. But the problem here is that these devices are heterogeneous which means they have different requirements. So standardization is required. It means having a common platform for diverse applications and is important as it will lead to a better interoperability But the process of standardization is quite complex and cannot be achieved so easily.

CHAPTER 2

REVIEW OF LITERATURE

In this thesis a systematic literature review was performed about the ‘Traffic Prioritization in an MQTT Gateway’ but it was found that no work related to this topic had been done earlier.

Search Execution:

1. First of all a search was performed using the selected search string and then the obtained results were sorted by relevance.
2. Then from each topic, the full citation, abstract of the paper and the full text were retrieved.
3. Finally the topics matching the inclusion criteria were examined further. In case the abstract of the article/paper did not contradict the exclusion criteria, it was fully read. If the paper/article was still conforming to the study selection criteria, it was selected.

Accordingly, first the keyword MQTT-SN Gateway was used for search and it fetched about 132 papers. Then in order to make the search more specific, keyword ‘MQTT-SN Gateway in Internet of Things’ was used and it fetched about 98 papers. Then to further narrow down the search area, the keyword ‘Aggregating MQTT-SN Gateway in Internet of Things’ was used and finally yielded about 37 papers.

Therefore, the literature review given below includes articles on Internet of Things as well as some of the 37 papers selected by means of inclusion exclusion principle.

2.1 Internet of Things (IoT): A Literature Review (Madakam et al. 2015) - This paper gives an idea about the different types of architectures of IoT. The main problem of IoT is that it is so vast that there is no uniform architecture has been proposed for it. This paper also gives details of some of the proposed architectures for IoT. Some of the architectures discussed here are:

- European FP7 Research Project.
- ITU Architecture: A five layered architecture.
- IoT Forum Architecture: It divides IoT into three parts.

2.2 Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges (Khan, Rafiullah et al. 2012) - This paper has discussed the evolution of internet towards the present day Internet of Things and how just human-human communication have been transformed into human-human, human-device and device-device communication.

This paper has also put forward architecture of Internet of Things that consists of five layers. Some applications have been highlighted like prediction of natural disasters, smart cities, smart homes, industry applications, etc.

Certain key challenges that are being faced by Internet of Things have been discussed. Some of them are:

- Naming and Identity Management- Since Internet of Things connects innumerable objects, each object should have an identity that is unique, over the internet. Therefore an effective mechanism for naming as well as the identification of things is required.
- Standardization and Interoperability-There are many vendors that provide services that cannot be accessed by others. So standardization is a must for interoperability of all devices in Internet of Things.
- Information Privacy-Internet of Things uses different enabling technologies like RFID, 2D and since all kinds of day-to-day life objects will have tags for the identification purposes that will obtain the information about a particular object, privacy must be insured.

2.3 Architecture of Things and its key Technology Integration based on RFID (Zang, Minghui et al. 2012) - This paper puts forward the concept architecture of Internet of Things consisting of six layer which has been derived from the working flow of Internet of Things and the basic architecture consisting of three layers. This paper also presents a Zigbee based automatic recognition system which is developed by integrating wireless sensor networks and radio frequency identification.

2.4 A Survey on Application Layer Protocols for the Internet of Things (Karagianis, Vasileios et al 2015) - This paper discusses the protocols that are used for communication purposes in Internet of Things at the application layer. Different protocols have been studied and compared in order to know how well they are suited for Internet of Things on the basis of factors such as how much energy they consume, reliability, etc.

Several factors influencing the selection of protocols at the application layer have been identified and the most important of them are: consumption of battery, computational speed, ability to communicate with other devices.

2.5 The Internet of Things vision: Key Features, Applications and Open Issues (Borgia Eleanore 2014) – This paper suggests three different phases with the help of which real world interacts with the digital world.

This paper also gives an idea about key features of IoT:

- General requirements and features:
- Communication requirements: are related to the data or traffic which the IoT devices generate as well as transmit.

2.6 Spamming the Internet of Things: A Possibility and its probable solution (Razzak, Faisal 2011) – This paper throws some light on the possibility of spamming the Internet of Things and also proposes a feasible solution for the same. Here the main idea is that spammers can utilize 2D barcodes so as to trick internet users to view content over the internet that is unrelated to them and even ruin the authenticity of the original content. Therefore 2D bar codes are most vulnerable to spamming. To avoid spamming, the paper puts forward a technique to guarantee the integrity of content as well as establishes the identity of creator of the content. All this is achieved by using of Digital Signatures in order to digitally sign the content that is present inside the 2D barcodes.

2.7 Internet of Things- Promise for the future? An Introduction (Coetzee et al 2011) - This paper presents the advancement of utilization of internet starting from computers to humans and now finally to things, allowing the development of many new applications and services. Some application areas have also been highlighted.

This paper also throws some light on Fleisch's value drivers. This paper also discusses some challenges that are being faced by Internet of Things at present and need to be addressed. The challenges that have been discussed include- privacy, standardization, interoperability, etc. It also highlights the efforts made by some multinational companies like IBM, Microsoft, etc in recognizing the commercial potential of internet of things.

2.8 Security of the Internet of Things: perspectives and challenges (Jing Qi et al 2014) - This paper has considered the three levels of architecture of internet of things comprising of – perception, transport and application layer and has focused on the security issues of each of these layers. The various issues discussed are as follows-

1. Perception Layer-

- Security issues of RFID technology-uniform coding, conflict collision, privacy protection, etc.
- Security issues of wireless sensor networks.
- Problems of heterogeneous integration.

2. Transport Layer-

- Wi-Fi security analysis.
- Ad hoc security analysis issue.
- Core network

3. Application Layer-

- Security Threats.
- Service interruption and attack issue.
- Investigate audit issues.

This paper also discusses the security issues of various applications of internet of things such as smart home, smart city, etc. It also compares the issues of security between traditional network and internet of things and finally concludes that internet of things resides in more dangerous surrounding environment having very limited resources, thus lightweight solutions should be used for the security of internet of things.

2.9 MQTT-S – A Publish/Subscribe Protocol For Wireless Sensor Networks (Hunkeler, Urs et al 2008) - This paper throws light on MQTT-S, a modification of the Message Queue Telemetry Transport protocol in order to meet constraints of Wireless Sensor Networks. The publish/subscribe standard meets many of the requirements for the purpose of WSNs communication since it can hide the network topology and permit delivery of data on the basis of an individual device's interests rather than its address. An important advantage of MQTT-S over other Internet of Things protocols is that MQTT-S is based on a well-established publish/subscribe protocol already widely used. Implementation of MQTT-S has brought forth many challenges of WSNs. The implementation also shows that this protocol can be easily implemented on devices that have only limited resources.

2.10 Securing Smart Maintenance Services: Hardware-Security and TLS for MQTT (Lesjak, Christian et al 2015) - According to this paper a secure end to end connection is required between deployed devices and the remote maintenance service provider for the process

of remote data acquisition for smart maintenance services. In this paper the authors studied a use case of AVL Particle Counter (APC) and the MQTT Information Broker (MIB) and then investigated the client authentication problem in order to the Message Queue Telemetry Transport protocol secure. The proposed design utilizes TLS concept in order to append a secured layer of communication underneath Message Queue Telemetry Transport protocol. The proposed system also utilizes a hardware security controller which performs client authentication by means of TLS.

2.11 Semantic Data Extraction over MQTT for IoT centric Wireless Sensor Networks (Wagle et al, 2016) -According to this paper semantic extraction of data is encouraged by several of the functionalities provided by MQTT. These functionalities include message retention, which makes it easy to add new devices as well as makes it easy to integrate these new devices into the loop. Availability of topics for the purpose of subscription and publishing leaves the procedures of data routing largely redundant. Data organization is done at the protocol level; therefore, no heavy mechanisms are required for channelling the data to designated buffers that are present at the program level. The Quality of Service and Last Will and Testament present options for reliability for MQTT, hence making it usable in a number of situations that face various kinds of constraints. An important benefit of semantic data extraction by means of user initiated events that are asynchronous is that a training set which predefined is not needed. The system can learn by itself and does not require any prior knowledge of the system. Thus, designing such a kind of system does not need precise knowledge of the process, since the algorithm can extract data of significance on its own. A problem with this technique is that it takes a lot of time to generate a training set of its own.

2.12 Handling Mobility in IoT applications using the MQTT protocol (Luzuriaga, Jorge et al 2015) - This paper focuses on providing a method the in order to adapt the Message Queuing Telemetry Transport protocol to mobile scenarios. It will be advantageous to the developers as they will not have to explicitly take into consideration the changes in the point of attachment to the network. The proposed method continues to use publish subscribe approach but it separates the data generation process by process of data sending. This is based on intermediate buffering technique. The separation or decoupling permits the recovery when communication channel goes through disruption periods, even if the disruption periods are quite frequent and last for a few seconds, a condition where Transmission Control Protocol (TCP) cannot recover from. For the

implementation purposes, to generate workload for the system of messages queuing, a test application called `mqttperf` has been developed. This application uses the Paho library which is MQTT protocols open source implementation. This paper then measures the variability in jitter and information loss in an extended wireless network. It was observed that the mean jitter introduced by using this approach was in the range of 35 to 38 seconds. It was also seen that this approach guaranteed that there was no loss in information during hand-off of the publisher subscriber node, hence making the messaging system that is based on Message Queuing Telemetry Transport protocol robust as well as capable of guaranteeing delivery of messages without any losses in presence of mobility of publisher node. Losses in messages would only be present in case the roaming time tends to infinity, creating a situation which is likely to cause system buffer capacity to be overloaded and memory leaks.

2.13 Authorization Mechanism for MQTT-based Internet of Things (Niruntaskurat, Aimaschana et al 2016) - This paper presents the design as well as the implementation of a mechanism for authorization for the Message Queuing Telemetry Transport-based Internet of Things service. The framework OAuth 1.0a has been used and modified so as to adapt the restrictions faced by Internet of Things devices. At authorization time, a single credential set consisting of Token Secret and Access Token Access are passed over an insecure and unencrypted channel between the authorization endpoint and the device and these are susceptible to be stolen. But the mechanism for authorization used in this paper makes use of two credential sets. The second credential set consists of Device Secret and Device ID is sent to the users and is then embedded offline into the memory of the device. By means of real service and experiments, the mentioned process guarantees to work as proposed. Moreover, the user experience is not affected by the overhead that is incurred.

2.14 Análise de Desempenho de Brokers MQTT em Sistema de Baixo Custo (Performance Analysis of MQTT Brokers in Low Cost System) (Torres, Rocha et al 2016) - This paper presents a performance analysis (CPU usage, memory consumption and message throughput) of MQTT brokers in a low-cost hardware, the Raspberry Pi 2 Model B. The objectives of the analysis are to ascertain which MQTT broker implementation is best suited to the limitations of the hardware and to verify if the Raspberry Pi 2 is actually able to function as a gateway in a sensor and actuator network for the internet of things (IoT). The results showed that the Raspberry Pi 2 can handle large number of connections and that the implementation in Erlang

(eMQTT) obtained the results in data throughput, while the implementation in C obtained the lowest CPU load and memory consumption.

2.15 Multi-Protocol Transport Layer QoS: An Emulation Based Performance Analysis for the Internet of Things (Wilcox, James et al 2016) - This paper demonstrates that wisely chosen transport protocols can increase the efficiency of resource usage of a network under specific network conditions. Selecting real time transport protocols in real time makes possible the achievement of a distributed embedded system having different actors capable of reacting to application specified Quality of service as well as varying network conditions. vNET, which is a custom, visualization based, distributed network emulation test bed has been presented as well as validated using an Message Queue Telemetry Transport (MQTT) performance analysis before it was used to validate the premise of multi-protocol transport layer QoS.

2.16 Lightweight Internet Protocols for Web Enablement of Sensors using Constrained Gateway Devices (Bandyopadhyay, Soma et al 2013) - Lightweight Internet protocols are nowadays greatly being used in ubiquitous environment in order to optimize the usage of resources of constrained devices such as a smart mobile gateway. This paper puts forward a study on the different such protocols in order to optimize network resources, the usage of energy, and computation cost of a constrained gateway device. Feature wise categorization and comprehensive analysis of existing dominant protocols, such as MQTT (message queue telemetry transport), CoAP (constrained application protocol) have been provided so as to achieve improved understanding of the existing issues as well as gaps in this domain. This paper also identifies the best suited application areas for each protocol on the basis of results corresponding to the typical requirement of resources as well as performance attributes.

2.17 The Data Distribution Service: The Communication Middleware Fabric for Scalable and Extensible Systems-of-Systems (Corsaro, Angelo, et al 2012) - This paper throws light on the DDS protocol of Internet of Things. Various components in the DDS standard along with the Quality of service policies have been studied. The authors have also provided the guidelines for efficient and scalable SoS by means of DDS integration.

2.18 Toward better horizontal integration among IoT services (Al Fuqaha, Ala et al 2015) - This paper throws light on the major shortcomings of the current IoT protocols and also suggests a rule-based intelligent gateway with the help of which the gap between existing IoT protocols will be bridged in order to enable the effective integration of horizontal IoT services. This

intelligent gateway does enhance the protocol fragmentation in IoT context but does not address the cause of fragmentation. This paper proposes an enhanced MQTT protocol version that mitigates the problems prevalent in the existing MQTT protocol. No work has been done on MQTT gateway to prioritize traffic.

2.19 Internet of Things: A Survey on Enabling Technologies, Protocols and Applications

(Al Fuqaha, Ala et al 2015) - This paper gives an overview of the Internet of Things (IoT) including its protocols, enabling technologies and application issues. A thorough summary of the application issues and protocols has been provided along with some of the key challenges that are being faced by Internet of Things. The authors have also discussed big data analytics, fog and cloud computing in context of Internet of Things.

2.20 A Scalable and Sustainable Web of Buildings Architecture (Bovet, Gerome 2015)

-The authors discuss how Web technologies in context of smart buildings are beneficial to make the application level homogenous, resulting in intelligent/smart and reusable entities. Emphasis has also been laid on how the REST architectural style can be applied to all the levels, in order to homogenize the entire ecosystem. It also discusses the MQTT gateways, both transparent as well as aggregating. But no work has been done on prioritizing traffic on these gateways.

2.21 IoT integration on Industrial Environments (Diaz-Cacho, Miguel, et al 2015)

-This paper proposes that the performance of IoT devices that are inserted into an IP backbone including industrial environments may be proved by IoT smart gateways without degrading the network load. An important feature of this IoT gateway is the usage of selected data structures along with the implementation of deadband models. The data that is sent across shared IP networks can therefore be organized, selected and concentrated with the help of the IoT gateways. The results of simulation performed using Networked Control Systems show that it is feasible to apply these models to the gateway.

2.22 IoT Home Gateway for Auto-Configuration and Management of MQTT Devices (S.M

Kim et al, 2015) - This paper puts forward a proposal for an IoT Home Gateway which supports the abstracted device data in order to remove heterogeneity, provides for device discovery by DPWS and also the auto configuration for constrained devices like Arduino. Moreover, the IoT Home Gateway gives lightweight information delivery by means of MQTT protocol. The implementation results which control the device in accordance to the home energy saving scenario have also been shown. In order to satisfy the user's requirements regarding the energy

management in home, an overall architecture of IoT service has been proposed. This proposed architecture consists of the Internet of Things Home Gateway so as to collect data from devices, the Web Based Service Definition Engine in order to define the required services of the user along with the IoT Service. IoT service in terms of home domain is in need of easy and effective ways to manage a number of devices and appliances. Therefore, the home environment greatly needs a gateway which is capable of providing dynamical registration of devices as well their discovery.

2.23 Secure MQTT for Internet of Things (IoT) (Meena Singh et al, 2015) - The authors of this paper designed as well as implemented secure MQTT protocols (SMQTT and SMQTTSN) having a new secure publish command called “Publish”. This command publishes the encrypted data by means of optimizing some parameters as well as computation algorithms. Moreover, the security analysis of SMQTT i.e. secure MQTT protocol under different attack scenarios have been studied and the feasibility of SMQTT for the purpose of distributed Pub-Sub architecture is proposed on an end-to-end basis.

2.24 Integrating MQTT and ISO/IEEE 11073 for Health Information Sharing in the Internet of Things (Yuri F. Gomes et al, 2015) - This paper discusses the use of the lightweight MQTT (Message Queue Telemetry Transport) protocol along with the ISO/IEEE 11073 standard. It also suggests some new ways of connecting PHDs in home networks and the Internet by the use of MQTT Brokers that will in turn reduce the amount of data traffic. Besides the various advantages for the purpose of constrained resource devices, MQTT publish/subscribe communication model has some other desirable features like the automatic discovery of devices by means of Brokers. In terms of connected health, the proposed architecture gives a new set of scenarios, wherein MQTT brokers can be used for the purpose of sharing health information in home networks as well as with the Internet.

2.25 Correlation Analysis of MQTT Loss and Delay According to QoS Level (S. Lee, H. Kim, D. Hong and H. Ju, 2013) - In this paper, the authors have analysed the process of MQTT message transmission consisting of real wireless/wired publish client, subscribe client as well as broker server. By means of transmitting messages through 3 QoS levels with different sizes of payloads, they have captured packets in order to analyse the end-to-end delays as well as message loss. The experiments were performed in a kind of realistic network environment than the existing simulated environment. Moreover, the authors have also suggested the results of

correlation analysis of the end-to-end delay and the message loss under different QoS levels and payloads. The results of experiment suggest that the end-to-end delay is notably associated with message loss under different payloads.

3.1 PROBLEM FORMULATION

Message Queue Telemetry Transport protocol is extremely light weight and is used to connect small devices to constrained networks. The main job of MQTT is to collect data generated by devices. As its name suggests the main purpose of this protocol is telemetry or in other words remote monitoring. It is a publish/subscribe messaging transport i.e. it simply lets the receivers also called subscribers let the publisher know that they are interested and the receiver or publisher in turn stores their addresses in order to know where to send which message.

It collects device data and transfers the same to IT infrastructure. It is used by applications like Facebook Messenger. Since one does not want to lose data, so this protocol runs over TCP, which ensures reliability. MQTT finds its use in various applications, for instance to monitor a huge oil pipeline in order to check leaks or any kind of vandalism, power usage monitoring, intelligent gardening, lighting control, etc.

Concept

MQTT uses the publish/subscribe pattern to connect parties that are interested in communicating with each other. The publisher (sender) sends a message to a particular topic for which a number of subscribers (receivers) are waiting in order to receive the message. The subscribers as well as the publishers are autonomous; they therefore do not need to be aware of each other's presence.

Components of MQTT

- **Client** – A client can be any publisher or subscriber which connects itself to a broker rather, a centralized broker over a network. It's important to note that MQTT has both servers and clients. Clients can either be persistent or they can be transient. A client's session with the broker is maintained by the persistent clients while the broker does not track the transient clients.
- **Broker** – The software that receives all the messages from the clients that act as publishers and forwards them towards the clients that act as subscribers. Since the broker can result in a single point of failure or become the bottleneck, it is therefore clustered for the purpose of scalability and reliability.

- **Topic** – Topics are endpoints to which the different clients connect. Topics are quite simple strings that are hierarchical in nature and are encoded using UTF-8, delimited by a forward slash. Topics are case sensitive. Topics have two levels:
 - Single level: building /+/ humidity
 - * building /kitchen/ humidity
 - * building /living room/ humidity
 - Multiple level (only at the end): building / room/ #
 - * building /kitchen/wall/ temperature.
- **Connection** – MQTT can be used by clients in light of TCP/IP. 1883 is the standard port used by brokers and it is not a protected port. Port 8883 is utilized by brokers who support TLS/SSL. The broker and the clients depend on digital certificates for secure correspondence. MQTT is streamlined for correspondences over networks where bandwidth or data transfer capacity is at a premium or where there is irregular network connection. However MQTT requires a basic system like TCP/IP, which gives a lossless association capacity and it is very much complicated for extremely simple, and minimal cost devices like WSN. So a somewhat different IoT protocol for WSN is created. It was called MQTT-SN which a publish/subscribe protocol created for the purpose of wireless sensor networks. This protocol can be considered as a form of MQTT which has adjusted itself to the bizarreness of a remote correspondence environment. Three types of MQTT-SN components are present namely clients, gateways (GW), and forwarders.

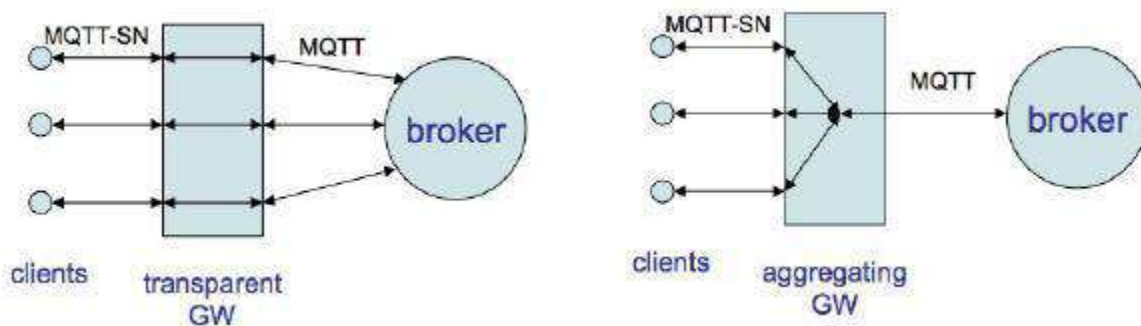


Fig.10.Types of MQTT gateways (Stanford et al, 2008).

MQTT Gateway

Transparent Type: For each and every connected client rather than MQTT-SN client a gateway will be composed and it will maintain a connection to the MQTT server.

Aggregating Type: Instead of having a MQTT association for each associated client, an aggregating gateway will have just a single MQTT association with the server. All exchange of messages between an aggregating gateway and an MQTT-SN client finish at the gateway. The gateway then chooses which data will be offered to the server. In spite of the fact that its execution and implementation is far more complicated than the one of a transparent gateway, an aggregating gateway might be useful if there be a WSN with extremely large number of nodes since it lessens the number of MQTT associations that the server needs to support simultaneously.

Problem- As the number of nodes in a network grows, so does the blocking delay (or queuing delay) suffered by packets of some MQTT-SN nodes. However, there are time-critical applications for which this delay is not tolerable. Therefore a prioritization of MQTT-SN nodes is required so that the nodes having a high priority are blocked for minimum possible time. We propose to prioritize traffic in an MQTT gateway by using different scheduling algorithms, which shall mitigate this delay.

In order to prioritize traffic in an MQTT-SN Gateway, the following concepts have been used:

1. Aggregating Gateway

An aggregating gateway has a single MQTT association with the server instead of having an MQTT association for each associated client as is the case with transparent gateway. All exchange of messages between an aggregating gateway and an MQTT-SN client finish at the gateway. The gateway then chooses which data will be offered to the server. In spite of the fact that its execution and implementation is far more complicated than the one of a transparent gateway, an aggregating gateway is useful if there is a WSN with extremely large number of nodes since it lessens the number of MQTT associations that the server needs to support simultaneously.

Since an aggregating gateway is used; therefore only one connection will be established with the MQTT broker. MQTT-SN gateway actually acts as a forwarder that simply forwards data packets from different sensor nodes towards the MQTT broker.

Here aggregating gateways are used because it is not feasible to maintain multiple connections with the server as in case of transparent gateways where the server has to support number of MQTT connections concurrently.

2. Differentiated Services

In Differentiated Services traffic is classified into a set of classes and a priority based treatment is provided according to these classes.

Since the aim is to prioritize data, therefore the concept of Differentiated Services has been used in which a packet's class can be assigned in the packet directly. It divides traffic into a small number of classes and allocates resources on a per-class basis. The Differentiated Services (DS or DiffServ) architecture was developed in response to the need for relatively simple, coarse methods of providing different levels of service for Internet traffic. The advantage of DS is that many traffic streams can be aggregated into one of a small number of behaviour aggregates, forwarded using the same PHBs (per hop behaviour) at the router, thereby simplifying the processing and associated storage. It offers services for traffic on a per-class basis rather than on a per-flow basis at the internal nodes.

3. Abstract distribution of nodes

We have taken into consideration an abstract distribution of nodes where the sensor nodes gathering information are arranged in different abstract classes. An abstract class is a group of sensor nodes which serve or belong to the same traffic class. For each traffic class, there will be a single priority group. It means that all the nodes belonging to a particular traffic class will have the same priority.

4. Buffers

The concept of buffers is used. Buffers are used to temporarily store traffic/ data packets before they can be forwarded. Let us assume that $T_1, T_2, T_3 \dots T_n$ are buffers with different traffic classes where T_i is the i th traffic class, then for some traffic classes i and j $T_i >_p T_j$ for $i < j$ i.e. higher priority will be assigned to lower values.

To understand the need of buffers we will consider the pigeon- hole principle. Let us assume that there are n pigeonholes and each pigeonhole can contain only one pigeon. Now let us suppose we have m pigeons such that $m > n$. It means that some pigeonholes contain more than one pigeon.

This is analogous to our need of buffer for all traffic classes. In other words each buffer can contain more than one data packets. Data packets coming from different sensor nodes need to wait for their turn be sent to the MQTT broker. Therefore buffers are required to store data packets/traffic until they are forwarded towards the broker.

When packets arrive at a server, they have to be processed and transmitted. A server can only process one packet at a time. If packets arrive faster than the rate at which the server can process them (such as in a burst transmission) the server puts the packets into the queue (also known as the buffer) until it can process them. Delay can also vary from packet to packet. As a queue begins to fill up due to traffic arriving faster than it can be processed, the amount of delay a packet experiences going through the queue increases.

The number of slots required per buffer will be decided on the basis of delay. Suppose there are two buffers B1 and B2 of length L1 and L2 respectively such that $L1 < L2$.



Fig.11. Buffers

Again suppose that the server takes 1 time unit to process each packet such that all other packets have to wait in the buffer for their turn to be processed. Therefore in B1 third packet will have to wait for two time units before it is processed. Likewise in B2, fifth packet will have to wait for four time units for its turn to be processed.

Hence two cases arise:

1. Less buffer space /capacity: If the buffer capacity is less as in B1, as more and more packets arrive at the buffer, they will be dropped.
2. More buffer space /capacity: If the capacity of the buffer is quite large as in B2, packets will have to face worst case wait i.e. they will have to wait for quite a long time for their turn to be processed. The maximum queuing delay is proportional to buffer size. The longer the line of packets waiting to be transmitted, the longer the average waiting time is.

So in order to handle the above stated problems, an optimal length buffer needs to be found in order to limit both packet dropping as well as waiting time of packets.

Basic queuing theory assumes traffic/ packet arrivals are Poisson distributed with rate λ . The service times are exponentially distributed with rate μ . The ratio of λ/μ is known as utilization ρ . If this ratio (utilization) is greater than 1, then packets arrive faster than they can be served, and so the line will grow without bound. If the ratio is less than 1, then the line will reach some steady state on average.

By using all of the above mentioned concepts, we propose to design an MQTT-SN Gateway Model so as to prioritise data arriving at the Gateway.

3.2 OBJECTIVES OF THE STUDY

The objectives of this study are as follows:

1. To study the packet delay/latency in an un-prioritized MQTT IoT gateway using Matlab.
2. To add priorities to MQTT packets streams using Matlab.
3. To establish and study various priority scheduling algorithms (FIFO, RR) on MQTT packets, in an MQTT gateway using Matlab.
4. To analysis and compare the delay/latency on packets in an MQTT gateway (with and without priorities) using Matlab and implementation.

3.3 RESEARCH METHADODOLOGY

The research will be carried out in the following steps:

1. Installation of Matlab and study the concepts of Publish/Subscribe message model.
2. Develop MQTT gateway model in Matlab and measure packet delays/latency in the gateway.
3. Implement priorities in MQTT packets and addition of packet scheduling (FIFO, RR) algorithms to the MQTT gateway model.

3.3.1 Simulation Tool Used

MATrix LABoratory or MATLAB is a programming package that is particularly designed for fast and easy scientific computation. It actually has a large number of inbuilt functions for different computations. It has many toolboxes that are designed for different research disciplines.

MATLAB is a generic simulator which is used for simulating MQTT gateway model. MQTT is an IoT protocol which sits on top of a TCP/IP stack as show in the figure below. As such, MQTT packets are encapsulated by the TCP/IP packets in an IoT. MATLAB is available

for all the popular platforms (OSX, Linux, and Windows). Here, we will use Matlab on a Windows platform for building and simulating our model.

3.3.2 Algorithm Steps

Step 1: Collect information: In this step, we are focusing on collecting the information about MQTT gateway (GW), MQTT protocol specification, and creation of aggregating MQTT gateway (without priorities) model in matlab.

Step 2: Create packet-delay graph: Delay is main factor; we need to focus on when simulating and analysing basic MQTT GW model. From the randomly generated traffic an average delay graph of MQTT-SN node packets will be found. A comparison between basic MQTT-GW and prioritized MQTT-GW will be made using this graph.

Step 3: Create Prioritized GW model: In this step we create two types of GW each with a unique packet scheduling algorithm, like Round Robin (RR) and FIFO.

Step 4: Generate packet-delay graph of prioritized GW: In this step we create delay graphs, for each GW of step 3, as was done in step 2.

Step 5: Comparative analysis: Here we compare the graphs generated in step 4 with that of the graph generated in step2 in order to see how each scheduling algorithm in a MQTT-GW has an impact on the delay of priority nodes.

Step 6: Summary: A conclusion is drawn about the best suited scheduling algorithms.

DATAFLOW DIAGRAM OF THE PROPOSED DESIGN

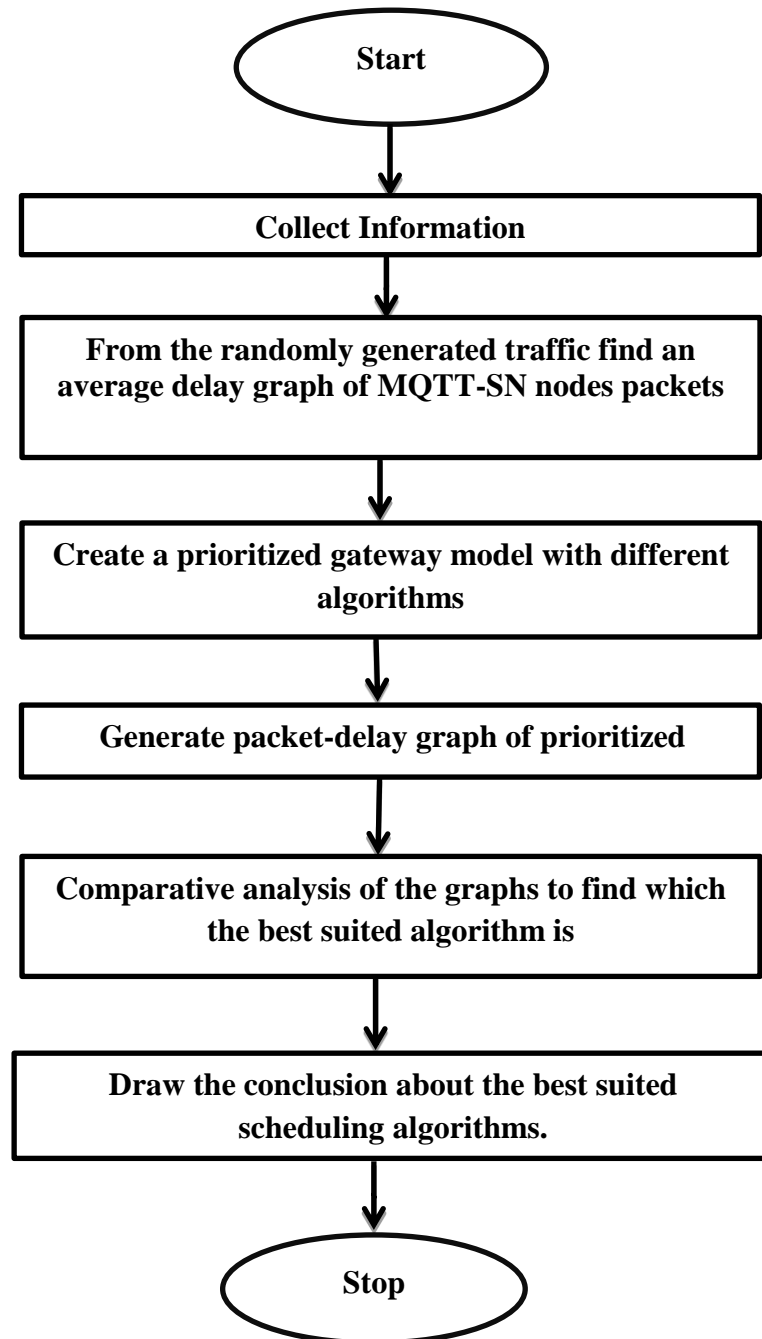


Fig.12. Dataflow Diagram of Proposed Method

4.1 ASSUMPTIONS

Size of each packet is equal to a TCP packet.

The queue lengths are same.

Traffic arrivals are Poisson Distributed.

4.2 SIMULATION PARAMETERS

Parameters Used	<ol style="list-style-type: none">1. Entities Departed from Server2. Server Utilization3. Average Waiting Time of Packets
Simulation Time	50s

Table.1. Simulation Parameters

4.3 SIMULATION

We have added priorities to MQTT packet streams and used two algorithms namely Priority Scheduling algorithm and Round Robin algorithm on an MQTT Gateway in order to prioritize the traffic arriving at the gateway. The packets are arriving randomly (Poisson's Distribution) and the size of each packet is same (equivalent to TCP packet). We have created an artificial dataset of traffic arriving at an MQTT-SN Gateway using Wireshark.

Priority based Packet Scheduling

Packets with three different priority levels are generated by the three packet generators. Each packet has attributes representing the priority and the task execution time. The tokens are sorted according to ascending priority in the Priority-Based Task Queue just before the CPU. The tokens are then passed to the CPU for task execution.

Results and Displays

Following scopes present the simulation results:

High priority tasks completed

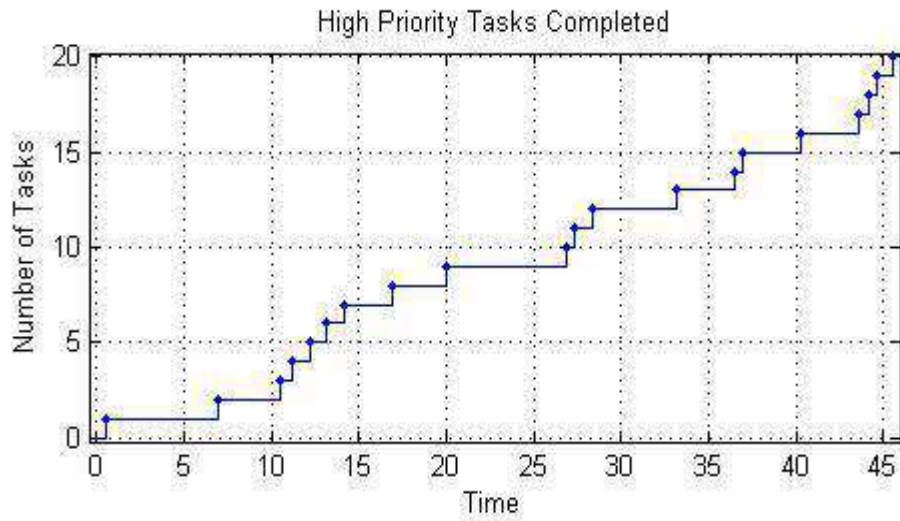


Fig.13. High priority tasks completed

As soon as a high priority packet arrives at the server, it is served. Here the service time of the server is 0.5, so the server will take time $T=0.5$ to complete the processing of the packet.

Medium Priority Tasks Completed

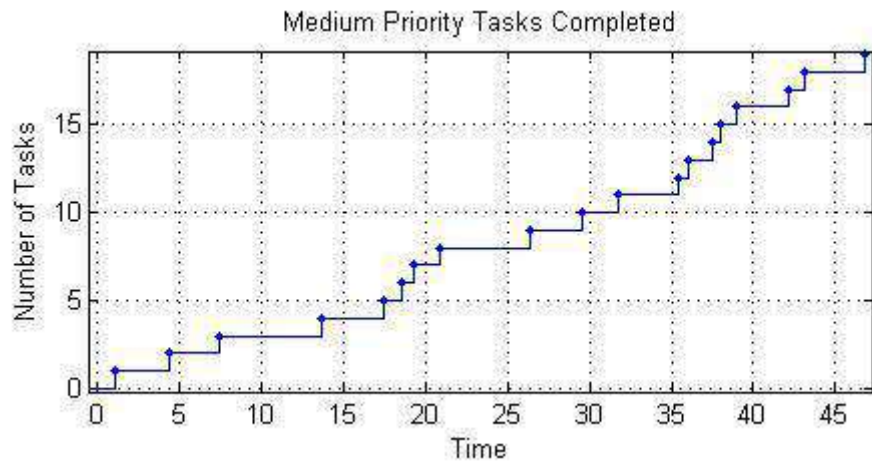


Fig.14. Medium priority tasks completed

Medium priority packets will be processed only after high priority packets have been served by the server. In this plot a medium priority packet will be served at time $T=1$. Here also the service time is equal to 0.5 but the first 0.5 time units will be used to serve a high priority packet that arrived before this packet, so it will be processed at time $T=1$.

Low Priority Tasks Completed

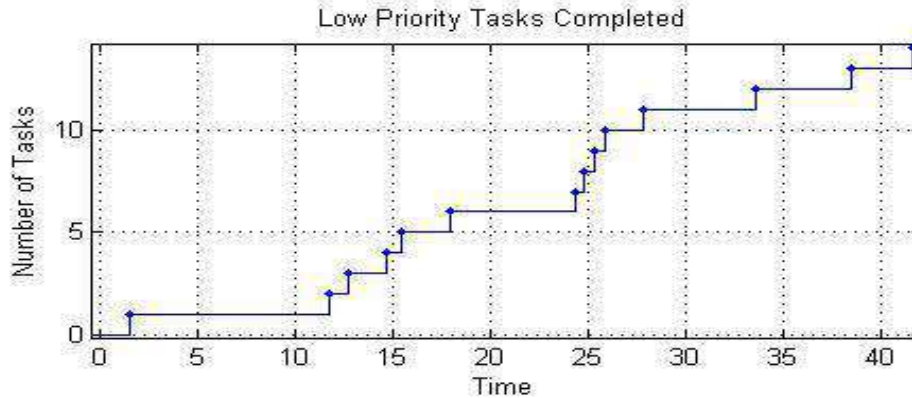


Fig.15. Low priority tasks completed

The low priority packets will be processed only after the high and medium priority packets have been processed. So, in this graph the low priority packet will be served at time $T=1.5$.

Priority Values for Tasks

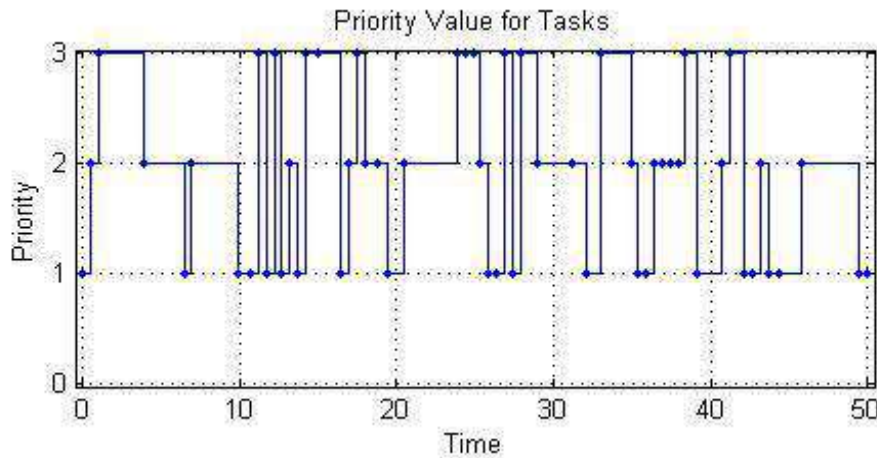


Fig.16. Priority Values for Tasks

This graph indicates when a packet having high-priority task (with priority 1) pre-empts a low-priority packet (priority 2) and when a packet with low-priority task pre-empts a still lower packet (priority 3).

Packets Departed from the Server

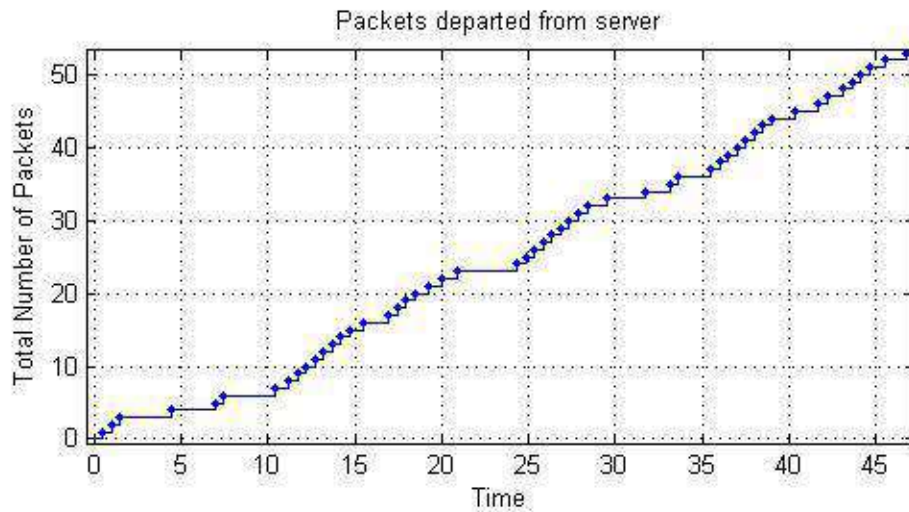


Fig.17. Packets Departed from the Server

Until $T=0.5$, no packets depart from the server. This is because it takes 0.5 seconds for the server to process the first packet. Starting at $T=0.5$, the given graph is a staircase plot. The stairs of the plot have height equal to 1. This is because the server processes only one packet at a time, so packets depart one at a time. The stairs have width equal to the constant service time of the server, which is 0.5 seconds.

Server Utilization

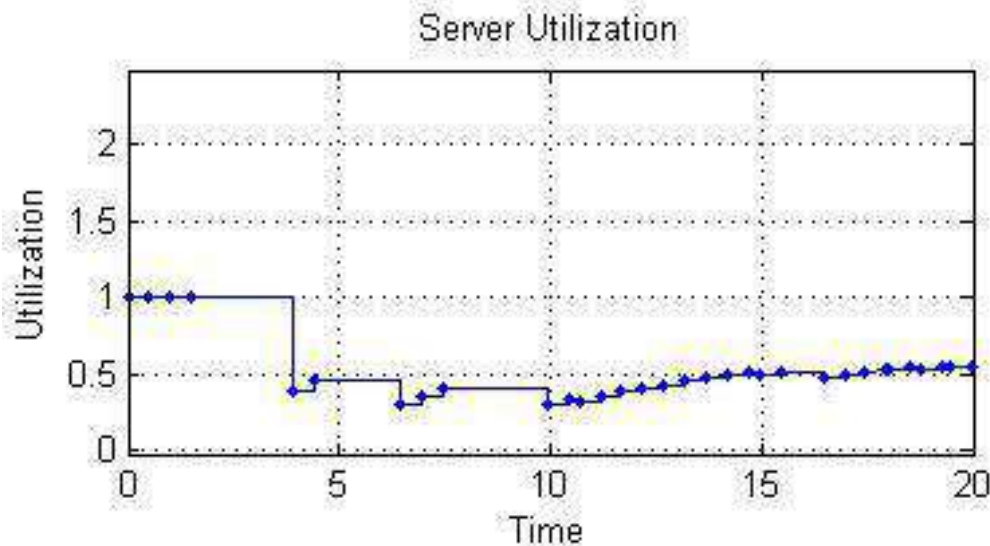


Fig.18. Server Utilization

This graph shows that at the beginning ($T=0$) the server utilization is 1 i.e. the server is completely busy. At time $T=4$, the server utilization decreases. The utilization of the server decreases if the intergeneration time of packets is larger than the service time (here service time being 0.5) because the server has idle periods between entities. After this the server utilization varies (increases/ decreases) according to the intergeneration time.

Average Waiting Time

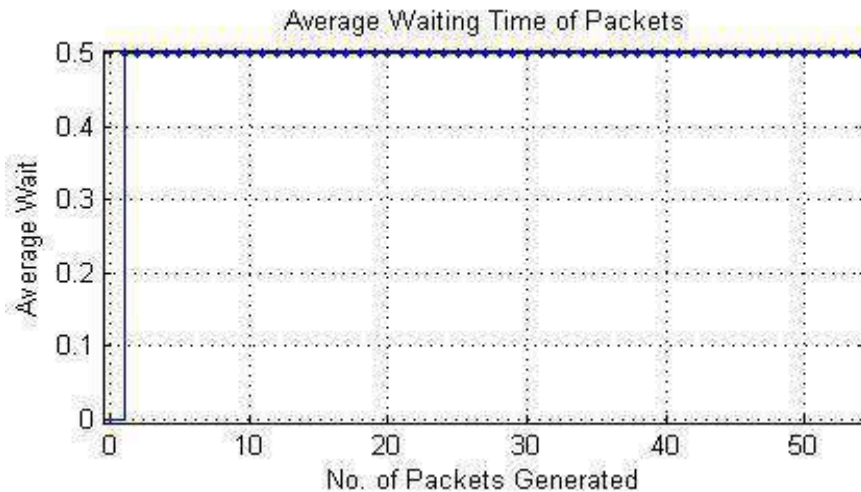


Fig.19. Average Waiting Time

From the given plot, it is clearly understood that only the first packet gets service as soon as it arrives. Rest of the packets have to wait for their turn to get processed. The average waiting time in the server does not change after the first departure from the server because the service time (0.5 in our case) is fixed for all departed entities.

Round Robin Scheduling Algorithm

Packets are generated by the three different packet generators. Each packet has attributes representing the priority and the task execution time. The packets are then sent to their respective FIFO queues whose output is fed to an input switch, wherefrom the packets are finally sent to the server for execution.

Results and Displays

Following scopes present the simulation results:

Waiting Time of Packets

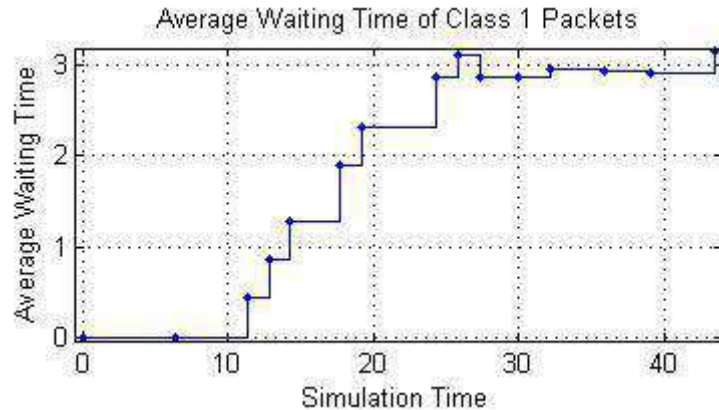


Fig.20. Average Waiting Time of Class 1 Packets

This plot gives the average waiting time of packets from class 1 that get buffered in FIFO Queue 1. It shows that as the simulation time increases the average waiting time of packets also increases. At simulation time $T=0$, the average waiting of a packet is also zero. Then at $T=11.3$, waiting time = 0.45 and so on. The average waiting time increases if the intergeneration time of packets is smaller than the service time because the queue gets longer.

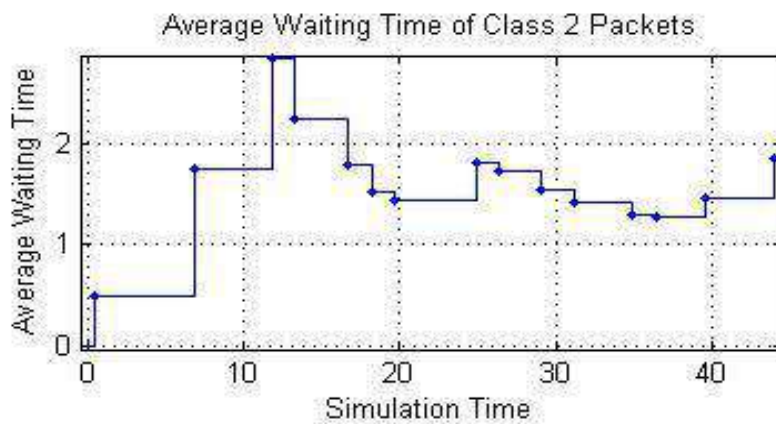


Fig.21. Average Waiting Time of Class 2 Packets

This plot gives the average waiting time of packets from class 2 that get buffered in FIFO Queue 2. Here the first packet from class 2 packet generator has an average waiting time of 0.5 (this is because first the packet of class 1 will be processed and then the packet from second class will get a chance to be served). Then as the simulation time increases, the average waiting time also increases.

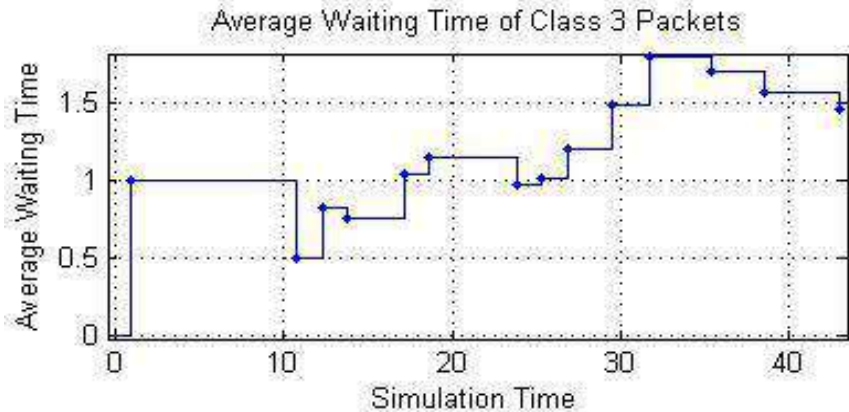


Fig.22. Average Waiting Time of Class 3 Packets

This plot gives the average waiting time of packets from class 3 that get buffered in FIFO Queue 3. Here the first packet from class 3 packet generator has an average waiting time of 1 (this is because first the packet of class 1 will be processed, followed by a packet of second class and then finally the packet from second class will get a chance to be served). Then as the simulation time increases, the average waiting time also increases.

Server Utilization

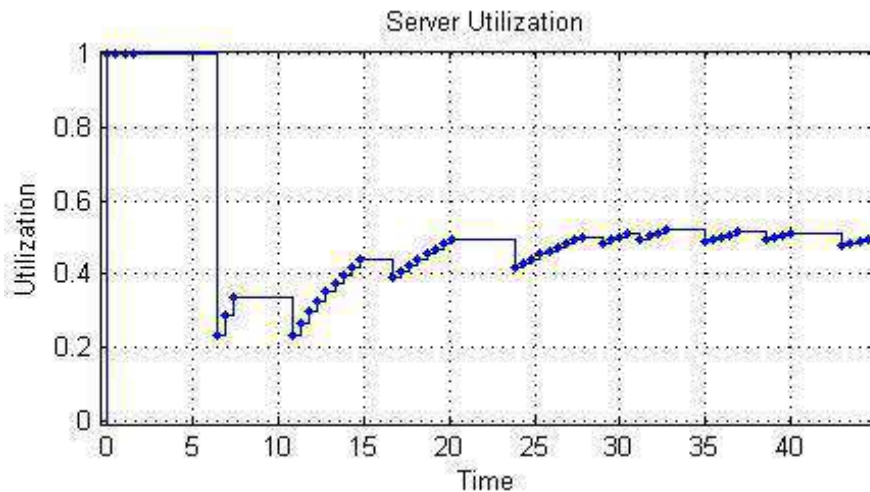


Fig.23. Server Utilization

This graph shows that right from the time $T=0$, the server utilization is equal to 1 i.e. the server is completely busy and it continues to be fully utilized. Then the utilization of the server decreases at $T=6$. The utilization of the server decreases if the intergeneration time of packets is larger than the service time (here service time being 0.5) because the server has idle periods between entities. After this the server utilization varies (increases/ decreases) according to the intergeneration time.

Packets departed from server

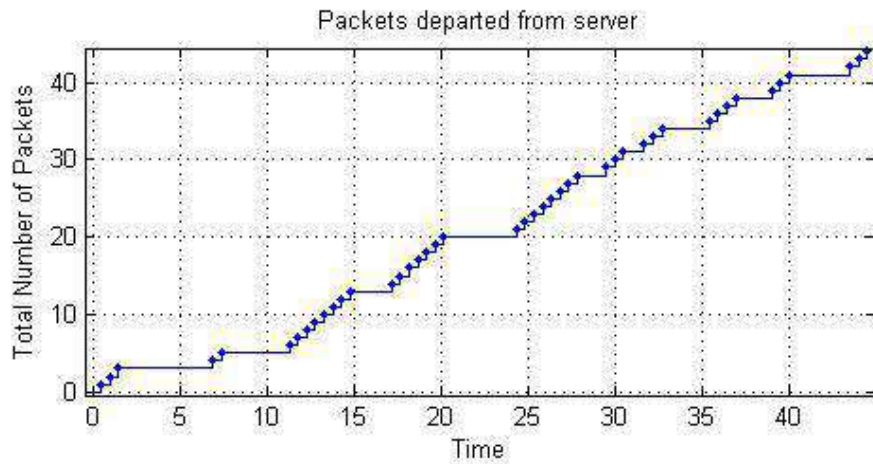


Fig.24. Packets departed from server

Here also the plot is a staircase one. At time $T=0.5$, first packet departs from the server. Again, the height of the step is 1 as only one packet is processed at a time.

Average Waiting Time

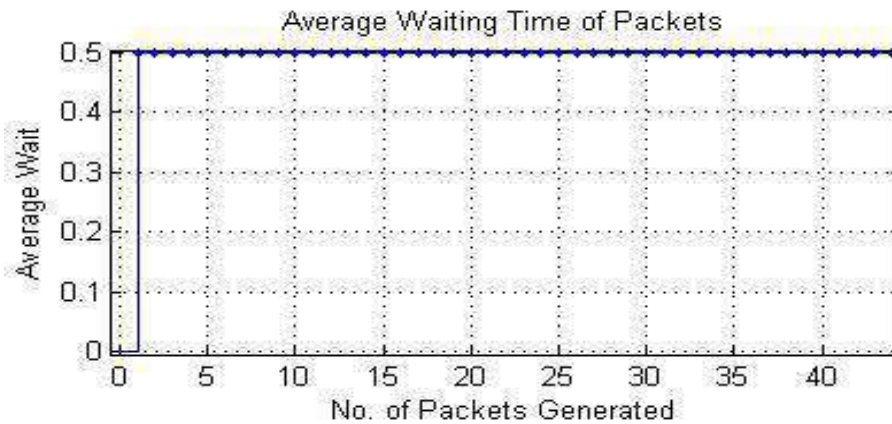


Fig.25. Average Waiting Time

From the given plot, it is clearly understood that only the first packet gets service as soon as it arrives. Rest of the packets have to wait for their turn to get processed. The average waiting time in the server does not change after the first departure from the server because the service time (0.5 in our case) is fixed for all departed entities.

An Unprioritized MQTT-SN Gateway Model

In this case incoming sensor data is not prioritised by using any scheduling algorithm. Sensor data is forwarded to the server in the same order in which it arrives. No attempt is made to prioritise data forwarded by different sensors deployed in the sensor fields.

Results and Displays

Following scopes present the simulation results:

Packets departed from server

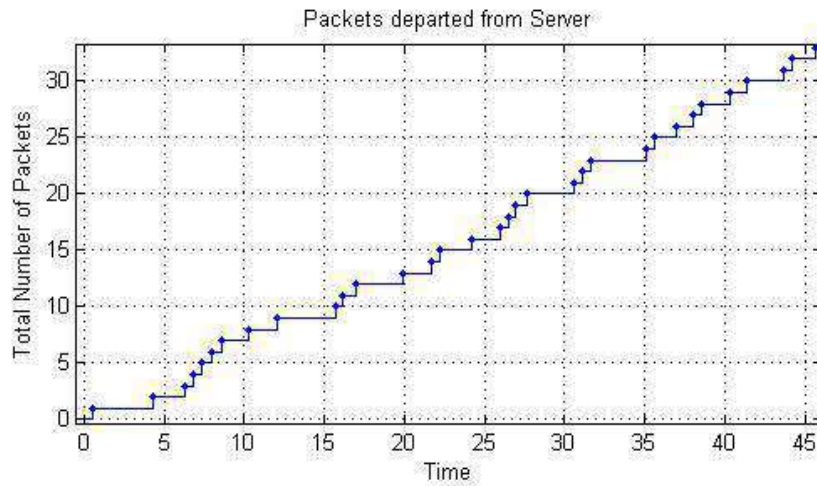


Fig.26. Packets departed from server

Again, the plot is a staircase one. Since the processing time of the server is equal to 0.5 seconds, therefore the first packet departs from the server at time $T=0.5$ seconds. Again, the height of the step is 1 as only one packet is processed at a time.

Server Utilization

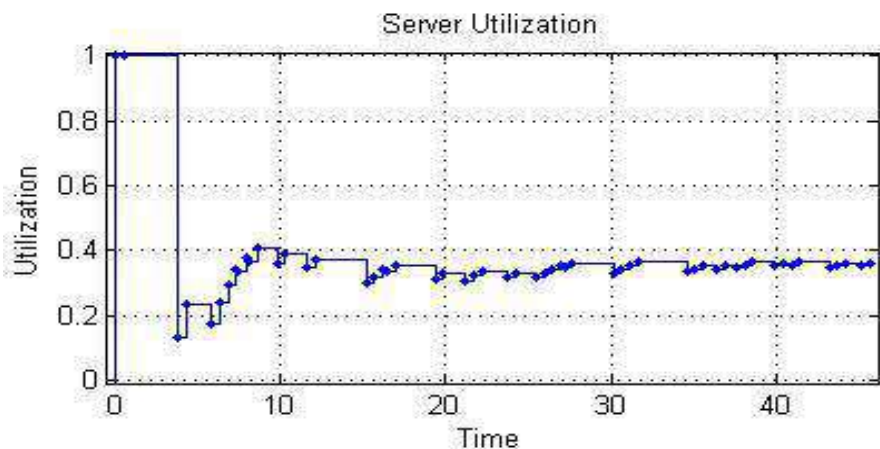


Fig.27. Server Utilization

This graph shows that right from the time $T=0$, the server utilization is equal to 1 i.e. the server is completely busy and it continues to be fully utilized. Then the utilization of the server decreases at $T=3.7$ seconds. The utilization of the server decreases if the intergeneration time of packets is larger than the service time (here service time being 0.5) because the server has idle periods between entities. After this the server utilization varies (increases/ decreases) according to the intergeneration time and fluctuates between 0.13 and 0.4.

Waiting Time of Packets

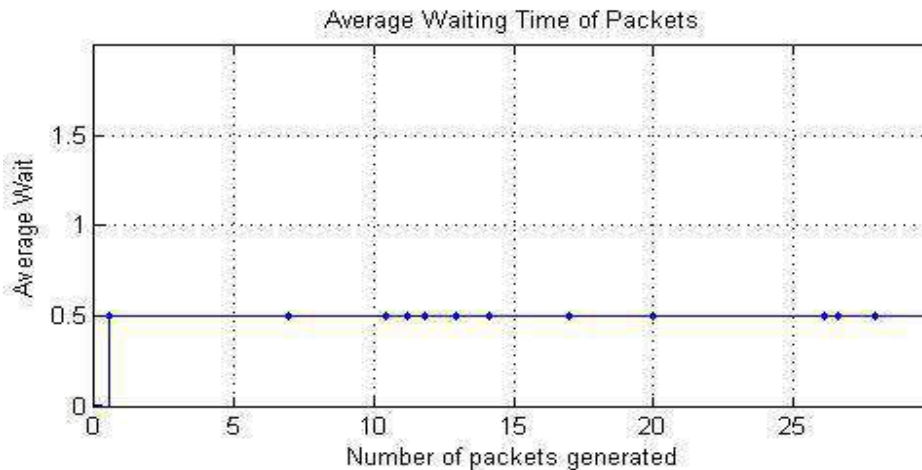


Fig.28. Waiting Time of Packets

From the above given graph, it is clear that only the first packet is processed as soon as it arrives. Rest of the packets have to wait for their turn to get processed. The average waiting time in the server does not change after the first departure from the server because the service time (0.5 in our case) is fixed for all departed entities.

4.4 COMPARISON OF RESULTS

The results are shown on the basis of three parameters. These are Packets Departed from the Server, Server Utilization Time and Average Waiting Time. The simulation is done using Matlab R2013a.

Packets departed from server

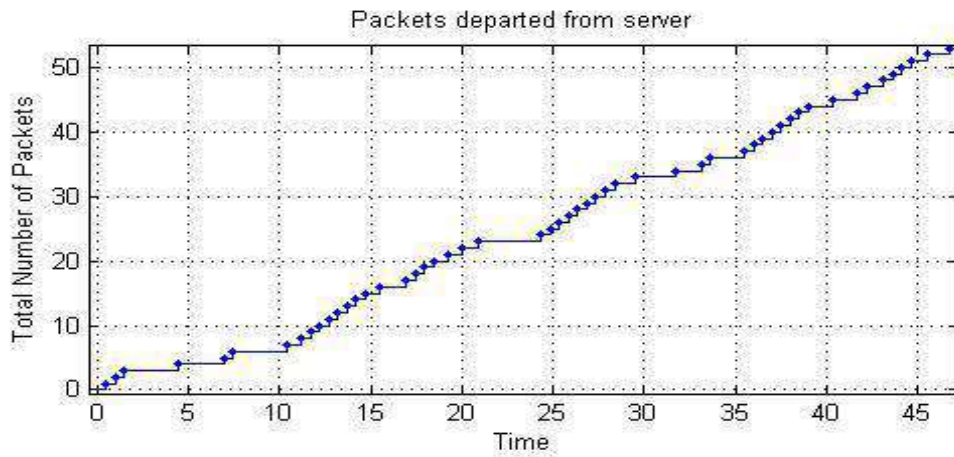


Fig.29. Packets Departed from Server in Priority Scheduling Algorithm

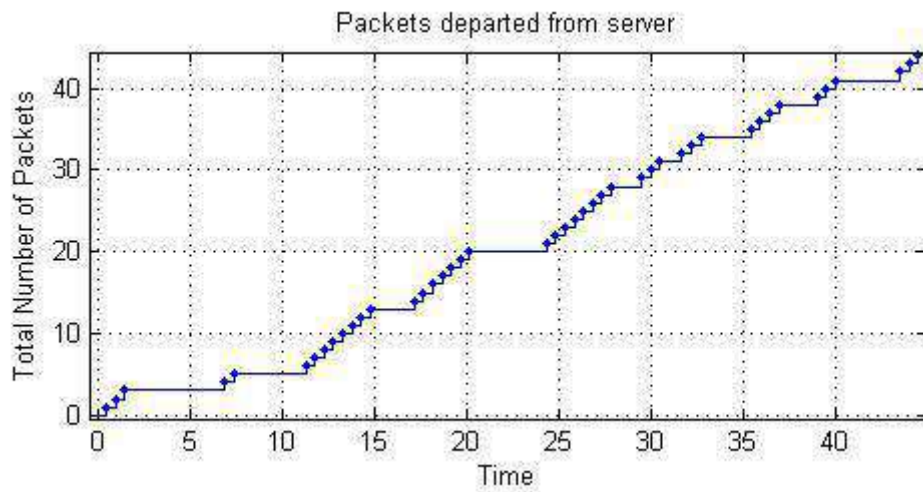


Fig.30. Packets Departed from Server in Round Robin Algorithm.

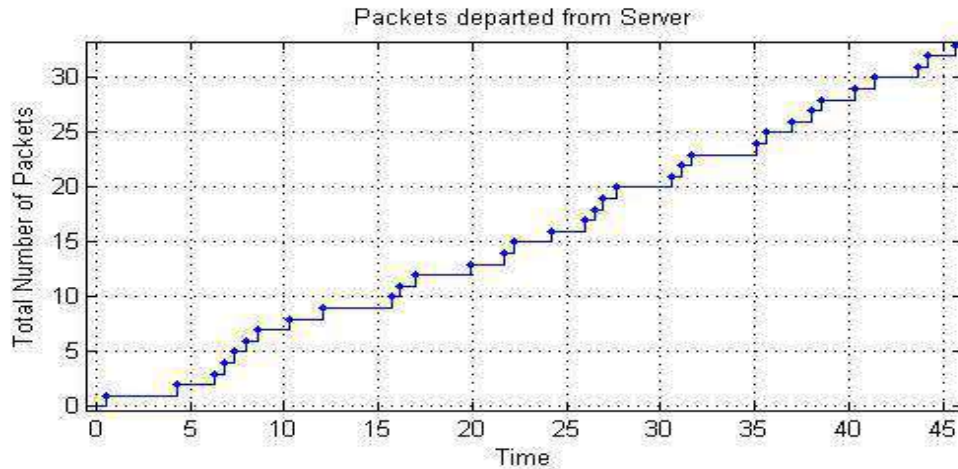


Fig.31. Packets Departed from Server in Unprioritized MQTT-SN Gateway

A comparison between the above given three plots shows that:

1. Maximum number of packets are departed from the server when Priority Scheduling Algorithm is used. Here time $T= 25$, 25 packets are departed.
2. In case of Round Robin Algorithm, at time $T=25$ about 22 packets are departed by the server after processing them.
3. In the graph where no scheduling algorithm is used, at time $T= 25$ almost 16 packets are departed from the server.

Hence in terms of packets departed from the server, best results are obtained using Priority Scheduling Algorithm.

Server Utilization

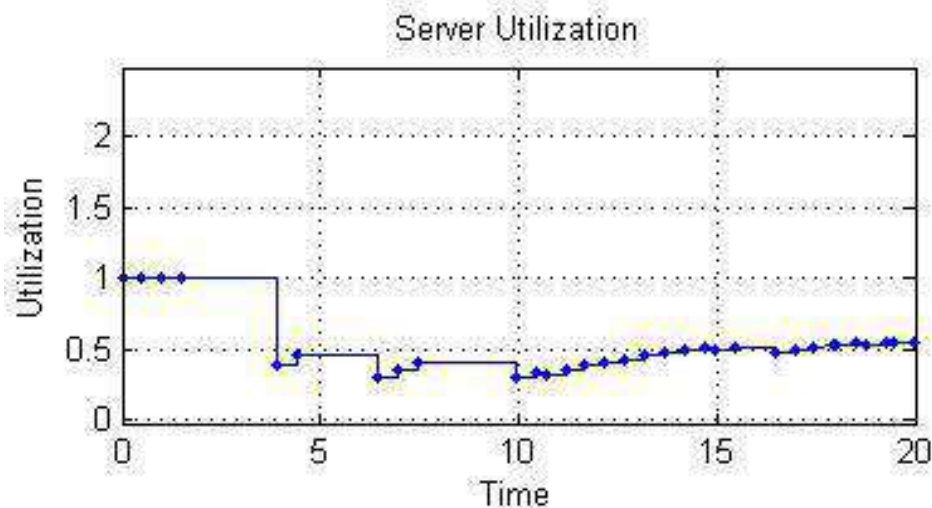


Fig.32. Server Utilization Time in Priority Scheduling Algorithm

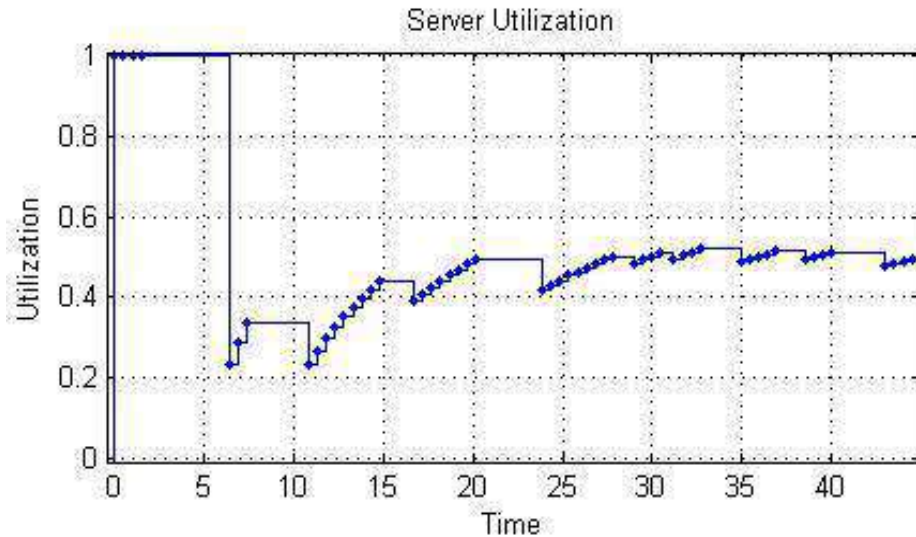


Fig.33. Server Utilization Time in Round Robin Algorithm

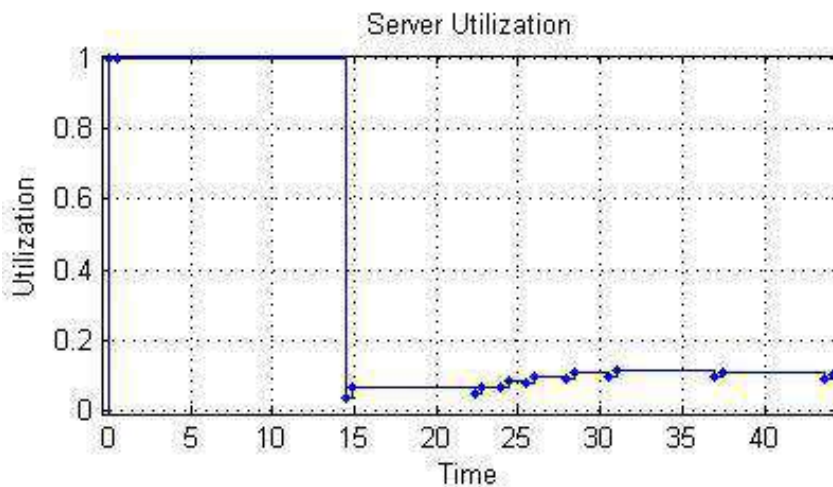


Fig.34. Server Utilization Time in Unprioritized MQTT-SN Gateway

From the above given three graphs, it is clear that:

1. In case of Priority Scheduling Algorithm the server is continuously busy (value being 1) till time $T=3.9$. After that it becomes somewhat free and its utilization decreases to about 0.383. Thereafter the utilization fluctuates between 0.24 and 0.48. It means that it is then available for processing more packets. From time $T=10$ s, the utilisation of the server is getting steady around 0.5 i.e. there is a monotonic trend.
2. In case of Round Robin Algorithm, the server is completely busy for about $T=6.5$ seconds. Then the utilization decreases and from time, $T= 10$ onwards there is a rising

and falling trend in utilization. This ringing effect in server utilisation in Round Robin Algorithm is because of the fact that Round Robin is switching between different queues providing a chance for a packet to be processed from each queue.

3. In case of unprioritized MQTT-SN Gateway, the server is fully busy for about $T=14.5$ seconds. Here the server is completely busy in the beginning and that too for a long time. Then its utilization decreases to about 0.13. After that the utilization continues to be in between 0.17 and 0.36.

Hence in terms of server utilization, Priority Scheduling Algorithm gives the best results as in that case server is not 100% busy for a long time and therefore can process other waiting packets.

Average Waiting Time of Packets

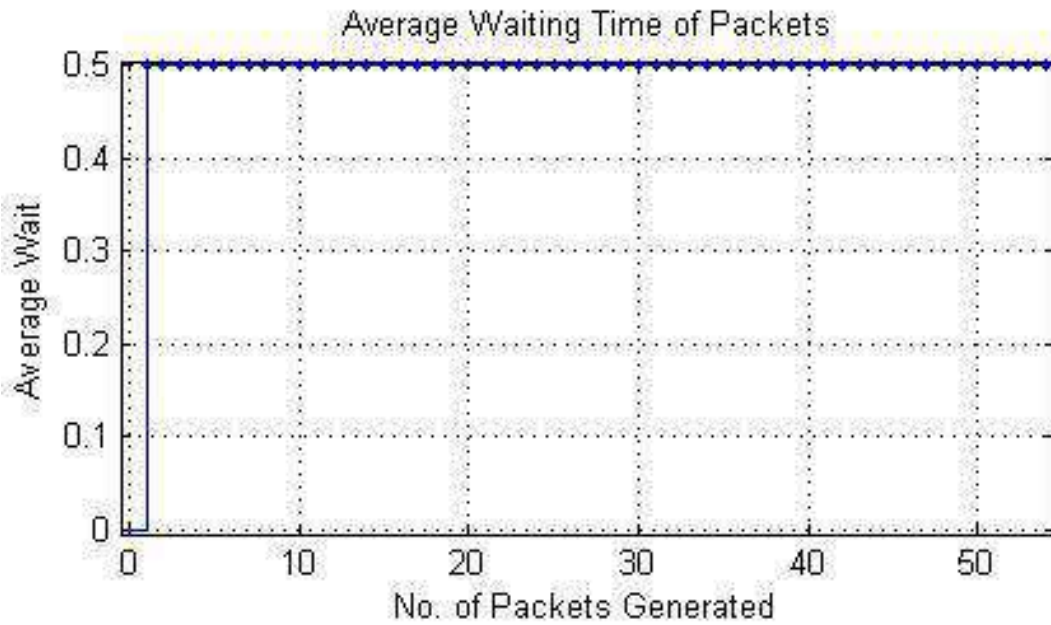


Fig.35. Average Waiting Time in Priority Scheduling Algorithm

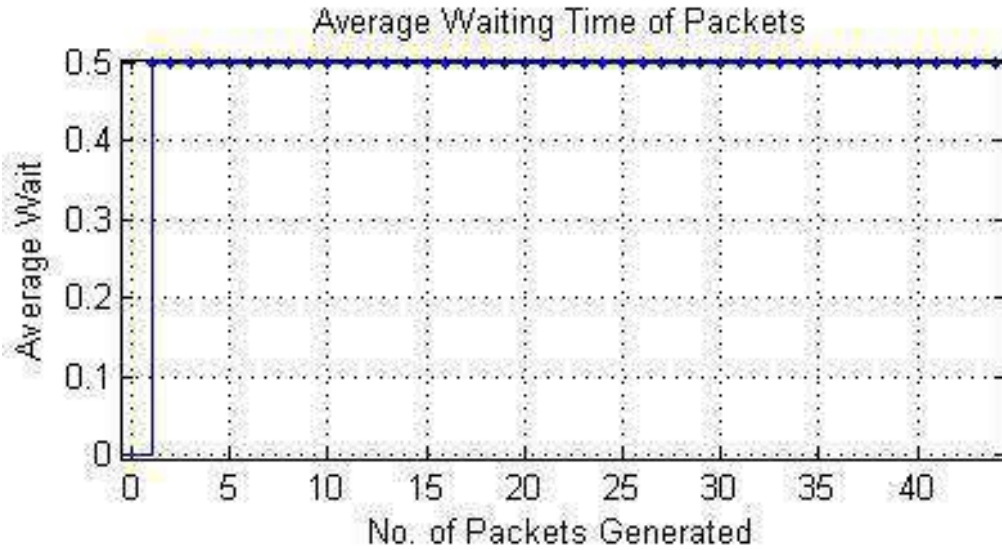


Fig.36. Average Waiting Time in Round Robin Algorithm

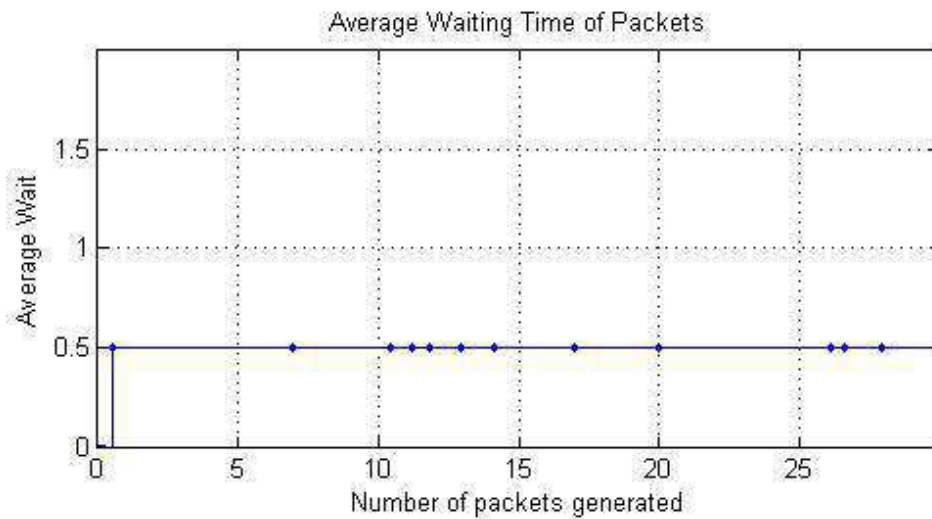


Fig.37. Average Waiting Time in Unprioritized MQTT-SN Gateway

From the above given plots it is clear that in all the three cases the average waiting time in the server does not change after the first departure from the server because the service time (here being 0.5) is fixed for all departed packets.

Conclusion

From the above given comparisons, it can be concluded that for traffic that is random and sensitive in time constraints as well as for low power devices, Priority Scheduling Algorithm is better suited.

It however does not mean that Round Robin is not suitable. If the traffic arrives at a uniform rate and starvation is not desirable, then Round Robin Algorithm is better suited.

Hence both of these scheduling algorithms have their own benefits and can be used in conditions to which they are better suited.

4.5 Hardware Implementation

For the purpose of hardware implementation a NodeMCU has been used. It is an open source IoT platform and runs on ESP8266 Wi-Fi system on chip (SoC). This NodeMCU acts both as a client as well as a server. Three sensor nodes have been used to send data to the MQTT-SN Gateway. ThingSpeak has been used in order to analyse as well as visualize data that has been uploaded using MATLAB. ThingSpeak is an open source IoT application used to store and retrieve data from sensors with the help of HTTP protocol. NodeMCU acts as a server between the sensor nodes and the MQTT-SN Gateway. It also acts as a client between MQTT-SN Gateway and ThingSpeak.

First of all, an Arduino Development Environment (Arduino 1.8.2) will be installed on the system. After installation, NodeMCU needs to be configured with the Arduino Development Environment and all the required libraries like ThingSpeak Communication Library for Arduino and ESP8266, Onewire, etc need to be installed.

NodeMCU will be connected to the system with the help of a USB cable through a USB port. Then a sketch (created with the help of a C program) will demonstrate how to set up a Gateway server that receives requests from clients. A ThingSpeak account needs to be created for the purpose of visualizing and analysing sensor data. The ThingSpeak support toolbox allows one to use the desktop MATLAB so as to analyse as well as visualize the data that is stored on ThingSpeak.com. After creating an account in ThingSpeak and logging in, a channel needs to be created. This channel is used to store the data that is collected by ThingSpeak application. Then in channel settings, the required fields need to be filled and then the channel needs to be saved. After that the Channel ID and the API Keys thus obtained need to be copied to the sketch. This sketch will then be compiled and uploaded to the MQTT Broker on ThingSpeak cloud in order to analyse and visualize the data sent by the sensor nodes. After the sketch is successfully uploaded to ThingSpeak, the Node MCU is assigned a unique IP address. Different sensor nodes can then send their collected data to the IP address assigned to NodeMCU which therefore acts as a server

between the sensor nodes and the MQTT-SN Gateway on the system. The Arduino IDE has a feature called serial monitor which is a separate pop-up window. It acts as a separate terminal which communicates by means of sending and receiving serial data. The data sent by different sensor nodes is reflected on this serial monitor.

Here, since two scheduling algorithms (First in First out and Round Robin) are being used, therefore two different sketches need to be created in order to visualize how different priority data sent by different sensor nodes will be treated.

First In First Out

NodeMCU esp8266 is configured with Arduino Development Environment (Arduino 1.8.2). Then a sketch written for First in First out Scheduling Algorithm is compiled and uploaded to ThingSpeak cloud so that data can be visualized and analysed. Three sensor nodes (mobile phones) are used to send data which is reflected on the serial monitor.

The data of three different priorities (priority 1, 2 and 3) is sent by the three sensor nodes. As soon as a sensor node sends data to the ThingSpeak cloud, it is represented as a dot in the graph. As more and more data is sent, dots start appearing in the graph with the value of data and in this way a graph is created.

Three different graphs are created for data of different priority as shown below:



Fig.38. Graph showing sensor data of priority 1



Fig.39. Graph showing sensor data of priority 2



Fig.40. Graph showing sensor data of priority 3

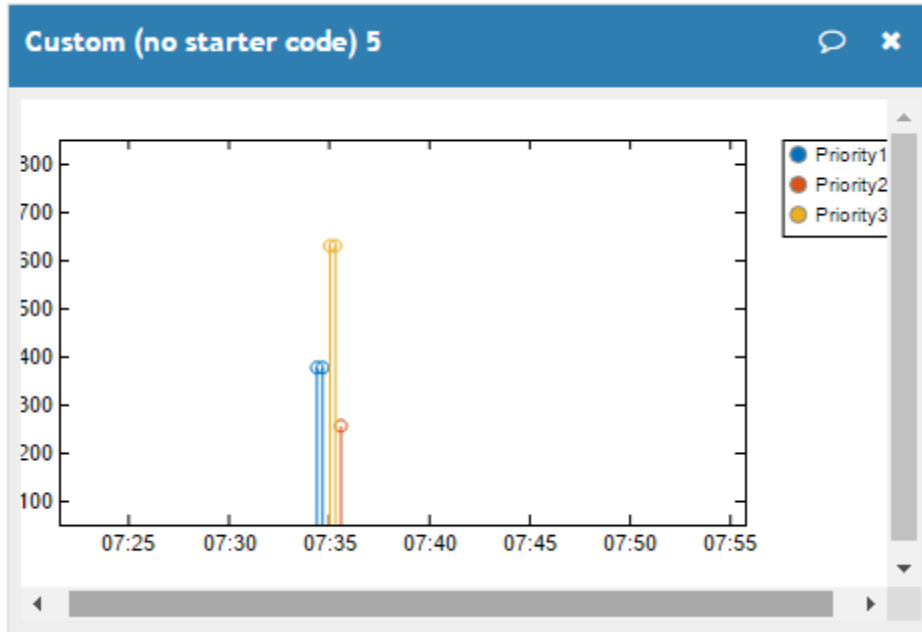


Fig.41. Arrival of sensor data in First in First out Scheduling Algorithm

Round Robin Scheduling Algorithm

In this case a sketch written for Round Robin Scheduling Algorithm is uploaded. The rest of the procedure is same as that for First in First out Scheduling Algorithm. Here also three graphs are created for three different priorities of sensor data.

The above mentioned graphs are given below:

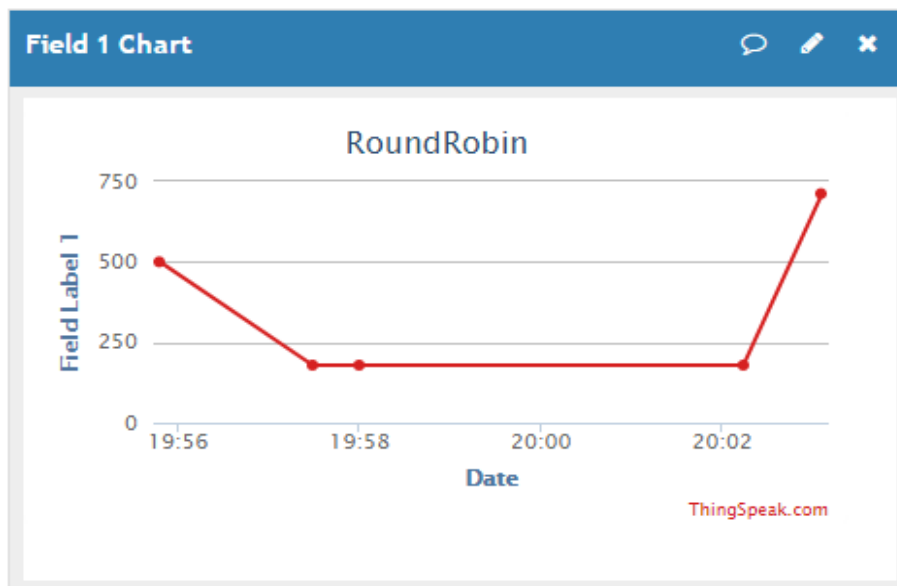


Fig.42. Graph showing sensor data of priority 1



Fig.43. Graph showing sensor data of priority 2

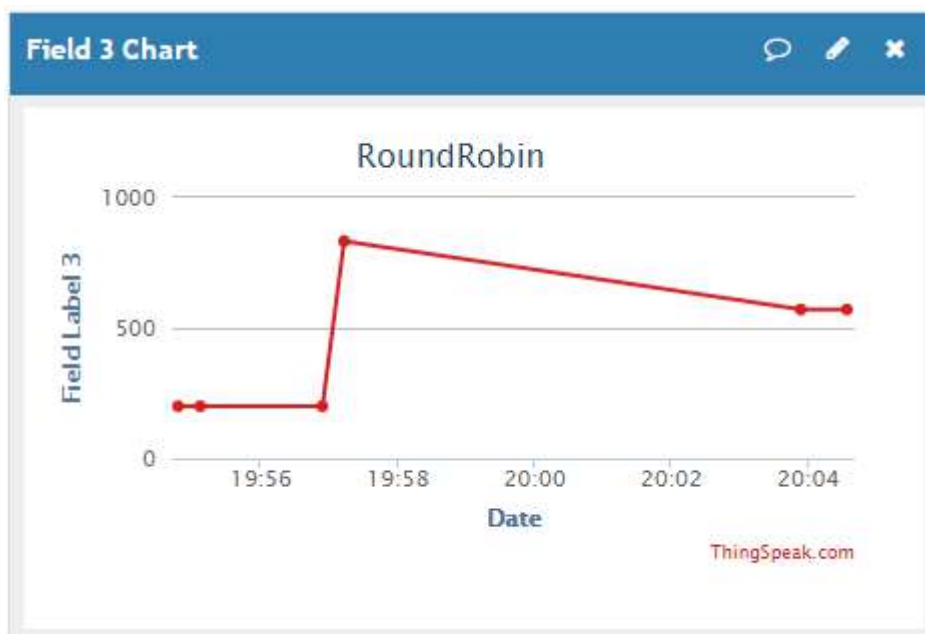


Fig.44. Graph showing sensor data of priority 3



Fig.45. Arrival of sensor data in Round Robin Scheduling Algorithm

5.1 CONCLUSION

Message Queue Telemetry Transport protocol is a very light weight and therefore is used for the purpose of connecting small devices to constrained networks. It collects data from devices and then the same is sent to IT infrastructure. Applications like Facebook Messenger, etc use this protocol. Since losing data is not desirable, so this protocol runs over TCP, which guarantees reliability. MQTT finds its use in various kinds of applications, for instance it is used to monitor a huge oil pipeline in order to check leaks or any kind of vandalism. It is also used in power usage monitoring, intelligent gardening, lighting control, etc. The only problem with MQTT is that there is no mechanism for prioritising the sensor data arriving at an MQTT-SN Gateway. This report proposes a mechanism in order to prioritise data at an MQTT-SN Gateway so that high priority sensor data such as data from a sensor monitoring a patient's heart rate gets better Quality of Service (QoS) as compared to low priority data such as data from a sensor monitoring temperature changes. Two scheduling algorithms (First in First out and Round Robin) have been used for the purpose of prioritising data. Several parameters have been set on the basis of which the above mentioned algorithms have been compared in order to determine which of the two is better. According to the parameters that have been taken for comparison Priority Scheduling Algorithm seems to be better suited. If however the traffic flow is uniform, then Round Robin will be better suited. No work has earlier been done on this topic.

5.2 FUTURE SCOPE

In this thesis, we examined the applicability and adoption of multiple scheduling algorithms in gateways for enabling the proliferation of the IoT, besides mitigating various challenges. As such, we studied various aspects of scheduling.

As a future work, we would like to explore the gateways which may run a multiple IoT stacks, in order to cater for heterogeneous IoT protocol stacks running on nodes. For example, for situations when nodes may be running different IoT stacks and a gateway allows them to:

1. talk to each other;
2. forward messages from nodes to the server (via the internet).

As such, a gateway may be running instances of brokers (for Publish/Subscribe type protocols) and servers (for Request/Response type protocols). Moreover, the gateway does not have a sensing or actuation capability and is simply a message forwarder (between nodes and server or between nodes with different IoT stacks). As such, as a future work, we would like to explore a gateway with a multi-protocol stack, allowing two messaging domains (P/S and R/R) to communicate.

Moreover, as a future work, we would like to study a gateway, as it can become a bottleneck for message traffic, or become a single-point-of-failure. We may mitigate this by adding redundant gateway devices, which can result in scalability and reliability (to avoid bottlenecks and single-point-of-failures). The number of gateway devices that are needed to create a scalable and reliable system is a design characteristic, and as such is decided by the system designers.

Reference for an article

1. Madakam, Somayya, R. Ramaswamy, and Siddharth Tripathi. "Internet of Things (IoT): A Literature Review." *Journal of Computer and Communications* 3.05 (2015): 164.
2. R.Khan, S. Ullah Khan, R.Zaheer and S.Khan."Future internet: the internet of things architecture, possible applications and key challenges." *Frontiers of Information Technology (FIT), 2012 10th International Conference on.* IEEE, 2012.
3. M. Zhang, F. Sun, and X. Cheng. "Architecture of internet of things and its key technology integration based-on RFID." *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on.* Vol. 1. IEEE, 2012.
4. V. Karagiannis, P.Chatzimisios, F.V.Gallego and J. A. Zarate. "A survey on application layer protocols for the internet of things." *Transaction on IoT and Cloud Computing* 3.1 (2015): 11-17.
5. E. Borgia. "The Internet of Things vision: Key features, applications and open issues." *Computer Communications* 54 (2014): 1-31.
6. F. Razzak. "Spamming the Internet of Things: A Possibility and its probable Solution." *Procedia computer science* 10 (2012): 658-665.
7. L. Coetzee, and J. Eksteen. "The Internet of Things-promise for the future? An introduction." *IST-Africa Conference Proceedings, 2011.* IEEE, 2011.
8. Q. Jing, A. V. Vasilakos, J. Wan, J. Lu and D. Qiu. "Security of the internet of things: Perspectives and challenges." *Wireless Networks* 20.8 (2014): 2481-2501.
9. A. Al-Fuqaha , M. Guizani . M. Mohammadi , M. Aledhari and Moussa Ayyash."Internet of things: A survey on enabling technologies, protocols, and applications." *IEEE Communications Surveys & Tutorials*17.4 (2015): 2347-2376.

10. S. Schneider. "Understanding the protocols behind the internet of things." *Electronic Design*, <http://electronicdesign.com/iot/understanding-protocols-behind-internet-things> [09.10.2013] (2013).
11. Farooq, M. U., et al. "A Review on Internet of Things (IoT)." *International Journal of Computer Applications* 113.1 (2015).
12. P. Pande, Prajakta, and A. R. Padwalkar. "Internet of Things—A Future of Internet: A Survey." *International Journal* 2.2 (2014).
13. L. Atzori, A. Iera, and G. Morabito. "The internet of things: A survey." *Computer networks* 54.15 (2010): 2787-2805.
14. R. Alur, E. Berger, A.W. Drobnis, L. Fix, et al. "Systems Computing Challenges in the Internet of Things." *arXiv preprint arXiv:1604.02980* (2016).
15. A. Whitmore, A. Agarwal, and L.D. Xu. "The Internet of Things—A survey of topics and trends." *Information Systems Frontiers* 17.2 (2015): 261-274.
16. J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami. "Internet of Things (IoT): A vision, architectural elements, and future directions." *Future Generation Computer Systems* 29.7 (2013): 1645-1660.
17. U. Hunkeler, H. L. Truong, and A. S. Clark. "MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks." *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*. IEEE, 2008.
18. C. Lesjak, D. Hein, et al. "Securing smart maintenance services: Hardware-security and TLS for MQTT." *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*. IEEE, 2015.
19. J.E. Luzuriaga, J.C. Cano, et al. "Handling mobility in IoT applications using the MQTT protocol." *Internet Technologies and Applications (ITA), 2015*. IEEE, 2015.

20. A. Niruntasukrat, C.Issariyapat, et al. "Authorization mechanism for MQTT-based Internet of Things." *Communications Workshops (ICC), 2016 IEEE International Conference on*. IEEE, 2016.
21. X. Li, J. Lu, et al. "A queue scheduling approach to QoS support in terminal communication access network." *Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016 12th International Conference on*. IEEE, 2016.
22. A. Stanford-Clark, and H. L. Truong. "MQTT for sensor networks (MQTT-S) protocol specification." *International Business Machines Corporation version 1* (2008).
23. A. Corsaro, and D.C.Schmidt. "The Data Distribution Service-The Communication Middleware Fabric for Scalable and Extensible Systems-of-Systems." INTECH Open Access Publisher, 2012.
24. S.Wagle."Semantic data extraction over MQTT for IoTcentric wireless sensor networks." *Internet of Things and Applications (IOTA), International Conference on*. IEEE, 2016.
25. J. Wilcox, D. Kaleshi, and M. Sooriyabandara. "Multi-Protocol Transport Layer QoS: An Emulation Based Performance Analysis for the Internet of Things."
26. S.Bandyopadhyay and A.Bhattacharyya. "Lightweight Internet protocols for web enablement of sensors using constrained gateway devices." *Computing, Networking and Communications (ICNC), 2013 International Conference on*. IEEE, 2013.
27. A. Al-Fuqaha, A.Khreishah, M. Guizani, A. Rayes and M. Mohammadi "Toward better horizontal integration among IoT services." *IEEE Communications Magazine* 53.9 (2015): 72-79.
28. Bovet, Gerome. *A Scalable and Sustainable Web of Buildings Architecture*. Diss. Telecom ParisTech, 2015.
29. M. Diaz-Cacho, E. Delgado, P. Falcon and A. Barreiro "IoT integration on industrial environments." *Factory Communication Systems (WFCS), 2015 IEEE World Conference on*. IEEE, 2015.

30. M. Singh , M.A. Rajan ; V.L. Shivraj ; P. Balamuralidhar. "Secure mqtt for internet of things (iot)." *Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on.* IEEE, 2015.
31. S.M Kim, H.S Choi and W.S Rhee. "IoT home gateway for auto-configuration and management of MQTT devices." *Wireless Sensors (ICWiSe), 2015 IEEE Conference on.* IEEE, 2015.
32. Y.F Gomes, D.F.S Santos, H.O. Almeida and A. Perkusich. "Integrating MQTT and ISO/IEEE 11073 for health information sharing in the Internet of Things." *Consumer Electronics (ICCE), 2015 IEEE International Conference on.* IEEE, 2015.
33. S. Lee, H. Kim, D. Hong and H. Ju "Correlation analysis of MQTT loss and delay according to QoS level." *Information Networking (ICOIN), 2013 International Conference on.* IEEE, 2013.

Reference to a web page

http://www.h3c.com.hk/Technical_Support/1828P04-6W182/10/201408/839064_1285_0.htm.

ABBREVIATIONS

IoT	Internet of Things
RFID	Radio Frequency Identification
EPC	Electronic Product Code
NFC	Near Field Communication
WSN	Wireless Sensor Networks
WiFi	Wireless Fidelity
LAN	Local Area Network
D2D	Device to Device
S2S	Server to Server
CoAP	Constrained Application Protocol
HTTP	Hypertext Transfer Protocol
UDP	User Datagram Protocol
QoS	Quality of Service
AMQP	Advanced Message Queuing Protocol
XMPP	Extensible Messaging and Presence Protocol
DDS	Distributed Data Service
MQTT	Message Queue Telemetry Protocol
FIFO	First in First out

FCFS First Come First Serve

RR Round Robin