

**DESIGN AND IMPLEMENTATION OF ENHANCED MESSAGE  
QUEUING TELEMETRY TRANSPORT PROTOCOL FOR  
INTERNET OF THINGS**

A Thesis Submitted to the  
University of Petroleum and Energy Studies

**For the Award of  
Doctor of Philosophy**  
**in**  
**Computer Science and Engineering**

*By*

**ANKIT KHARE**

JUNE – 2020

Supervisor(s)

**DR. RASHMI SHARMA**

**&**

**DR. NEELU J. AHUJA**



UNIVERSITY WITH A PURPOSE

**School of Computer Science**  
**University of Petroleum & Energy Studies**  
**Dehradun - 248007: Uttarakhand**

**DESIGN AND IMPLEMENTATION OF ENHANCED MESSAGE  
QUEUING TELEMETRY TRANSPORT PROTOCOL FOR  
INTERNET OF THINGS**

A Thesis Submitted to the  
University of Petroleum and Energy Studies

**For the Award of  
Doctor of Philosophy  
in  
Computer Science and Engineering**

*By*

**ANKIT KHARE**

SAP ID: 500031710

JUNE - 2020

Supervisor

**DR. RASHMI SHARMA**

Assistant Professor(SG), School of Computer Science

Co-Supervisor

**DR. NEELU J. AHUJA**

Professor, School of Computer Science



UNIVERSITY WITH A PURPOSE

**School of Computer Science**

**University of Petroleum & Energy Studies**

**Dehradun - 248007: Uttarakhand**

---

**I Dedicate My Ph.D. Thesis to**

My Loving Parents, In-Laws, Wife and My Guide

**Dr. Rashmi Sharma, and**

**Dr. Neelu J. Ahuja**

for their endless support, blessings and guidance.

---

## DECLARATION

I declare that this thesis, which I submit to University of Petroleum and Energy Studies, Dehradun, for examination in consideration of the award of a higher degree Doctor of Philosophy in School of Computer Science is my own personal effort. Where any of the content presented is the result of input or data from a related collaborative research program this is duly acknowledged in the text such that it is possible to ascertain how much of the work is my own. Furthermore, I took reasonable care to ensure that the work is original, and, to the best of my knowledge, does not breach copyright law, and has not been taken from other sources except where such work has been cited and acknowledged within the text.




**Signature of the Candidate:**\_\_\_\_\_

SAP ID:500031710

Date:\_\_\_\_\_

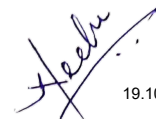
## THESIS COMPLETION CERTIFICATE

This is to certify that the thesis entitled "**Design and Implementation of Enhanced Message Queuing Telemetry Transport Protocol for Internet of Things**" by **Ankit Khare, (SAP ID: 500031710)** in partial completion of the requirements for the award of the Degree of Doctor of Philosophy in School of Computer Science is an original work carried out by him under our joint supervision and guidance. It is certified that the work has not been submitted anywhere else for the award of any other diploma or degree of this or any other University.



10/10/2020

**Dr. Rashmi Sharma**  
Assistant Professor (SG)  
School of Computer Science  
UPES, Dehradun



19.10.2020

**Dr. Neelu Jyoti Ahuja**  
Professor  
School of Computer Science  
UPES, Dehradun

---

# Abstract

The physical world is integrated with Internet of Things(IoT) and computer systems that collaborate with each other to operate smoothly from remote areas. The environment is monitored with sensors and the data collected is analysed for decision making.

There are several devices equipped with sensing and processing devices. Such devices enable monitoring and controlling of various data collection devices. Healthcare applications do find wide usage of IoT for monitoring and attending patients located remotely and providing emergency medical aids to those who need it remotely. Availability of IoT devices has led to flexible scalable and interoperable solutions in wide areas of applicability. The devices exploit the potential of cloud storage to manage and store the data which is used to enhance the utility of services from where it has been collected.

The existing research and industries have come up with many architecture and tools (hardware and software) in order to provide support and impetus to IoT. There exist compatibility issues between the devices and implementation of applications so there is a need of new solution to easily control different kind of smart devices. Developing a new communication method that is capable of ousting these issues is needed and is demand of current scenario.

There is a requirement of one or more communication protocols between several devices for message passing and information exchange. Protocols are designed to operate in low bandwidth considering the limited resources of IoT devices. This also resolves the issue of unsatable communication and high latency networks.

Several sublayers engage and interact using the different communication protocols are Message Queuing Telemetry Transport(MQTT), Constrained Application Protocol (CoAP), Zig-Bee, Extensible Messaging and Presence Protocol(XMPP), Advanced Message Queuing Protocol(AMQP), HyperText Transfer Protocol (HTTP) and Representational State Transfer (REST).

A well-known light weight protocol that is developed for message transferring process and communicating among IoT devices is MQTT. The message transfer takes place from publisher to subscriber and the broker manages the communication system. The only issue with MQTT is that its not safe without authentication and leads to information loss and network latency.

---

The current work initially integrates the ID technique to minimize the information (message) loss in the server. This is achieved by using message ID and device ID in the message passed. The proposed work monitors the messaging order in the network. If any part of message is missed then a request is generated to respective device to resend the missed message. The second major concern which is security is also ensured by applying suitable methods to improvise the loopholes.

The vulnerabilities in the software applications are addressed using fuzzy testing techniques. Weight based fuzzy methods has been proposed to automatically analyze the network packets. A comparison of experimental results with existing methods is presented and evaluated based on parameters like latency per message, MQTT connection time, message loss rate and end-to-end delay.

The results showed that the proposed methodology reduces the message transfer time to 0.86s and the integrated ID has the connection time of 22.95s. The overall result shows less time as compared to existing methods. The weighted fuzzy technique showed higher performance as compared to the other established methods. The proposed work showed reduction of loss in message as 0.207 percentage over against 0.22 percentage considering the payload pf 3000 messages.

The proposed method stored and processed the numerical data in form of device ID for identification that required less time to process the data as compared to existing methods which uses other information (user name, token, etc.).

---

# Acknowledgment

I bow my head humbly to pay heart felt regards to Almighty God for giving me the strengths and blessing in completing this thesis.

There are quite a few people that have helped me in one way or another to the completion of this work. It is with great pleasure, I would like to thank all of you from very deep inside.

Foremost, I would like to express my sincere gratitude to my thesis supervisors Dr. Rashmi Sharma and Dr. Neelu J. Ahuja, for picking me up as a student at the critical stage of my career and the continuous support of my Ph.D study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis.

Besides my supervisors, I would like to thank Chancellor Dr. S. J. Chopra, Vice chancellor Dr. Sunil Rai and Dean Dr. Manish Prateek at the University of Petroleum and Energy Studies for their encouragement, suggestions and valuable support for my research work.

I would like to express my special thanks to Dr. J K Pandey, Associate Dean R&D, Dr. Kiran Kumar Assistant Dean R&D and Dr. Rakhi Ruhel, Program Manager-Ph.D, University of Petroleum and Energy Studies, for assistance during my research work. I am grateful to the University of Petroleum and Energy Studies, for giving me an opportunity to pursue my research.

I would like to thank all the Heads of School of Computer Science Departments and colleagues in the university. I would also like to thank my teachers, friends and well wishers Dr. Nitin, Mr. Tapan Pancholi, Dr. Monit Kapoor, Dr. Hanumat Sastry G, Mr. Sunil Kumar, Dr. Ankit Kumar Jain, Dr. Prakash G.L, Mr. M. V. Kamal, Dr. Tanupriya Chaudhary, Dr. Ved Prakash and Mr. Pushpendra Kumar for their insightful comments and encouragement.

Finally, I would like to express my gratitude to my family for all of the love, support, encouragement, and prayers they have sent my way along this journey. I am eternally indebted to my loving parents and in-laws for all the sacrifices they have made on my behalf. I would like to express sincere gratitude to my beloved wife Harshita Kant who believed in me and provided encouragement during challenging times. Your unconditional love and support in the moments when there was no one to answer my queries has helped me immensely.



# Contents

<b>Declaration</b>	<b>ii</b>
<b>Thesis Completion Certificate</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>Acknowledgment</b>	<b>vi</b>
<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Motivation . . . . .	4
1.3 Objective . . . . .	4
1.4 Problem Domain . . . . .	5
1.5 Scope of the Work . . . . .	6
1.6 Chapter Organization . . . . .	6
<b>2 LITERATURE SURVEY</b>	<b>8</b>
2.1 Background . . . . .	8
2.2 IoT Authentication and User Accessibility . . . . .	9
2.2.1 User Access . . . . .	9
2.2.2 Security and Integrity . . . . .	9
2.2.3 Data Locality . . . . .	9
2.2.4 Replication and Recovery . . . . .	9
2.2.5 Investigative Support . . . . .	9
2.2.6 Third Party Add-ons . . . . .	10
2.3 Security Issues and Challenges in IoT . . . . .	11
2.3.1 Security Issues in Different Layers of IoT . . . . .	13
2.3.2 Security issues in MQTT Protocol . . . . .	15
2.4 Data Transfer Mechanism in IoT . . . . .	19

2.4.1	Comparison of SOAP and REST Web Services . . . . .	20
2.4.2	Comparison of MQTT and CoAP Protocol . . . . .	21
2.4.3	Security of the IoT applications . . . . .	21
2.5	Comparative Discussion Based on IoT Application . . . . .	23
2.6	Related Study of IoT Protocols and Challenges . . . . .	23
<b>3</b>	<b>COMMUNICATION ARCHITECTURE</b>	<b>45</b>
3.1	Architecture of IoT . . . . .	45
3.2	Lightweight Protocols in IoT . . . . .	46
3.2.1	Constrained Application Protocol(CoAP) . . . . .	47
3.2.2	Simple Object Access Protocol (SOAP) . . . . .	48
3.2.3	Message Queue Telemetry Transport Protocol (MQTT) . . . . .	50
3.3	Connection Establishment in MQTT Protocol . . . . .	51
3.3.1	Communication in MQTT after Establish Connection . . . . .	51
3.3.2	Steps for Subscription . . . . .	51
3.3.3	Basis Packet Formats in MQTT Protocol . . . . .	51
3.4	Various Types of MQTT packets . . . . .	56
3.4.1	CONNECT Packet . . . . .	56
3.4.2	CONNACK Packet . . . . .	58
3.4.3	PUBLISH Packet . . . . .	59
3.4.4	SUBSCRIBE Packet . . . . .	59
3.5	Computing Performance . . . . .	61
<b>4</b>	<b>NOVEL MESSAGE TRANSMISSION AND SECURITY METHOD</b>	<b>62</b>
4.1	Design and Development . . . . .	62
4.2	Structure of the Proposed Approach . . . . .	63
4.3	Proposed Method . . . . .	63
4.3.1	Data Pre-Processing . . . . .	66
4.3.2	IoT Devices . . . . .	66
4.3.3	Integrated ID Method . . . . .	70
4.3.4	Mathematical Derivation of Integrated ID Method . . . . .	72
4.3.5	Scapy and Weighted Fuzzy System . . . . .	74
<b>5</b>	<b>IMPLEMENTATION</b>	<b>79</b>
5.1	System Requirements . . . . .	79

---

5.2	Output 1: Real Time Control of Devices with Message Ids . . . . .	82
5.3	Output 2: Generation of Message Id along with Device Id . . . . .	83
5.4	Output 3: Connection of Sensor devices to MQTT broker . . . . .	84
5.5	Output 4: Communication Between Sensor device and Control device . . . .	85
5.6	Output 5: Connection Process between sensor devices and control devices .	86
5.7	Output 6: Published data of sensor device with control device . . . . .	87
5.8	Output 7: Connection Establishment between All Components on MQTT .	88
5.9	Output 8: Connection Time and Latency between MQTT Packets . . . . .	89
5.10	Output 9: Integrated ID Generation . . . . .	90
5.11	Output 10: Working Application with MQTT Broker . . . . .	91
<b>6</b>	<b>RESULT ANALYSIS AND DISCUSSION</b>	<b>92</b>
6.1	Results Based on Security and Message Payload . . . . .	92
6.2	Comparative Analysis based on the Latency and Connection time . . . . .	97
<b>7</b>	<b>CONCLUSION AND FUTURE SCOPE</b>	<b>103</b>
7.1	Conclusion . . . . .	103
7.2	Future Scope . . . . .	105
	<b>BIBLIOGRAPHY</b>	<b>106</b>
	<b>LIST OF PUBLICATIONS</b>	<b>124</b>

## List of Figures

1.1	IoT Basic Architecture Describing Layer and Its Components . . . . .	2
2.1	IoT Computing Resources . . . . .	8
2.2	IoT Pre-processing Mechanism . . . . .	10
2.3	IoT Communication Procedure . . . . .	11
2.4	Security Issues in IoT . . . . .	13
2.5	Working Model MQTT . . . . .	17
3.1	IoT Service Architecture . . . . .	45
3.2	Process of Messages Transportation in CoAP . . . . .	48
3.3	SOAP Architecture . . . . .	49
3.4	MQTT Architecture . . . . .	50
3.5	Communication in MQTT . . . . .	52
3.6	Steps for Subscription . . . . .	52
3.7	Basic Packet Format MQTT . . . . .	53
3.8	Commands Type in MQTT Packet . . . . .	54
3.9	Control Field in Connect Packet . . . . .	56
3.10	Format Variable Header in Connect Packet . . . . .	56
3.11	Length in Variable Header . . . . .	57
3.12	Flags Position . . . . .	57
3.13	Variable Header and Payload in Connect . . . . .	58
3.14	Connect MQTT Packet . . . . .	58
3.15	Connect Acknowledgement Values . . . . .	59
3.16	Publish Packet MQTT . . . . .	60
3.17	Subscribe Packet MQTT . . . . .	60
3.18	IoT Services and Application . . . . .	61
4.1	Overall System Structure . . . . .	64
4.2	Flow Diagram of Integrated ID Data Pre-Processing . . . . .	65
4.3	Data Pre-processing and Categorization . . . . .	67
4.4	Overview of MQTT Method . . . . .	69
4.5	Message Transaction between Sensor and Control Device . . . . .	71

---

4.6	Communication between Devices . . . . .	72
4.7	Architecture of Fuzzy Framework . . . . .	78
5.1	Adafruit cloud applets data along with control device buttons and data id field	82
5.2	Detection of Data Loss in Id Field . . . . .	83
5.3	Connection of Sensor Device to MQTT Broker(Adafruit IO Server) . . . . .	84
5.4	Data of Sensor Device1 along with Data on Control Device . . . . .	85
5.5	Data of Sensor Device2 with Connection Process . . . . .	86
5.6	Sensor Device Data with Control Command and Control Device data with Feed id . . . . .	87
5.7	Connection Establishment on MQTT Protocol . . . . .	88
5.8	Connection Time and Latency . . . . .	89
5.9	Message ID Generation in Integrated ID Method . . . . .	90
5.10	Working Model of Proposed Method . . . . .	91
6.1	Message Loss of the Proposed Method . . . . .	96
6.2	End to End Delay Vs Number of Request . . . . .	96
6.3	Delay Vs Message Payload . . . . .	97
6.4	Latency for Number of Messages . . . . .	99
6.5	Connection Time in Different Methods . . . . .	100
6.6	Program Memory Size for Proposed Method . . . . .	101

## List of Tables

2.1	<b>Comparative Study on Web Services Protocol</b> . . . . .	20
2.2	<b>Comparative Study on MQTT and CoAP</b> . . . . .	22
2.3	<b>Comparison of Lightweight Protocols in IoT [16, 18, 21]</b> . . . . .	24
2.4	<b>Related Work Discussion</b> . . . . .	39
3.1	<b>Control Flags in MQTT Packet</b> . . . . .	55
6.1	<b>Limitations of Existing Methods</b> . . . . .	92
6.2	<b>Experimental Design</b> . . . . .	93
6.3	<b>Publisher to Subscriber Delay with Message Payload</b> . . . . .	94
6.4	<b>Source to Destination Message Loss with Payload</b> . . . . .	95
6.5	<b>Environment Setup</b> . . . . .	98
6.6	<b>Time Taken by First Message to Process</b> . . . . .	101
6.7	<b>Wrong Message Rate</b> . . . . .	102

# Chapter : 1

## INTRODUCTION

### 1.1 Introduction

The Internet of Things(IoT) generally relates to the link between objects which include but are not limited to a variety of sensors, portable devices, software systems, and several tags utilized for the radio-frequency identification (RFID)[1]. This interconnection puts IoT to use to allow the transfer of data among these objects. Implying on identifiable objects called “things” [2]. the IoT enables communication through a worldwide framework of the Internet.

In the definition, “Internet” refers to the communication channel, which is put to use by the objects and objects are referred to as “things”. This mechanism in its entirety does not usually deal with the geographical location. Put simply, the IoT can be addressed as the object of smart intelligence utilized for interaction that is achieved through unique addressing schemes [3].

The ability to bring together a set of heterogeneous objects to set up a meaningful connection between physical and virtual entities, lays down the principle of IoT. Human life in today’s times makes it an essential part due to the present needs.

It is availed using different integration and exists in many forms [3,4]. A few areas that can be integrated with IoT are intelligent transportation, e-learning, digital health systems, and industrial manufacturing with logistics [5, 6].

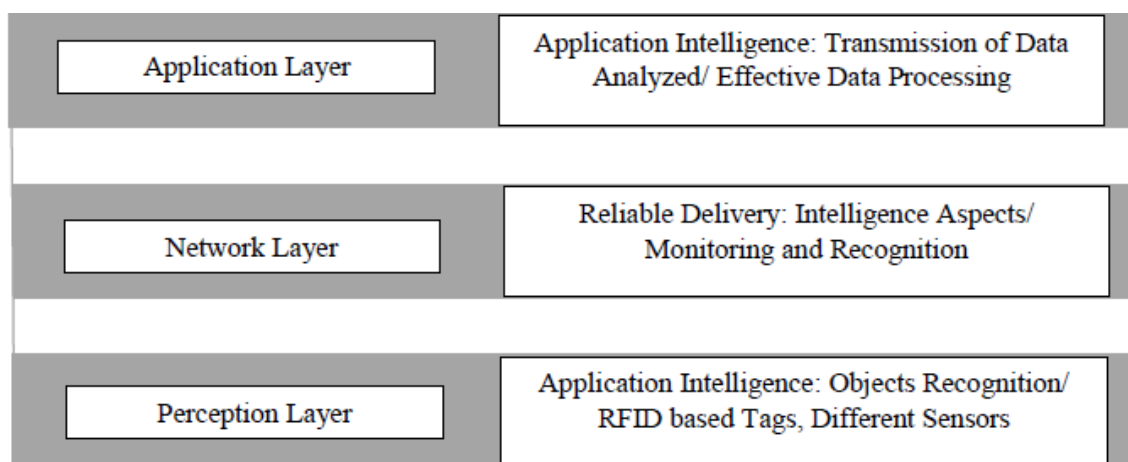
As stated earlier, the purpose is to serve interaction between divergent objects. The main purpose of IoT is the interaction between different objects. Three major lying

aspects are the platform, architecture and operating systems. It is not necessarily the same for all the objects [7].

There are basically three layers: application layer, network layer and perceptual layer as shown in Figure 1.1.

The responsibility of the perception layer is for object sensing, reading and identification. The perception layer includes sensors, RFID tags and bar code readers. The network layer is accountable for accessing, reading and storing data from the perception layer. It is a point of junction for communication. The application layer serves the bridge between IoT and industry.

An assembly of physical devices called “things” is what makes up the internet of things (IoT). These things are embedded with electronics, software, sensors, actuators etc. These smart devices are capable of communicating with each other without any human interference required to serve the purpose of Monitoring-Analysis-Control [104].



**Figure 1.1:** IoT Basic Architecture Describing Layer and Its Components



Network traffic in IoT typically comprises of the two parts:

1. Telemetry: Telemetry means sending the aggregated data from the pool of sensors to the server.
2. Tele-Command: Tele-command stands for sending/receiving the commands across a network to control IoT devices.

One of the most crucial elements in the IoT system is the message transmission system, which is accountable for controlling the devices and collection of data [104]. The control is achieved by sending commands. Therefore, it becomes vital to have one or more communication protocols while message passing between IoT devices. Due to the restricted memory, compute, power and storage, IoT devices are resource-constrained in nature [92]. Hence, the protocols are designed to operate in lower bandwidth and higher latency networks. To work efficiently in these environments, lightweight protocols are designed for IoT devices.

The process of data collection and computation is available using the network. One main characteristics of the network is low bit rate and adequate use of resources like compute power etc.

IoT has a number of messaging protocols put to use for communication. Some of these include Extensible Messaging and Presence Protocol(XMPP), Advanced Message Queuing Protocol(AMQP), Lightweight M2M(LWM2M), Constrained Application Protocol(CoAP), ZigBee, Message Queue Telemetry Transport Protocol(MQTT) and web services like Simple Object Access Protocol(SOAP) and Representation State Transfer(REST) etc. [104].

Message Queue Telemetry Transport Protocol (MQTT) can be a good choice among

the other available protocols because of the smaller footprints and lesser payload, ideal for resource-constrained environments [104,105].

## 1.2 Motivation

The main motivation of our dissertation is to design and implement an enhanced and efficient lightweight method for MQTT protocol of IoT. Therefore, making the communication scenario efficient, which can mitigate the message loss and maintain the ordering of those messages. MQTT protocol has three level of QoS (QoS0, QoS1 and QoS2). Higher level of QoS are more reliable but involve higher latency and high bandwidth requirements and lower level of QoS does not guarantee to delivery of message.

This dissertation provides an empirical and algorithmic study for the efficient framework development in terms of protocol for the authentication, networking and integration of message transmission with standardization in the IoT environment. So, Integrated ID method improve the efficiency of MQTT protocol by minimizes the message loss during transmission.

## 1.3 Objective

To Design and Implement Enhanced Lightweight Method of MQTT for Internet of Things Devices.

Sub-Objectives:

1. To identify issues in existing lightweight protocols during information exchange in IoT devices.

2. To design a lightweight method for message transmission to minimize the message loss in MQTT protocol.
3. To enhance the existing method for securely delivering the message with adaptability of the framework.

## 1.4 Problem Domain

Based on the research carried about in this field or relating to the field, the problem statements are:

1. Being plotted in different places, IoT enabled devices communicate with each other to generate large amount of data making it protection is difficult. Therefore, an effective mechanism is required to avail verification of security and protection of data being generated by IoT devices, network protocol and applications [92,105]
2. A better exploration and adoption is required in the existing protocols used in IoT. and the need of data exchange mechanism becomes important to simplify the process and the system [90].
3. There is a need for mechanism, which efficiently handles the packet delivery rate along with the message loss. These parameters become crucial because these influence the comprehensive performance of the system [104].
4. There is the need of an empirical evaluation for designing and development of the protocols for the authentication, networking and interoperability. It should adopt the security standards for data communication in the case of IoT environment and systems [92].

## **1.5 Scope of the Work**

The main aspect of this dissertation is to design implement an enhanced lightweight protocol with resource sharing in an IoT environment.

It is intended to be widely used in all areas including hospital management system, university, library, smart home etc. The work also proposes to enable using it with different new era technologies like cloud, big data, artificial intelligence etc. The demand and the popularity of IoT is increasing day by day. New services using smart physical and virtual objects are solving problems like never before. The other side of the dissertation scope highlights the high security requirements and needs for effective data communication.

## **1.6 Chapter Organization**

The complete organization including chapter 1 with the structure and the brief highlights are as follows:

Chapter 1 Introduction completely covers the basic background of the IoT along with the focus on the motivation of the dissertation highlighting the objectives. It also explores the concurrent ideas behind the objectives.

Chapter 2 Elaborates and enhance the related work in the direction of the dissertation theme. The main approach is to explore the related research in the area of IoT with security and performance aspect of MQTT and other lightweight protocols.

Chapter 3 Illustrates the overview of the existing lightweight protocols and their needs in internet of things. It also provides the systematic comparison and evaluation.

Chapter 4 Discusses the proposed approach, algorithm, flowchart, block diagram and

the procedure of method used.

Chapter 5 Gives the complete explanation of the design, implementation structure, system requirements along with the list of examples. The complete requirement, specifications along with the concurrent design requirements have been analyzed and discussed.

Chapter 6 Covers the complete elaboration of the results. Results are explored and discussed based on analysis and enhancement observed with final discussion is based on the comparison of the results from the existing research.

Chapter 7 Provides the concluding remarks along with the insights on the future directions.

Followed by References and Publications.

## Chapter : 2

### LITERATURE SURVEY

#### 2.1 Background

Internet of Things (IoT) provides a better way of resource sharing by using different user and access level protocols without using any human and computer interfacing. It has been accessed and applied through different communication protocols. with data aggregation and specialization, these protocols works in different communication scenarios.

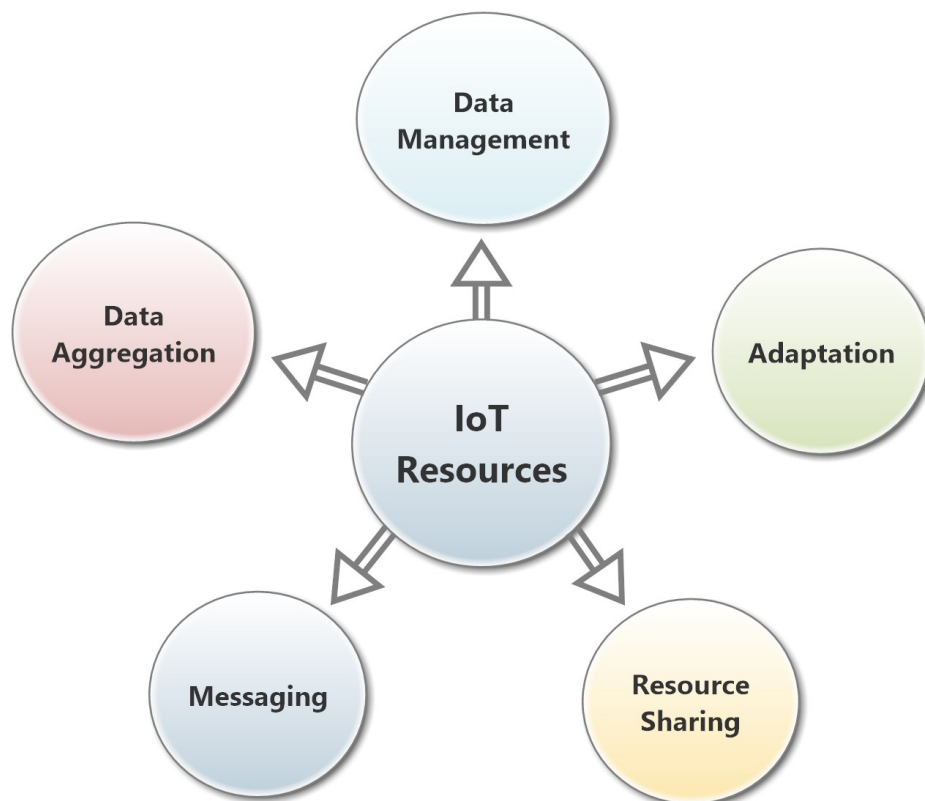


Figure 2.1: IoT Computing Resources

## **2.2 IoT Authentication and User Accessibility**

The IoT authentication and accessibility system have been discussed based on the protocol access and variation mechanism.

### **2.2.1 User Access**

User access is an important aspect in terms of IoT computing. As it should be under the control of the administrative authority. Hence, the level of access should be determined in such a way that it could handle the data regulation with the user access.

### **2.2.2 Security and Integrity**

In cloud computing, data security and integrity are an important concern. Some data constraints should apply and follow in the complete transaction.

### **2.2.3 Data Locality**

There should be some concern in the data locality with the proper surety of their confidential data. as the location of the user-stored data is unknown.

### **2.2.4 Replication and Recovery**

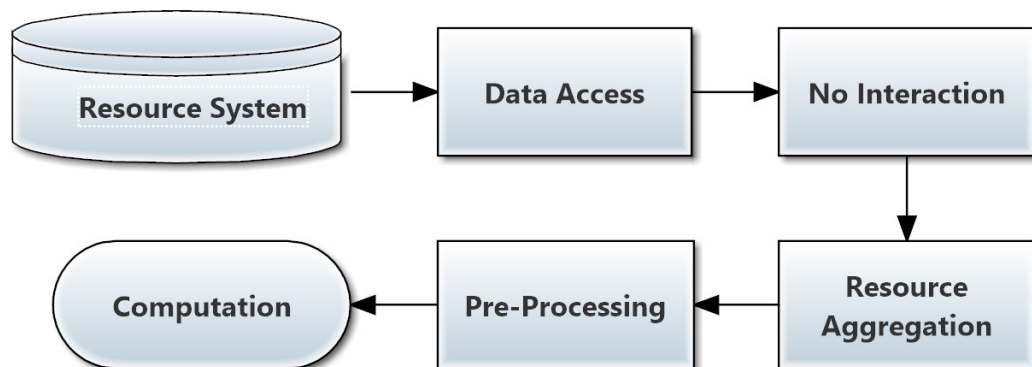
There is a need of proper data replication and recovery when required. Further, it can be activated whenever the data is stored on the location for particular services.

### **2.2.5 Investigative Support**

There should be some activity for the detection of illegal events. It should be found in such manner that it could be called and invoked accordingly.

### 2.2.6 Third Party Add-ons

There should be some protocols for the third-party add-ons as it can be vulnerable from the latest threats and achieving applicable adaptability for controlling it.

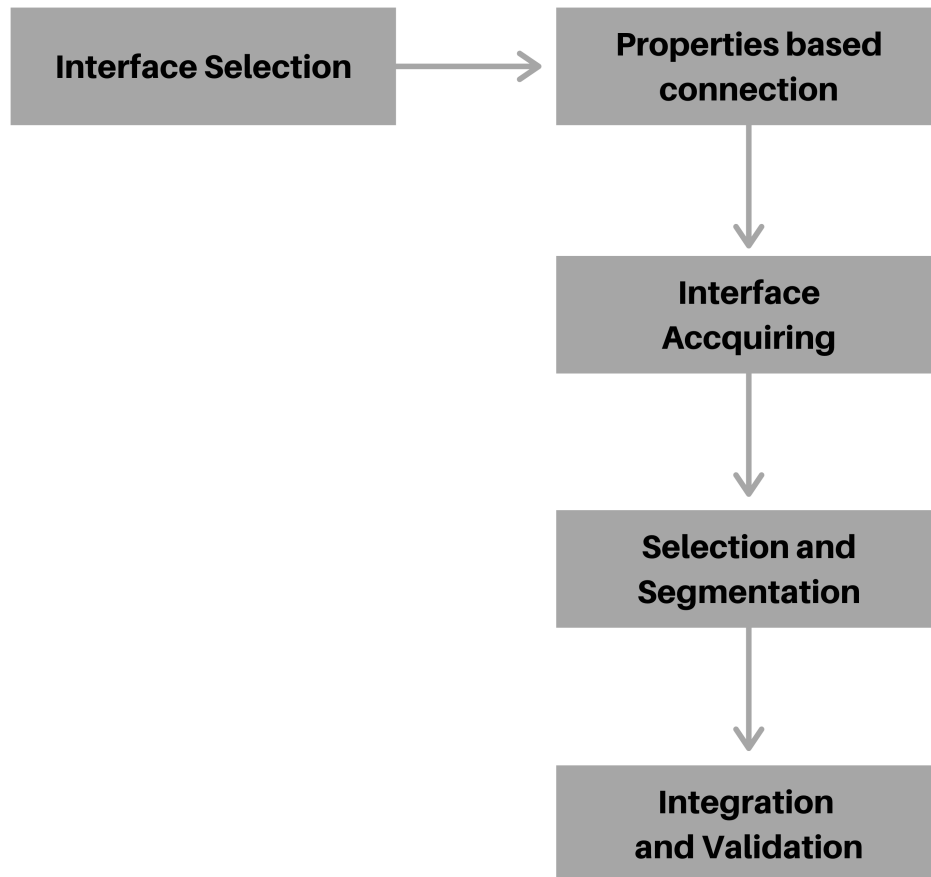


**Figure 2.2:** IoT Pre-processing Mechanism

Figure 2.2 shows the IoT pre-processing mechanism, It includes the mechanism for data access with similar pairs of activity for data sharing. It come up with the data aggregation, pre-processing, and computation process.

Figure 2.3 shows the procedure of data communication in IoT. It shows the data selection with the property's characterization mechanism. first it will select the interface then make a connection based on properties and acquire an interface. after selection and segmentation of data finally it checks the integrity and validity of the data.





**Figure 2.3:** IoT Communication Procedure

## 2.3 Security Issues and Challenges in IoT

The growth of information technology has also increased some new security issues. Whereas traditional security issues are becoming more severe. There are many resource-constrained devices like sensors, wearable devices, and other smart devices. It means they have less computing power and battery capacity. Therefore, they are lacking in security and robustness.

The data present in IoT devices is a personal information about any particular person and owning that information of users in detail. In addition, the privacy of the user

data is a serious concern, which need to be addressed in case of misuses of personal information [106]. Therefore, we need to lay down dedicated procurement and privacy rules that declassify and encrypts sensitive data before storing and distancing payloads of IoT data from information so that it cannot be used to identify any individual.

Moreover, the cached and other data which is no longer required should be disposed securely. Some security concerns are also due to management, like for small companies it is difficult to build secure applications for mobile and web based IoT applications due to small budget and insufficient manpower [107].

In addition, most of the manufacturers tend to focus more on getting the device or application on the market as soon as possible to start growing their user base and get more funding, but this action could risk a major security breach.

Backdoor problems are generated because of software vulnerabilities. Consequently, an attacker exercise plantation of backdoor in victim's system without any artifact. Also, other security mechanisms like antivirus or IDS are not applicable in IoT because of their restricted amount of computational power. IoT is also vulnerable to Malware; Due to the 24\*7 connectivity to the internet, it works as a hotbed to spread the malware [108].

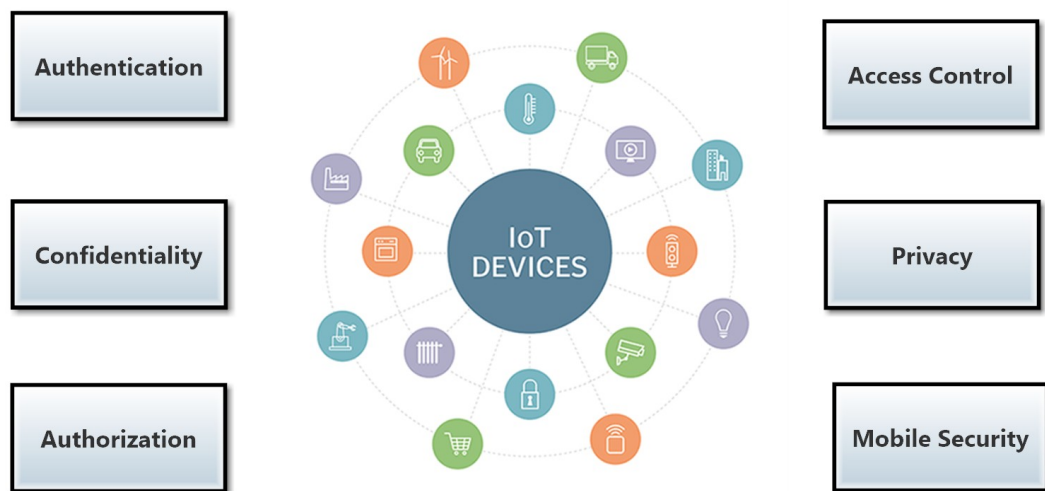
Even though IoT peripheral devices depends on wireless medium technology such as Ethernet and Bluetooth technology for end to end communication, such technology should either use essential updates and improvements to fix vulnerabilities already present in software or use updated version of the framework to take care of the security issues quickly.

a) The IoT system relies on the encryption methods used between the communicating nodes to facilitate end-to-end reliable communication between peers by the use of

unique keys.

b) Security layer contains all the devices that attempts to gain access to the IoT network, depending on the preferences of the company for successful implementation of the authentication process of IoT devices.

c) The third and important layer is the control of the communication path and division of communication paths by creating a group of devices based on types of network and functionalities (using Routers and Switches). Aside from monitoring other devices on the network, Bluetooth and IP-based firewalls may be powerful defensive measures for enforcing cyber threats.



**Figure 2.4:** Security Issues in IoT

### 2.3.1 Security Issues in Different Layers of IoT

#### Perception Layer

The technologies used in perception layer are RFID (Radio Frequency Identification), WSN (Wireless Sensor Network) etc. Some of the possible threats are addressed [108] below-

- a) Malicious Node: Attackers adds a malicious node to the victim's system through which they can get access to transmit malicious data over the IoT network to infect and get control of the entire IoT system. For such purposes attackers uses backdoor.
- b) Node capture: The peripheral devices in the network gateway are more prone to be captured which may results in the leakage and alteration of important information. This can results into risk of whole system security.
- c) Replay attacks: The Attackers sends previous messages to the receiving node of the IoT network in order to bypass the authentication process and network trust.

#### **Network and Transformation Layer**

The possible threats in these layers are related to confidentiality, integrity of the system. The main threats includes network intrusion, eavesdropping, DoS/DDoS attack, Man-in-the-middle attacks etc. Some of the possible threats are addressed below [109][110] –

- a) Scalability: IoT consists of large number of peripheral devices and these devices may enter and leave the network many times, thus network congestion can be challenging threat, also lack of authentication and authorization, and access control will be challenging.
- b) Heterogeneity: Due to diversified platforms of IoT, it is difficult to take care of the security challenges. As various technologies are involved, network coordination among different platforms and security of protocols are difficult to preserve.
- c) Data Revelation: The attackers may try to get important information from the IoT network. As lot of peripheral devices have large amount of data, it is quite easy to hack such device by using some information retrieval mechanisms.

### **Application Layer**

This layers requirement of security level depends on the application. The application requirements results in complicated and hard to secure application tasks. Some of the possible threats are addressed below [111] –

- a) Privacy: Individual privacy must be guaranteed for every connection. There can be two categories for privacy preservation. Data unionization and data collection policy [106]. Data collection policy can ensure the amount and type of information is restricted, and preservation of privacy. For data anonymity, cryptography protection and obfuscation of data is required. The obfuscation of data relation can remove direct relation between the owner and data.
- b) Data Management: As the data collection is huge, its complexity is growing rapidly and the resources are being exhausted. Such mechanisms may also incur loss of data.
- c) Device Identification and Authentication: To prevent any illegal access, each application should be able to perform authentication. Authorization should also be implemented as attackers may intervene communication and collect the data packets by making some change in routing table in the gateway. Even if Secure Socket Layer technique is already applies, the attackers are able to bypass the forged certificate.
- d) Specific Vulnerabilities of the Applications: There exist vulnerabilities in IoT systems, which can be used by attackers that may results in development of some modules (injection of some unwanted code) of which user is unaware.

### **2.3.2 Security issues in MQTT Protocol**

Message Queue Telemetry Transport (MQTT) protocol works on TCP/IP and provides the lightweight, less bandwidth, and high reliability to the IoT system.

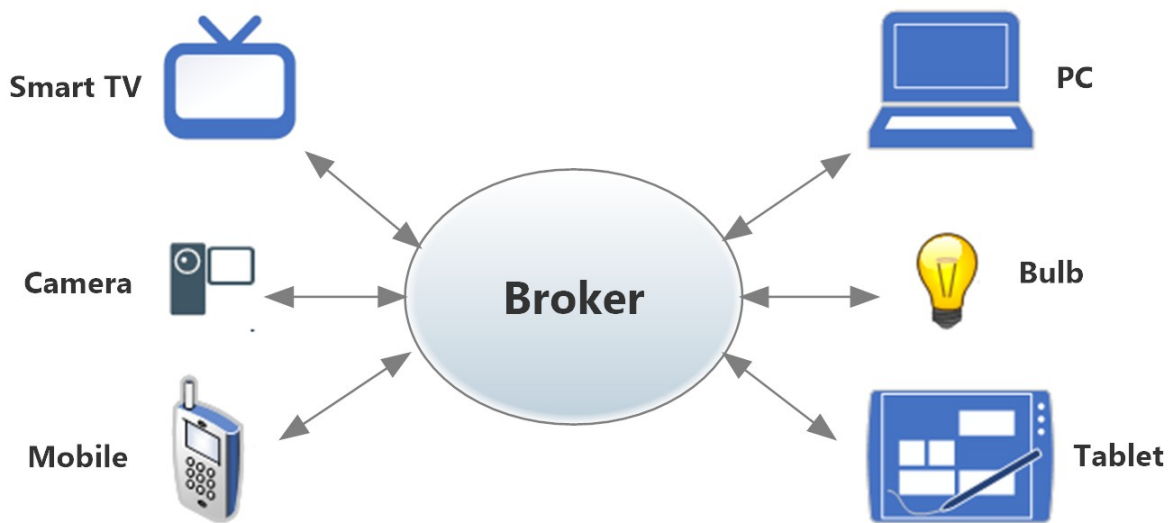
The MQTT protocol consists of a broker and the MQTT client. The MQTT client consists of a subscriber and the publisher. In the IoT network, the message packets communicate through the MQTT protocol. The main purpose of the broker is to transmit message packets to the MQTT client. MQTT protocol consists of various types of messages like CONNECT, DISCONNECT, and PUBLISH etc..

In CONNECT message packets create a connection between the client and broker. After that, the MQTT client transmits data to the broker using the Publish command. Then, the DISCONNECT command is used to disconnect the TCP/IP session between the client and the broker [114,116,118].

Quality of service (QoS) can specify every connection according to the overhead between client and broker. QoS is classified according to the cost defined as fire and forget, acknowledged delivery, assured delivery. Fire and forget delivery consists of message transmitted only once.

The client and broker do not acknowledge the delivery of the message. Acknowledged delivery consists of message packets sent multiple times up to when it receives acknowledgment delivery of the message. The assured delivery consists of a handshake between the broker and client so that only a single copy of the message is received during the handshaking process [119].

MQTT protocol consists of a broker that acts as an intermediate between the IoT devices and MQTT client. However, the smart device forgets to receive the message from the broker. Then, the broker stores these message packets in their memory. Hence, a large number of message packages store in the broker and degrades its performance [115]. Fig shows the general architecture of the MQTT protocol including the IoT devices and user or client.



**Figure 2.5:** Working Model MQTT

MQTT protocol design for the low processing devices. Therefore, The MQTT generally develops for minimizing the processing power of IoT devices. The MQTT protocol consists of various security requirements in terms of confidentiality, integrity, authorization.

1. **Issues in Confidentiality:** Confidentiality refers to safeguard the secret, sensitive information from unauthorized users. Privacy can be assured using encryption. However, the MQTT protocol does not encrypt the packets between the communicating channels. Hence, the adversary can easily obtain data information from the eavesdrop packets.
2. **Issues in Authentication:** Authentication refers to the process in which the appropriate or genuine user can able to access the server; otherwise, the system denied to use these services. In general, the MQTT protocol, CONNECT message packet uses to transmit username and password for authentication. MQTT protocol uses TCP protocol for forwarding the packets. Therefore, these message packets cannot encrypt by communication. Hence, these packets can

easily extract by the adversary. Then, the adversary can able to un-authorized access to the system [117].

3. Issues in Authorization: Authorization is a process in which the appropriate authorized device can access the server; otherwise, it denied these services. So, every authorized user or client can subscribe and publish all topics without the use of proper authorization. The topic or message is implemented on the broker side to restrict the client to subscribe or publish by authorized topic [113].
4. Issues in Integrity: In MQTT protocol, the untrusted user can access the sensitive data to the MQTT broker while the TLS connection is not using by the MQTT protocol. MQTT supports MAC, Digital Signature, and checksum to provide the integrity mechanism in power constraint devices.
5. Eavesdropping Attack: The adversary capture packets or essential credential communicating between the broker and IoT smart devices. After that, the adversary can use these credentials to unauthorized access to the MQTT broker.
6. Man in the Middle Attack: The adversary listens to the communication between the authentication server and the IoT devices. After that, the adversary eavesdrop packets between the communication channels. Then, the adversary modifies these eavesdropping packets or essential credentials and sent them to the credential to gain access to the system.
7. Replay Attack: In the replay attack, the adversary eavesdropping packets between the communicating channels. After some time, the adversary sends these genuine packets to unauthorized access to the system.



8. Rogue Attack: The IoT devices and user communicates with each other using the authentication server. The adversary brief that the attacker node is genuine and the user can interact with this node. After that, the user or device grants authentication token to these malicious nodes. Hence, the attacker node able to unauthorized access to the system.
9. Node capturing Attack: In the node capture attack, the adversary capture the IoT device. After that, the adversary extracts the stored secret information from the capture device. Then, the attacker uses this secret information to unauthorized access to another system in the communicating channel [113].
10. DOS Attack: In Denial of service attack, the adversary transmits many packets to the communicating network. These large numbers of packet create jam in the communicating channel. Hence, a genuine packet cannot communicates with the authenticating server. Due to this, the authenticating server cannot available to the legitimate user or IoT device.

## **2.4 Data Transfer Mechanism in IoT**

In IoT there are different kinds of architecture connections that can be used for the data transfer purpose. These are as follows:

- Device to Device(D2D)
- Device to Gateway(D2G)
- Gateway to Data System(G2D)
- Between Data Systems

**Table 2.1: Comparative Study on Web Services Protocol**

PARAMETER	SOAP	REST
Protocol used	Protocol	Architectural Style
Data format	XML format	Plain text, JSON, HTML, XML, etc.
Usability	Can't use REST	Can use SOAP web services.
Standards	Strictly followed.	Loosely followed
Bandwidth requirement	More Resources and bandwidth	Less bandwidth and resource
Security	Describes its own security.	Inherits security measures from the underlying transport.
Preference	Less preferred	More preferred

### 2.4.1 Comparison of SOAP and REST Web Services

Web services provides a common platform that allows multiple applications to build on various programming languages to have the ability to communicate with each other so it's a standardized medium to propagate communication between the client and server. The major two categories inculcated in web services are simple object access protocol (SOAP) and REST (Table 2.1).

Web services use SOAP for sending XML data between applications and REST is one way communication and the connection is intermittent so server needs to wait. on the basis of comparison it can be concluded that protocol domination is not possible because of the layover possibility. As it is very clear that the advantages and disadvantages vary in each and every protocol. The advantage of SOAP is the security provisioning and disadvantage is the heavy weight protocol [14, 15]. REST

is lightweight but there is some security concern is in their communicative operations. So, SOAP is better in security as comparison to the REST protocol.

### **2.4.2 Comparison of MQTT and CoAP Protocol**

Some of the standard protocols have been used in the IoT for the better functioning. For resource- constrained devices MQTT and CoAP protocols are used [16, 17]. Some features of MQTT and CoAP protocols are as follows:

- Supports open standards.
- It is more suitable as comparison to HTTP for the constrained devices.
- Communication supported is asynchronous.
- IP has been used.

Table 2.2 shows the parametric variations and the comparative analysis.

### **2.4.3 Security of the IoT applications**

The web-based application security may be determined with different security features along with the communication protocols.

#### **Multi-Tenancy**

The implementation of MQTT-SN support low cost device implementation. It mainly focuses in the traditional MQTT. If it is compared with CoAP the automation and design can be separated automatically.

#### **Data Operability**

It is the term for the data operation along with the requirements needed for the communication in terms of the data acquisition and following the prerequisite for the object communication.

**Table 2.2: Comparative Study on MQTT and CoAP**

Parameter	MQTT	CoAP
Mode of communication	M2M communication with central broker	Direct M2M communication
Functional communication protocol	Many to many communication protocol	One to one communication protocol
Architecture	Broker architecture	REST architecture
Communication protocol	TCP	UDP
Model	Publish/Subscribe model	Request/Response model
Header size	2 Bytes	4 Bytes
QoS	3 level	2 level
Security	Transport layer security/ secure socket layer	Datagram transport layer security
Messaging	Asynchronous communication model	Both (Asynchronous & Synchronous Model)
Number of message type	16	4
Type of communication	Remote communication	Local communication
Speed of transmit cycle	Slower	Faster
Resource discovery	Flexible topic subscription	Stable resource discovery mechanism

## 2.5 Comparative Discussion Based on IoT Application

The lightweight protocols comparison is shown in Table 2.3. based on the comparative study and discussion it is clear that as the security protocol REST and HTTP uses HTTPS. DTLS has been used by MQTT-SN, CoAP, and LWM2M. CoAP also uses IPSEC. MQTT-SN and MQTT uses TLS. In MQTT and MQTT-SN data size is small. It has 2-byte header file. The size may vary form 4 byte for the MQTT-SN. Same like MQTT-SN CoAP has 4 bytes header file. XML documents as their data in case of LWM2M and SoAP. Large message data size is in the case of HTTP.

## 2.6 Related Study of IoT Protocols and Challenges

In [23] Aman et al., Suggested the security under limited resources is the main concern in IoT adopted frameworks and systems. They have suggested that the IoT devices low cost nature may arise the chances of attacks.

In [24] Pramukantoro et al., Discussed the challenges in case of developing IoT middleware. They have suggested that limited computing capacity is the major concern in single middleware. It is also suggested that due to the enhancement in the middleware cluster scalability can be improved. Their objective is to enhance the scalability and for that, they have applied a Redis message broker. The number of clients they have considered are 100, 500, 1000 and 1500 clients.

In [25] Huang et al., Discussed about the gaps bridging between physical and digital world in case of IoT. They have suggested fog computing as the key component for this purpose.

In [26] Alqinsi et al., Discussed about the uninterruptible power supply (UPS) mon-

**Table 2.3: Comparison of Lightweight Protocols in IoT [16, 18, 21]**

	MQTT	MQTT-SN	CoAP	LWM2M	REST	HTTP	SOAP
Full Form	Message queue telemetry transport	MQTT for Sensor Networks	Constrained application protocol	Light weight M2M	Representational state transfer	Hyper text transfer protocol	Simple object access protocol
Architecture	Publish/Subscribe	Request/Response	Request-Response, Publish-Subscribe	Request/Response	Request/Response	request/response	request/response
Need of broker	required, send devices communicate via broker	Required, send and receive messages.	No broker required.	Context broker	not required, end devices direct communicate	Hybrid broker can be used.	WebSphere Message Broker
Transport protocol	TCP/IP	UDP	UDP, TCP	UDP	TCP/IP	UDP	HTTP and SMTP etc
Security protocol	TLS	DTLS	IPSEC or DTLS	DTLS	HTTPS	HTTPS	It is already secure.

Scope	device to cloud cloud to cloud	client inside network, the broker is outside on Internet	Device to Gate-way, Gate-way to Device	Device to de-vice, De-vice to gateway	Device to cloud, cloud to cloud	Device to de-vice, De-vice to gateway	Device to cloud, Cloud to cloud
Design Methodology	Protocol is data centric.	protocols suited for sensors network like ZigBee, Z-Wave	Generic web protocol for special requirements	Protocol is data centric.	Protocol is data centric	Protocol is document centric.	protocol is document centric
Message size	Small, binary with 2Byte header.	2 or 4 bytes Header, smallest 2 bytes, Largest 65535 bytes	4 Bytes	XML document	Small, in different formats	Large, ASCII format.	XML document
Service levels	3	5	2	-	-	1	-
Implementation	remotely perform service enablement and application management	implementation on low-cost, battery-operated devices	smart energy and building automation	enables device management for M2M	used over nearly any protocol	Used by the World Wide Web	protocol that allows windows and Linux to communicate using HTTP

itoring system by the consideration of MQTT protocol. They have suggested that it can be helpful in solving UPS monitoring issues. It is also adaptable for the large infrastructure. and explored the proof of small size message data. In this work Arduino microcontroller is used. In addition, web server as the raspberry, MQTT broker, MQTT subscriber and database. They have monitored the UPS parameter using a web-based application.

In [27] An and Kim, Discussed about the information-centric networking (ICN). They have explored their approach in terms of wireless domain. It has been implemented in terms of light weighted protocols used for communication. ICN-based content delivery scheme has been presented for the IoT. It shows the efficiency in terms of supporting it as for the seamless hand-off.

In [28] Chippalkatti et al., Discussed the issue of traffic in the urban areas. They have proposed an architecture. Their architecture is useful for the small module and for the efficient handling. They have designed an efficient system. It can be helpful in the time management and real time monitoring system. It has been used for finding the empty slots. And could be used for car parking anywhere.

In [29] Celia and Cungang, Discussed IoT as a heterogeneous system. They have suggested that the key management security is important for the authentication. It is useful in different security operations. They have presented an interactive and non-interactive key management protocol. It has been presented to reduce the communication cost. Their approach is found to be useful in case of various attacks.

In [30] Anthi et al., Discussed the security and privacy issues in IoT. They have suggested that these devices hold crucial information like username and passwords. So, they have suggested the need of intrusion detection system (IDS). And it can be helpful



in monitoring IoT ecosystems. They have proposed the initial stage development of Pulse IDS model. It adopts the machine learning capabilities. It can be helpful in identifying the denial of service attacks.

In [31] Bellagente et al., Discussed about the traditional healthcare approaches. The drawbacks suggested by them are mainly the higher cost. In this paper, it is mentioned that the connected health paradigm may help in reducing the cost. They have suggested that with the help of IoT personalized treatments can be possible efficiently and presented an approach based on the message-oriented middleware.

This approach is useful in the interactions for the CH actors. They have used JavaScript object notation for the data modelling with the advanced message queuing protocol and implemented a proof-of-concept prototype. Their approach is capable in providing a feasible solution.

In [32] Chandrappa et al., Discussed about the security monitoring system. They have presented an efficient IoT smart security system. It is capable in monitoring and informing the authorized person if the intrusion is detected. The presence of intruder alarm is also provided by this system.

They have developed a theft control unit (TCU) based on IoT. So, they named this system as IoT-TCU. Their alarm rising and message sending system is found to be successful.

In [33] Gao et al., Proposed a secure lightweight which is based on MQTT. It is IoT based thing-centered system and categorized in three parts. In the first phase critical functionalities identification has been performed. Secure pair designing and binding has also performed in this stage.

In the second phase end-end to encryption. Finally, a strong authentication mecha-

nism has been designed for different types of attacks. They have implemented their approach on a prototype of SecT on a USD 10 Raspberry Pi Zero W. Their results support their approach.

In [34] Iftikhar et al., Proposed a method for the fusion of the smart devices. They have used GPS and the IoT devices for the location finding and message delivery. Their proposed approach is suitable for warning, review and request messages and evaluated their framework on single and multiple recipient framework.

In [35] Kamalraj and Sakthivel, Discussed about the IoT system in case of children safety. It is mentioned in this work that there are several issues that can be handled by the IoT based systems. They have used sensors along with the IoT components.

They have suggested that by using sensor traceability an alert message can be generated for tracing the location and synchronization of children and parents. Alcohol and smoke gas sensor can be helpful in finding the abnormal conditions of the child. It is suggested that by the help of different input data measurement and decisions children's can be save.

In [36] Irmansyah et al., Discussed the development of information technology in terms of IoT. They have designed a heart rate monitoring device with low cost. It is enabled with the SMS notification alert system. Their sensor is capable in converting the heart rate into the bit per minute and the database is relying on the TCP IP communications server.

Doctors and patients can check the information through the website. Their system has the capability of warning system through the SMS when the bit per minute is low by 60 and above 100. Their system is helpful for the doctors in the supervision.

In [37] Landge and Satopay, Discussed about the security aspect in case of connected and embedded device through the internet. They have mentioned the need of security in case of several embedded devices. And used MD5 that has the capability of satisfying the security concern in case of IoT. They have proposed an autonomous mechanism to enable MD5 in case of IoT for embedded device when it is connected through the internet.

In [38] Kulik and Kirichek, Discussed about the problems in case of industrial IoT. They have raised various industrial protocol for this. They have developed a software architecture for the data formats transformation for the problem raised. They have also proposed data formats transformation between different industrial protocols and conducted message conversion for the related means.

In [39] Leshem et al., suggested the need of robust security requirements and the need because of the high messaging volume. The high messaging volumes generate the probability of attack. Therefore, there is the need of robust security requirement. They have suggested the need of encryption mechanism along with the frequently changes in the required keys.

They have used key distribution and the constant key construction. The limitations of IoT devices as the poor memory, storage, and processing bandwidth. Moreover, there is a need of IoT-devices as the unique key per the conversation. Their proposed protocol is mainly based on the probability analysis. They have suggested that it ensures a common key. It also capable in pairing the IoT devices. Their approach shows the strength in the security protocol for IoT networks.

In [40] Lenk et al., Discussed about the IoT devices. The geographic information is needed in many devices like vehicles. They have suggested the use of block ciphers

in case of storing geographic information and proposed a new algorithm. It has been used for storing geographic spatial information. It has been used successfully in IoT devices and cloud.

In [41] El Kashef and Barakat, Discussed IoT in terms of monitoring, controlling and for the security properties. They have proposed an intelligent alarm system. And adopted machine learning techniques. They have used different sensors and Arduino microcontroller. It can be notified based on email message based on real time scenario. Their results show that their approach has the capability of achieving high performance with low false positive rate.

In [42] Kurera and Navoda, Discussed about the exchange data mechanism for the IoT devices. The several of the IoT devices are low power devices and they have suggested that their design supports to use it as the low-power devices. It is also the need of less electric power in this case and the need of secure data transmission. They have calculated the cost computation in case of encryption and decryption of messages.

Therefore, they have suggested that all secure transmission protocols not need to be IoT friendly. They have proposed a secure data transmission protocol for the IoT devices specially low-power. Inherited the features of Kerberos and onetime password concepts.

They have used symmetric key cryptography and achieved good security performance in case of strong authentication and providing the secure data transmission among different nodes.

In [43] Lin et al., Discussed IoT in terms of client devices as well as the client gateway. It provides services for transforming the raw streaming. It can be helpful in the extraction of different valuable information. They have suggested that RabbitMQ service

cluster can be used to connect the clients as well as the other cloud services. Their proposed mechanism supports the monitoring system for the resource usage by the client. Their system is capable in generating throttling rules as soon as such activities can be identified. Their results have shown feasible solutions with less overhead.

In [44] Kim et al., Discussed for the integration of industrial message protocol. They have suggested some minor changes in case of the existing system and the manufacturing message specification protocol provides communication in case of network support the industry. The main drawback is in the controlling of the multiple devices as the single message unable to control multiple things.

In [45] Kamal et al., Discussed IoT in terms of security protocols in case of limited resources and scalability. They have suggested that due to space limitations several cryptographic solutions are not feasible. They have proposed a light-weight protocol. Based on this correlation coefficient has been calculated.

According to the authors the value of the correlation coefficient is directly proportional to the secured data transfer. They have suggested that the adversarial node detection in case of lower value. Their results show 97 percent correlation achieved in case of adversarial node.

In [46] Kumar and Raza, Discussed about the MQTT protocol in terms of IoT and suggested the need of MQTT implementations in terms of load balancing in case of multiple brokers. It is mentioned that the information sharing between brokers increases in case of larger number of clients.

They have proposed a topic structure for the location-dependent data. It is also incorporable in case of handling location-dependent data. They have also introduced distributed brokers and gateways. It is also helpful in case reducing broker load and

in supporting the heterogeneous brokers. Proposed system is effective in terms of of prototype implementation and theoretical evaluation.

In [47] Madritsch and Klinger, Explored the use of computing cluster using IoT Technologies. First, they have explored the goal of development and their expected application. Then in the next phase is in the design and implementation of the group of students in detail.

Authors have demonstrated by the experimentation for the practical use of the cluster. They have used the message passing interface and open multi-processing methods for exploring the ability and functionality.

In [48] Mukhopadhyay et al., Discussed about the current anti-theft systems. It is mentioned that the current system lacks the tracking and monitoring function. They have discussed these systems in terms of vehicle. Novel security system have been proposed. It is mainly based on the wireless communication and a low-cost Bluetooth module.

They have elaborated the model and It consists of GSM which has been used for sending messages and have employed a keypad password. It is capable in controlling the safety locker door and controls the seat belt wearing. It has been connected to the Bluetooth module as well as the alarm system. It has the capability of transmitting an alert signal that is helpful in sending the alerts.

In [49] Narang et al., Discussed about the IoT in terms of intelligent system. They have discussed the advantages and disadvantages with the exploration and analysis in terms of current approaches. It also shows the fundamental and security requisites, the chances, and their current impact.

In [50] Nasir and Kanwal, Discussed the authorization and security aspects in terms of IoT. And several algorithms have been proposed. They have discussed a RFID based mechanism as the previous approach and suggested that in this mechanism disclosure attack is possible. They have proposed an approach based on the related approach to protect against the disclosure attack.

In [51] Ni et al., Discussed about the causality's discussion regarding the children and the patients. They have proposed an IoT based intelligent life monitoring system. It is the combination vehicle interior environment and emergency monitoring system. They have designed a demo system based on the above mechanism.

In [52] Oak and Daruwala, Discussed about the information sharing between two devices in the communication. It is considered as the layered system. They have suggested that security protocol is not inbuilt in MQTT by default. Their main aim is to enhance the security features in case of MQTT protocol. They have also compared the MQTT protocol with the encryption algorithms.

In [53] Peniak and Franeková, Discussed about the secure communication in terms of embedded devices with the conjunction of IoT. They have mainly focused on the MQTT. Their approach has the capability for achieving the secure communication.

In [54] Meera and Rao, Discussed about the IoT system along with the MQTT protocol for the enhancement and enrichment in the communication.

In [55] Chen et al., Discussed the mobile payment protocol in context of IoT system. They have explained that the existing mobile payments system authentication mechanism suffers from heavy workload. And presented a lightweight protocol for solving this situation. For this they have proposed a unidirectional certificate less proxy re-signature scheme.

They have provided a new mobile payment protocol. Their proposed protocol supports the improvement in the computational cost. They have also proved the security concern through CDH problem. It proved to be efficient in case of resource-limited smart devices in IoT.

In [56] Harbi et al., Discussed interconnection between objects in the IoT devices. They emphasized that data aggregation may plays an significance role in massive amount of data and requirement is high in case of privacy of sensed data. They have proposed an efficient cryptographic scheme. It has been helpful in securing aggregation and transmission of sensed data in case of wireless sensor network.

Their proposed approach is based on the elliptic curve cryptography and message authentication code. It is used in end-to-end security mechanism. Their results show that the performance of the proposed approach is better in comparison to the related work.

In [57] Huynh-Van et al., Discussed about the over-the-air (OTA) programming for the IoT system. They have discussed that main limitations in the programming development is the memory resources and the battery energy and concern about the security. They have applied advance encryption standard (AES) algorithm on the Deluge on TinyOS 2.x have been presented.

They have provided the runtime enhancement on the IoTs devices like Telosb mote on TinyOS 2.x also. And deployed the solution based on real IoT based systems.

In [58] Choi et al., Discussed authentication process as the important and crucial aspect in IoT devices. Study and analysis have been performed in case of authentication protocol.



In [59] Eldefrawy et al., Discussed about the security concerns in case of IoT technology. They have discussed about the several security concerns regarding the authentication scheme. And discussed about the industrial IoT (IIoT) technology. They have mentioned that there is lack of security resources in case of IIoT.

They have proposed an improved user authentication protocol for the IIoT system. And achieved secure remote user authentication. It does not require timestamping. Their protocol only needs Hashing and XOR. They have used Tmote Sky node for experimentation. They have also used Scyther verification tool.

In [60] Mamour and Congduc et al., Discussed about LoRa approach along with the long-range radio technology. Long-range radios commonly expel the multifaceted nature of keeping up a multi-jump connect with middle of the road hubs for handing-off data. In any case, even with the expanded range, 1-jump network can be hard to accomplish in certifiable sending situation, particularly for remote what's more, rustic zones where thickness of doors is low and where gadgets/passage are normally sent for a particular application. They portray a 2-bounce LoRa way to deal with consistently expand a sent LoRa organize so as to diminish both parcel misfortunes and transmission cost.

They have presented a shrewd and battery-operated transfer gadget that can be included after an organization crusade to straightforward give an additional jump between the remote gadgets and the door. They have introduced a battery-operated smart relay-device. It can be helpful in the remote devices and the gateway.

In [61] Gebremichael et al., Discussed about the secure group communication. They have discussed about the security insurance in case of secure group communication and adopted fast symmetric-key encryption. This supports a lightweight group key

establishment scheme. They have used one-time pad mechanism. Their scheme is convenient and helpful in case of IoT group applications. In this applicability, the nodes are resource-constrained. They have suggested that this scheme for resource-constrained nodes.

In [62] Hribar and DaSilva, Proposed an updating mechanism. It is capable in learning the collected content of information. It has the capability of frequency reduction. They have used the correlated information and based on this proposed mechanism has been evaluated. Their result shows the increasing efficiency through their approach.

In [63] Mostafa et al., Discussed the quality control arrangements in the direction of IoT security and system.

In [64] Muhammad et al., Discussed M2M initiative in terms of IoT. They have presented the implementation of secure MQTT. And mentioned about the security for the different IoT applications. They have considered the smart home real world application for testing.

In [65] Munsadwala et al., Discussed about the exponential growth and causes for air pollution. They discussed the issue of IoT in terms of the GPS sensor system and used MQTT protocol, which plays an important role in data acquisition unit. They also have the dashboard for visualizing the stored data and given email alerts, activity log, verification of geographical location and system authentication features. on the basis of that they suggested that, the authorized user has the ability to control from a remote location.

In [66] Poulter et al., Addressed the secure remote update protocol. They have applied the method for the formation of a secure transmission of messages. explained the development of a Python wrapper for SRUP. They recommended that there is a need

to write less code with minimal effort by using this feature.

In [67] Alhazmi and Aloufi, Addressed and analyzed the architecture, applications and need of IoT. Recommended the adoption of a conventional cloud or IoT focused data center. They pointed out that the main issue with cloud-based data center is that it is very far away, hence fog-based IoT was suggested. They recommended that it could be helpful to analyze quickly, so it is good for applications that are time sensitive.

Indicated that the adaptability of various aspects of security would be beneficial. It could be dependent on available resources and adaptability with various output constraints. The respective challenges were discussed alongside the performance analysis. They are based on MQTT protocol for their proposed protection scheme. In terms of sensors, they have addressed IoT with the GPS system. The protocol used MQTT was useful in the data acquisition unit.

In [104] Hwang HC. et al., Designed reliable message transmission system which is based on MQTT protocol. The proposed design extended the payload area of MQTT protocol for data verification in reliable message transmission system. The authors studied the background such as IoT environment, transmission systems, their importance and several design issues. Various protocols were also a part of literature discussion. Comparison of HTTP, XMPP, CoAP and MQTT protocols was also presented in work.

The proposed methodology of the work included assigning sequence numbers to all messages. The assigned sequence number was requested by each client before sending message to receiver. Towards receiver side each client checked the sequence number of the received message and ask for the re-transmission to the sender on finding a missing message.

The research described reliable message transmission system based on MQTT protocol and verified through experiment. Fast message delivery and guaranteed ordering of the messages for IoT environment was ensured experimentally.

Table 2.4 shows the methodological comparison based on the method and results achieved. It shows the elaboration based on the method and approaches. It has been discussed and analyzed based on the results and discussion.

**Chapter Summary:** MQTT protocol is used for communication within an IoT environment that functions on top of TCP/IP. MQTT is a messaging protocol that uses the publish-subscribe communication model. In this protocol clients themselves do not require updates, causing a reduction of used resources which makes this protocol optimal for use in low bandwidth environment.

The protocol functions on a server called broker, pushes updates to MQTT clients and clients will not send messages directly to each other instead relying on the broker. Every MQTT message contains a topic and is organized in a tree-like structure to which the clients can subscribe or publish.

MQTT can be a good choice when compared to other protocols like HTTP, CoAP, XMPP etc. because it has smaller footprints (MQTT has very short message header and smallest packet size of two bytes) so suitable for resource-constrained environment.

**Table 2.4: Related Work Discussion**

S.No	Reference	Method	Approach	Results Achieved
1	[68]	S-MQTT	They have proposed a secure MQTT protocol version. The encryption algorithm they have used is elliptic curve cryptosystem. They have also proposed a multi-tier authentication system. This provides the extra security layer for the data theft prevention.	Their results shows that by the use of ECC with CP/KP-ABE has the better and secure lightweight communication.
2	[69]	Publish-Subscribe message broker	They have described about the refactoring process. It can be helpful in different levels of latency.	They have suggested that their event handler has the capability of achieving end-to-end latency of an average of 50ms.
3	[70]	SensPnP	They have presented a plug-and-play (PnP) for the problem of solving third part integration. Their presented approach has the capability of communication protocols for the heterogeneous embedded peripheral. They have also proposed automatic driver management algorithm for IoT device with the connected sensors. Their experimentation has been performed on the real test-bed.	Their results shows the efficiency in terms of minimal memory footprint, energy consumption reduction along with the PnP time reduction. Their results shows that the cost is also less.

4	[71]	Human actors in IoT	They have provided an impactful analysis and study on the requirements and design for the Cyber-Physical System (CPS) which is mainly for the integration of the human actors. They have also presented a comprehensive human-integration framework. It is a part of multiagent IoT middleware. They have also presented the capabilities of this middleware based on the human integration.	Their approach suggested the capability for the data model capability with the agent-based IoT middleware architecture.
5	[72]	MQTT L2 network	They have suggested that the establishment of large number of communication in the IoT arises the need of information centric network (ICN). It has been done for the IP-based communication cooperation. Their performance has been evaluated based on different parameters.	Their results show the reduction in the variation and the communication delay.
6	[73]	Message-based IoT systems	suggested that there is the need of security in the case of critical data processing. They have introduced a LUCON system for this. It is a data-centric security policy framework. It has been used for the distributed systems as well as for the message controlling system. Their policy enforcement has been considered for the message routes verification. They have also evaluated the performance in terms of run time impact.	Their run time impact is better in terms of related research.

7	[74]	Secure vaults authentication in IoT	<p>They have suggested that the mutual authentication services and security system is an important aspect of the IoT system. They have presented mutual authentication mechanism based on multi-key. They have suggested the need of secure vault and the vault equalization should be maintained based on the equal sized keys. It has been shared between the server and the IoT device for the vault change mechanism for the security. They have used Arduino device and concern their feasibility for the IoT devices and covered with the memory and computational power constraints.</p>	Their results supports their approach.
8	[75]	Fuzzy-based fog computing	<p>This work suggested that due to the large volume of data increase in IoT devices of healthcare, increases the data traffic values along with the consideration in different burdens of the cloud computing environment. There are several high network latency along with the high service latency with the large data transmission may increase the overhead tedious and unmanageable. They proposed a method based on the fuzzy-based fog computing architecture. It has been applied based on the fuzzy data packet allocation (FDPA) algorithm with fog services.</p>	There mechanism has the capability of providing data packet allocation based on different virtual machines.

9	[76]	Fuzzy c-means algorithm for filtering spam messages	They have suggested that the email plays an important role in data communication with the ease of fast and effective communication. So, there is attack possibility in the communication media during email. It is also through the spam messages. Spamming is the utilization of informing or electronic informing framework that send immense measure of information. Spam regularly fills the web with numerous duplicates of a message and are sent to various beneficiaries over and again without their solicitation and desires to open them. They have analyzed analyze different machine learning techniques. It is with the combination of with and without feature selection. It has been applied for the spam classification.	Their results support the approach.
10	[77]	IoT protocols under constrained network	They have suggested that there are several communication protocols have been developed for helping in efficient communication in IoT devices. IoT working mechanism intended to run the applications with constrained resources. A quantitative assessment of Their objective is to assess the performance of IoT messaging protocols. It is in case of restricted wireless access network. The use case considered here are M2M specifications.	In this, experimented different messaging protocols belongs to IoT for the performance comparison. Their approach supports the dynamic selections of the protocols.



11	[78]	MQTT protocol in IoT environment	They have suggested that MQTT is highly used protocol in IoT. IT supports transport layer security (MQTT-TLS). Because of the number of autorizing rule it may infeasible. So they have proposed MQTT thing-to-thing security (MQTT-TTS).	Their results supports the approach.
12	[79]	Middleware technology for laboratory environmental monitoring	According to the authors IoT is an important aspect in the computer network. They have focused on the integration of IoT and middleware technology. They have classified IoT researches as internet-oriented, things-oriented , and semantic-oriented. They have provided the IoT middleware for laboratory environmental monitoring.	Their result shows that their proposed IoT can efficiently monitor the light and sound in the laboratory environment. It shows high accuracy.
13	[80]	Light weight security framework	They have suggested that IoT comprises of complex network. It is of smart devices. It consist of large number of nodes. They have suggested that the solution for energy dissipation is not possible in traditional cryptography techniques. They have used RSSI values. It is used for the generation of the fingerprints. It has been used for the correlation coefficient matching.	They have used light-weight protocols for maintaining the security.

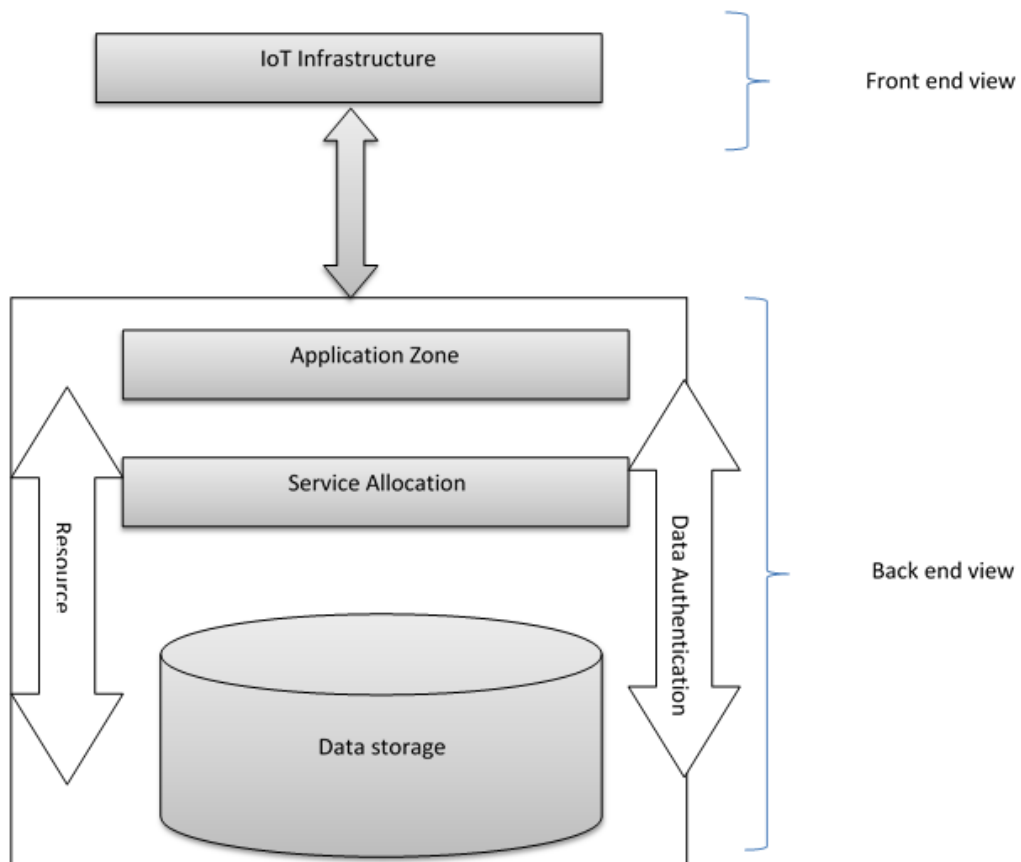
14	[81]	Messaging models in IoT	Analyzed the technologies, architectures, communication templates, has been used for the communication messaging models. They have suggested that there is no such approach for the messaging model. It allows application-layer protocols for building the IoT. Their main focus is for the selection of the messaging models and application-level protocols. It is for the critical and noncritical systems in IoT. They have also compared the interrelated characteristics of IoT.	Their technique was tested along with the corporate system example.
15	[82]	Automated authentication credential derivation	Author has suggested that the number of embedded systems and resource constrained devices have been increased due to IoT trends. and most of the devices used default settings and authentication credentials, so not secure enough. They have presented a device configuration approach. It is capable in automatic authentication credentials derivations. They have also highlighted the security benefits.	They have highlighted the security and threat analysis based on the successful attempts in threat identification.
16	[83]	Object authentication for cyber security in IoT	Authors have suggested that cyber security is an important concern. Due to efficient and technological use of internet makes the way of living effortlessly. It is also possible due to the use of artificial intelligent system.	They have provides the idea for the intelligent and artificial system in case of IoT system.

## Chapter : 3

# COMMUNICATION ARCHITECTURE

### 3.1 Architecture of IoT

Figure 3.1 shows the architecture of Internet of Things (IoT) computing. It clearly shows the front and back end architecture of IoT. It shows the area of different service application, along with the service, storage and infrastructures area. It also shows the data management and security layer based on user demand and its use.



**Figure 3.1:** IoT Service Architecture

## 3.2 Lightweight Protocols in IoT

Internet Protocol (IP) based IoT definition and standardization has been done through the internet engineering task force (IETF) [18]. Following discussion is based on the working groups:

1. IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) work group: It is concern with the IPv6 adaption with the IEEE 802.15.4 as there are several amounts of host for the connection [85].
2. Routing Over Low power and Lossy networks (ROLL) work group: It is focused on the routing design techniques which are used for the resource constrained devices networks with high packet drop rate [85].
3. The Constrained RESTful Environment (CoRE) work group: It provides the application layer definition protocol. It is used for the resource-constrained devices which also called as constrained application protocol (CoAP) [19]. It belongs to the HTTP protocols. The main dependency of this protocol is dependent on the representational state transfer (REST) architecture [20].
4. Message Queue Telemetry Transport (MQTT) protocol: IBM and OASIS [21] introduced this protocol. It is helpful in providing the embedded connection. In first phase the connection is in between the applications and middleware's. In second phase, it is between the networks and communications. It supports the architecture that contains the main elements like publishers, subscribers, and a broker.

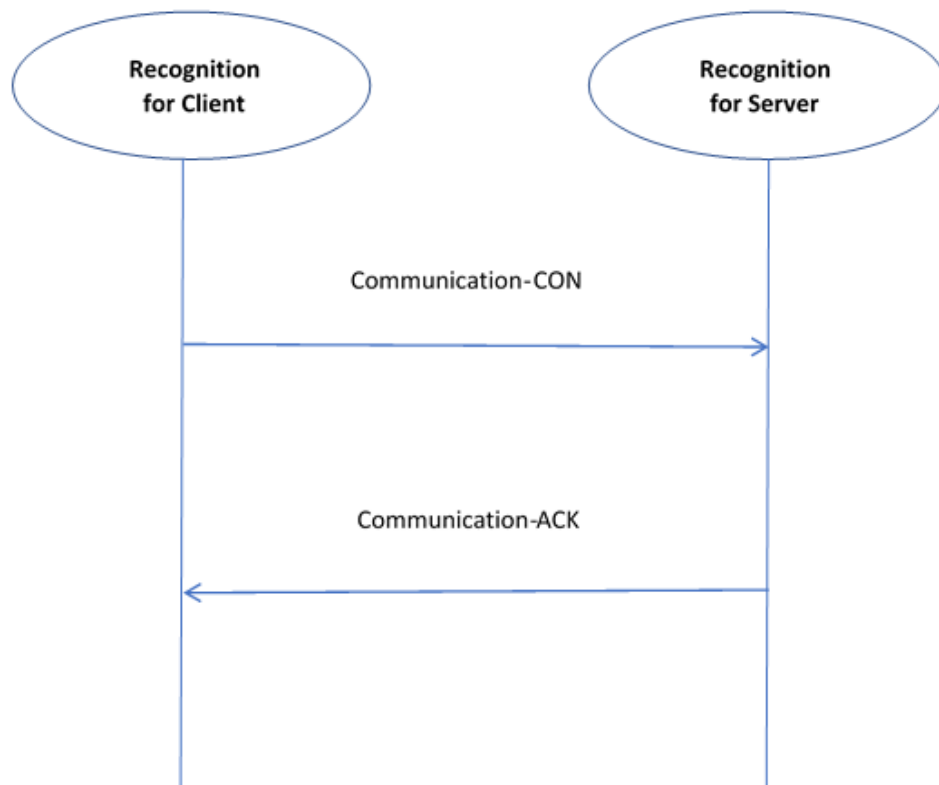
5. Advanced Message Queuing Protocol (AMQP): It is a session layer protocol. that is designed for the financial industry. TCP is required for the running and implementation. It provides publish/subscribe architecture. There are two fundamental parts: trade and lines. The trade distribute messages and conveys them to lines dependent on pre-characterized standards and conditions. Lines characterize the subjects and bought in by endorsers, which will get the information given by the sensors at whatever point they are accessible.

### **3.2.1 Constrained Application Protocol(CoAP)**

CoAP supports the interaction model in a way to handle the data to support the variability in the context of data. It supports the client/server model. The main properties are as follows [16, 19]:

- The lower layer is the message layer. It is adopted on the user datagram protocol. It also supports the asynchronous switching. The notation used here are CON for the confirmable, NON for the non-confirmable, ACK for the acknowledgement and RST for the Reset. These above four messages are very important in message layer.
- In case of upper layer, the request/response protocol helps in the main communication, which can be able to communicate with the lower layer. Figure 3.2 shows the process of message transportation in CoAP[19].

The main responsibilities of CoAP is reliability, messages duplication and data communication. In this duplication and reliability may be adopted and handled by the message. Request/response layer is for the handling of communication part [16, 19].



**Figure 3.2:** Process of Messages Transportation in CoAP

### 3.2.2 Simple Object Access Protocol (SOAP)

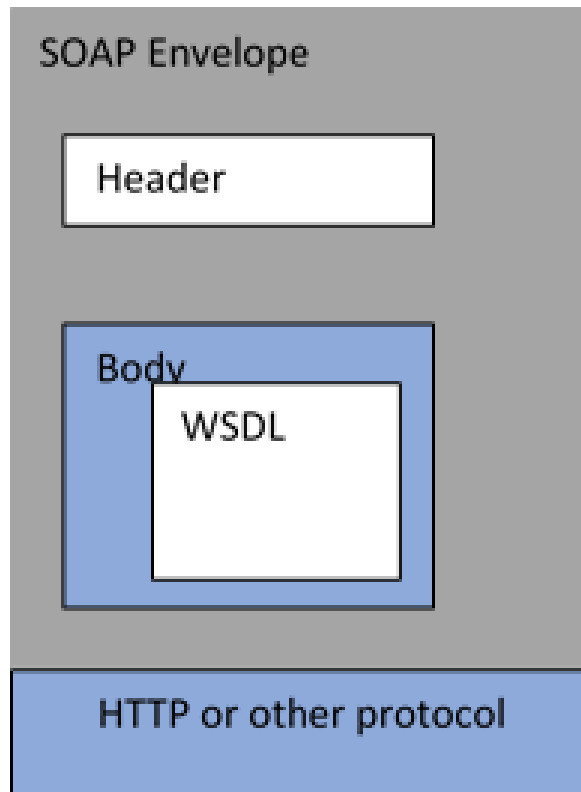
Then web services come in the picture. It is helpful in enhancing the communication. As the name implies it is mainly dedicated for the services provided by the web. It is basically of two types:

- Representational state transfer (REST): It is an architecture for designing of network applications. HTTP protocol is used for the client server architecture [8].
- Arbitrary web services: This is the combination of operations in arbitrary nature along with the SOAP messages.

Communication is possible in web service and client through message exchange. It is similar to the working mechanism of HTTP. Their message format relies next to the simple object application protocol (SOAP). REST message uses HTTP, XML formats [9] and it follows the JavaScript object notation (JSON). SOAP is an XML

specification application.

Figure 3.3 shows the architecture of SOAP[10]. It supports the architecture that



**Figure 3.3:** SOAP Architecture

is capable in envelope inclination [10]. It is a complete variant of JavaScript object notation (JSON) and REST.

SOAP message format consists of the following:

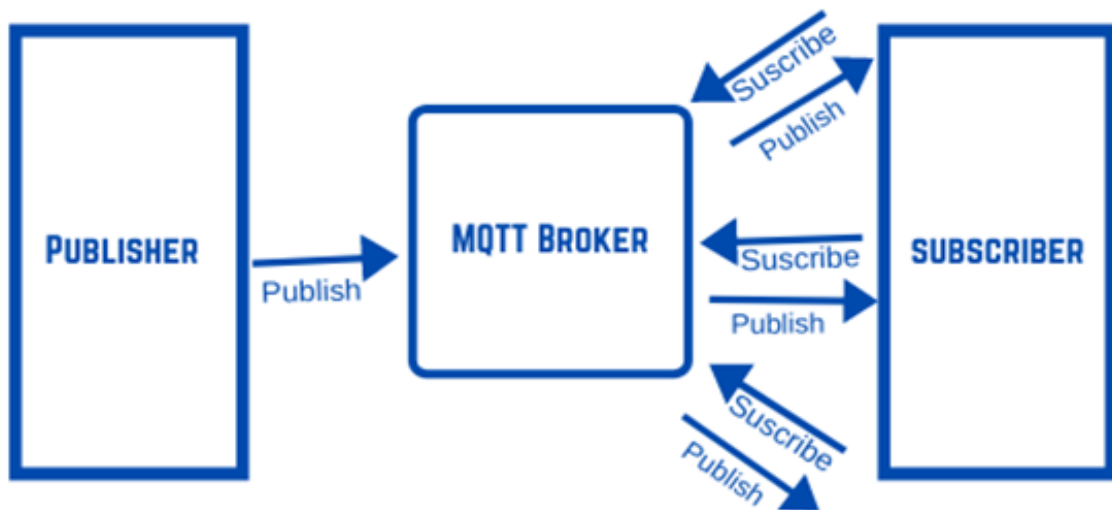
1. Envelope
2. Encoding style
3. Header
4. Body

### 3.2.3 Message Queue Telemetry Transport Protocol (MQTT)

IoT having some standard protocols for handling the communication and messaging. MQTT and CoAP are two main popular protocols in IoT. These are the most effective protocols for resource- constrained devices [16, 21]. MQTT and CoAP protocols features are as follows:

- They support open standard.
- It is best suited for the resource-constrained devices.
- It supports asynchronous communication.
- For implementation and smooth operations, IP is needed.

Figure 3.4 shows the MQTT architecture. It has mainly three main components. These are publisher, subscriber and a broker [21, 22].



**Figure 3.4:** MQTT Architecture



### **3.3 Connection Establishment in MQTT Protocol**

Protocol MQTT is defined as a binary protocol, which has command and command acknowledgement format [112]. We can say that every time when client send a command to the broker, broker will send an acknowledgement where whole communication is based upon TCP/IP protocol.

The procedure for the same is, there will be a TCP connection and then MQTT connection establishment after which data transfer will take place. When the process complete TCP connection will terminate. The client needs to send commands to the broker for every function as a command and command acknowledgement based protocol.

#### **3.3.1 Communication in MQTT after Establish Connection**

In figure 3.5 Client will take following steps after connection establishment [112]:

#### **3.3.2 Steps for Subscription**

Figure 3.6 shows the steps for subscription [112]

#### **3.3.3 Basis Packet Formats in MQTT Protocol**

The MQTT packet consists header (two byte fixed + variable) and payload. Variable header and payload are not present in every packet but fixed header always present with size two byte. Figure 3.7 shows the basic format of packets[112].

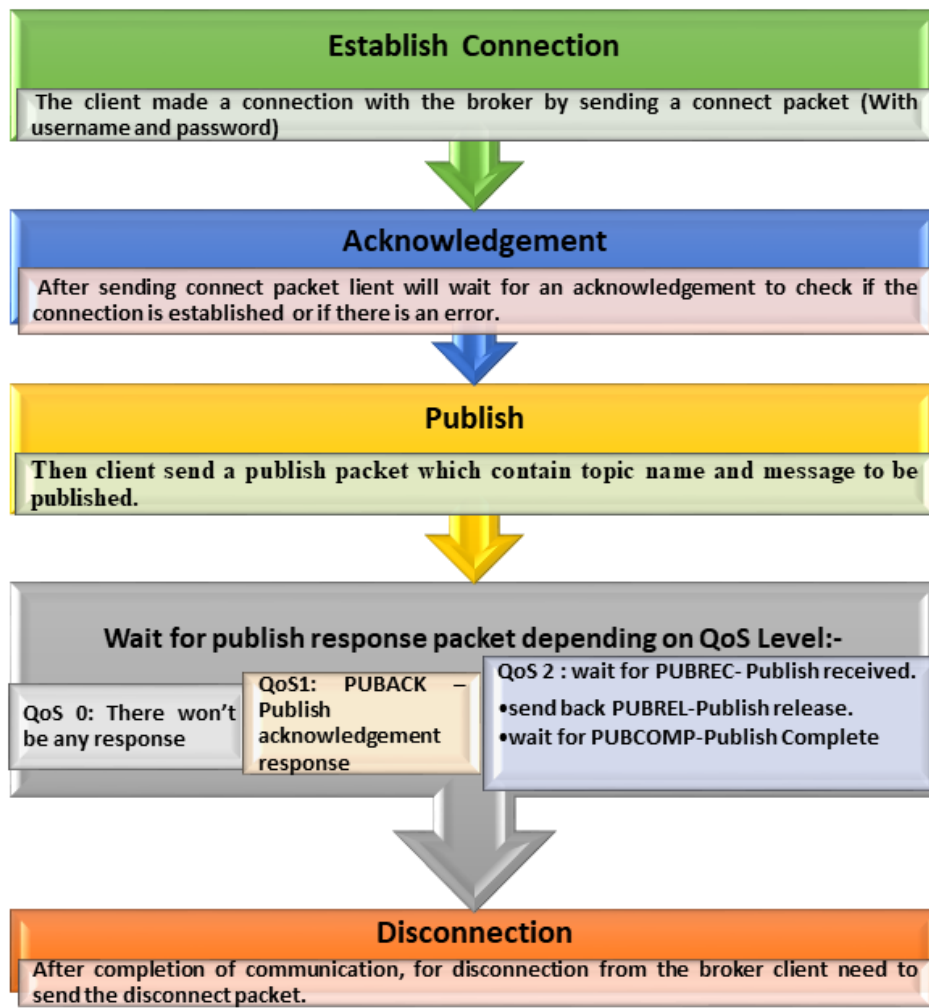


Figure 3.5: Communication in MQTT

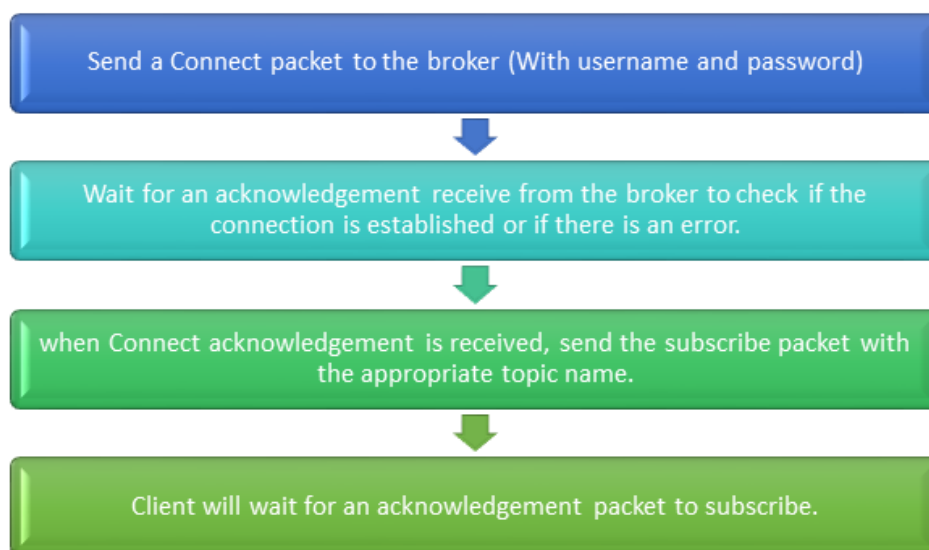
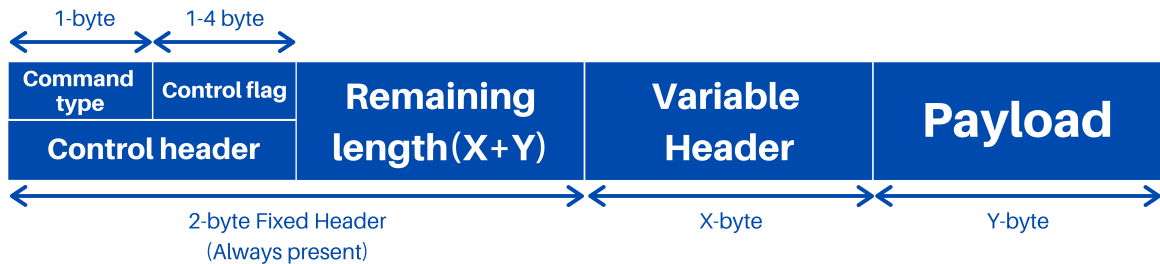


Figure 3.6: Steps for Subscription



**Figure 3.7:** Basic Packet Format MQTT

### Control Header

The 2-Byte fixed header contains the following fields:

#### Command Type

- The first byte is the control Field which is divided into four fields.
- The first 4 MSB bits are command type field, suppose the value of the connect command is 1. That means for connect command the connect type field should be 1 which is 0001 and for publish command the value is 3, so, the connect type field should be 0011. The following given below table will describe the Packets and their values. Figure 3.8 shows the basic command types in control field.

#### Control Flag Bits

The next 4 bits are known as control flag bits which are used by the publish command for the rest of the commands and are reserved with the value 0. For PUBLISH command the 0th bit denotes if the message that is published is retained the message. 1st and 2nd bits are used to select the quality of service if it is 0 or 1 or 2 and the 3rd bit denotes if it is a duplicate message. We can see the table given below for all these

COMMANDS	Name	Value	Direction of Flow	Description
	Reserved	0	Forbidden	Reserved
	Connect	1	Client to Server	Request to connect to server
	Connack	2	Server to Client	Connect Acknowledgment
	Publish	3	Client to Server or Server to Client	Publish Message
	Puback	4	Client to Server or Server to Client	Publish Acknowledgement
	Pubrec	5	Client to Server or Server to Client	Publish received (assured delivery part 1)
	Pubrel	6	Client to Server or Server to Client	Publish released (assured delivery part 2)
	Pubcomp	7	Client to Server or Server to Client	Publish complete (assured delivery part 3)
	Subscribe	8	Client to Server	Client subscribe request
	Suback	9	Server to Client	Subscribe Acknowledgment
	Unsubscribe	10	Client to Server	Unsubscribe request
	Unsubpack	11	Server to Client	Unsubscribe acknowledgement
	Pingreq	12	Client to Server	Ping request
	Disconnect	13	Server to Client	PING response
	Reserved	15	Forbidden	Reserved

Figure 3.8: Commands Type in MQTT Packet

control packets. Table 3.1 describe control flags[112].

### Remaining Length

The second byte of the fixed header, which is the remaining length, contains total length of variable header and payload. Remaining length can use up to 4 bytes in which each byte uses 7 bits for the length and the MSB bit being a continuation flag. Suppose, if the continuation flag bit of a byte is 1, it means the next byte is also part of the remaining length and if the continuation flag bit is 0, it means that byte is the last one of the remaining length. E.g. if the variable header length is 10 and the payload

**Table 3.1: Control Flags in MQTT Packet**

Control Packet	Fixed Header Flags	Bit 3	Bit 2	Bit 1	Bit 0
CONNECT	Reserved	0	0	0	0
CONNACK	Reserved	0	0	0	0
PUBLISH	Used in MQTT 3.1.1	DUP1	QoS2	QoS2	RETAIN3
PUBACK	Reserved	0	0	0	0
PUBREC	Reserved	0	0	0	0
PUBREL	Reserved	0	0	1	0
PUBCOMP	Reserved	0	0	0	0
SUBSCRIBE	Reserved	0	0	1	0
SUBACK	Reserved	0	0	0	0
UNSUBSCRIBE	Reserved	0	0	1	0
UNSUBACK	Reserved	0	0	0	0
PINGREQ	Reserved	0	0	0	0
PINGRESP	Reserved	0	0	0	0
DISCONNECT	Reserved	0	0	0	0

length is 20, then remaining length should be 30.

### Variable Header

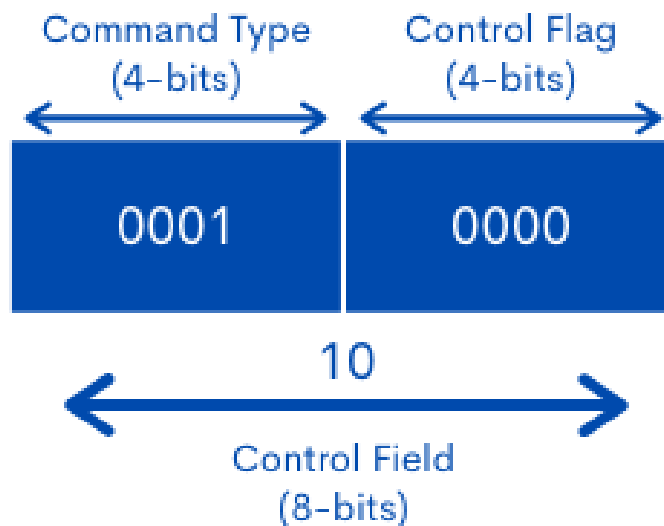
A variable header is not present in all the MQTT packets. There are some MQTT commands or messages which uses this field to provide additional information or flags and they vary depending on the packet type. A packet identifier is common in most of the packets types.

**Payload:** The packet contains a payload in the end of the packet which is optional and may varies with the type of the packet. The main function of this field is to contain data which is being sent. E.g. for CONNECT packet the payload is client ID and ‘username and password’ if they are present. And for PUBLISH packet, it is the message to be published.

### 3.4 Various Types of MQTT packets

#### 3.4.1 CONNECT Packet

The first byte of the connect packet will be 10 as the value of CONNECT command is 1, the first 4 MSB will be 1 and there are no flags so the next 4 bits will be 0. Figure 3.9 shows the control field in connect packet.



**Figure 3.9:** Control Field in Connect Packet

Protocol Name Length	Protocol Name	Protocol Level	Connect Flag Byte	Keep Alive	Client ID Length	Client ID	Username Length	User Name	Password Length	Password
<b>Variable Header</b>					Payload					

**Figure 3.10:** Format Variable Header in Connect Packet

In the variable header, there is always a protocol name and for this, the first 2 bytes should mention the length of the protocol name followed by the protocol name, as in the given below, the protocol name is MQTT which is of length 4.



**Figure 3.11:** Length in Variable Header

The server identifies the MQTT traffic and if found invalid protocol then the server may reject the connection and Thus, we cannot give any name to a protocol. The protocol level determines which version of MQTT it supports as for version 3.1.1, the protocol's level is 4. and if the protocol is not supported by the server it disconnects by sending an acknowledgement with return code 01.

Bit	7	6	5	4	3	2	1	0
	<b>User Name Flag</b>	<b>Password Flag</b>	<b>Will Retain</b>	<b>Will QoS</b>		<b>Will Flag</b>	<b>Clean Session</b>	<b>Reserved</b>
	X	X	X	X	X	X	X	0

**Figure 3.12:** Flags Position

Next two bytes are used to mention the keep alive duration in seconds. For 60 seconds, the value will be 003C in hex. After the variable header, there will be the payload and it will contain client id, username and password. In the given below case there is no username and password, so only client Id will be present.

It's easy to determine the remaining length. If the total bytes used for variable header and payload is counted and it is 17. So the remaining length is 17. Figure 3.14 shows the final connect packet

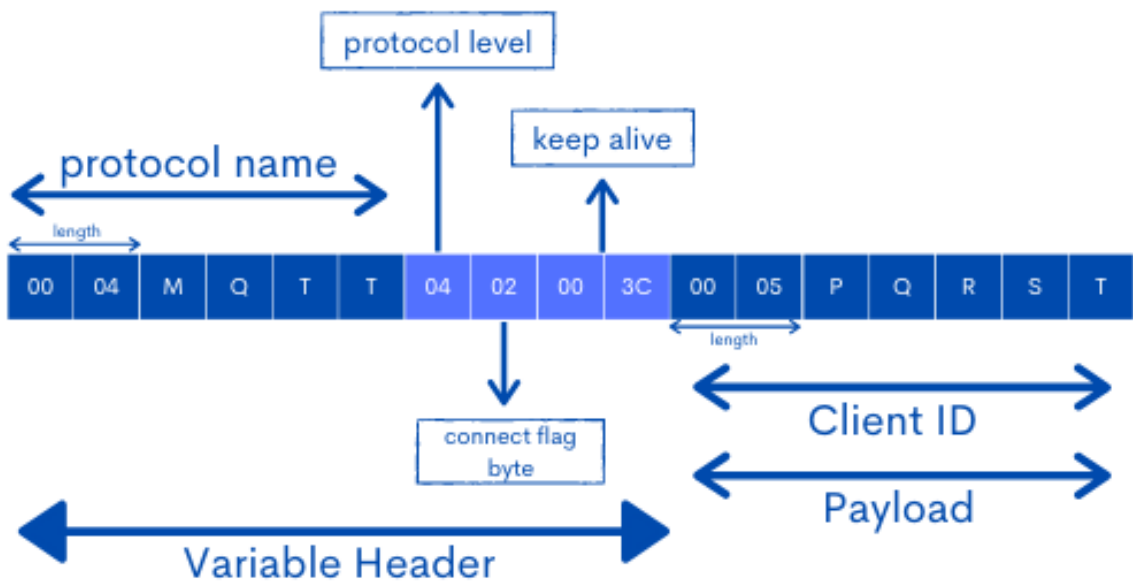


Figure 3.13: Variable Header and Payload in Connect

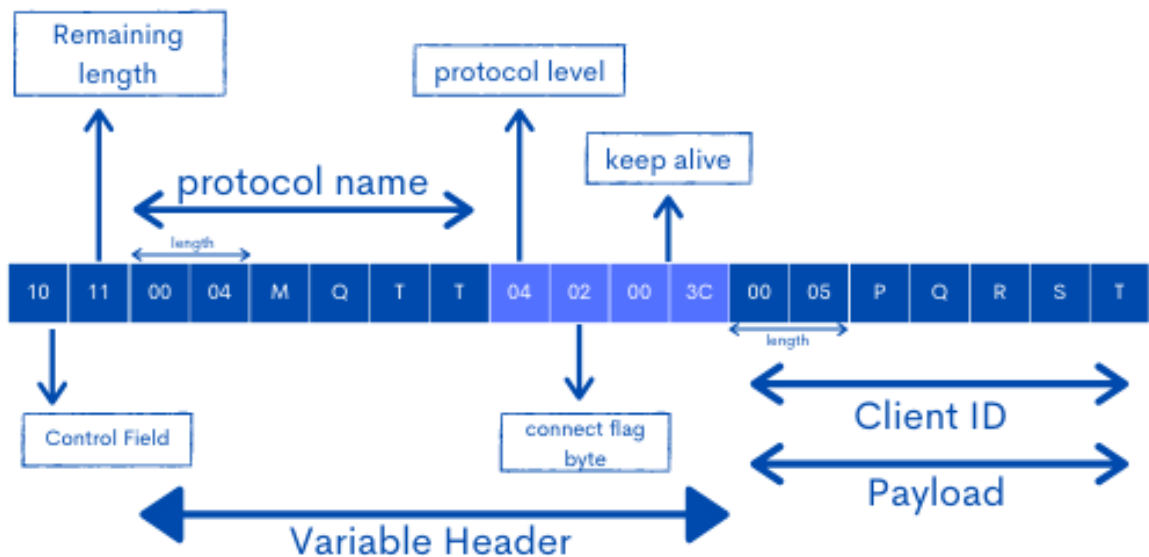
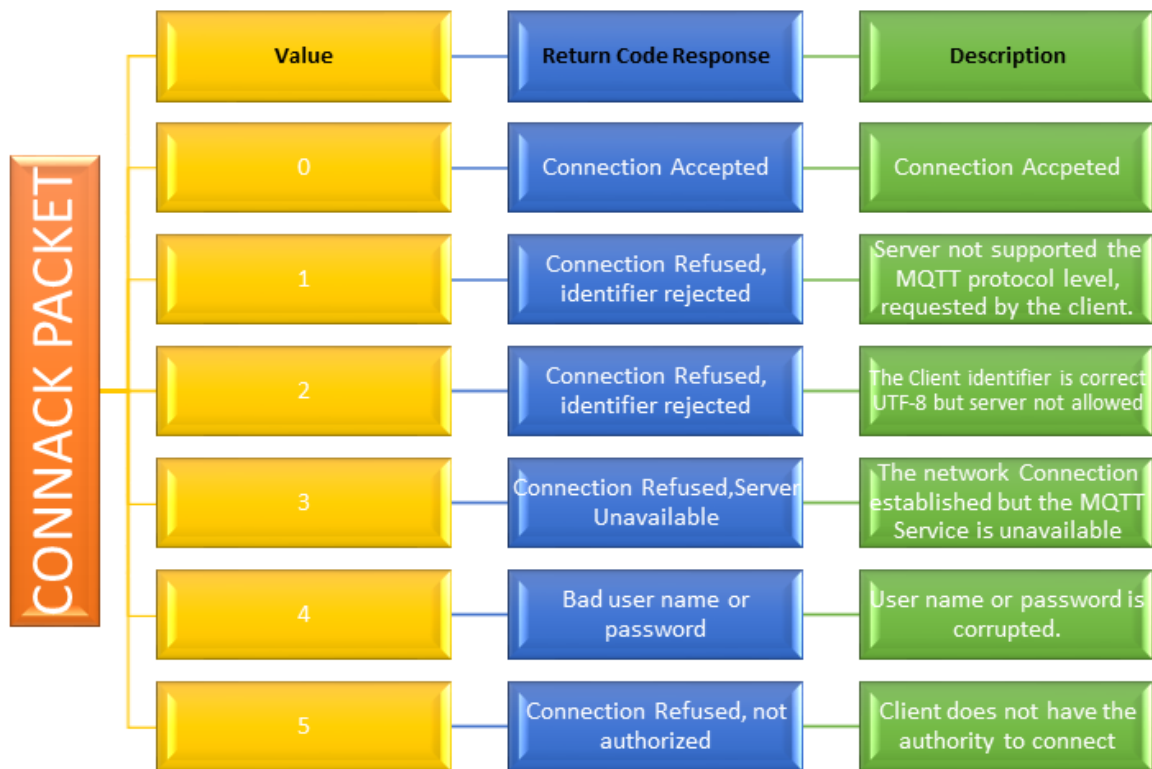


Figure 3.14: Connect MQTT Packet

### 3.4.2 CONNACK Packet

Once the connect packet is sent, and the broker receives the connection it will send back the acknowledgement as CONNACK. variable header of CONNACK will show that the connect return code. Thus, we can know the reason behind the rejection. The





**Figure 3.15:** Connect Acknowledgement Values

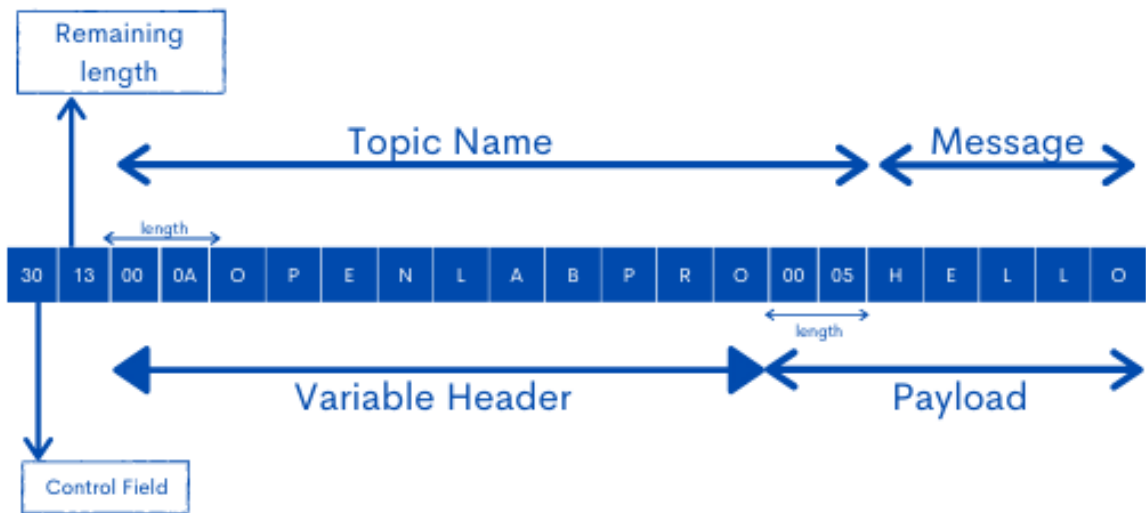
values and other descriptions can be shown in the following ways:

### 3.4.3 PUBLISH Packet

For publishing HELLO to the OPENLABPRO, we need PUBLISH packet Command value, which is three. The QoS level 0 with and without retaining the message control flag will be zero. In the variable header section, first two bytes define the topic length and then followed by topic. Similarly, in payload section first two bytes define the length of the message followed by the actual message.

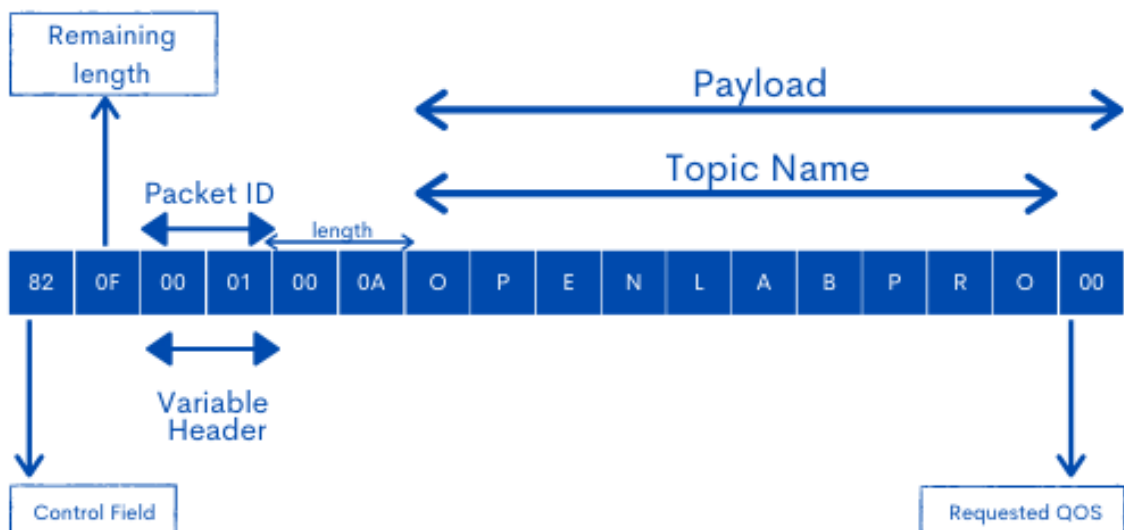
### 3.4.4 SUBSCRIBE Packet

When the message is published and if there are any subscribers for that topic, they will receive the message. For subscribe to a topic the client needs to send the SUBSCRIBE



**Figure 3.16:** Publish Packet MQTT

packet. The Command value of Subscribe packet is 8, and the Control flag is reserved which is 2. The variable header contains a 16-bit packet ID. In addition, payload, there will be the topic to subscribe followed by requested QoS level. To subscribe to the topic OPENLABPRO with QoS 0 as we can see below:

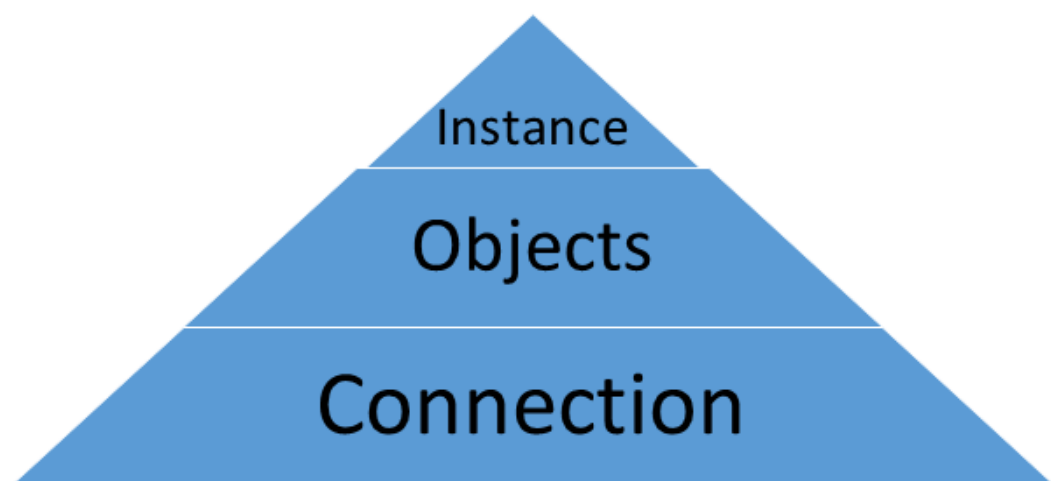


**Figure 3.17:** Subscribe Packet MQTT

### 3.5 Computing Performance

High-performance IoT computing through the cloud computing-based data center is very far. But if it is suggested through the fog based IoT. It can be helpful in the way for which it is applicable for high-priority applications. It is recommended regarding the adaptability of different security aspects. It can be based on the resource's availability and adoptability with different performance constraints.

The prospects are also discussed in terms of respective challenges along with the performance analysis. It can be suggested in terms of the MQTT protocol. The technique which is in the form of different communication protocols and the ability to generate the prospect of other conventional-issue are relied on the communication protocol. The main theme is to generate the communication criteria for the arena of different IoT based system for the regeneration of the transmission media and their smart devices. The services and applications are shown in Figure 3.18.



**Figure 3.18:** IoT Services and Application

## **Chapter : 4**

# **NOVEL MESSAGE TRANSMISSION AND SECURITY METHOD**

### **4.1 Design and Development**

The Internet of Things ( IoT) is rising technology nowadays in the latest trends and scenarios. That is applicable in many ways such as intelligent transportation, wearables, telemedicine, and intelligent home to smart city. Different kind of devices are plotted at different places in mentioned applications, but effective method to protect the data are required here.

IoT devices are resource-restricting devices and need a lightweight method to improve security. Thus, fuzzy technique are used to find the vulnerabilities in the system by passing unexpected information on the input data and then analysis the system. This work deals with the weighted based fuzzing technique to increase the efficiency of IoT system. The Scapy method is used in the weighted technique and this method operates in the block based protocol. This makes it simple and automatically to recalculate the length of fields.

Message queuing telemetry transport (MQTT) protocol is used as a lightweight protocol for the exchange of information. Therefore it is used to evaluate the fuzzy technique based security method.

The objective of this research is to develop and design the model of Integrated Identity technique to minimize message loss and test it in IoT environment. Integrated ID developed on the push message service of MQTT. It is a method of combining the

hexadecimal number in the actual message and the ID is used to retrieve the data in the edge devices if the message is lost.

## **4.2 Structure of the Proposed Approach**

The smart devices increase drastically through the internet and growing rapidly. There are different constraint-based connection showing the structure to abide with different sensors for monitoring it. Several problems occur due to increased number of devices. The main problem arise is the high message loss. It is due to the different messaging and transmission techniques used. This requires effective method to minimize the message loss in the various devices with the monitoring system along with the security aspects.

The system structure is designed and developed for providing data security of the uploaded stream along with the message loss control. This situation provokes the data adaption with better controlling mechanism that may be helpful in effective data handling also.

## **4.3 Proposed Method**

Internet linked devices growing over the years, now billions of devices, including smartphone, sensors, energy meter etc., are connected through the internet. This causes the data traffic, which leads to message loss in control devices. Hence, an efficient method is required for lightweight protocol that can transfer data without loss.

The overall system structure is shown in Figure 4.1. In order to transfer data effectively Integrated ID approach is proposed here whose flowchart is explained in figure 4.2.

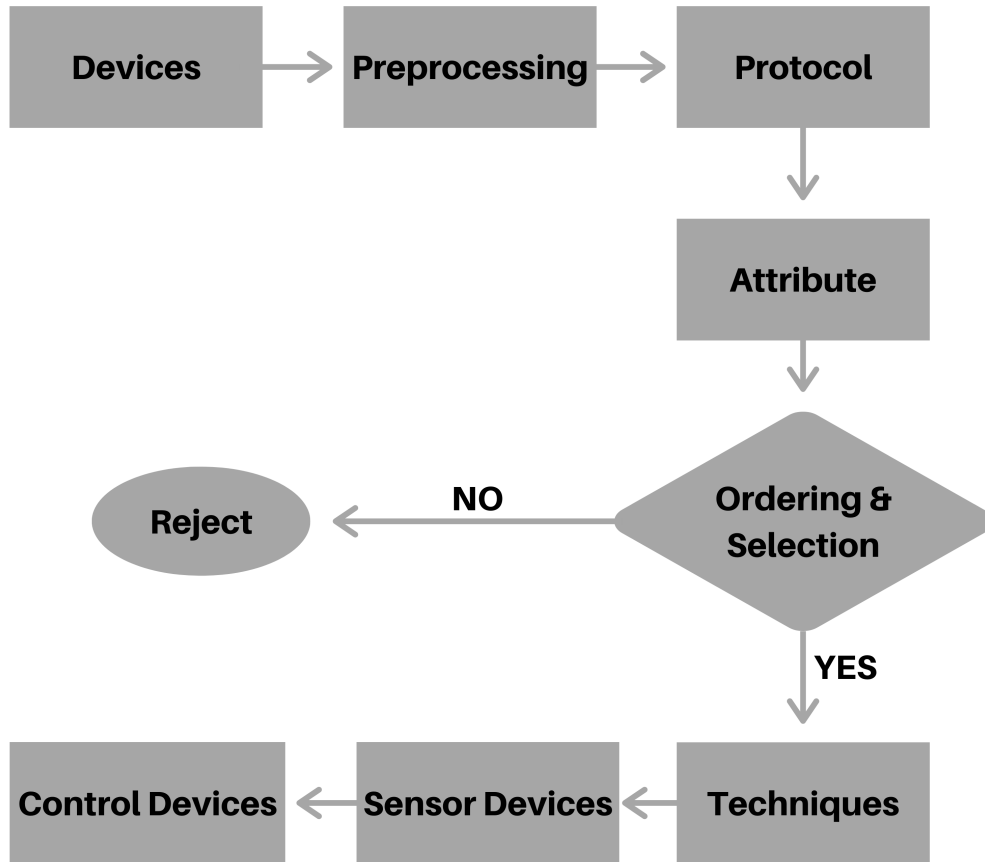
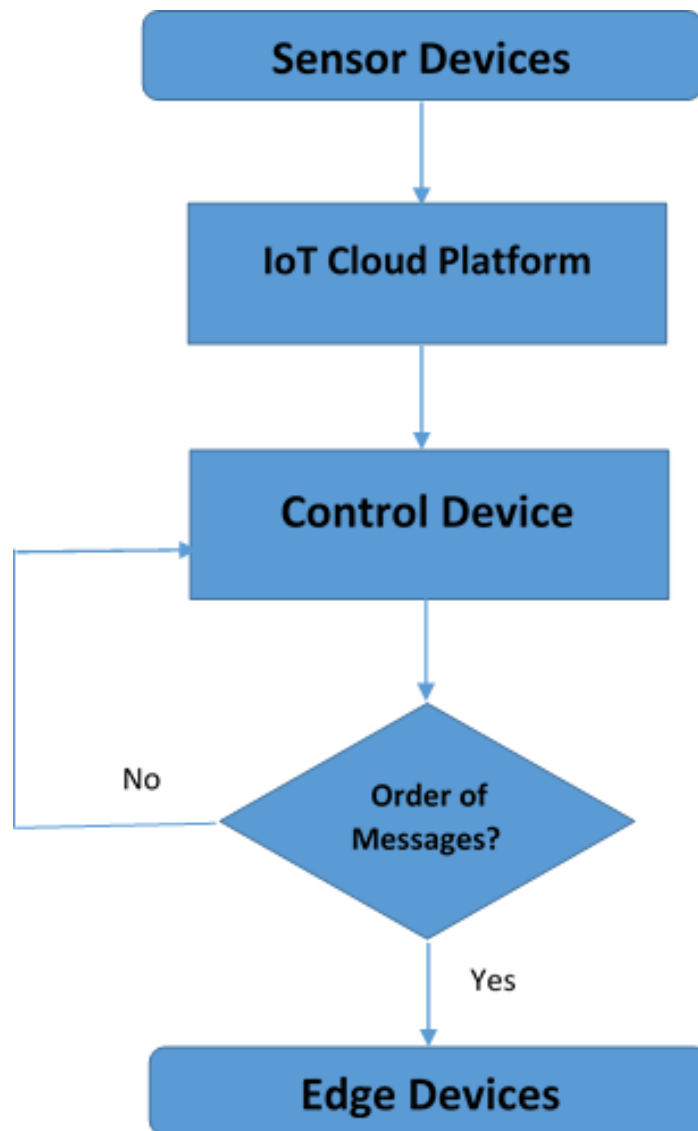


Figure 4.1: Overall System Structure

Let us consider room temperature as a monitoring factor to test the effectiveness of the proposed method. Sensor devices measure the room temperature and transfer it to the server. Further server sends back to the control devices. The control devices turn on/off those sensor devices using message-transferring system.

The Integrated ID technique helps to recover the lost message from the control device. The ID number of messages generate in the hexadecimal order and transmit along with the temperature data and device ID (attached to the last part of the data). In case of data loss, the control devices send the request signal to the specific devices to recover the data. This method is explained in following section.

**Sensor devices:** Collecting and transforming the signals such as light, vibration



**Figure 4.2:** Flow Diagram of Integrated ID Data Pre-Processing

etc. into electrical signal and then transferring them to micro-controller.

**IoT Platform:** Micro-controller receives the data from the sensor and process it then send to the IoT cloud platform which is MQTT broker(i.e Adafruit IO).

**Control Device:** MQTT broker sends the data to the control device(i.e. Raspberry Pi), control device then checks the ordering of the messages. if any loss of data then it will send the request to re-transmit the data otherwise send the command to the edge device.

### **4.3.1 Data Pre-Processing**

This chapter deals with an efficient approach based on the enhancement in the existing protocol and selection. In the first phase, data pre-processing is applied on the basis of sensor devices and monitoring allocations.

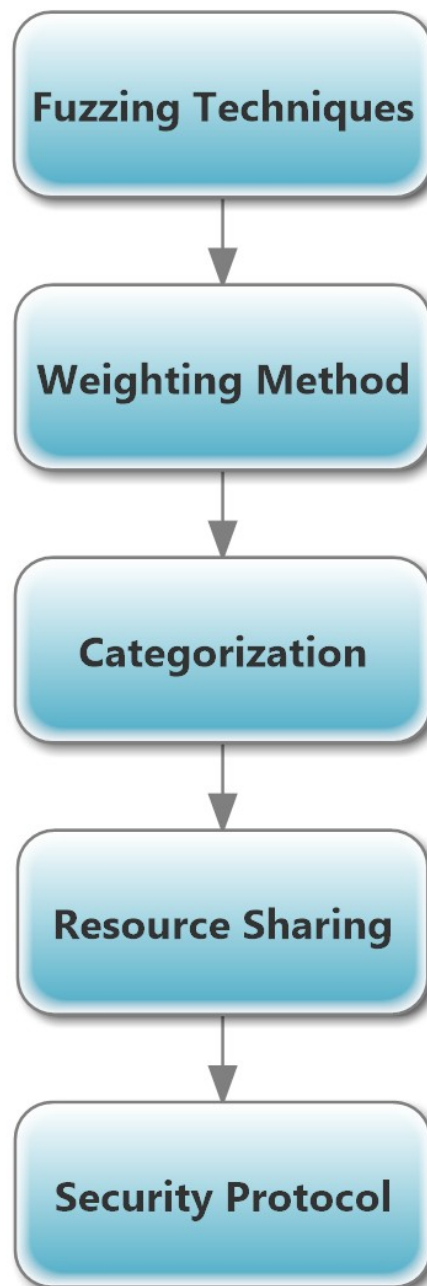
The message consists the temperature data of the subject and order of ID is added in the message in hexadecimal value. Along with the cumulative measure in the message ID is added in the prefix of the data. The device representation and temperature data allocation come in this phase. Moreover, weighted technique and method operation are also applied in block-based protocol, which is helpful in the recalculation of the field length and other control field automatically.

### **4.3.2 IoT Devices**

Two types of IoT devices are used here: such as controller and sensor. For message transmission through MQTT protocol. The Raspberry Pi 3 works as a controller and Arduino board as a sensor module.

The proposed method analyse in the transmission technique through MQTT protocol. Arduino collects the data of room temperature and Raspberry Pi 3 display the data and control the sensor devices (like turning on/off). Mostly smartphones are used as both control and sensor devices, and can be connected to more control devices for large applications. To maintain the reliable message transmission between sensor and control devices, the ratio of these devices is set with 1: N devices.





**Figure 4.3:** Data Pre-processing and Categorization

### **Arduino**

Arduino is an open source board consists of the physical programmable memory (microcontroller) and code can be uploaded to the Arduino by mean of software (IDE). Arduino does not need the separate hardware to upload the code. It follows the standard format to break the function of microcontroller that helps to access more

package. This is popular open source electronic board and suitable to evaluate the proposed method [86].

### **Raspberry pi 3**

Raspberry Pi 3 is the credit card size computer, which supports many functions from word or spreadsheet to games. This device supports all input devices (keyboard, mouse, etc.) and execute it on the Linux distribution. Raspberry Pi 4 model B is the latest version, which is low cost and uses to test the proposed method [87].

### **MQTT Broker**

International Business Machines (IBM) corporation developed MQTT protocol, which is both instant messaging and lightweight broker messaging protocol [88]. This technique is platform independent and supports most popular programming languages. This messaging protocol is suitable for the mobile pushing message due to its simplicity and scalability. This pushing message technique is used by many enterprises in an android phone and server-side also. Working of mentioned technique is shown in figure 4.4 MQTT [89]. MQTT protocol works on the application level, especially designed for resource-constrained devices [90, 91]. The publishing and subscribing technique are used in this protocol, as a broker device.

For instance, if the client sends the message denoted as M related to the topic T, then the message is sent to all client, subscribed to the topic T. Similar to the HTTP, MQTT process is based on the Transmission Control Protocol (TCP) with Internet Protocol (IP) as its underlying layers. The reliability of the protocols is ensured by following three Quality of Service (QoS) levels.

Level 0: Message delivered once without acknowledgment

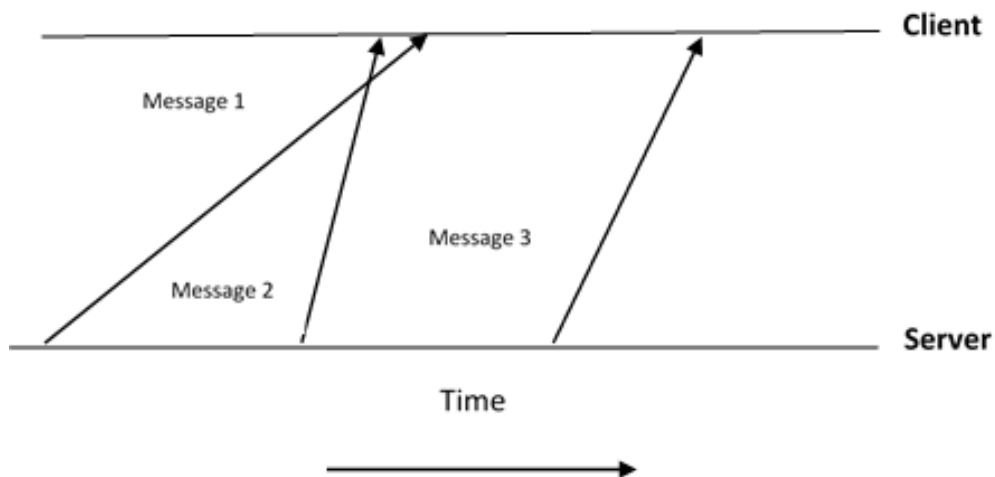


Figure 4.4: Overview of MQTT Method

Level 1: Receive acknowledgment.

Level 2: Four-way handshake mechanism executes to deliver the message.

MQTT expose as the REST resource and broker will define the state not MQTT. The last payload is identified as the resource value and after the analysis; the topic can be exposed by changing the single value. The topic is created based on the pure REST style with the help of HTTP POST and request for topic contains both topic name and payload. For instance, the topic's name is taken as temperature, and then the process is conducted as follows:

- There is a need of analysis of last published value. This analyses based on the HTTP GET request execution at /topics/temperature;
- It publishes a value. It should achieve by the HTTP PUT execution.
- The HTTP protocols allow it for the MQTT topic for the querying of state. It can help using HTTP caching protocol. The method provides only the best level of QoS. Hence, it is used HTTP protocol.

### 4.3.3 Integrated ID Method

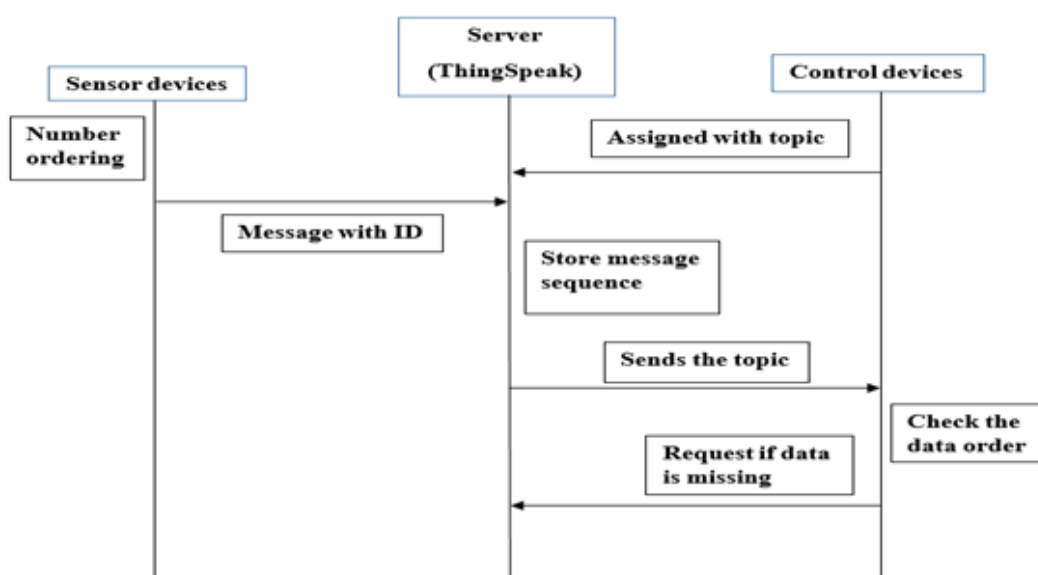
The sensor devices send the data to the system and cloud passes the received message to the control devices. The control devices turn on/off the sensor devices and this data is used in various fields like medical, home usage, etc. The sensor sends the message to the server and there might be a loss of data in the server [104]. Therefore, the message is added with hexadecimal numbers for analysis the order of the message and based on that missing data is requested.

ID denotes the order of the message and last field of the message from the sensor devices is added with 2-bit data to address the devices. There might be many devices connected to the control devices. The device number is used to identify missing data of device. The requested signal then sent from the cloud to that device to get the data. The ID number is generated for each message before publishing the data with the device number. The data field number is in the manner of hexadecimal numbers and the value is generated from 0000. Therefore, it has the capacity to transfer many data.

MQTT is the topic-based protocol, which uses the character strings to support the hierarchical topics. This technique can facilitate the multiple topics which is discussed with examples.

For instance, temperature monitoring in floor “F4” of the building in the Room “R2” and its data transferred in the hierarchical form “wsn/sensor/F4/R2/temperature”. The data separated using “/” and wsn denotes the Wireless Sensor networks. The data are replaced by using a wildcard as “wsn/sensor/F4/+temperature”, which can now access any sensor to monitor the different room temperature in the 4th floor.

This technique includes the data, such as “wsn/sensor/F2/R2/xxxxtemperatureyy”, as first four digits represent the message and last 2-bit denotes the device. If many devices



**Figure 4.5:** Message Transaction between Sensor and Control Device

connected to the control devices, then the message from the devices are identified using its device ID. It helps to recover the data by knowing the missing data from the device using message order.

The IoT sensor devices data is attached the hexadecimal value to the temperature data in prefix and 2-bit is added as the device ID in the last, which contains the order of messages. This helps to understand the message order and request made for the missing data. These message order, then attached to every message send to the server and message can be easily noted. The Raspberry Pi request the message with the device ID helps to understand the respective device. If the message is miss in the devices, then it requests to get the messages. The program is stored in the control devices and the server helps to analysis the data order. The server checks the number and it finds the missing data. Then it requests the sensor to send the message. This process is clearly explained by the Figure 4.5 and 4.6.

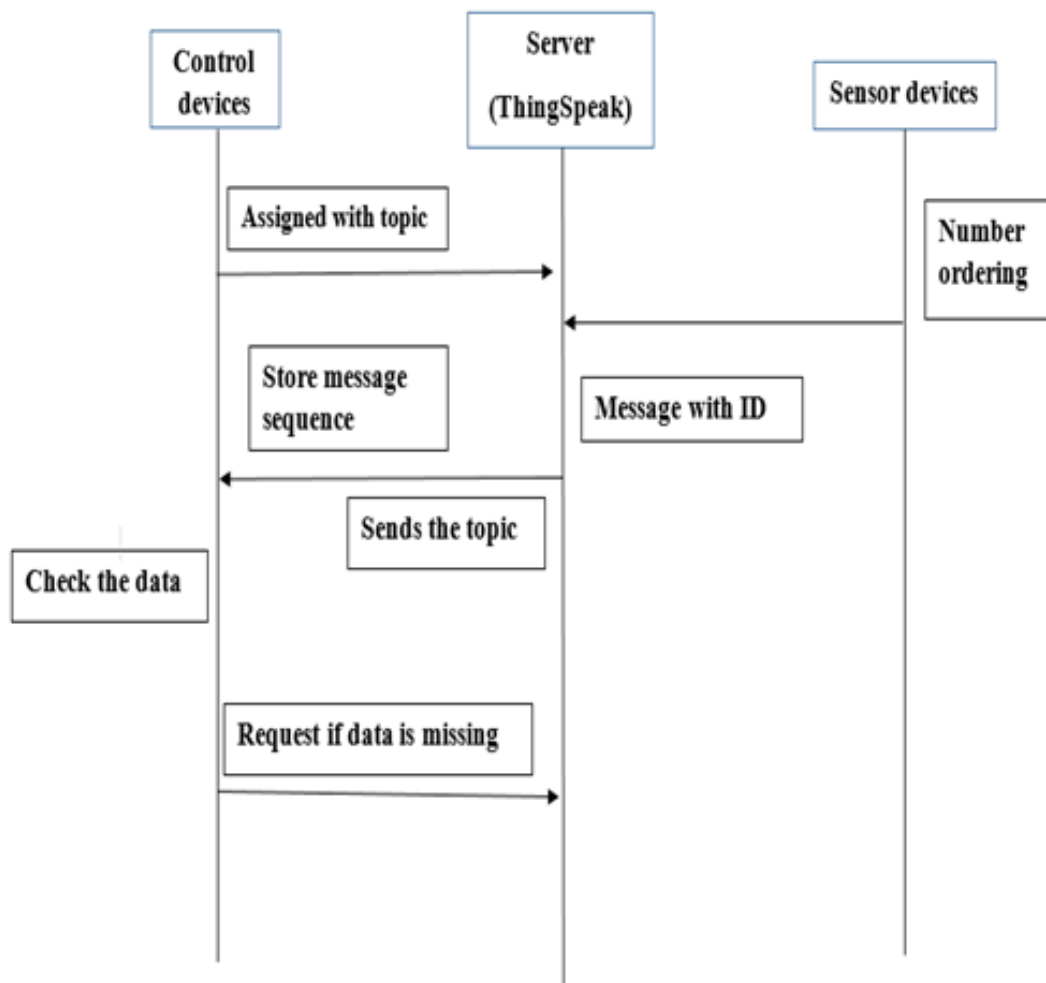


Figure 4.6: Communication between Devices

#### 4.3.4 Mathematical Derivation of Integrated ID Method

Once the temperature has been measured by the sensor in the room of R2 in the 4th floor of the building, the data has been combined with the ID in the control devices. This function is represented in the Eq. (4.1).

$$I_M.M.I_D \longrightarrow I_MMI_D \quad (4.1)$$

From the Eq. (4.1), M denotes the message and  $I_M$  denote the identity of the message,  $I_D$  denotes the identity of the devices that is used to check the order of the message.

For example, if the message is sent as 24°C and this is the first message send by the sensor then message is added with ID, as in the Eq. (4.2).

$$(0001).(24).(01) \longrightarrow 00012401 \quad (4.2)$$

The Eq. (4.2) can be generally denoted as in the Eq. (4.3). The first x denotes the digits of the message identity in the data and the y denotes the digits of the device identity.

$$xxxx.tt.yy \longrightarrow xxxxtt yy \quad (4.3)$$

The digits of x and y can be increased, depends on the monitoring time and number of devices in the room. If the temperature monitoring is plotted in the green field, there are more of sensor devices is need to plot in the environment and their ID digits can be increased. The *xxxx* is generated in the hexadecimal and *yy* is generated in the binary sequential order. In Hexadecimal we could use sixteen digits (0 to 9 and six more) since number of messages would be in large which the sensors generate and we can represents 16-bit in four hex digit And in hex easy to work rather than huge no of 0's and 1's Number of devices will be less as compare to messages so binary digits will be sufficient.

The control devices add the identity in the message, as this message can be sent to the server. The message is sent to the Adafruit IO and this will check the message transmission in the system. In the Adafruit IO server, the topic T has been checked and based on the topic the data is send to the edge devices. The topic T example is shown in the Eq. (4.4).

$$wsn/sensor/F4/R2/data \quad (4.4)$$

Where *wsn* denotes the network type, *sensor* is the temperature sensor, *F4* denotes

the 4th floor, R2 denotes the 2nd room in the building and data is given in the Eq. 4.3. The data is sent to the edge devices; the message order is checked in the edge devices. If there are any order missing in the message, then the request has been sent to the server to resend the message. This method helps to reduce the message loss in the system.

### 4.3.5 Scapy and Weighted Fuzzy System

**Fuzzy Method:** Fuzzy means things which are not very clear, fuzzy logic provides the best possible decision for given input. There are various components of fuzzy logic architecture.

**Rule Base:** This is a decision making system.

**Inference Engine:** It matches the current fuzzy input to each rule and based on that decides which rule is fired.

**Fuzzification:** It converts crisp input to fuzzy input set value.

**Defuzzification:** It converts fuzzy sets obtained by inference engine to crisp value.

**Fuzzy Process:** It is a way to discover bugs in software system by providing randomized inputs to the program to find test cases that cause a crash. It helps you to find critical bugs in the system. There are various fuzzy frameworks available (i.e. Radamsa, Sulley, Peach, SPIKE, Nodefuzz etc).

Fuzzers provide random input to the software, this may be in the form of network protocol, a file of certain format or direct user input.

Fuzzers are classified into two categories:

**Mutation-Based Fuzzer:** Apply mutations on existing data samples to create test.

**Generation-Based Fuzzer:** Create test cases from scratch by modelling target proto-



col or file format.

This technique allows automated generation of a template with the fields we want to test for each network packet. Fuzzers consider the fields and positions of the header for inserting data to perform fuzzing process.

**Fuzzing Processes:**

Step 1- Identify an objective

Step 2- Identify Entry Points

Step 3- Generating the Fuzzing Data

Step 4- Executing Test Cases

Step 5- Exception Monitoring

**Fuzzing MQTT Messages:**

- Select package(Connect, Publish etc.) from specification and fields that are of interest for inserting information.
- In MQTT most of the information is transmitted in Publish packet.
- Study variable header to select type of field and field's position in bytes, into which test cases are inserted.
- Look for control field which will be recalculated once test cases inserted (i.e add, delete etc so bytes will be change).

The proposed fuzzy system recalculates the control fields automatically using the block-based method. The framework of Boofuzz is the simple FTP and highly used now a days. This is successor to [92], which is highly influenced by SPIKE [93]. Sulley is actively developed fuzzing engines and fuzz testing framework. It has the collaboration of the multiple extensible components. It also exceeds the capabilities.

The capabilities are in the terms of published fuzzing technologies previously. It includes the public and commercial domain.

The framework aims to simplify the data representation, transmission and instrumentation. Sulley is named after the creature from Monster Inc., due to the creature is fuzzy. The advantages of boofuzzy is as follows: Easy and quick data generation, Instrumentation – in the manner of Target reset after failure, failure detection, and test data recording.

In this type of framework the complex protocols definition required is slow. As it requires the knowledge of tool itself. It also requires entire protocol specification. For this purpose, the weighted based approach would be useful. The instrument screens the correspondence dependent on the intermediary strategy. The arrangement of parameters done by the client whereby the parcels are sifted. The traffic is created between the customer and server, when client need to fluff. The predetermined parcels are expelled by the client and prepared. The. json weight worth is consequently created with the assistance of gave group.

MQTT publish layer of the packets. It is shown by the portion of the weighted value. Each field in the packet will appear and two factors are added namely recalculate and fuzzable. The user has to provide the fuzzy to a specific field of a package to change the fuzzable value. The true value need to assign by the user to recalculate the packet consistency. The verification value has been provided automatically by the tool. It measures all the fields. It is in the terms of the package. It recalculate the flag which set to true.

The computation complexity of the developed method is  $O(n)$  means that generated time is constant, independent of the generated weight value. The user no need to aware

of structure of any details in the tool or protocol, except the apply fuzzy and field that need to recalculate. In case, user want to make modification in the weight, do need for the special tool. That can be edited with the normal text editor if the structure of .json is maintained.

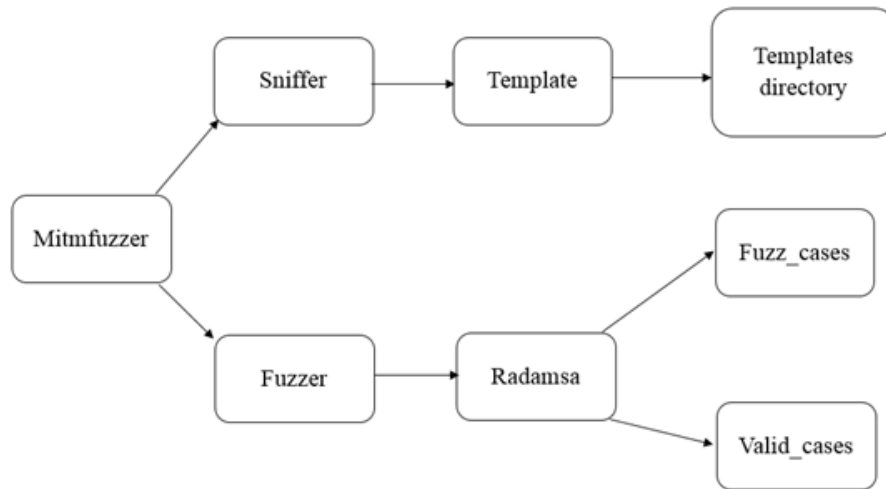
The development of the system is based on the same environment as in the research [92] and evaluated the proposed method. The architecture is shown in the Figure 4.7 and the weighted based fuzzy is used in the sniffer.

**Mitmfuzzer:** Mitmfuzzer is the driver that is used to call the remaining application function. The user enters the argument that is process by the python module argparse [94]. This provides the small interface to show the activity state of the tool.

**Sniffer:** Sniffer feature is the important function of the tool, which is present in the middle of the communication to monitor and select the data for filter and process. It also filter the data that are specified by the user for weight generation. The model is implemented using the Scapy [95]. Once it finds the package selected by the user, process it and sent it in a certain format that is written in python language. The method also provides the weight value for the module.

**Weight:** The package is received in module with certain format and process it generation of weight in .json format.

**Fuzzer:** It is a crucial and important module which process the monitoring, filter, addition and insertion of test case. The weight file is the input to the module that is generated in weight module. Using the iptables and nfquene [96, 97], monitoring the communication and transfer the packages that is not identified with the weight based on filtering. And matching packets are processed and its fields are compared with weight and check the one user specified. In case more than one field is identified, then



**Figure 4.7:** Architecture of Fuzzy Framework

module checks for the directory provided by the user to process. If the directory is not provided, then Radamsa has been called by the module [98], passing the parameter as file with the valid case like validcases/fieldname directory. It is a stock generator. It has been developed for software verification.

**Scapy:** Scapy is the library, which is used for the packet manipulation in the large number of network protocols. It plays an important role in the application core. The Scapy advantages is that it uses the block-based method to support the protocol. The fields in the package are modified, control field and length are recalculated automatically. The fuzzer has the packet, it sends to the Scapy to process. The structure of the packets is returned by the Scapy that is easy to change. The modification is made in the MQTT packets, Scapy takes the manipulated package that is inconsistency in the control field such as length field or checksum field. The control field is recalculated based on a block-based method and the data is encapsulated if it is original.

## Chapter : 5

### IMPLEMENTATION

#### 5.1 System Requirements

The tools used in this dissertation are Arduino IDE, Adafruit, and Raspberry Pi etc. The hardware and software requirement for the work implementation and experimentation are as follows:

**Hardware Requirement:**

1. Arduino Uno Board (Arduino Ethernet Shield, Arduino Uno USB Cable)
2. Raspberry Pi3 Model B
3. Micro SD Card -Minimum 16 GB
4. LAN Cables -Minimum four cables
5. RAM -Minimum 1GB suggested 2GB
6. Processor: -Intel or AMD x86-64 processor -Suggested with four logical cores
7. Disk Space: Including installation, it needs minimum of 1GB to 5GB for the workspace and the other applications.

**Software Requirement:**

1. Arduino IDE
2. Win 32 Disk Imager

3. Advanced IP Scanner
4. Putty
5. SD Card Formatter
6. IoT Platform(MQTT Broker): -Adafruit IO and ThingSpeak
7. Programming Language: -Python, C and C++
8. Operating System: Windows 7 or X and Raspbian (Raspberry Pi OS)
9. Graphics Requirements: No specific graphics card is required.

Main components that are used in implementation are Arduino Uno as sensor device, Raspberry Pi as control device and Adafruit IO works as a MQTT broker. which provide the complete integration to test the real time scenario.

**Arduino Uno:** Arduino is an open source board consists of the physical programmable memory (microcontroller) and code can be uploaded to the Arduino by mean of software (IDE). Arduino does not need the separate hardware to upload the code. It follows the standard format to break the function of microcontroller that helps to access more package. This is popular open source electronic board and suitable to evaluate the proposed method. Arduino boards are able to read inputs like- light on sensor, finger on button, twitter message etc. and turn into an output like—Activating motor, turning on LED, publishing online etc.

It has following advantages:

- Simple and inexpensive
- Supports cross platform
- Simple programming environment

**Raspberry Pi:** Raspberry Pi 3 is the credit card size computer, which supports many functions from word or spreadsheet to games. This device supports all input devices (keyboard, mouse, etc.) and execute it on the Linux distribution. Raspberry Pi 4 model B is the latest version, which is low cost and uses to test the proposed method.

It also has the following features:

- Enables people of all ages to explore computing and learn how to program in language like scratch and python.
- Capable of doing everything what computer can do like browsing internet, word processing, games etc.

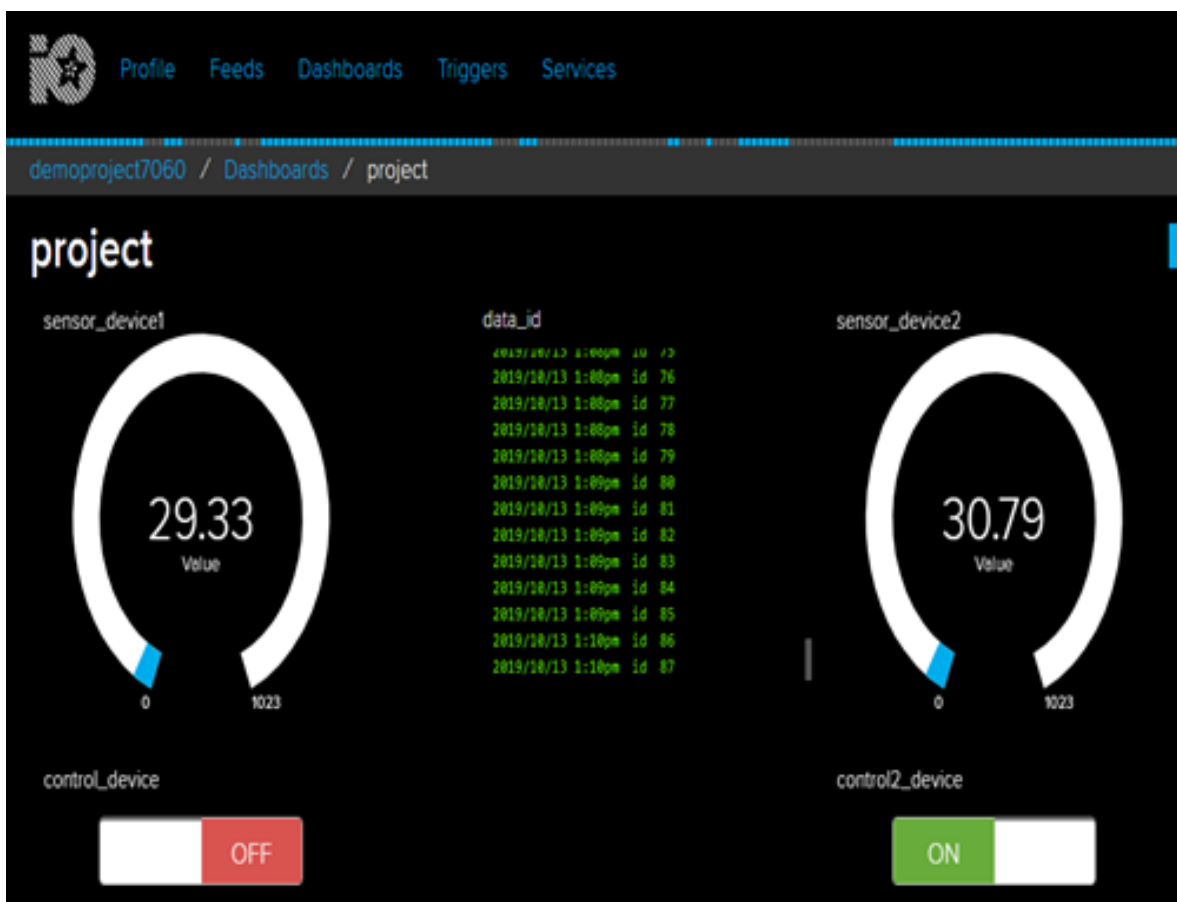
**Adafruit IO:** International Business Machines (IBM) corporation developed MQTT protocol, which is both instant messaging and lightweight broker messaging protocol. Adafruit IO is MQTT enabled cloud based IoT platform which display the data in real time. It make your project internet connected and can control motors, read sensor data etc.

It has following features:

- Connect project to web services like- twitter, RSS feeds, weather services etc.
- Connect project to IoT enabled devices and can use triggers.
- Free and Open Source

## 5.2 Output 1: Real Time Control of Devices with Message Ids

The below figure shows that the AC of sensor device1 room is off, because it has temperature below 30.00 degree Celsius and on the other hand and AC of room 2 is on as its sensor device2 shows temperature above 30 degree Celsius (as we have set in script of Raspberry pi). So, the below picture shows the real-time control of room1 and room2 AC with its data id (for detecting any kind of data loss occurs).



**Figure 5.1:** Adafruit cloud applets data along with control device buttons and data id field



### 5.3 Output 2: Generation of Message Id along with Device Id

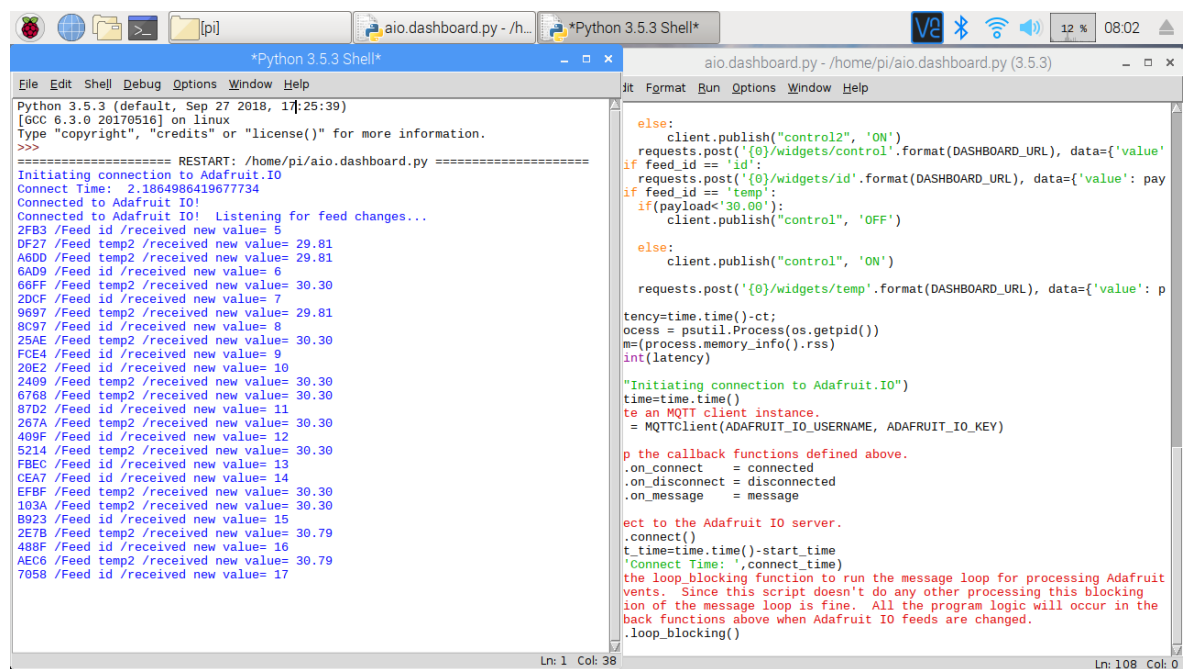
The below figure shows the detection of data loss in id field. It shows that the data loss recovery is possible from the sensor side. It shows the data id along with the sensor device1 and sensor device2.



Figure 5.2: Detection of Data Loss in Id Field

## 5.4 Output 3: Connection of Sensor devices to MQTT broker

The below figure shows the connection of sensor device (via ethernet shield) to Adafruit server. It shows the connection mechanism through the sensor device which is through the Ethernet shield. The connection is provided through Arduino IDE to the Adafruit server. It can allow the multiple input and output feeds.



```

Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/pi/aio.dashboard.py =====
Initiating connection to Adafruit.IO
Connect Time: 2.1864986419677734
Connected to Adafruit IO!
Connected to Adafruit IO! Listening for feed changes...
2FB3 /Feed id /received new value= 5
DF27 /Feed temp2 /received new value= 29.81
A6DD /Feed temp2 /received new value= 29.81
6AD9 /Feed id /received new value= 6
66FF /Feed temp2 /received new value= 30.30
2DCf /Feed id /received new value= 7
9697 /Feed temp2 /received new value= 29.81
8C97 /Feed id /received new value= 8
25AE /Feed temp2 /received new value= 30.30
FCE4 /Feed id /received new value= 9
20E2 /Feed id /received new value= 10
2409 /Feed temp2 /received new value= 30.30
6768 /Feed temp2 /received new value= 30.30
87D2 /Feed id /received new value= 11
267A /Feed temp2 /received new value= 30.30
409F /Feed id /received new value= 12
5214 /Feed temp2 /received new value= 30.30
FBEC /Feed id /received new value= 13
CEA7 /Feed id /received new value= 14
EFBf /Feed temp2 /received new value= 30.30
103A /Feed temp2 /received new value= 30.30
B923 /Feed id /received new value= 15
2E7B /Feed temp2 /received new value= 30.79
488F /Feed id /received new value= 16
AEC6 /Feed temp2 /received new value= 30.79
7058 /Feed id /received new value= 17

```

```

else:
    client.publish("control2", 'ON')
requests.post('{0}/widgets/control'.format(DASHBOARD_URL), data={'value': 'value'})
if feed_id == 'id':
    requests.post('{0}/widgets/id'.format(DASHBOARD_URL), data={'value': 'value'})
if feed_id == 'temp':
    if(payload<'30.00'):
        client.publish("control", 'OFF')
    else:
        client.publish("control", 'ON')
requests.post('{0}/widgets/temp'.format(DASHBOARD_URL), data={'value': 'value'})

tency=time.time()-ct;
ocess = psutil.Process(os.getpid())
m=(process.memory_info().rss)
int(latency)

"Initiating connection to Adafruit.IO"
time=time.time()
te an MQTT client instance.
= MQTTclient(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)

p the callback functions defined above.
.on_connect = connected
.on_disconnect = disconnected
.on_message = message

ect to the Adafruit IO server.
.connect()
t_time=time.time()-start_time
'Connect Time:',connect_time)
the loop_blocking function to run the message loop for processing Adafruit
vents. Since this script doesn't do any other processing this blocking
ion of the message loop is fine. All the program logic will occur in the
back functions above when Adafruit IO feeds are changed.
.loop_blocking()

```

Figure 5.3: Connection of Sensor Device to MQTT Broker(Adafruit IO Server)

## 5.5 Output 4: Communication Between Sensor device and Control device

The below figure shows the data communication of sensor device1 along with data on control device side. here all the sensors are connected to the Arduino Uno Board and the control device raspberry pi, which control all the sensors through commands

```

// float t = DHT.read
x= analogRead(sens
temp_val = (x*(5.
temp_val = (temp_val
Serial.println(temp
long now = millis
if (now - lastMsg
lastMsg = now;
++value;
sprintf(msg, "%d
if (!sensor_device
Serial.println(F
) else {
Serial.println(F
data_id.publish
)
}

//if(Serial.available
//{
// char q=Serial.read();
// if(q=='1')
// {
// Serial.println("publishing data to leds/pi!");
// sensor_device.publish("ON");
// }
// else if(q=='0')
// {
// Serial.println("publishing data to leds/pi!");
// sensor_device.publish("OFF");
// }
}

825F /Feed id /received new value= 5
F7AF /Feed temp2 /received new value= 29.81
653C /Feed id /received new value= 6
4E74 /Feed temp2 /received new value= 28.84
B712 /Feed temp2 /received new value= 30.79
4F0B /Feed id /received new value= 7
9E49 /Feed id /received new value= 1
9FD4 /Feed temp /received new value= 29.33
A387 /Feed id /received new value= 2
5831 /Feed temp /received new value= 29.81
B30E /Feed temp /received new value= 43.50
00AE /Feed id /received new value= 3
E71A /Feed temp /received new value= 56.70
938F /Feed id /received new value= 4
7687 /Feed temp /received new value= 32.26

```

Figure 5.4: Data of Sensor Device1 along with Data on Control Device

## 5.6 Output 5: Connection Process between sensor devices and control devices

The below figure shows the data of sensor device2 with its connection process. It shows the integration between sensor and control devices. sensor device communication with raspberry pi through the Arduino IDE. And raspberry pi sending the command to the sensor device via adafruit server

The figure displays three windows illustrating the connection process:

- Arduino IDE:** Shows the code for the sensor device (sensor\_device2\_mqt\_with\_pi3). The code includes headers for SPI, Ethernet, Adafruit MQTT, and DHT. It defines a DHT11 sensor on pin 2 and a WiFi Access Point with MAC address 0x0E, 0xAD, 0x5E, 0xEF, 0xFE, 0xED. The IP address is 192.168.43.92. The COM port is COM3.
- Python 3.5.3 Shell:** Shows the execution of the Python script. It displays the connection process to the Adafruit IO server, including the connection time (1.3038127422332764) and the successful connection to the Adafruit IO server. It also shows the received data for the temp2 feed: 28.35, 1, 2, 29.33, 29.81, 3, and 29.81.
- Python Script (aio.dashboard.py):** Shows the code for the control device. It subscribes to the three pi-dashboard feeds (temp, id, temp2). It defines a disconnected function and a message function. The message function updates the physical dashboard depending on the channel (temp2, id, temp) and publishes the control signal (OFF or ON) to the control2 feed. It also sends a request to the dashboard to update the widget.

Figure 5.5: Data of Sensor Device2 with Connection Process

## 5.7 Output 6: Published data of sensor device with control device

The below figure shows the sensor device data with its control command and also shows same data on control device side with its id and feed name. It shows the control command with the data with the control device.

The screenshot displays an IDE with the following components:

- Code Editor:** Shows Arduino code for an MQTT client. Key lines include:
 

```

Adafruit_MQTT_Publish sensor_device = Adafruit_MQTT_Publish(mqtt, MQTT_USERNAME "/t/temp");
Adafruit_MQTT_Publish data_id = Adafruit_MQTT_Publish(mqtt, MQTT_USERNAME "/t/id");
Adafruit_MQTT_Subscribe control_device = Adafruit_MQTT_Subscribe(mqtt, MQTT_USERNAME "/t/control");

```
- Terminal (Python 3.5.3 Shell):** Shows the MQTT connection process and a list of received messages:
 

```

Connect Time: 2.1864986419677734
Connected to Adafruit IO!
Connected to Adafruit IO! Listening for feed changes...
2FB3 /Feed id /received new value= 5
DF27 /Feed temp2 /received new value= 29.81
A6D0 /Feed temp2 /received new value= 29.81
6AD9 /Feed id /received new value= 6
66FF /Feed temp2 /received new value= 30.30
2DCF /Feed id /received new value= 7
9697 /Feed temp2 /received new value= 29.81

```
- COMS Window:** Shows a serial monitor with the following output:
 

```

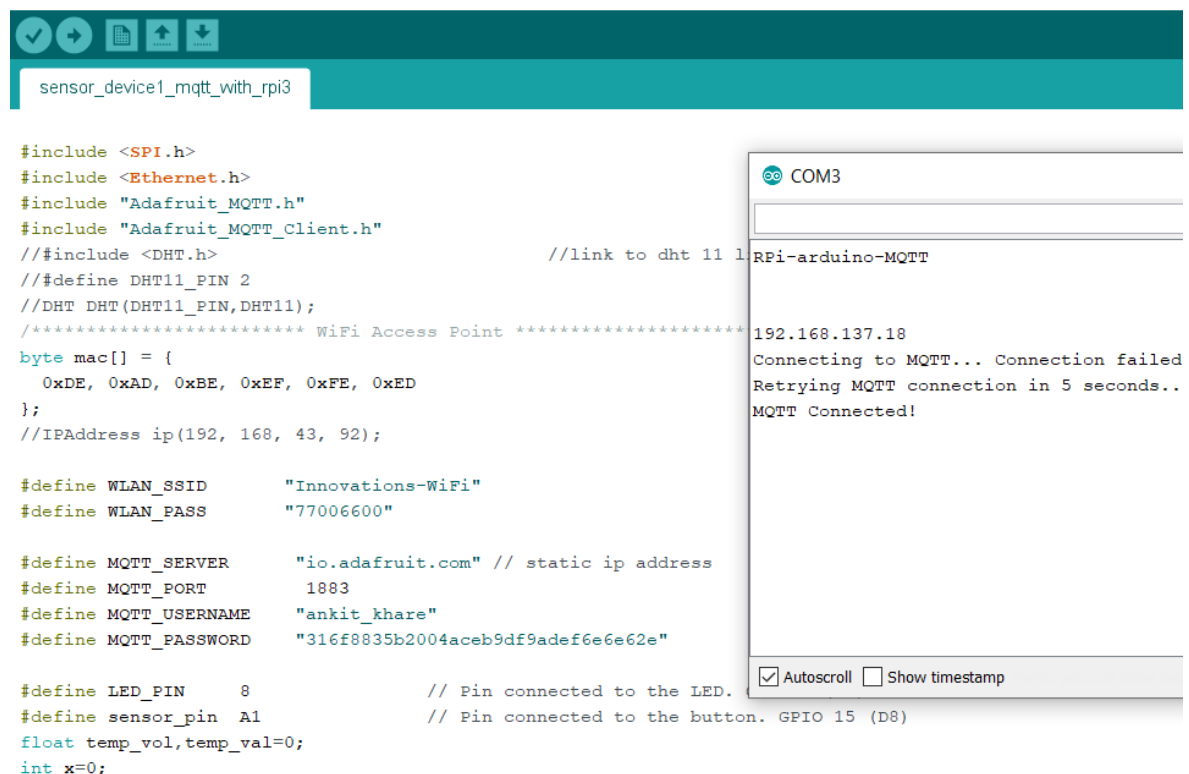
30.79 OK! data published
Got: OK
30.79 OK! data published
Got: OK
30.79 OK! data published
Got: OK
30.79 OK! data published
Got: OK
30.79 OK! data published
Got: OK
30.79 OK! data published
Got: OK

```

**Figure 5.6:** Sensor Device Data with Control Command and Control Device data with Feed id

## 5.8 Output 7: Connection Establishment between All Components on MQTT

Below figure shows the connection establishment of Adafruit IO server with Raspberry Pi and Arduino, based on MQTT protocol.



```

sensor_device1_mqtt_with_rpi3

#include <SPI.h>
#include <Ethernet.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
//#include <DHT.h> //link to dht 11 1
//#define DHT11_PIN 2
//DHT DHT (DHT11_PIN,DHT11);
/***** WiFi Access Point *****/
byte mac[] = {
  0x0E, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
//IPAddress ip(192, 168, 43, 92);

#define WLAN_SSID      "Innovations-WiFi"
#define WLAN_PASS      "77006600"

#define MQTT_SERVER    "io.adafruit.com" // static ip address
#define MQTT_PORT      1883
#define MQTT_USERNAME   "ankit_khare"
#define MQTT_PASSWORD   "316f8835b2004aceb9df9adf6e6e62e"

#define LED_PIN        8 // Pin connected to the LED.
#define sensor_pin     A1 // Pin connected to the button. GPIO 15 (D8)
float temp_vol,temp_val=0;
int x=0;

```

```

COM3
Rpi-arduino-MQTT
192.168.137.18
Connecting to MQTT... Connection failed
Retrying MQTT connection in 5 seconds..
MQTT Connected!
 Autoscroll  Show timestamp

```

**Figure 5.7:** Connection Establishment on MQTT Protocol

## 5.9 Output 8: Connection Time and Latency between MQTT Packets

Below figure shows the connection time and latency when first MQTT packet is processed. It shows that how much time it takes to process the first packet from sensor device to control device

The figure shows two windows. The left window is a Python 3.5.3 Shell with the following output:

```
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/pi/Desktop/project/aio.dashboard.py =====
Initiating connection to Adafruit.IO
Connect Time: 1.1651172637939453
Connected to Adafruit IO!
Connected to Adafruit IO! Listening for feed changes...
E65D /received Temperature(Feed temp) value= 500.00/ 01
latency= 0.0502774715423584
4B66 /received Temperature(Feed id) value= 357/ 01
latency= 0.08586001396179199
D721 /received Temperature(Feed temp) value= 416.42/ 01
latency= 0.0522153377532959
294D /received Temperature(Feed id) value= 358/ 01
latency= 0.04889369010925293
```

The right window shows the Python script `aio.dashboard.py` with the following code:

```
s.exit(1)

ssage(client, feed_id, payload):
    Message function will be called when a subscribed feed l
    art_time=time.time()
    ientID=""
    vice_id="01"
    r x in range(1,5):
        clientID=random.choice(string.alphanumeric)

    int(clientID, '/received Temperature(Feed {0}) value= {1

    =time.time()
    tency=ct-start_time;
    rocess = psutil.Process(os.getpid())
    em=(process.memory_info().rss)
    int("latency=", latency)

    feed_id == 'temp2':
        if(payload<'28.00'):
            client.publish("control2", 'OFF')

        else:
            client.publish("control2", 'ON')
            requests.post('{0}/widgets/control'.format(DASHBOARD_I
            requests.post('{0}/widgets/id'.format(DASHBOARD_URL),
            if feed_id == 'temp':
```

Figure 5.8: Connection Time and Latency

## 5.10 Output 9: Integrated ID Generation

Below figure shows the working process of Integrated ID Method. four digits hexadecimal numbers are used here for message id as a prefix and two digits binary numbers as a suffix are used to identify device id here. sensors are sending the temperature data here along with their message and device id as prefix and suffix with actual value.

```

*Python 3.5.3 Shell*
File Edit Shell Debug Options Window Help
330D /received Temperature(Feed temp) value= 140.27/ 01
FRE6 /received Temperature(Feed temp) value= 161.29/ 01
36D3 /received Temperature(Feed id) value= 143/ 01
0524 /received Temperature(Feed temp) value= 141.74/ 01
654D /received Temperature(Feed id) value= 144/ 01
696F /received Temperature(Feed temp) value= 150.05/ 01
5C50 /received Temperature(Feed id) value= 145/ 01
940B /received Temperature(Feed id) value= 146/ 01
5054 /received Temperature(Feed temp) value= 159.82/ 01
CB65 /received Temperature(Feed temp) value= 138.81/ 01
9B90 /received Temperature(Feed id) value= 147/ 01
CD28 /received Temperature(Feed id) value= 148/ 01
2FF0 /received Temperature(Feed temp) value= 160.80/ 01
2A9B /received Temperature(Feed temp) value= 139.78/ 01
5861 /received Temperature(Feed id) value= 149/ 01
BA76 /received Temperature(Feed temp) value= 156.40/ 01
5504 /received Temperature(Feed id) value= 150/ 01
DB79 /received Temperature(Feed id) value= 151/ 01
A398 /received Temperature(Feed temp) value= 170.58/ 01
6E18 /received Temperature(Feed id) value= 152/ 01
0898 /received Temperature(Feed temp) value= 147.61/ 01
F423 /received Temperature(Feed id) value= 153/ 01
8C15 /received Temperature(Feed temp) value= 130.01/ 01
5D0C /received Temperature(Feed id) value= 154/ 01
C739 /received Temperature(Feed temp) value= 155.43/ 01
30E8 /received Temperature(Feed id) value= 155/ 01
ED71 /received Temperature(Feed temp) value= 133.43/ 01
F671 /received Temperature(Feed id) value= 156/ 01
58F9 /received Temperature(Feed temp) value= 142.72/ 01
5FF5 /received Temperature(Feed temp) value= 152.98/ 01

aio_dashboard.py - /home/pi/Desktop/project/aio_dashboard.p
File Edit Format Run Options Window Help
# Message function will be called when a subscribed feed has
start_time=time.time()
clientID=""
device_id="01"
for x in range(1,5):
    clientID+=random.choice(string.alphanum)

print(clientID,'/received Temperature(Feed {0}) value= {1}').

ct=time.time()

if feed_id == 'temp2':
    if(payload<'28.00'):
        client.publish("control2", 'OFF')
    else:
        client.publish("control2", 'ON')
        requests.post('{0}/widgets/control'.format(DASHBOARD_URL)
elif feed_id == 'id':
    requests.post('{0}/widgets/id'.format(DASHBOARD_URL), dat
elif feed_id == 'temp':
    if(payload<'30.00'):
        client.publish("control", 'OFF')
    else:
        client.publish("control", 'ON')

requests.post('{0}/widgets/temp'.format(DASHBOARD_URL), d

```

Figure 5.9: Message ID Generation in Integrated ID Method



## 5.11 Output 10: Working Application with MQTT Broker

Below figure shows the integration of sensor device and raspberry pi with adafruit MQTT broker. data is published on MQTT broker as well as control device which helps to identify the ordering and loss of any data. This is a complete application of Integrated ID method which is used to mitigate the message loss.

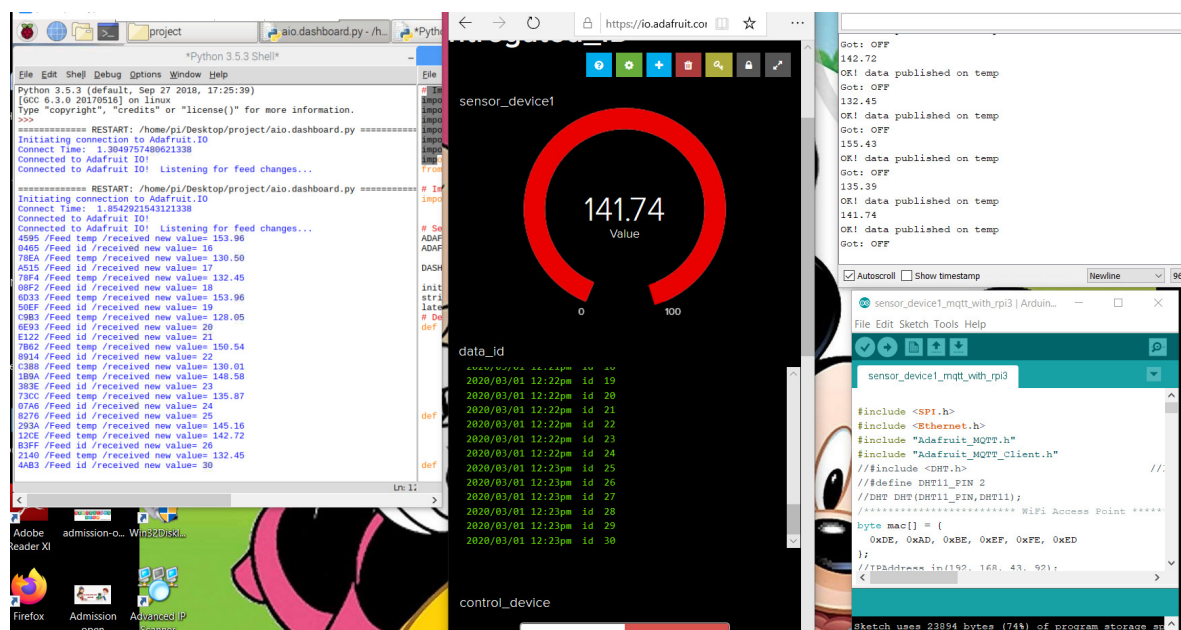


Figure 5.10: Working Model of Proposed Method

## Chapter : 6

### RESULT ANALYSIS AND DISCUSSION

The experimental setup consists of the different stages: sensor nodes, open-source mosquito broker/server, and Adafruit IO. The temperature data is measured from the different rooms in the building as the temperature data are sensed with the help of sensors. These data are sent through the MQTT gateway and stored in the Adafruit IO server.

The proposed method is also compared with the existing method in the same environment. Table 6.1 shows the limitations of existing methods. The sources considered here are from [90, 99-103]. The requirements for the experimental setup are provided here in the following Table 6.2.

**Table 6.1: Limitations of Existing Methods**

Method	Limitation
Vehicle Tracking System [99]	Traceability is low
Crypto-Hash-Modules [100]	Large overhead
Lightweight Method [101]	Low efficiency Low Tracebility
MinT-I [102]	Higher energy Consumption
Openness and Expandability [103]	Bottleneck problem
Load Balancing method [90]	Low efficiency Low tracebility

#### 6.1 Results Based on Security and Message Payload

The IoT devices are growing and used in a different field for continuous monitoring and controlling. The major issue in the IoT is security and these devices are deployed

**Table 6.2: Experimental Design**

Apparatus	Quantity	Provider	RAM	HDD	Processor	Operating system
Sensor Device	Temperature sensor(5 pcs)					
MQTT broker	MQTT broker: Computation instance	Raspberry Pi 3	1 GB	32 GB	4× ARM Cortex-A53 at 1.2 GHz	Noobs
Receiver node	Receiver node 1	Adafruit IO				

in different places, which makes it difficult to protect. These devices are low constraint devices and requires lightweight security protocol with efficiency.

The objective of this research is to improve the security of IoT devices without much affecting the efficiency. MQTT is a simple and lightweight protocol, which is capable to support thousands of clients. Due to the efficiency of MQTT, it is highly used in the IoT application. The fuzzy technique is used to verify the security in IoT devices. In this research, the weighted based fuzzy method is proposed to reduce the load of the user and increase efficiency. This method is completely automated to weight the field to test each network packets. Due to the fuzzification process message can be extracted at any platform employing defuzz [92]. The data can be well secured by fuzzified packets during transmission and transferred through a well-known MQTT broker in the current architecture to reduce data loss. The experiment is carried out in the given scenario and the result is measured. Results are compared and investigated to check the efficiency of the proposed approach.

**Table 6.3: Publisher to Subscriber Delay with Message Payload**

Message payload set by Paho API (in byte)	Publisher-to-Subscriber delay at QoS level-0 [16] (sec.)	Publisher-to-Subscriber delay at QoS level-1 [16] (sec.)	Publisher-to-Subscriber delay at QoS level-2 [16] (sec.)	Fuzzy Method (sec.)
1000	183	227	310	174
2000	293	421	526	286
3000	385	494	580	318
4000	498	580	697	447

The proposed method is compared with the conventional method [90] and experimented in the same scenario. The publisher to subscriber delay is measured for the different message payload and the result is compared with the conventional method, as shown in Table 6.3. The weighted fuzzy method has a low delay as compared to the existing method at different payload. For 4000 bytes of payload, the weighted fuzzy method sends the message in 447 s and QoS level-2 method is to send in 697 sec.

The source to destination loss of message with different payload has been measured for the existing and proposed method, as shown in Table 6.4. There are three levels of QoS in the existing method [90] and these are compared with the proposed one.

The message loss of the proposed fuzzy method is equal to the QoS level-2 at 1000 bytes of payload.

The proposed method has a low message loss as compared to the three-level of conventional method. For 3000 bytes of the message payload, the proposed method has a message loss of 0.223 % as compared to the conventional method of QoS-level-2 0.24 %.

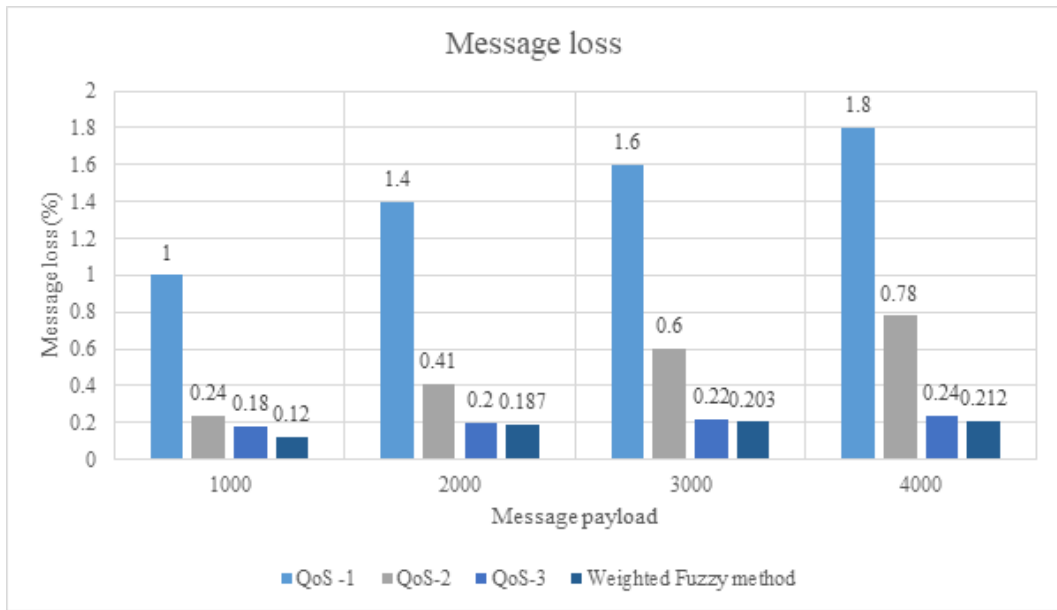
**Table 6.4: Source to Destination Message Loss with Payload**

Message payload set by Paho API (in bytes)	messages loss at QoS level-0 (In %)	messages loss at QoS level-1 (In %)	messages loss at QoS level-2 (In %)	Fuzzy Method (In %)
1000	1	0.24	0.18	0.18
2000	1.4	0.41	0.2	0.195
3000	1.6	0.6	0.22	0.207
4000	1.8	0.78	0.24	0.223

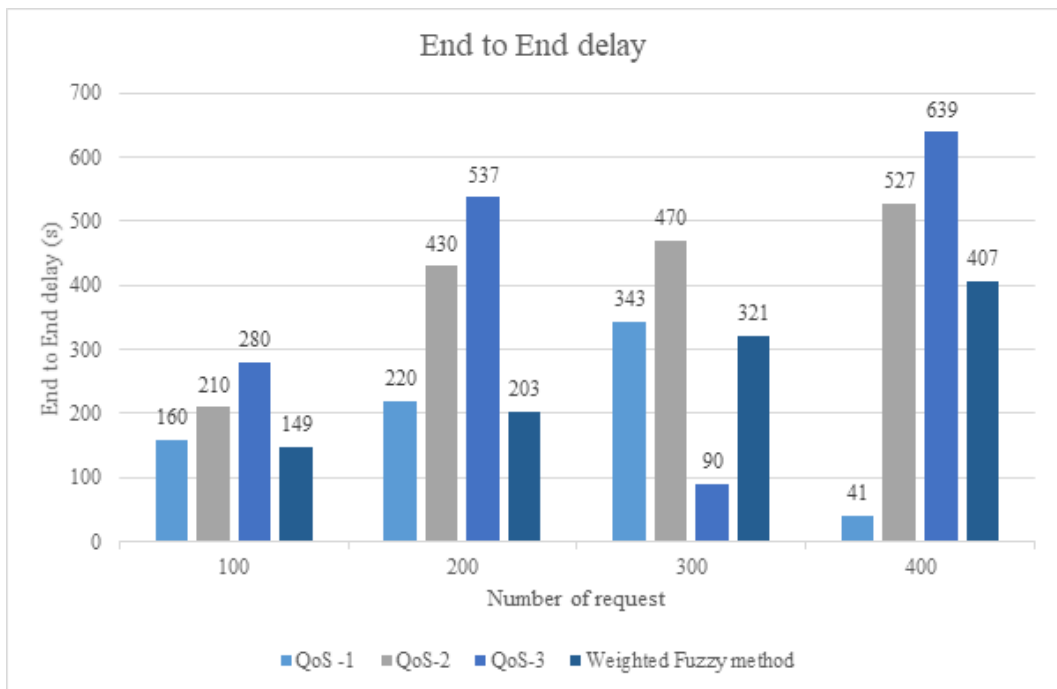
The message loss of the proposed and conventional method are calculated for different payloads as shown in Figure 6.1. This shows that the proposed method has a low message loss compared to the existing method at a different level of the message payload. The message loss has been much reduced by the proposed method compared to the conventional method.

The end-to-end delay for three QoS levels [90] and fuzzy system at different numbers of requests are measured and plotted as a graph in Figure 6.2. The different number of requests are made in the experiment and calculated the end-to-end delay. It shows the proposed method has low delay compared to the existing method at a different number of requests. At the 300 requests, the QoS level-2 has low delay as compared to the proposed method. The delay of the proposed method is low for the different level of request. For the 400 requests, the proposed method has 407 sec delay compared to the QoS level-2 at 639 sec delay.

The end-to-end delay is measured for the proposed method at a different level of the message payload and compared with the conventional method. The end-to-end delay for the different payload is plotted as graph in the Figure 6.3. The different level of payload is sent through the system and measured the end-to-end delay. The



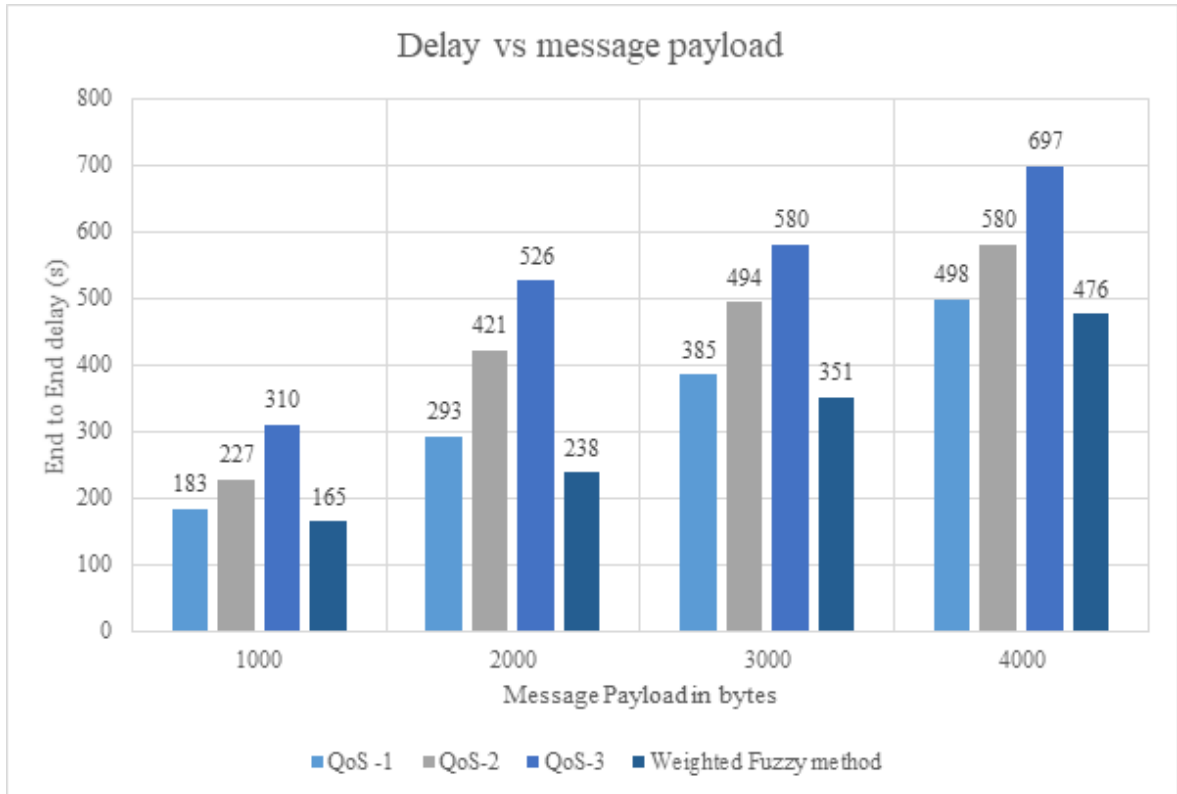
**Figure 6.1:** Message Loss of the Proposed Method



**Figure 6.2:** End to End Delay Vs Number of Request

fuzzy system has less delay compared to the existing method at different payload. For the 4000 bytes of payload, the proposed method has the end-to-end delay of 476 s compared to the QoS level-3 has the delay of 697 sec.

It shows that the efficiency of proposed method is high compared to the conventional method due to the fuzzy method.



**Figure 6.3:** Delay Vs Message Payload

## 6.2 Comparative Analysis based on the Latency and Connection time

Most of the home appliances are connected through the Internet and controlled remotely using edge devices. The number of devices connected to the internet is growing high and it needs a reliable messaging technique. The method aims to reduce the loss of messages by analyzing the order of messages. The proposed method is evaluated using the Raspberry pi 3 and the message is sent through the devices. The unstable conditions created in the clumsy network simulator [104] and the message are analyzed on both sides of devices. Table 6.5 shows the devices and its supported applications.

**Table 6.5: Environment Setup**

Item	Server	Sensor Devices	Control Devices
Operating System (OS)		Arduino IDE on Microsoft Windows	Raspbian on Raspberry Pi 3
Relational Database Management System (RDBMS)	Adafruit IO		
Message broker	Adafruit IO	Adafruit IO	Adafruit IO

The latency of the proposed method is calculated for different messages that is presented in Figure 6.4. Latency is lower for the first message and it varies with different messages. This latency is considerable for the IoT application and it measures in milliseconds.

In the research [105], OAuthing securely transmitted the message and this involved in protecting the data. These methods don't share the data with the third-party application and additionally provide privacy to the user. This technique is compared with OAuthing research [105], in terms of time, latency, and program memory. The research [105] involved in user registration and then tried to connect the client and the user. The message was assigned with the ID and continuously monitored. The message loss minimized with the help of the token and the data stored in the separate cloud.

The connection time is based on the device initialization, and the evaluated connection time of different method is compared with other techniques (like OAuthing) in research [105]. Mosquitto technique has a connection time of 24.5 ms and the integrated ID has a connection time of 22.95 sec. This time is less compared to conventional methods.



The existing method of OAuthing [105] involved in initializing the third-party app and other gateways, caused a rise in the connection time. The proposed method of integrated ID will not allow the third party to involve, so processing time decreased. The IDs were stored in the variable memory of the device instead of cloud storage, which results in decreasing the latency of the proposed method. Figure 6.5 shows the connection time of various techniques.

The first connection time is much higher for the OAuthing method due to its privacy protection algorithm. The further connection of OAuthing having a less computational time of 35.9 ms.

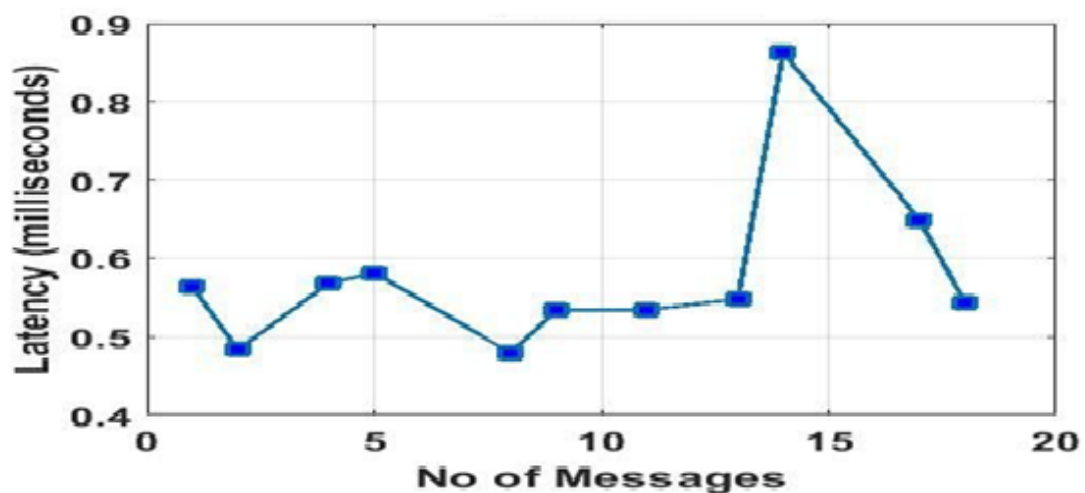
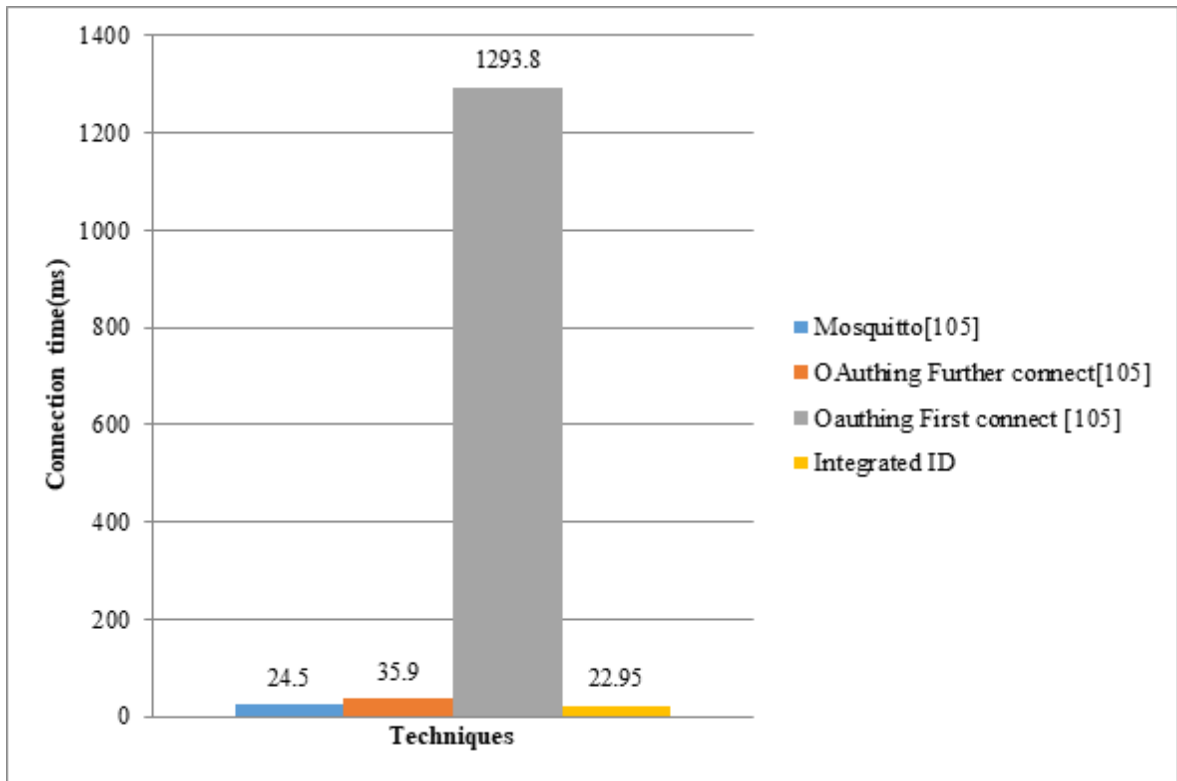


Figure 6.4: Latency for Number of Messages

The program memory gives the space required by the method to store and less program memory helps to provide more space for variable memory. The program memory of the proposed and conventional method is shown in Figure 6.6. The Integrated ID method requires lower memory than the existing method OAuthing [105].

The existing method store the data in the cloud and the integration of the cloud increases the program memory. The Integrated ID technique uses the numerical data

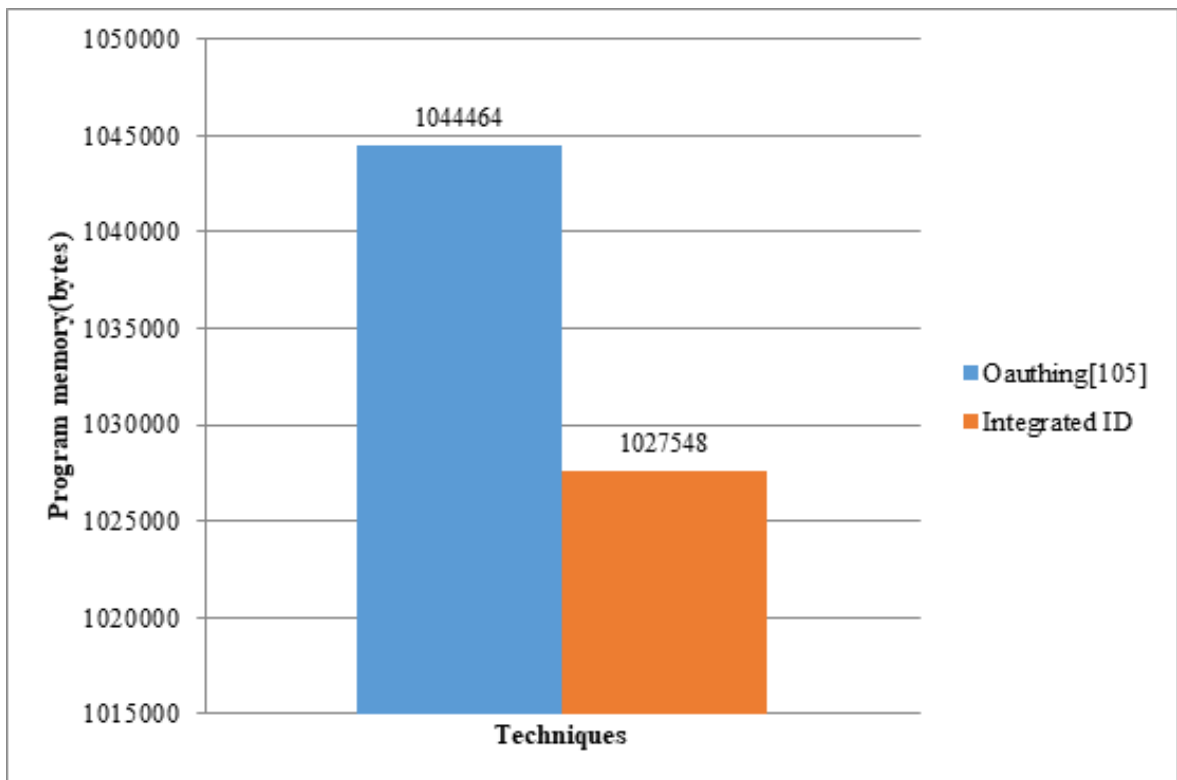


**Figure 6.5:** Connection Time in Different Methods

for identifying message loss and this requires very less space that can be stored in the variable memory of the devices. The proposed method stores and processes the numerical data (device ID) for identification which requires less time to process than other existing methodologies with other information (user name, token, etc.).

IoT messaging technique is used to transfer messages across several devices and some devices may have lower resources. These kinds of resources need the lightweight program for its function. This Integrated ID method shows that the proposed method has a low program memory compared to the other method [105].

The time requires for the first message is calculated and compared with the standard method. The comparison of time taken to process the first message between existing and proposed is shown in Table 6.6. The proposed method requires less time compared to other methods and the proposed method requires only 0.86 sec for transmission of



**Figure 6.6:** Program Memory Size for Proposed Method

first messages. The existing methods involve registering the data before transferring the message. The device ID in the Integrated ID identifies the user, as it helps to improve the performance of the message process in less time. This helps to reduce the time taken for the first message to process and achieve low time for the first message compares to conventional methods.

**Table 6.6:** Time Taken by First Message to Process

Technique	Time (sec)
MQTT [105]	12.03
OAuthing [105]	12.71
Integrated ID	0.86

The wrong message rate is measured for the Integrated ID and the common MQTT technique in Table 6.7. This shows that the proposed method has less than 1 % wrong

**Table 6.7: Wrong Message Rate**

Methods	Wrong message rate (In %)
MQTT	4
Integrated ID	less than 1

message rate and it also has high efficiency. However, the proposed integrated ID has been tested in the unstable network and this shows that the wrong message rate in 9This shows that the proposed method has higher performance compared to the conventional methods. This method can be applied to the IoT technique for efficient transfer of messages, helps in monitoring the environment.

## **Chapter : 7**

### **CONCLUSION AND FUTURE SCOPE**

#### **7.1 Conclusion**

In the Internet of Things (IoT) connected devices are rapidly growing and control devices are connected to the sensor to monitor certain conditions. In the Internet of Things, devices are resource-constrained and the lightweight method is required to enhance the security and mitigate loss of the message.

In this thesis, our main objective is to mitigate the message loss during the message transmission in-between IoT devices. Additionally, monitoring the network without affecting integrity of messages and quality of service (QoS) in the system.

The first objective is to identify issues in existing lightweight protocols during information exchange in IoT devices. To fulfill the first objective comparative study of existing lightweight protocols in IoT is analyzed based on multiple parameters (Architecture, Need of Broker, Transport Protocol, Security, Scope, Design Methodology, Packet Size, Service levels, Application, etc.). After comparative study, MQTT protocol identified as it has a QoS mechanism for each sort of message. However, it is very hard to ensure message delivery and ordering of messages.

Hence, the second objective addresses the effective solution to minimize the loss of information during message transfer among devices. This objective is achieved by designing the model of the Integrated Identity (ID) technique that minimizes message loss. After the execution of this new model on the IoT environment following outcome has achieved:

Connection time of the method is low 22.95 ms, proposed method transfer the first message within 0.86 Sec with satisfactory latency and less storage memory as compared with existing methods. Proposed method has the lower wrong message rate (less than 1%) than the existing message exchange method.

Integrated ID method is lightweight which is suitable for the majority of the devices in IoT.

After message loss minimization, the next objective is a network monitoring and integrity of information without degrading the performance of the system. Hence, fuzzy techniques pave a solution since they can be used to find vulnerabilities in a system. Such techniques pass unpredicted inputs first and afterwards, analyze the system. Proposed fuzzy technique utilizes a scapy method to operate in block-based protocol. The length and other control fields of the network are recalculated automatically with less duration.

Comparison with conventional methods for communication shows that the proposed fuzzy technique based method has higher performance. End-to-end delay in proposed method is 476 sec while existing method is 498 sec on the payload of 4000 bytes. The loss of messages in proposed method is 0.207%, while in the existing method the message loss is 0.22% at 3000 message payloads as concluded by experimental results.

## 7.2 Future Scope

Proposed framework has some limitations. Reporting of errors is one of the most significant constraints. Proposed work has implemented using MQTT protocol; this can extend to allow confirmation of a wider range of network protocols used by IoT devices. Future suggestions based on the thesis are as follows:

1. The possible future directions of this method involve in increases the security in IoT based on data encryption technique, the lightweight technique can be applied to decrease latency, and power consumption of the IoT devices is needed to be optimized.
2. This work can be extended with the combination of machine learning and other clustering algorithms.
3. The parameters can be extended in the future for reporting errors.
4. In future work, the security method can be developed (eg. authentication and authorization of device) in message transmission that can ensure the privacy of the user.

## Bibliography

- [1] Ashton, K. (2009). That ‘internet of things’ thing. *RFID journal*, 22(7), 97-114.
- [2] Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., Wolff, S. (2009). A brief history of the Internet. *ACM SIGCOMM Computer Communication Review*, 39(5), 22-31.
- [3] Atzori, L., Iera, A., Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15), 2787-2805.
- [4] Garcia-Morchon, O., Falck, T., Heer, T., Wehrle, K. (2009, July). Security for pervasive medical sensor networks. In *2009 6th Annual International Mobile and Ubiquitous Systems: Networking and Services, MobiQuitous* (pp. 1-10). IEEE.
- [5] Miorandi, D., Sicari, S., De Pellegrini, F., Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad hoc networks*, 10(7), 1497-1516.
- [6] Da Xu, L., He, W., Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4), 2233-2243.
- [7] Oen HM. Interoperability at the Application Layer in the Internet of Things (Master’s thesis, NTNU).
- [8] Guinard D, Ion I, Mayer S. In search of an internet of things service architecture: REST or WS-\*? A developers’ perspective. In *International Conference on*



- Mobile and Ubiquitous Systems: Computing, Networking, and Services 2011  
Dec 6, 326-337.
- [9] Haghi, M., Thurow, K., Stoll, R. (2017). Wearable devices in medical internet of things: scientific research and commercially available devices. *Healthcare informatics research*, 23(1), 4-15.
- [10] Weerawarana, S., Curbera, F., Leymann, F., Storey, T., Ferguson, D. F. (2005). *Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more*. Prentice Hall PTR. 2005.
- [11] Song Y. *Security in Internet of Things*. 2013.
- [12] Lee C. A survey of the World Wide Web evolution with respect to security issues. *PeerJ Preprints*; 2017.
- [13] Hammami R, Kacem YH, Souissi S, Bellaaj H, Kacem AH. Weighted Priority Queuing: A New Scheduling Strategy for Web Services. *Information Technology and Computer Science*, 2017, 11-17.
- [14] Nastic, S., Truong, H. L., Dustdar, S. (2015). Sdg-pro: a programming framework for software-defined iot cloud gateways. *Journal of Internet Services and Applications*, 6(1), 21.
- [15] Batalla, J. M., Gonciarz, F. (2019). Deployment of smart home management system at the edge: mechanisms and protocols. *Neural Computing and Applications*, 31(5), 1301-1315.

- [16] Brachmann M, Garcia-Morchon O, Kirsche M. Security for practical coap applications: Issues and solution approaches. GI/ITG KuVS Fachgesprch Sensornetze (FGSN). Universitt Stuttgart. 2011.
- [17] Hunkeler U, Truong HL, Stanford-Clark A. MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. In2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COM-SWARE'08) 2008 Jan 6 (pp. 791-798). IEEE.
- [18] Sheng, Z., Yang, S., Yu, Y., Vasilakos, A. V., McCann, J. A., Leung, K. K. (2013). A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. *IEEE Wireless Communications*, 20(6), 91-98.
- [19] Chen X. Constrained application protocol for internet of things. URL: <https://www.cse.wustl.edu/jain/cse574-14/ftp/coap>. 2014 Apr.
- [20] Moritz G, Golatowski F, Timmermann D. A lightweight SOAP over CoAP transport binding for resource constraint networks. In2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems 2011 Oct 17 (pp. 861-866). IEEE.
- [21] Mishra, A., Mohapatro, M. (2019). An IoT framework for bio-medical sensor data acquisition and machine learning for early detection. *International Journal of Advanced Technology and Engineering Exploration*, 6(54), 112-125.
- [22] Lee S, Kim H, Hong DK, Ju H. Correlation analysis of MQTT loss and delay according to QoS level. InThe International Conference on Information Networking 2013 (ICOIN) 2013 Jan 28 (pp. 714-717). IEEE.

- [23] Aman MN, Chua KC, Sikdar B. A light-weight mutual authentication protocol for IoT systems. In *GLOBECOM 2017-2017 IEEE Global Communications Conference 2017 Dec 4* (pp. 1-6). IEEE.
- [24] Pramukantoro ES, Wulandari JR, Yahya W, Nurwarsito H. A Cluster Message Broker in IoT Middleware using Ioredis. In *2018 International Conference on Sustainable Information Engineering and Technology (SIET) 2018 Nov 10* (pp. 247-251). IEEE.
- [25] Huang C, Liu D, Ni J, Lu R, Shen X. Reliable and Privacy-Preserving Selective Data Aggregation for Fog-Based IoT. In *2018 IEEE International Conference on Communications (ICC) 2018 May 20* (pp. 1-6). IEEE.
- [26] Alqinsi P, Edward IJ, Ismail N, Darmalaksana W. IoT-Based UPS Monitoring System Using MQTT Protocols. In *2018 4th International Conference on Wireless and Telematics (ICWT) 2018 Jul 12* (pp. 1-5). IEEE.
- [27] An D, Kim D. ICN-Based light-weighted mobility support in IoT. In *2018 27th International Conference on Computer Communication and Networks (ICCCN) 2018 Jul 30* (pp. 1-2). IEEE.
- [28] Chippalkatti P, Kadam G, Ichake V. I-SPARK: IoT Based Smart Parking System. In *2018 International Conference on Advances in Communication and Computing Technology (ICACCT) 2018 Feb 8* (pp. 473-477). IEEE.
- [29] Celia L, Cungang Y. (WIP) Authenticated key management protocols for Internet of Things. In *2018 IEEE International Congress on Internet of Things (ICIOT) 2018 Jul 2* (pp. 126-129). IEEE.

- [30] Anthi E, Williams L, Burnap P. Pulse: an adaptive intrusion detection for the internet of things. *Living in the Internet of Things: Cybersecurity of the IoT 2018* (pp. 1-4).
- [31] Bellagente P, Depari A, Ferrari P, Flammini A, Sisinni E, Rinaldi S. M 3 IoT—Message-oriented middleware for M-health Internet of Things: Design and validation. In *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) 2018 May 14* (pp. 1-6). IEEE.
- [32] Chandrappa DR, Pavan HR, Sagar MV, Dakshayini M. IoT Security Solution to Avoid Theft. In *2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) 2018 Mar 22* (pp. 1-3). IEEE.
- [33] Gao C, Ling Z, Chen B, Fu X, Zhao W. SecT: A Lightweight Secure Thing-Centered IoT Communication System. In *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS) 2018 Oct 9* (pp. 46-54). IEEE.
- [34] Iftikhar R, Memon M, Hafiz T, Jaffri R. A Novel framework for location based messages. In *2018 5th International Multi-Topic ICT Conference (IMTIC) 2018 Apr 25* (pp. 1-7). IEEE.
- [35] Kamalraj R, Sakthivel M. A Hybrid Model on Child Security and Activities Monitoring System Using IoT. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA) 2018 Jul 11* (pp. 996-999). IEEE.

- [36] Irmansyah M, Madona E, Nasution A, Putra R. Low Cost Heart Rate Portable Device for Risk Patients with IoT and Warning System. In 2018 International Conference on Applied Information Technology and Innovation (ICAITI) 2018 Sep 3 (pp. 46-49). IEEE.
- [37] Landge IA, Satopay H. Secured IoT Through Hashing Using MD5. In 2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB) 2018 Feb 27 (pp. 1-5). IEEE.
- [38] Kulik V, Kirichek R. The Heterogeneous Gateways in the Industrial Internet of Things. In 2018 10th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT) 2018 Nov 5 (pp. 1-5). IEEE.
- [39] Leshem G, David E, Domb M. Probability Based Keys Sharing for IOT Security. In 2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE) 2018 Dec 12 (pp. 1-5). IEEE.
- [40] Lenk A, Marcus P, Pova I. GeoFPE: Format Preserving Encryption of Geospatial Data for the Internet of Things. In 2018 IEEE International Congress on Internet of Things (ICIOT) 2018 Jul 2 (pp. 172-175). IEEE.
- [41] El Kashef A, Barakat N. Intelligent Alarm System to Protect Small, Valuable Items. In 2018 International Conference on Computer and Applications (ICCA) 2018 Aug 25 (pp. 326-330). IEEE.

- [42] Kurera C, Navoda D. Node-to-Node Secure Data Transmission Protocol for Low-power IoT Devices. In 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer) 2018 Sep 26 (pp. 1-7). IEEE.
- [43] Lin SC, Wu PH, Lu HT, Chuang SH, Wang WJ. Dynamic throttling for IoT streaming hub services on multi-tenant cloud environment. In 2018 IEEE International Conference on Applied System Invention (ICASI) 2018 Apr 13 (pp. 544-547). IEEE.
- [44] Kim DH, Lee HY, Kim DS. Enhanced industrial message protocol for real-time IoT platform. In 2018 International Conference on Electronics, Information, and Communication (ICEIC) 2018 Jan 24 (pp. 1-2). IEEE.
- [45] Kamal, M. (2018). Light-weight security and data provenance for multi-hop Internet of Things. *IEEE Access*, 6, 34439-34448..
- [46] Kumar S, Raza Z. A K-means clustering based message forwarding model for Internet of Things (IoT). In 2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence) 2018 Jan 11 (pp. 604-609). IEEE.
- [47] Madritsch C, Klinger T. Work in progress: Computing cluster using IoT technologies. In 2018 IEEE Global Engineering Education Conference (EDUCON) 2018 Apr 17 (pp. 1428-1431). IEEE.
- [48] Mukhopadhyay D, Gupta M, Attar T, Chavan P, Patel V. An Attempt to Develop an IOT Based Vehicle Security System. In 2018 IEEE International Symposium

- on Smart Electronic Systems (iSES)(Formerly iNiS) 2018 Dec 17 (pp. 195-198). IEEE.
- [49] Narang S, Nalwa T, Choudhury T, Kashyap N. An efficient method for security measurement in internet of things. In2018 International Conference on Communication, Computing and Internet of Things (IC3IoT) 2018 Feb 15 (pp. 319-323). IEEE.
- [50] Nasir H, Kanwal N. Prevention of Disclosure Attack on a Mutual Authentication Protocol Using RFID Tag in IoT. In2018 International Conference on Applied and Engineering Mathematics (ICAEM) 2018 Sep 4 (pp. 136-139). IEEE.
- [51] Ni F, Wei J, Shen J. An Internet of Things (IoTs) based Intelligent Life Monitoring System for Vehicles. In2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) 2018 Oct 12 (pp. 532-535). IEEE.
- [52] Oak A, Daruwala RD. Assessment of Message Queue Telemetry and Transport (MQTT) protocol with Symmetric Encryption. In2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC) 2018 Dec 15 (pp. 5-8). IEEE.
- [53] Peniak P, Franeková M. Extended Model of Secure Communication for Embedded Systems with IoT and MQTT. In2018 International Conference on Applied Electronics (AE) 2018 Sep 11 (pp. 1-4). IEEE.

- [54] Meera MS, Rao SN. Comparative analysis of IoT protocols for a marine IoT system. In 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) 2018 Sep 19 (pp. 2049-2053). IEEE.
- [55] Chen Y, Xu W, Peng L, Zhang H. Light-Weight and Privacy-Preserving Authentication Protocol for Mobile Payments in the Context of IoT. *IEEE Access*. 2019 Jan 21; 7, 15210-21.
- [56] Harbi Y, Aliouat Z, Refoufi A, Harous S. Efficient End-to-End Security Scheme for Privacy-Preserving in IoT. In 2019 International Conference on Networking and Advanced Systems (ICNAS) 2019 Jun 26 (pp. 1-6). IEEE.
- [57] Huynh-Van D, Le-Thi-Chau N, Ngo-Khanh K, Le-Trung Q. Towards an Integration of AES Cryptography into Deluge Dissemination Protocol for Securing IoTs Reconfiguration. In 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF) 2019 Mar 20 (pp. 1-6). IEEE.
- [58] Choi SK, Ko JS, Kwak J. A Study on IoT Device Authentication Protocol for High Speed and Lightweight. In 2019 International Conference on Platform Technology and Service (PlatCon) 2019 Jan 28 (pp. 1-5). IEEE.
- [59] Eldefrawy MH, Ferrari N, Gidlund M. Dynamic User Authentication Protocol for Industrial IoT without Timestamping. In 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS) 2019 May 27 (pp. 1-7). IEEE.



- [60] Mamour DI, Congduc PH. Increased flexibility in long-range IoT deployments with transparent and light-weight 2-hop LoRa approach. In2019 Wireless Days (WD) 2019 Apr 24 (pp. 1-6). IEEE.
- [61] Gebremichael T, Jennehag U, Gidlund M. Lightweight IoT Group Key Establishment Scheme from the One Time Pad. In2019 7th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (Mobile-Cloud) 2019 Apr 4 (pp. 101-106). IEEE.
- [62] Hribar J, DaSilva L. Utilising Correlated Information to Improve the Sustainability of Internet of Things Devices. In2019 IEEE 5th World Forum on Internet of Things (WF-IoT) 2019 Apr 15 (pp. 805-808). IEEE.
- [63] Mostafa B. Monitoring Internet of Things Networks. In2019 IEEE 5th World Forum on Internet of Things (WF-IoT) 2019 Apr 15 (pp. 295-298). IEEE.
- [64] Muhammad A, Afzal B, Imran B, Tanwir A, Akbar AH, Shah G. oneM2M Architecture Based Secure MQTT Binding in Mbed OS. In2019 IEEE European Symposium on Security and Privacy Workshops (EuroSPW) 2019 Jun 17 (pp. 48-56). IEEE.
- [65] Munsadwala Y, Joshi P, Patel P, Rana K. Identification and Visualization of Hazardous Gases Using IoT. In2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) 2019 Apr 18 (pp. 1-6). IEEE.

- [66] Poulter AJ, Johnston SJ, Cox SJ. pySRUP–Simplifying Secure Communications for Command Control in the Internet of Things. In2019 IEEE 5th World Forum on Internet of Things (WF-IoT) 2019 Apr 15 (pp. 273-277). IEEE.
- [67] Alhazmi OH, Aloufi KS. Fog-Based Internet of Things: A Security Scheme. In2019 2nd International Conference on Computer Applications Information Security (ICCAIS) 2019 May 1 (pp. 1-6). IEEE.
- [68] Rahman A, Roy S, Kaiser MS, Islam MS. A Lightweight Multi-tier S-MQTT Framework to Secure Communication between low-end IoT Nodes. In2018 5th International Conference on Networking, Systems and Security (NSysS) 2018 Dec 18 (pp. 1-6). IEEE.
- [69] Rocha R, Ferreira LL, Maia C, Souto P, Varga P. Improving the performance of a Publish-Subscribe message broker. In2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC) 2019 May 7 (pp. 91-92). IEEE.
- [70] Roy SK, Misra S, Raghuwanshi NS. SensPnP: Seamless Integration of Heterogeneous Sensors with IoT Devices. IEEE Transactions on Consumer Electronics. 2019 Mar 6;65(2):205-14.
- [71] Sahinel D, Akpolat C, Görür OC, Sivrikaya F. Integration of Human Actors in IoT and CPS Landscape. In2019 IEEE 5th World Forum on Internet of Things (WF-IoT) 2019 Apr 15 (pp. 485-490). IEEE.
- [72] Sasaki Y, Yokotani T, Mukai H. Proposals on IoT Communication through MQTT over L2 Network and their Performance Evaluation. In2018 Interna-

- tional Conference on Innovations in Information Technology (IIT) 2018 Nov 18 (pp. 30-35). IEEE.
- [73] Schütte J, Brost GS. LUCON: data flow control for message-based IoT systems. In 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) 2018 Aug 1 (pp. 289-299). IEEE.
- [74] Shah T, Venkatesan S. Authentication of IoT device and IoT server using secure vaults. In 2018 17th IEEE International Conference on Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) 2018 Aug 1 (pp. 819-824). IEEE.
- [75] Shukla S, Hassan MF, Jung LT, Awang A. Fuzzy-based Fog Computing for Real-Time Data Transmission in Healthcare Internet-of-Things. International Conference on Green Computing and Internet of Things (PP. 104-108). IEEE.
- [76] Singh AK, Bhushan S, Vij S. Filtering spam messages and mails using fuzzy C means algorithm. In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) 2019 Apr 18 (pp. 1-5). IEEE.
- [77] Sreeraj S, Kumar GS. Performance of IoT protocols under constrained network, a Use Case based approach. In 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT) 2018 Feb 15 (pp. 495-498). IEEE.

- [78] Su WT, Chen WC, Chen CC. An Extensible and Transparent Thing-to-Thing Security Enhancement for MQTT Protocol in IoT Environment. In2019 Global IoT Summit (GIoTS) 2019 Jun 17 (pp. 1-4). IEEE.
- [79] Sun T, Xu Y, Li J, Zhang H. Research on Internet of Things Middleware Technology for Laboratory Environmental Monitoring. In2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS) 2018 Aug 10 (pp. 544-547). IEEE.
- [80] Thenmozhi S, Ranjitha JB, Anu S. Light Weight Security Framework for IoT. 5th International Conference on Advanced Computing and Communication Systems (ICACCS) 2019 Mar 15 (pp. 546-549).
- [81] Tkachenko V, Goriushkina A, Kolisnyk M. Communication Messaging Models in IoT/WoT: Survey and Application. In2018 International Scientific-Practical Conference Problems of Info communications. Science and Technology (PICST) 2018 Oct 9 (pp. 417-422). IEEE.
- [82] Ulz T, Pieber T, Steger C, Holler A, Haas S, Maticsek R. Automated Authentication Credential Derivation for the Secured Configuration of IoT Devices. In2018 IEEE 13th International Symposium on Industrial Embedded Systems (SIES) 2018 Jun 6 (pp. 1-8). IEEE.
- [83] Verma K, Jain N. IoT Object Authentication for Cyber Security: Securing Internet with Artificial intelligence. In2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) 2018 Feb 24 (pp. 1-3). IEEE.

- [84] Zilani KA, Yeasmin R, Zubair KA, Sammir MR, Sabrin S. R 3 HMS, An IoT Based Approach for Patient Health Monitoring. In 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2) 2018 Feb 8, (pp.1-4).IEEE.
- [85] Gluhak, A., Krco, S., Nati, M., Pfisterer, D., Mitton, N., Razafindralambo, T. (2011). A survey on facilities for experimental internet of things research. *IEEE Communications Magazine*, 49(11), 58-67..
- [86] Ali AS, Zanzinger Z, Debose D, Stephens B. Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection. *Building and Environment*. 2016 May 1; 100, 114-26.
- [87] Foster, S. W., Alirangues, M. J., Naese, J. A., Constans, E., Grinias, J. P. (2019). A low-cost, open-source digital stripchart recorder for chromatographic detectors using a Raspberry Pi. *Journal of Chromatography A*, 1603, 396-400.
- [88] La Marra A, Martinelli F, Mori P, Rizos A, Saracino A. Improving MQTT by inclusion of usage control. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage* 2017 Dec 12, 545-560.
- [89] Luzuriaga JE, Perez M, Boronat P, Cano JC, Calafate C, Manzoni P. Improving mqtt data delivery in mobile scenarios: Results from a realistic testbed. *Mobile Information Systems* 2016.

- [90] Roy DG, Mahato B, De D, Buyya R. Application-aware end-to-end delay and message loss estimation in Internet of Things (IoT)—MQTT-SN protocols. *Future Generation Computer Systems*. 2018 Dec 1; 89, 300-16.
- [91] Santamaria AF, De Rango F, Serianni A, Raimondo P. A real IoT device deployment for e-Health applications under lightweight communication protocols, activity classifier and edge data filtering. *Computer Communications*. 2018 Sep 1; 128, 60-73.
- [92] Hernández Ramos S, Villalba MT, Lacuesta R. MQTT Security: A Novel Fuzzing Approach. *Wireless Communications and Mobile Computing: 2018*.
- [93] Biyani A, Sharma G, Aghav J, Waradpande P, Savaji P, Gautam M. Extension of SPIKE for encrypted protocol fuzzing. In *2011 Third International Conference on Multimedia Information Networking and Security* 2011 Nov 4, 343-347.
- [94] <https://docs.python.org/3.4/library/argparse.html>
- [95] Bansal, S., Bansal, N. (2015). Scapy-a python tool for security testing. *Journal of Computer Science Systems Biology*, 8(3), 140.
- [96] Jones, A. (2004). Netfilter and IPTables—a structural examination. SANS Institute Reading Room site.
- [97] <https://www.netfilter.org>
- [98] <https://www.ee.oulu.fi/roles/ouspg/Radams>
- [99] Gowtham P, Arunachalam VP, Vijayakumar VA, Karthik S. An Efficient Monitoring of Real Time Traffic Clearance for an Emergency Service Vehicle Using IOT. *International Journal of Parallel Programming*. 2018, 1-27.

- [100] Yeh, K. H. (2016). A secure IoT-based healthcare system with body sensor networks. *IEEE Access*, 4, 10288-10299.
- [101] Diogo P, Lopes NV, Reis LP. An ideal IoT solution for real-time web monitoring. *Cluster Computing*. 2017 Sep 1;20(3), 2193-209.
- [102] Jeon S, Jung I. Experimental evaluation of improved IoT middleware for flexible performance and efficient connectivity. *Ad Hoc Networks*. 2018 Mar 1;70, 61-72.
- [103] Ciuffoletti A. Occi-IoT: an API to deploy and operate an IoT infrastructure. *IEEE Internet of Things Journal*. 2017 Jul 31;4(5), 1341-8.
- [104] Hwang HC, Park J, Shon JG. Design and Implementation of a Collaboration Messenger System Based on MQTT Protocol. In *Advances in Computer Science and Ubiquitous Computing 2015*, 513-519.
- [105] Fremantle P, Aziz B. Cloud-based federated identity for the Internet of Things. *Annals of Telecommunications*. 2018 Aug 1;73 (7-8), 415-27.
- [106] Zhang, Z. K., Cho, M. C. Y., Wang, C. W., Hsu, C. W., Chen, C. K., Shieh, S. (2014, November). IoT security: ongoing challenges and research opportunities. In *2014 IEEE 7th international conference on service-oriented computing and applications*, 230-234.
- [107] K. Tabassum, A. Ibrahim and S. A. El Rahman, "Security Issues and Challenges in IoT," *2019 International Conference on Computer and Information Sciences (ICCIS)*, Sakaka, Saudi Arabia, 2019, 1-5.

- [108] Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun, Hui-Ying Du, “Research on the architecture of Internet of Things”, in: Proceeding of 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), 2010, 484-487.
- [109] S. Sicari, A. Rizzardi, L.A. Grieco, A. Coen-Porisini, “Security, privacy and trust in Internet of Things: The road ahead”, *Computer Networks Journal*, Vol. 76, 2015, 146–164.
- [110] Christos Stergioua, Kostas E. Psannisa, Brij B. Guptab, Yutaka Ishibashic, “Security, privacy efficiency of sustainable Cloud Computing for Big Data and IoT”, *Sustainable Computing: Informatics and Systems*, Vol. 19, 2018, 174–184.
- [111] Zhihong Yang, Yingzhao Yue, Yu Yang, Yufeng Peng, Xiaobo Wang, Wenji Liu, “Study and application on the architecture and key technologies for IoT”, in Proceeding of 2011 International Conference on Multimedia Technology ( ICMT), 2011, 747-751.
- [112] <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>
- [113] Niruntasukrat, A., Issariyapat, C., Pongpaibool, P., Meesublak, K., Aium-supucgul, P., Panya, A. (2016, May). Authorization mechanism for mqtt-based internet of things. In 2016 IEEE International Conference on Communications Workshops (ICC), 290-295.
- [114] Dinculeană, D., Cheng, X. (2019). Vulnerabilities and limitations of MQTT protocol used between IoT devices. *Applied Sciences*, 9(5), 848.



- [115] Anthraper, J. J., Kotak, J. (2019). Security, Privacy and Forensic Concern of MQTT Protocol. Privacy and Forensic Concern of MQTT Protocol, March 19, 2019.
- [116] Hernández Ramos, S., Villalba, M. T., Lacuesta, R. (2018). Mqtt security: A novel fuzzing approach. Wireless Communications and Mobile Computing, 2018.
- [117] Authentication with Username and Password - MQTT Security Fundamentals,<https://www.hivemq.com/blog/mqtt-security-fundamentals-authentication-username-password/>
- [118] Authorization - MQTT Security Fundamentals,<https://www.hivemq.com/blog/mqtt-security-fundamentals-authorization/>.
- [119] Amaran, M., Rohmad, M., Adnan, L., Mohamed, N., Hashim, H. (2018). Lightweight security for MQTT-SN. International Journal of Engineering and Technology (UAE), 7(4), 223-226.
- [120] Haripriya, A. P., Kulothungan, K. (2019). Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things. EURASIP Journal on Wireless Communications and Networking, 2019(1), 90.

## List of Publications on this Research Work

### 1. Scopus Indexed Journals

1. **Ankit Khare, Rashmi Sharma, Neelu Jyoti Ahuja**, *Experimental Investigation of Integrated ID Method to Mitigate Message Loss in IOT Control Devices*, Journal of Engineering Science Technology (JESTEC) Vol. 15, No.1(Feb 2020) (pp.32-45).
2. **Ankit Khare, Rashmi Sharma, Neelu Jyoti Ahuja**, *Secure Message Transfer in Internet of Things Environment using Weighing based Fuzzy Technique*, International Journal Of Scientific Technology Research(IJSTR) Vol. 9, Issue 01, (Jan 2020)
3. **Ankit Khare, Rashmi Sharma, Neelu Jyoti Ahuja**, *Analysis of Various Light Weight Protocols in Internet of Things-A Comparative Study*, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Vol. 8 Issue-6C, (Apr 2019).

### 2. Refereed International Conference

4. **Ankit Khare, Rashmi Sharma, Neelu Jyoti Ahuja**, *Approach based Analysis of Internet of Things on Computational and Security Aspects*, in Intelligent Systems Conference (IntelliSys) 3-4 September 2020 Amsterdam, The Netherlands.  
[Accepted]



### Document Information

<b>Analyzed document</b>	Final thesis.pdf (D81050029)
<b>Submitted</b>	10/8/2020 3:15:00 PM
<b>Submitted by</b>	Rashmi Sharma
<b>Submitter email</b>	rashmi.sharma@ddn.upes.ac.in
<b>Similarity</b>	3%
<b>Analysis address</b>	rashmi.sharma.upes@analysis.urkund.com

### Sources included in the report

<b>SA</b>	<b>1512499702-TS.pdf</b> Document 1512499702-TS.pdf (D55292850)		2
<b>SA</b>	<b>IEEE conference final paper 24-02-2020.docx</b> Document IEEE conference final paper 24-02-2020.docx (D64307515)		2
<b>W</b>	URL: <a href="http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html">http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html</a> Fetched: 10/8/2020 3:17:00 PM		1
<b>W</b>	URL: <a href="https://www.researchgate.net/publication/326168098_Application-aware_end-to-end_de...">https://www.researchgate.net/publication/326168098_Application-aware_end-to-end_de ...</a> Fetched: 12/10/2019 12:01:21 AM		4
<b>J</b>	<b>Internet of Things : a gateway centric solution for providing IoT connectivity</b> URL: 4a0b57c8-4818-423d-ab44-86d0c38921ba Fetched: 2/17/2019 1:48:24 AM		3
<b>SA</b>	<b>socialaware-mobile-wireless_PDFa.pdf</b> Document socialaware-mobile-wireless_PDFa.pdf (D33157226)		2
<b>SA</b>	<b>P.Arivubrahan-Paper-IJITEE.doc</b> Document P.Arivubrahan-Paper-IJITEE.doc (D80181375)		2
<b>W</b>	URL: <a href="https://www.astesj.com/publications/ASTESJ_040131.pdf">https://www.astesj.com/publications/ASTESJ_040131.pdf</a> Fetched: 6/3/2020 11:00:14 PM		1
<b>SA</b>	<b>thesis_report.pdf</b> Document thesis_report.pdf (D55577552)		1
<b>W</b>	URL: <a href="https://en.wikipedia.org/wiki/Internet_of_Things">https://en.wikipedia.org/wiki/Internet_of_Things</a> Fetched: 10/8/2020 3:17:00 PM		2
<b>W</b>	URL: <a href="https://www.groundai.com/project/survey-of-communication-protocols-for-internet-of...">https://www.groundai.com/project/survey-of-communication-protocols-for-internet-of ...</a> Fetched: 12/14/2019 2:05:39 PM		1
<b>W</b>	URL: <a href="https://arxiv.org/pdf/1804.01747">https://arxiv.org/pdf/1804.01747</a> Fetched: 10/3/2019 6:56:09 AM		3