Fall 2021

Performance Evaluation of IoT Protocols for Environment Monitoring

Nishat Sultana
University of Windsor

Performance Evaluation of IoT Protocols for Environment Monitoring

By

Nishat Sultana

A Thesis

Submitted to the Faculty of Graduate Studies

through the Department of Electrical and Computer Engineering

in Partial Fulfillment of the Requirements for

the Degree of Master of Applied Science

at the University of Windsor

Windsor, Ontario, Canada

2021

Performance Evaluation of IoT Protocols for Environment Monitoring

by

Nishat Sultana

APPROVED BY:

____
A. Jaekel

School of Computer Science

____
E. Abdel-Raheem

Department of Electrical and Computer Engineering

____
K. Tepe, Advisor

Department of Electrical and Computer Engineering

August 5, 2021

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

For the last few decades, environmental pollution has created adverse effects on humans and the ecosystem. The pollutant of natural origin or man-made may cause diseases, allergies, and widespread damages to humans, animals, and food crops. The environmental issues could be generated by pollution of all kinds, i.e. air pollution, water pollution, and climate changes. For example, the wildfires incidents in Canada have a massive influence on air pollution since the caused devastation has increased significantly over the past years.

An environmental surveillance and monitoring system can be an effective tool to minimize the concern. However, developing a system for continuous interaction is a challenge due to the lack of communication coverage in far and isolated areas as well as power constraints. In this work we undertake a performance evaluation of an environment monitoring system applying the use of protocols and systems like Internet of Things (IoT), Message Queuing Telemetry Transport (MQTT), and Constrained Application Protocol (CoAP). This has the potential of being the leading technology since it makes machine-to-machine communication possible with minimum requirements.

The proposed prototype allows the fixed ground node located on a remote site to communicate with a moving node like a drone. The transmitted data packets were analyzed based on overheads, latency. The Packet Delivery Rate reaches 90% for MQTT even with a 600-meter distance between the two nodes. Bandwidth usage of CoAP is around 85 bits/s with 5000 data packets transmission. The designed system aimed to demonstrate the merits of the selected IoT protocols.

ACKNOWLEDGEMENTS

I sincerely want to express my gratitude to my thesis advisor Dr. Kemal Tepe for his support, guidance, and advice to complete the thesis. I would like to thank my committee members, Dr. A. Jaekel and Dr. E. Abdel-Raheem for their valuable comments and suggestions. I would like to extend a big thanks to Dr. Sabbir Ahmed for all the knowledge and support provided. And finally, I am grateful to my loving family for all the encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IoT	Internet of Things
6LowPAN	IPv6 over Low -Power Wireless Personal Area Networks
MQTT	Message Queuing Telemetry Transport
CoAP	Constrained Application Protocol
XMPP	Extensible Messaging and Presence Protocol
DTLS	Datagram Transport Layer Security
LwM2M	Lightweight Machine-to-Machine
UAV	Unmanned Aerial Vehicle
HTTP	Hypertext Transfer Protocol
LoRaWAN	Long Range Wide Area Network
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
AMQP	Advanced Message Queuing Protocol
OSI	Open Systems Interconnection
QoS	Quality of Service
DDS	Data Distribution Service
NDN	Named Data Networking
OPC UA	Open Platform Communications- Unified Architecture
GSM	Global System for Mobile

EDGE	Enhanced Data GSM Environment
UMTS	Universal Mobile Telecommunications System
LTE	Long Term Evolution

CHAPTER-1
INTRODUCTION

1.1 Introduction

Our ecosystem and human lives are facing harmful impacts of environmental pollution for the last few decades, such as a variety of diseases, allergies, disorders, and even death. Damages have also occurred to our grown crops, lands, properties, and animals. The crisis in the environment is produced by pollutions of all kinds, i.e. air pollution, water pollution, and climate changes.

Air pollution is one of the most serious kinds of environmental threats. In 2014 the World Health Organization (WHO) estimated that every year air pollution causes the premature death of 7 million people worldwide. The studies published in March 2019 specified that the number may be about 8.8 million [1]. Typically, air pollution is a mix of gasses and particles that have reached a harmful concentration both outside and indoors. In this modern era due to the extreme progression in the industrial and transport sector as well as the thermal and nuclear power generation plants cause a serious threat to humans and all living surroundings. The recent severe incidents of wildfires took all the attention in the field and have a huge impact on air pollution.

1.2 Impacts of Wildfire

The large uncontrolled blaze of the wildfires swiftly spread out through natural and rural areas and are fed by wind and drylands. It could be caused by human activities or natural events. The three main components that need to be indicated to cause wildfires are fuel, oxygen, and heat, which are called fire triangles by the firefighters. Nevertheless, the natural increase in temperature delivers the accurate climate to initiate an ignition for wildfire. It is estimated by Environmental Protection Agency (EPA) that 10 to 15 percent of wildfires occur on their own in wildlife. The rest 85 to 90 percent result from human activities, which include lit cigarettes, and unattended camp and debris fires. Wildfires have obvious devastating impacts on people's properties and lives. Table 1.1 demonstrates the impacts of some major wildfire incidents in the last two decades.

Table 1.1: Impacts of Wildfire

Year	**Location**	**Description**
2016	Canada	Fort McMurtry is the costliest wildfire in Canadian history. The total cost of the wildfires exceeded $8.86 billion with around 90,000 inhabitants forced to flee their properties by the flames [12].
2017	Canada	The estimated damage cost of around $100 million due to wildfires near William Lake British Columbia [11].
2017	Canada	Elephant Hill wildfires in B.C caused $27 million in damages to people's homes, vehicles, and businesses [11].
2019	United States	So far over 5,819 fires have been recorded according to the US Forest Service and totaling an estimated 162,693 acres of burned land as of October 13 [13].
Each year since 2000	United States	An average of 7 million acres of burnt land each year, double the number of acres scorched by wildfires in the 1990s [15].

1.3 Problem Statement

According to the National Wildlife Federation report [16], western forests become combustible within a month of snowmelt completion, which occurs 1 to 4 weeks earlier than it did 50 years ago. Also, spring runoff earlier causing summer to heat up rapidly and extend further into fall. All these factors leading a longer fire season. In western North America, the summer temperature has increased 3.6 to 9 degrees Fahrenheit by mid-century, boosting evaporation rates, while precipitation is decreased by up to 15 percent. Therefore, the probability of fire occurrence has become higher due to increased drier conditions. 1.8-degree Fahrenheit increase in temperature is projected to lead to a 6 percent enhancement in lightning, which has increased 12 to 30 percent in the region by mid-century. All these factors are forcing the change of wildfire.

The concept of the continuous early measuring of the air quality, temperature, and concentration of harmful gases in the environment might be the most effective method to prevent. For Wi-Fi technology, the typical range of a common 802.11g network with standard equipment is on the order of tens of meters, which is insufficient for a larger area. To acquire additional range, repeaters or additional access points will raise the cost [17]. Also, Wi-Fi may slow down due to a lack of bandwidth. The Bluetooth allows short-range communication with a slow data transfer rate and poor security [18]. Cellular technology requires an infrastructure setup. Installation of antennas requires space and a foundation tower, which is costly, time-consuming, and requires higher effort for less accessible areas [19]. So, the technologies like Wi-Fi, Bluetooth, or cellular network would not be a suitable option due to their lack of coverage, limited accessibility to isolated far locations, and high-power consumption. As a result, the internet of things (IoT) has emerged to fulfill the communications for such an application.

1.4 Objective of Thesis

For a resolution of environmental monitoring, the best option would be an experimental setup of continuous communication between a ground node producing air quality information and a drone/unmanned aerial vehicle (UAV) that flies over the rural and inaccessible wild areas to collect data from these nodes and store with the help of IoT protocols. The drone broadcasts the information to a satellite, which will later send the data to an earth station, where the data will be available from a satellite message repository. Figure 1.1 demonstrates the overall system layout. The focus of this thesis work is enclosed inside the box on the left of the figure, which includes the data transmission from the ground node (1) to the receiver deployed on the UAV (2). The objective of the thesis is to develop the prototype with an up-to-date IoT-based machine-to-machine (M2M) communication technology and inspection the advantage and disadvantage.

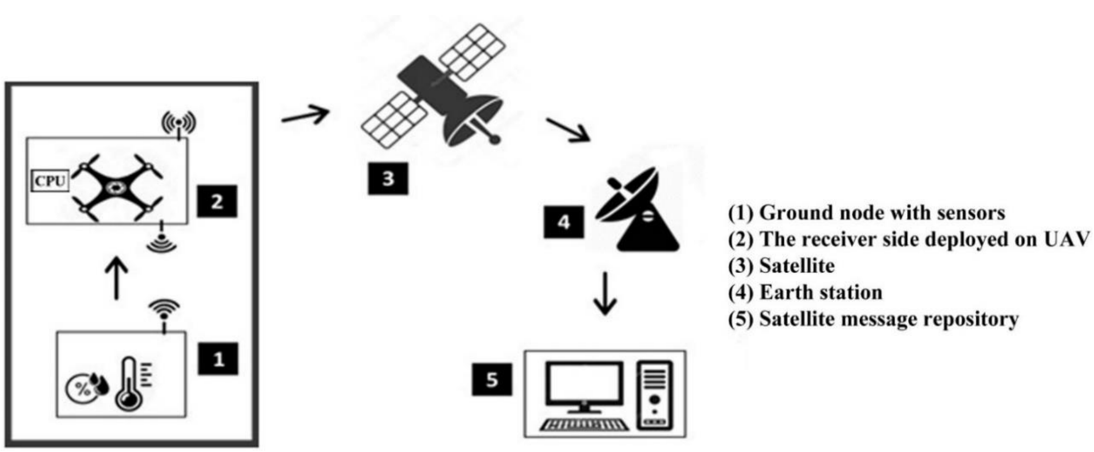

Figure 1.1: Overall System Layout

1.5 Contribution of Thesis

This thesis concentrates on the application of IoT protocols with a developed prototype which includes ground node with sensor and microcontroller as well as single-board computer deployed on the drone. The ground node is programmed to generate data packets with the sensor information and that will be later stored in the server installed on the single-board computer of the receiver side with the help of different IoT protocols. Packet Delivery Rate (PDR) is provided with the shortest distance between ground node and drone, with about 5000 samples for each selected distance used for the experiment. The different size of overheads creates dissimilar packet size for IoT protocols and transport layer protocols cause variations in latency. Microcontrollers are programmed using Arduino IDE, with the installed packages, Arduino SAMD Boards, DHT sensor library, Radiohead library, etc. All these are available in C language. The single-board computer Raspberry Pi uses New Out of the Box Software (NOOBS) as its OS as well as Mosquitto and Libcoap as a broker. Through utilizing the system developed in this thesis work, it is possible to compare the performance and capabilities of the popular IoT protocols, Message Queue Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP) as well as developing a new networking protocol to show some improvement.

1.6 Internet of Things

IoT is the network of connected physical objects, machines, and devices. These objects include vehicles, home appliances, communication devices, and other embedded electronics and sensors. IoT makes these devices and machines able to exchange data, and it provides dependable and consistent connectivity with minimum power consumption, low cost, and higher efficiency, which is consequential to a huge economic improvement [20]. According to the survey [21], the IoT industry will significantly increase and will be around $8.9 trillion, after it was upstretched from $2.99 trillion in 2014. This study also demonstrates that there will be more than 35 billion connected devices by the year 2021. By the year 2025, this quantity will rise and reach about 75 billion devices.

Figure 1.2 [21] shows the worldwide installed IoT-connected devices in billions. From this statistic, it is visible that IoT will have an incredible worldwide increase in the next decade because of its marvelous performance and glorious success. The use of IoT decreases the interaction between humans and machines while providing a direct connection between machines. This allows machines to communicate and data transfer with each other without human intervention. This is also known as machine-to-machine (M2M) Communication.

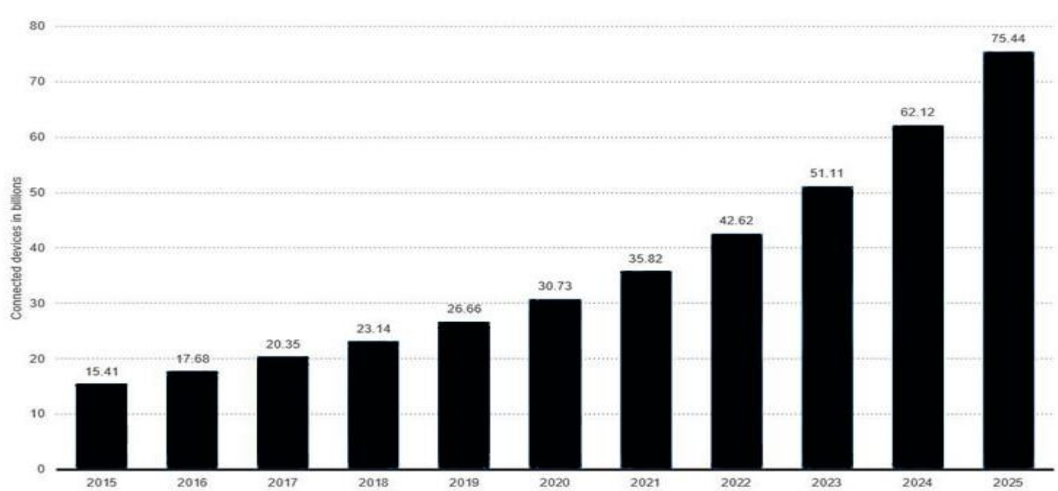

Figure 1.2 The Amount of IoT Connected Devices

1.7 IoT Protocols

The IoT system has three levels of architecture, they are, devices, gateway, and data system, where the data moves between these levels. The invisible language that allows the physical objects to communicate with each other consists of IoT standards and protocols. Usually, the general protocols used for personal computers, smartphones, or tablets may not be suitable for specific requirements of bandwidth, range, power consumption, etc. of IoT-based solutions [22]. This is the reason why several IoT network protocols have been developed, while the new ones are still evolving.

IoT standards and protocols can be generally classified into two distinct categories. The first one is the IoT network protocols. These protocols are used for connecting devices over the network. These communication protocols are typically used over the Internet. With the use of IoT network protocols, end-to-end data communication within the scope of the network is permissible. HTTP, LoRaWAN, Bluetooth, ZigBee are examples of IoT network protocols. The second category is the IoT Data Protocols. These are used for the connection of low-power IoT devices. These are the set of communication protocols that provide point-to-point communication with the hardware at the user side without any internet connection. The connectivity in these IoT data protocols is through wires, broadcasting, or cellular networks [23]. Some of the IoT data protocols are Message Queue Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), Advanced Message Queuing Protocol (AMQP), Extensible Messaging, and Presence Protocol (XMPP), etc.

1.8 Organization of Thesis

This section describes the organization of the remaining parts of the thesis. Chapter 2 reveals the background and related work for investigating IoT. The chapter addresses all the citation and related papers that involves previous work on different IoT protocols, which are relevant to this thesis work. Chapter 3 discusses the main proposal of the work, including the system design, instruments, and implementation illustrated in figures. The chapter also describes the architecture of a newly proposed networking protocol. Chapter 4 reflects the experimental results, measurements as well as compare them for the two IoT protocols. It also assesses between the newly proposed networking protocol and the existing two protocols. All the measurements are demonstrated using diagrams and tables. Lastly, chapter 5 provides the conclusion and the suggested future works.

CHAPTER-2
BACKGROUND AND LITERATURE SURVEY

2.1 Background

In the early 1980s, the concept of IoT was utilized at Carnegie Mellon University with a coke vending machine being the first internet appliance. The IoT-related concerns and activities came down to theoretical concepts, discussions, and individual ideas in the 1990s. 2000 to 2010 was a period of swift development when IoT projects began to succeed and got practical applications [22]. It is essential to analyze the features of IoT for a proper understanding of its potential. The literature review will be helpful to comprehend the different applications and aspects of the IoT protocols. This chapter will discuss the main features of IoT and related papers into different subsections about the experiment description, result, contributions of the prototypes.

2.2 Aspects Related to IoT Protocols

IoT is the concept of internet connectivity into everyday physical devices. These devices can interact and collaborate with others over the internet and can be monitored and operated. IoT protocols have made the evolution of M2M communication viable. To make a proper selection of a protocol that can be more suitable for an experimental scenario, it is essential to compare the features of the popular protocols. Table 2.1 reflects the comparison.

Table 2.1: Comparison of popular protocols [27],[28],[29],[30],[31],[32]

Features	MQTT	CoAP	HTTP	AMQP	XMPP
Lightweight protocol	Yes	Yes	No	No	Yes
Suited to resource-constrained devices	Yes	Yes	No	No	Yes
Bandwidth usage	Low	Low	High	High	High
Power consumption	Low	Low	High	High	Low
Asynchronous messaging	Yes	Yes	No	Yes	Yes
Quality of service	Supports	Supports	Do not support	Supports	Do not support

For an experimental scenario of continuous data transmission using constrained devices the CoAP and MQTT protocols are more suitable than others.

2.2.1 Message Queuing Telemetry Transport (MQTT)

One of the most favored protocols for IoT devices is MQTT, which collects data from various electronic devices and supports remote device monitoring, where all the clients connect to a common server, known as a broker.

Table 2.2: Features of MQTT protocol [35]

Properties	Description
Communication model	Publish-Subscribe
Application	Mostly used in devices that are economical as well as require less power and memory. For example, car sensors, smartwatches, microcontrollers, and text-based messaging apps.
Number of message types	16 message types
Application reliability	Supports three quality of service levels, such as, “Fire and forget”, “delivered at least once” and “delivered exactly once”.

2.2.2 Constrained Application Protocol (CoAP)

CoAP is a document transfer protocol, designed for the servers and clients to communicate through connectionless datagrams. Clients make requests to servers and servers send back responses.

Table 2.3: Features of CoAP [35]

Properties	**Description**
Communication model	Request-Response
Application	Used in constrained devices without consumption of extra RAM as well as requires less power, such as automation, mobiles, and microcontrollers.
Number of message types	4 message types
Application reliability	Supports two quality of service levels, such as the requests and response messages will be indicated as “confirmable” or “non conformable”.

2.2.3 Transport Layer Protocols

In computer networking the Open Systems Interconnection (OSI) model and TCP/IP model have layered architecture and provide similar functionalities. The transport layer is responsible to deliver the data to the appropriate application process on the host machines, this includes, forming data segments, and adding source and destination port numbers in the header of each transport layer data segment. Later it includes the source and destination IP address [36].

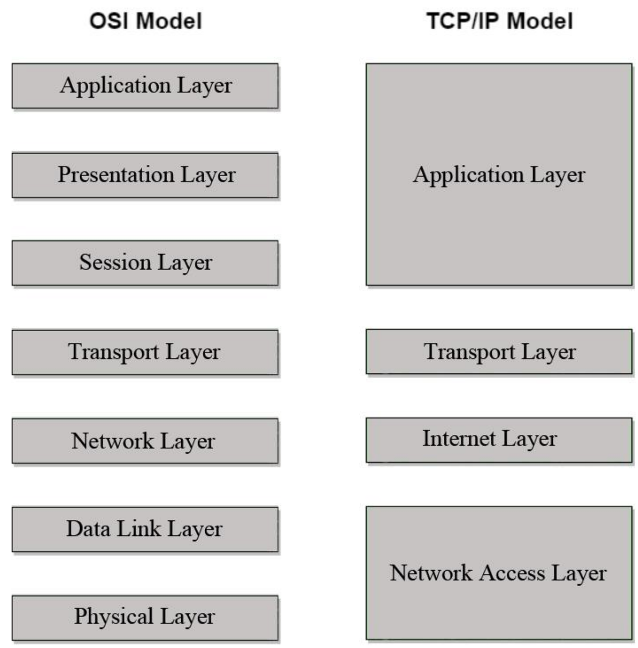

Figure 2.1: OSI and TCP/IP Model

TCP is a transport-layer protocol, in which a connection is established before data transmission begins. Data is sent without errors or duplication and is received in the same order as it is sent. MQTT protocol runs on top of TCP. UDP is another transport-layer protocol and is called an alternative to TCP, provides an unreliable datagram connection between applications. CoAP runs over UDP. Figure 2.2 shows the packet flow between sender and server.

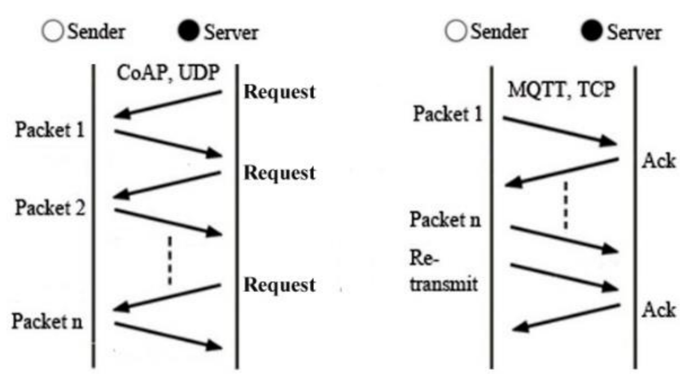

Figure 2.2: Data Packet Flow Over UDP and TCP

2.2.4 Packet Format

a) MQTT Message Format [37]:

The MQTT packets format includes three fields and those are fixed header, variable header, and payload. The fixed header field is present in all the packets, but the variable header and payload may or may not be present and the size is also variable.

Table 2.4 (a): MQTT Control Packet Structure

Fixed header (Present in all MQTT control packets)	**Variable header** (Present in some MQTT control packets)	**Payload** (Present in some MQTT control packets)
2 bytes	Variable size	Variable size

Table 2.4 (b): MQTT Fixed Header Structure

7	6	5	4	3	2	1	0	bit
Message type				Header flags				1st Byte
Remaining Length								2nd Byte

The message type represents a connection request type with 4 bits in length. Header flags include DUP, QoS, and RETAIN. Here, DUP is one bit (bit 3), QOS is 2 bit (bit 1 and bit 2), and RETAIN is one bit (bit 0). The variable header is not present in all MQTT control packets, and it has a different structure for different MQTT requests. Payload is actual information data that would be sent. but it is not present in all the MQTT control packets.

b) CoAP Message Format [38],[39]:

CoAP uses messages for the requests and responses by using a simple, binary, base header format. After the headers, any bytes are considered the message body, implied by the datagram length.

Version indicates the CoAP version number. Type is the message is of type Confirmable (0), Non-confirmable (1), Acknowledgement (2), or Reset (3). Token length maybe 0-8 bytes in length. CoAP Request/Response Code details are available in the reference. Message-ID is used to detect message duplication and to match message type.

Table 2.5: CoAP Header Structure

Offsets	**Octet**	**0**								**1**				**2**				**3**			
Octet	**Bit**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**..**	**..**	**15**	**16**	**..**	**..**	**23**	**24**	**..**	**..**	**31**
4	**32**	VER		Type		Token Length				CoAP Request/ Response Code				Message ID							
8	**64**	Token (0-8bytes)																			
12	**96**																				
16	**128**	Options (if available)																			
20	**160**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	Payload (if available)											

2.3 Related Works

This section describes the previously prepared works about IoT protocols related to this thesis but differs in experimental setup and focus of their contribution.

Table 2.6 (a): Summary of related work

Authors	Contribution	Experiment description	Result
Y. Chen, T. Kunz (2016) [3]	Supports medical applications by keeping a record of patients.	• Captured data from medical sensors worn by patients are transferred to a central server. • Sensors: heart rate, blood oxygen, skin conductivity voltage, patient accelerometer/orientation • The patient gateway consists of eHealth sensors. The central server is Arduino Uno revision 3.	• System packet loss 0-25%: MQTT, DDS consumed less bandwidth. CoAP, Custom UDP had the same packet loss as the network. • Network latency 100-400ms, system packet loss 0-25%: MQTT had a much higher latency
M. I. Yamin, S. Kuswadi, S. Sukaridhoto (2018) [4]	Focuses on the integration between robot platform and IoT protocol.	• 3 layers of robot platform UNR-PF are designed. • Layer 1: Services provided by multi-robot. Layer 2: Graphical User Interface. Layer 3: Registered operator • Performance of IoT protocols was observed on Wifi-Mesh network with a laptop as a broker	• MQTT: Received data bytes same with any number of robots, high transfer rate, CoAP: received bytes lower for lower no. of robots. • MQTT suits better for robot platform
A. Larmo, A.Ratilainen, J. Saarinen (2018) [5]	Provides a study on 5G massive IoT realization over an NB-IoT system.	• Simulated scenario: 7 base stations with 3 sectors network of 21 hexagonal cells. • NB-IoT carrier deployed in 900 MHz carrier with base stations tx power of 40W, 2 receiver and transmitter antennas. MTC devises transmission power of 0.2W,1 Rx, and 1 Tx antenna. • devices communicating over IoT stacks using UDP and TCP on transfer layer over NB-IoT	• Throughput: CoAP shows a higher quantity. • Service Availability: 2 MTC request/s for MQTT drops from 95%. CoAP drops from 99%. • Coverage: CoAP service has better coverage

Table 2.6 (b): Summary of related work

Authors	Contribution	Experiment description	Result
M. I. Urkia, A. Orive, A. Urbieta (2017) [7]	Focuses on the comparison of CoAP implementations to make it helpful for constrained platforms and adjust the scalability.	• The deployed industrial prototype uses Raspberry Pi-s as a platform for gateways and Industry 4.0 scenarios. • Connected via Wi-Fi through a local 56Mbps router. • Assessed CoAP implementations: Libcoap, Californium, smcp, CoAPy, microcoap, FreeCoAP, node-coap, CoAPthon,	• Interoperability test: only CoAPy interoperable • Latency: Faster server- libcoap, smcp, microcoap. Faster client-node-coap, Californium, CoAPthon • CPU and RAM: faster-libcoap, smcp, microcoap, FreeCoAP, RAM consumption 3.3MB.
C. Gündoğan, P. Kietzmann, M. Lenders, H. Petersen, T. C. Schmidt, M. Wählisch, (2018) [8]	Application of IoT protocols in a single-hop and multi-hop scenario.	• Single-hop topology: 70 nodes within the same radio range, 2 arbitrary nodes chosen. Multi-hop topology: 350 nodes spread evenly in a building. 50 M3 nodes (low-end), one A8 node (gateway) are chosen. • Software: RIOT version 2018. Hardware: ARM Cortex-M3 MCU with Atmel AT86RF231 transceiver and Gateway runs on a Cortex-A8 node.	• Single hop: MQTT and CoAP have a higher delivery rate and the lowest energy consumption. • Multi-hop: NDN performs better
D.Thangavel, X. Ma, A. Valera, H. X. Tan, C. K. Y. Tan (2014) [9]	Proposed a common middleware for programming interface of IoT protocols, extended to adaptive network conditions.	• Hardware: broker-laptop, publisher-BeagleBoard-xM, netbook-subscriber. Software: A wide Area Network (WAN) emulator. • On BeagleBoardxM common middleware implementation was deployed and was connected to a layer-2 switch through Ethernet.	• Delay: 0-25% packet loss-MQTT is better, High packet loss-CoAP better. • Overhead: 0-100% packet loss- CoAP shows less overhead. • Data transfer: 0-25% packet loss: QoS 2 messages are with more bandwidth.

Table 2.6 (c): Summary of related work

Authors	Contribution	Experiment description	Result
L. Durkop, B. Czybik, J. Jasperneit e (2015) [10]	Implementation of IoT protocols over Cellular network standards EDGE, UMTS, and LTE in a laboratory environment	• Over cellular network emulator Anritsu MD8475A, assessment of different IoT protocols is executed for GSM (2G), UMTS (3G), and LTE (4G). • The data source is connected via a mobile router to the radio interface of the cellular network emulator. The data sink is connected to the Ethernet interface of emulator.	• Periodic spikes in latency for OPC UA and MQTT over EDGE and UMTS. • Latency of CoAP rises every 1024 bytes over LTE, EDGE, UMTS. • Protocols based on TCP are better.
J.Esquiag ola, L. Costa, P. Calcina, G. Fedreches , M. Zuffo (2017) [33]	IoT platform based on framework SwarmOS (Costa 2015), explores dynamic cooperation and peer-to-peer communication of devices.	• Java programming language, version 8 is used. • Jetty web server is used to host, and a desktop core i5 computer is used to run the client-side of Tsung. The client devices include Intel NUC, Intel Edison, Intel Galileo.	Response time increment: Intel Galileo- after 70 requests/second, Intel Edison- After 200 request/second, Intel NUC- After 1500 requests/second.
T. Moraes, B. Nogueira , V. Lira and E. Tavares (2019) [34]	Provides the behavior and outcome of IoT protocols at network failure conditions.	• Two experiments are carried, one without network failure and the other with network failure, where data is redirected to an alternative route. • Algorithm 1 is used to inject failure on a network route and algorithm 2 is to repair. • For the system's logical connections, the components have been mapped differently on each protocol.	• ANOVA analysis: 95% confidence intervals- CoAP has throughput 37.75B/s, mean value message size 61 bytes/package. • Tukey test: fault injection- AMQP retransmits with high throughput. MQTT and CoAP have the lowest losses, 0.48%, and 1%

2.4 Summary

The performances of IoT protocols have been studied from several perspectives over many years. The above chapter summarizes the key concepts of the related works and an understanding of the system components. Dinesh Thangavel [9] proposed a common middleware design and implementation in his paper that supports different IoT protocols and provides a common programming interface. Publish-Subscribe architecture is used to generate the common API calls. Common middleware implementation was deployed and executed on publisher BeagleBoardxM and was connected to a layer-2 switch through ethernet. A Wide Area Network emulator application called Wanem was run on the subscriber netbook to match a lossy network connection. With 0-25% packet loss the transmission time of MQTT was better and QoS 2 messages occupy more bandwidth than QoS 0 and QoS 1. With 0-100% packet loss CoAP had less overhead. The paper of L. Durkop [10] demonstrates that the 3rd and 4th generations of cellular networks are promising coordinators for embedding a range of different devices into the IoT. The principal component is the Anritsu MD8475A emulator for GSM (2G), UMTS (3G), and LTE (4G). For a payload of 2500 bytes over EDGE, the transmission time for OPC UA has observed 840 ms. Whereas for MQTT with Q0S class 0 is 1312ms and for CoAP is 1775 ms. If the transport block size is larger than the IP packet size sent by the mobile device, the LTE concatenates the IP packets till the transport block length is reached. This causes the increment of latency for OPC UA at 4000 bytes, MQTT at 8100 bytes and CoAP rises at every 1024 bytes. TCP-based protocols accomplished a better performance. These studies illustrate that IoT technology must be selected carefully to design a prototype and also it is very important to check the system's precision.

CHAPTER-3
ARCHITECTURE AND DESIGN OF THE SYSTEM

3.1 Introduction

This chapter contains the steps that were taken to develop the experimental setup and operation. The thesis involves the following major steps:

1. Proper hardware setup both in the ground node and UAV side.
2. Use of selected brokers for MQTT protocol and CoAP.
3. Implementation executed in the selected testing location.
4. Development of a new improved networking protocol.

These areas are covered in depth in this chapter with images and specifications to successfully describe the project.

3.2 Design and Architecture

The proposed system design consists of a sender, receiver, and a Linux environment. The transmitter side sensor is connected to MCU which includes an on-board Lora module to transmit data. The receiver side deployed on the UAV is represented by another MCU connected to a small Linux computer that has a server installed in it. The MCUs are programmed using Arduino IDE software. The following Diagram 3.1 shows the design layout, where the ground node transmits packets to the receiver side. Both CoAP and MQTT protocols are used separately for this transmission and the performances are observed.

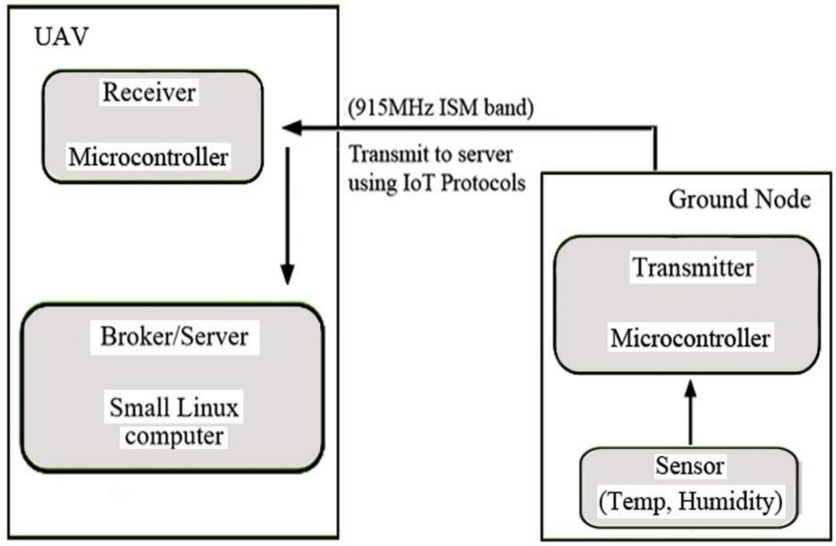

Figure 3.1: System Layout

3.3 Hardware Modules

This section describes the specifications of all the hardware used on both sender and receiver sides.

3.3.1 Adafruit Feather M0 RFM95 LoRa Radio

The microcontroller used is Adafruit Feather M0. The programming code is deployed using Arduino IDE. The specifications are shown in Table 3.1.

Table 3.1: Features of Adafruit Feather M0 [25]

Processor	ATSAMD21G18 ARM Cortex M0
Frequency	48 MHz and 3.3 V boot up voltage
Special chip includes	RFM95 LoRa radio module can be used for either 868MHz or 915MHz transmission/reception.

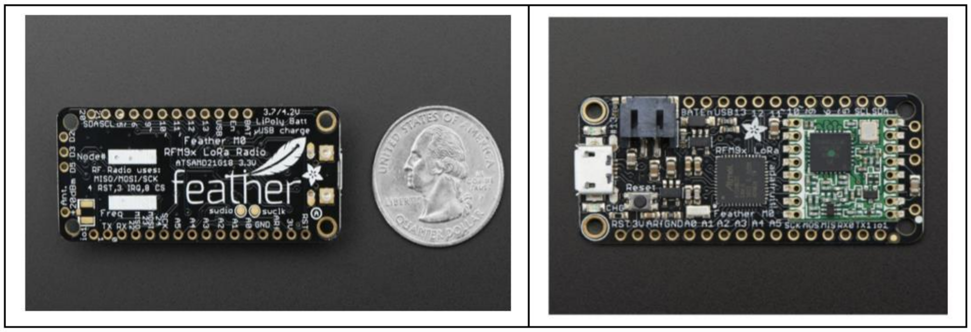

Figure 3.2: Adafruit Feather M0 RFM95LoRa Radio

Lora radios are not applicable for transmitting audio or video, but they work quite well for small data packet transmission.

3.3.2 DHT11 Sensor

The DHT11 sensor is used for sensing the temperature and humidity of the environment. The Arduino IDE requires a separate library to be installed for operating the sensor.

Table 3.2: Features of DHT11 sensor [40]

Operating voltage	3.5V to 5.5V
Operating current	0.3mA (measuring) 60uA (standby) with serial data output
Pin configuration	3 pins of power supply (Vcc), serial data output, and GND.
Accuracy	±1°C and ±1%

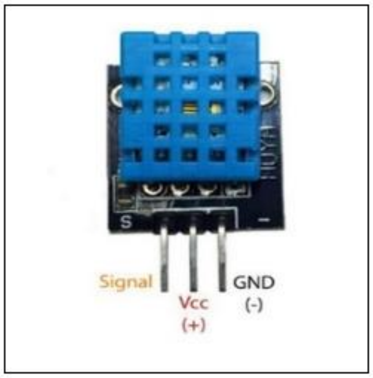

Figure 3.3: DHT11 Sensor

3.3.3 Lora Antenna Kit

For a longer ranger, the Lora antenna kit is used with the Lora module of the MCU. It is useful with some other wireless modules, such as BLE boards, WiFi, SiPy Sigfox, etc. The main specifications are shown in Table 3.3.

Table 3.3: Features of Lora Antenna [25]

Type of antenna	External
Including parts	uFL to RP-SMA antenna adapter cable and RP-SMA
Frequency	900MHz

Figure 3.4: Lora Antenna

3.3.4 Raspberry Pi

The Raspberry Pi is a nano Single Board Computer (SBC), which allows the execution of several variations of the GNU / Linux free operating system, particularly Debian, Raspbian, and compatible software as well as it also works with the Microsoft Windows operating system. The Raspberry Pi 3 Model B+ used in this experiment is the final revision in the Raspberry Pi 3 range. The main features are shown in Table 3.4.

Table 3.4: Features of Raspberry Pi 3 Model B+ [26]

Processor	Broadcom-BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC, 1.4 GHz
Memory	1 GB
Connectivity	USB, Ethernet, WiFi, HDMI, RCA, Bluetooth, PoE

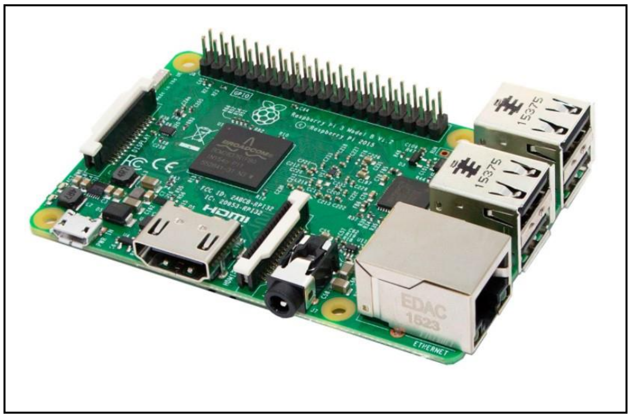

Figure 3.5: Raspberry Pi Model 3B+

3.4 Software Environments

This section describes all the software environments used in the prototype.

3.4.1 Software of Operating Instruments

Table 3.5 shows the software portion of the operating instruments.

Table 3.5: Software Environments of Instruments

Environment Name	**Installed Packages**	**Operating Instruments**
Arduino IDE	Arduino SAMD Boards (32-bits ARM Cortex-M0+) version 1.8.3	Adafruit Feather M0
	Adafruit SAMD Boards version 1.5.3	Adafruit Feather M0
	DHT sensor library	DHT11
	Radiohead library	Adafruit Feather M0
OS	NOOBS (offline and network install)	Raspberry Pi 3 B+

3.4.2 Broker for IoT Protocols

Mosquitto is an open-source message server that implements MQTT protocols [41]. Due to its documentation, good community support, and ease of installation, this has become one of the most popular MQTT brokers. It is suitable for use on all devices from low-power single-board computers to full servers.

Libcoap is used for setting up the environment for CoAP transmissions. Libcoap is a C implementation of a lightweight application protocol for devices that are simply constrained to their resources. It is designed to run on the embedded devices along with high-end computer systems with POSIX OS [42].

Figure 3.6(a) portrays the functionality of Mosquitto and Libcoap, as MQTT is many-to-many and CoAP is a one-to-one communication protocol. Figure 3.6(b) shows our obtained data at both brokers while data transmission.

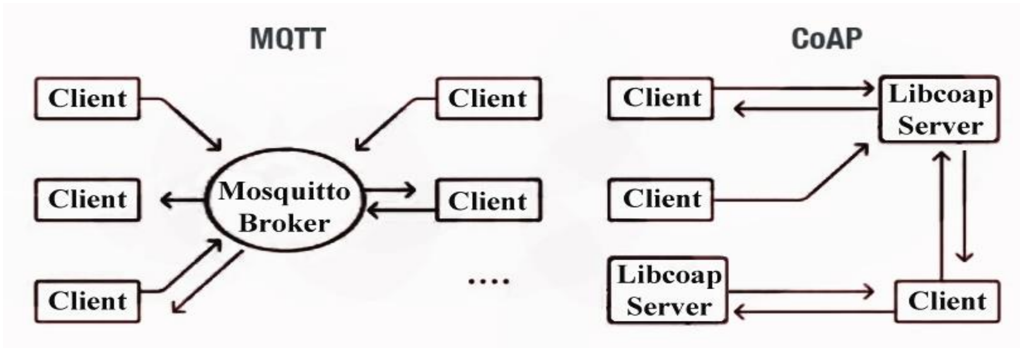

Figure 3.6(a): Functionality of Brokers

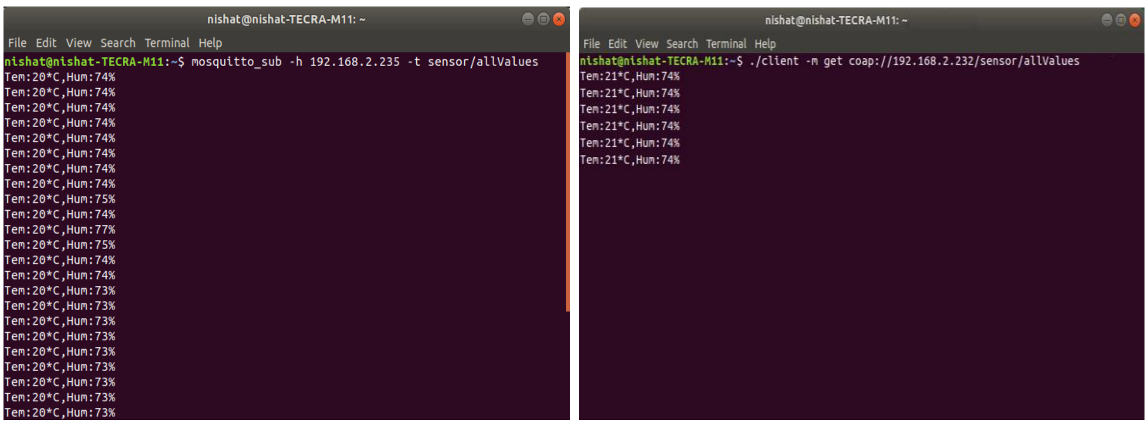

Figure 3.6(b): Brokers: (A) Mosquitto for MQTT (B) Libcoap for CoAP

3.5 Transmitter Side (Tx)

The transmitter side is the ground node with a DHT11 sensor, which is connected via serial port to an Adafruit Feather M0 MCU with an on-board RFM95W LoRa radio module, which is powered by a battery as a power source. The module is attached with a 900 MHz antenna and placed about 3 meters high above the ground for better transmission. The temperature and humidity data received from the sensor are prepared for transmission on the Feather MCU, then it transmits.

Figure 3.7: Transmitter Side (Tx)

3.6 Receiver Side (Rx)

The sensor data from the transmitter side is received by another MCU with a LoRa radio module and is attached with a 900 MHz antenna for better receiving. It is connected to Raspberry Pi (small Linux System PC) with a server in it and powered by a battery. In the case of MQTT protocol, the Mosquitto is installed as the broker on the PC to store the received sensor data. For CoAP transmission, Libcoap is installed.

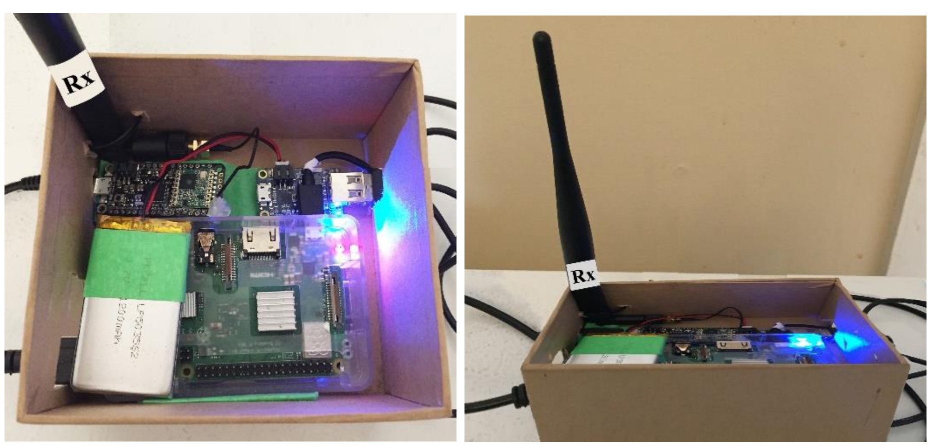

Figure 3.8: Receiver Side (Rx)

3.7 Setup with Receiver Side

A lightweight industrial drone was used for the UAV setup. All hardware modules for the receiver device are mounted on the drone, which includes, MCU, Raspberry Pi, and Lora antenna. A Lithium-ion polymer rechargeable battery with a power booster was also fitted on the unit as a power source to power the whole unit up.

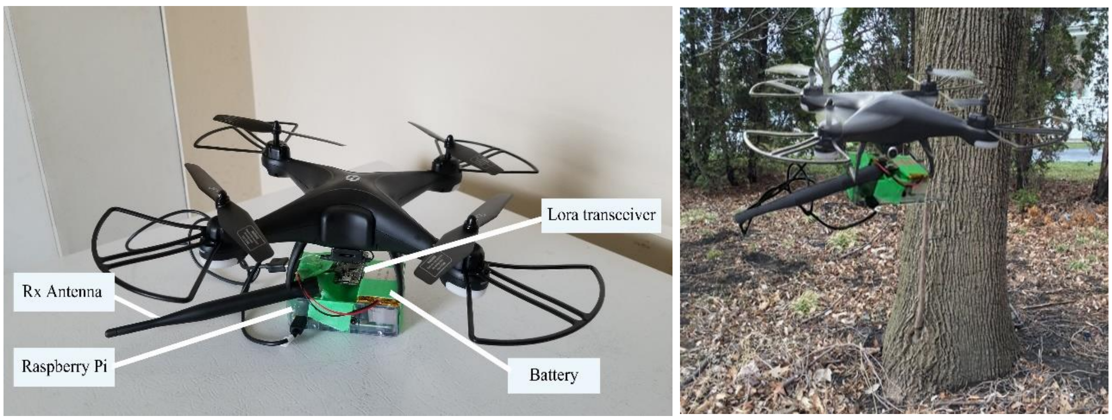

Figure 3.9: UAV Setup

The drone was tested in advance to check if it can manage all the extra weight included by all modules and antennas. The unit operated perfectly and was able to fly appropriately.

3.8 Testing Environments

The experiment was conducted in Malden Park near Malden hill, Windsor, Ontario. This 175-acre park located on the west side of Windsor features the highest hill in Essex County. The area consists of open space as well as wild trails and forests. The transmitter was placed near the parking lot of the park as shown in Figure 3.10 and the receiver was in different places depending on testing scenarios. This location was suitable for flying the drone properly with the mounted receiver mote. Data transmission for both CoAP and MQTT are performed, and real-time measurements were collected for each protocol.

Figure 3.10: Malden Park-West Side of Windsor

3.9 Developing new Customized Networking Protocol

After analyzing the two protocols MQTT and CoAP and comparing their performances an attempt has been taken to establish a new customized networking protocol.

It follows the Request-Response model for data transmission and uses UDP at the transport layer. When the server shows that is it available to the client, it sends all the data packets gradually with a low response time.

First, the server does not establish a connection with the client. Instead, the server just sends a datagram to show its availability to the client, using the sendto() function which requires the IP address of the destination/client as a parameter. Similarly, the client does not accept a connection from a server. Instead, the client just calls the recvfrom() function, which waits until data arrives from some server. Then it returns data packets to the server IP address. This is how the client can send a response to the server.

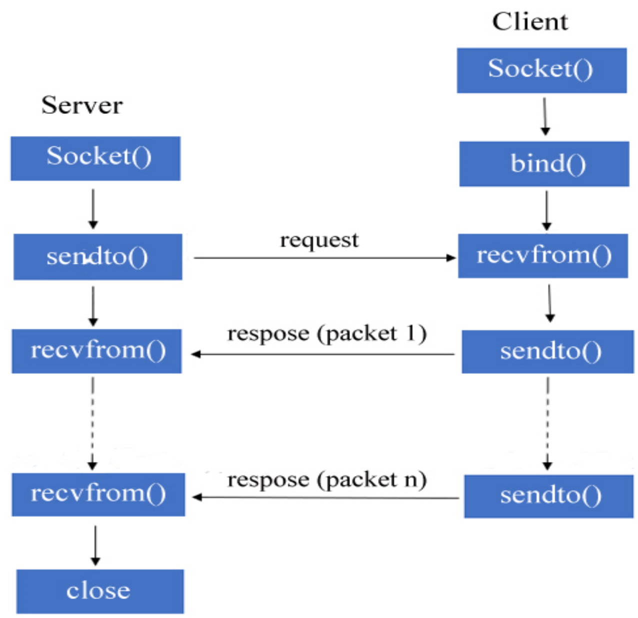

Figure 3.11: Flowchart of communication of Customized Networking Protocol

Figure 3.11 shows the steps of establishing the communication between the server and client-side. On the server-side:

- Creating a socket using the socket() function;
- Sending its availability with sendto() function.

The steps of establishing communication on the client side are as follows:

- Creating a socket with the socket() function;
- Binding the socket to an address of the socket;
- Receiving data and sending packets by means of recvfrom() and sendto() functions.

In the next chapter, the analysis of this new protocol comparing with MQTT and CoAP is discussed.

3.10 Summary

The proposed system design and methodology are explained in detail throughout this chapter for the experiment including all the hardware devices and software environments. The brokers used for IoT protocols and hardware setup is discussed thoroughly. Also, the testing environments and sites where the tests took place were briefly illustrated. Finally, the basic feature for the new customized network protocol design is explained in this chapter. So, the chapter provides a thorough understanding and platform for the experiment to be conducted.

CHAPTER-4
EXPERIMENT AND RESULT ANALYSIS

4.1 Introduction

Chapter 3 explained the design and system methods along with the main characteristics and test scenarios. In this chapter, the trials, measurements, and results are going to be discussed. This chapter undertakes observation of the performance and examines how differently the two protocols MQTT and CoAP react. To do so, the testing must be taken place in the testing environment mentioned in the previous chapter. Also, the two existing protocols will be compared with the newly developed networking protocol. In the experiment to analyze some parameters of the protocols, Wireshark is installed on the server-side. Wireshark is an open-source packet analyzer and provides network interface controllers where users can see all traffic in progress. It provides the information and different aspects of each data sample. It illustrates the length of various segments of data, delta time using a timestamp, and any lost packet as well as a retransmission. Microsoft Excel is used to record, calculate, and examine the data transmission captured for comparing different parameters and later used to draw graphs.

4.2 Parameters of Performance

The testing scenario includes the Packet Length, Latency, Packet Delivery Rate vs Distance, and Bandwidth Consumption for both protocols.

The transmitted data samples were analyzed for both protocols. Three types of data packets were sent, they are, overall sensor data, temperature data, and humidity data.

Overall sensor data contains both temperature and humidity values altogether. If a client wants both information, it is less time-consuming to be subscribed to a single topic, rather than requesting the sensor values separately.

4.2.1 Overhead of the Protocols

It is obtained from the analysis of the captured data that MQTT protocol is providing fixed bytes of MAC Header, IP Header, TCP Header, and a variable byte of payload in its data packet segments. On the other hand, CoAP is providing the fixed bytes of MAC Header, IP Header, UDP Header, and a variable byte of the payload.

Table 4.1: Overhead of data received for MQTT

MAC Header (bytes)	IP Header (bytes)	TCP Header (bytes)	Payload (bytes)
14	20	20	0-268,435,456

Table 4.2: Overhead of data received for CoAP

MAC Header (bytes)	IP Header (bytes)	UDP Header (bytes)	Payload (bytes)
14	20	8-16	0-65535

Tables 4.1 and 4.2 show the obtained values of headers in bytes for MQTT and CoAP respectively. The following graph shows the comparison of the packet length of both protocols.

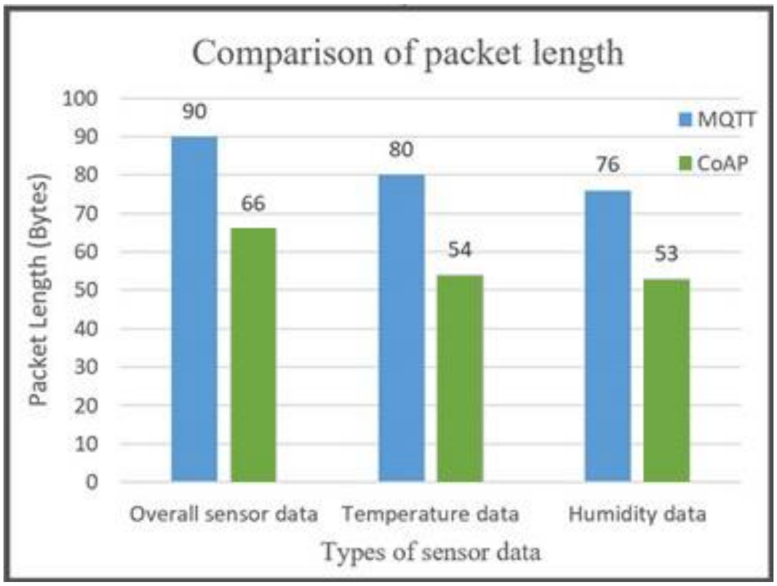

Figure 4.1: Packet Length Comparison for Both Protocols

Figure 4.1 illustrates the packet lengths for CoAP in three cases, which are 66 bytes, 54 bytes, and 53 bytes respectively, and they are much shorter than the packet length of MQTT transmission, which are 90bytes, 80bytes, and 76 bytes. Since CoAP runs over a less complex UDP transport protocol instead of TCP, provides an ability to reduce overhead.

4.2.2 Latency

Latency is the response time or transmission time. For each protocol data packets of three different lengths are transmitted. The latency is based on the time required to publish the data to the broker/server. To observe the time since the previous frame transmitted, applying timestamp is useful, therefore, it is possible to calculate the latency of every transmission that occurred.

More than 5,000 samples of data transmission for each protocol are taken, recorded, and examined. In Microsoft excel the data is logged and a graph of latency is drawn for MQTT and CoAP.

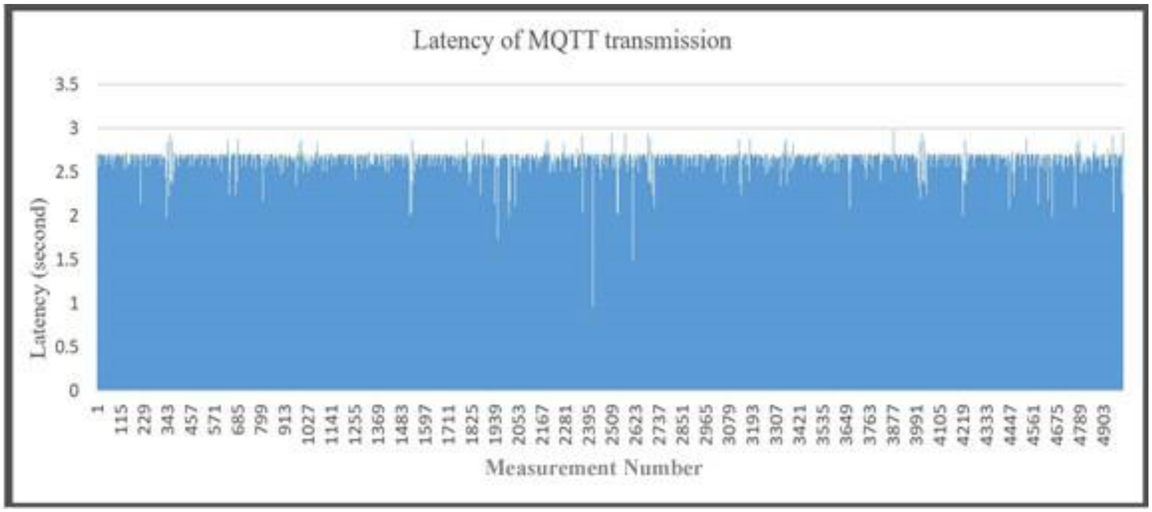

Figure 4.2: Latency of MQTT Data Transmission

Figure 4.2 demonstrates the data packet transmission for the MQTT protocol. The samples with higher latency are found mostly within the range from 2.5sec to 2.8sec.

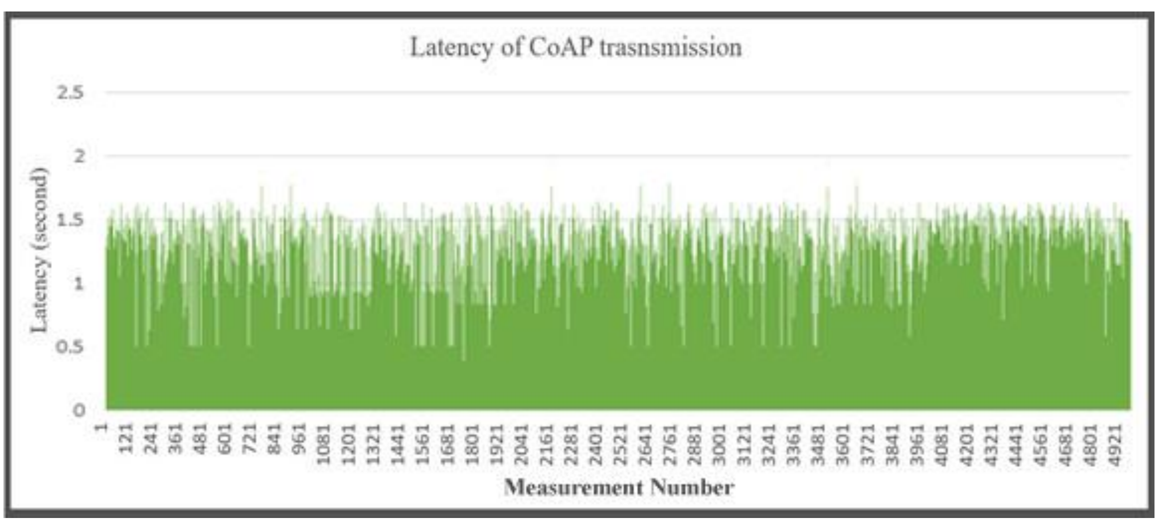

Figure 4.3: Latency of CoAP Data Transmission

Figure 4.3 demonstrates the 5000 samples of data transmission for CoAP. The samples with higher latency are detected within the range 1.6sec 1.4sec.

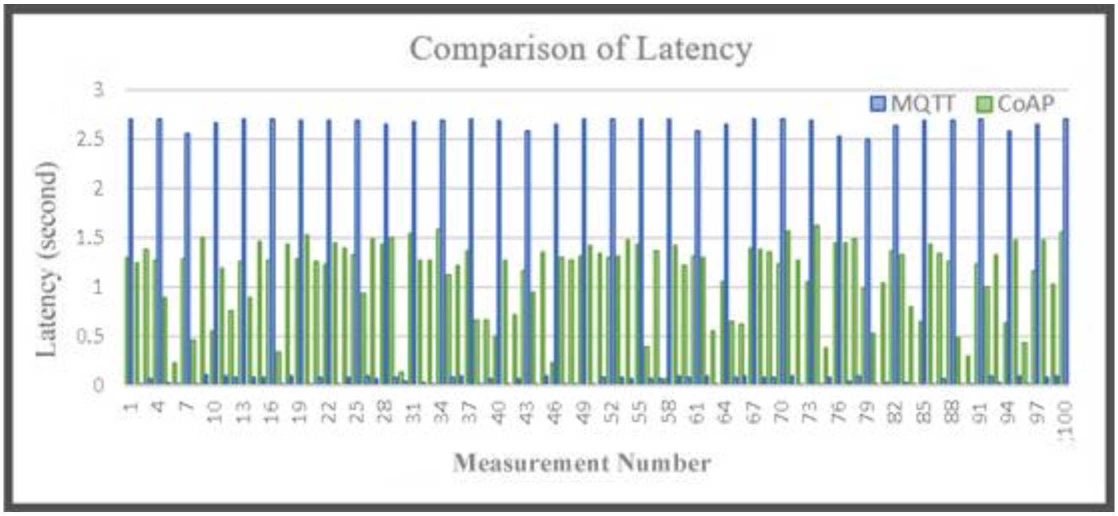

Figure 4.4: Comparison of Latency

Now, if a small portion is considered from the latency graph of both protocols, Figure 4.4 reflects that the fluctuation of transmission time for CoAP is within 0.8 sec to 1.6sec, which is not very significant. But MQTT shows a different scenario with a higher deviation in latency, which is called spikes. This is due to the feature of the sliding window of TCP. For MQTT transmission, the window size is reached when the data packets of 3 topics have been transmitted. Then it is forced to stop sending and the new window starts with a delay, causing the first packet of the next window to have a higher transmission time.

4.2.3 Packet Delivery Rate (PDR) vs Distance

MQTT protocol uses TCP. Due to that, an acknowledgment is being received after data transmission. But the broker/server will not send an acknowledgment message to the sender if the data is lost and not received. The feature of Quality of Service (QoS) 1 of the MQTT protocol causes a retransmission. Unlike MQTT protocol, the CoAP runs on UDP. As a result, there will be no acknowledgment message sent after the packet is being received. It is observable that the confirmable GET request messages are not responded to if data packets are lost. Analysis of the captured data packet transmission shows how many packets were lost while transmission. To determine PDR, it is possible to calculate the total number of successfully received packets over total transmitted packets. The formula for PDR is as follows,

$$PDR (\%) = (R / T) * 100$$

T = Total number of transmitted packets

R = Total number of successfully received packets.

In this test, PDR was examined for 100-meter, 200-meter, 300-meter, 400-meter, 500-meter, and 600-meter distances between the sender and the receiver. Both MQTT and CoAP were implemented separately.

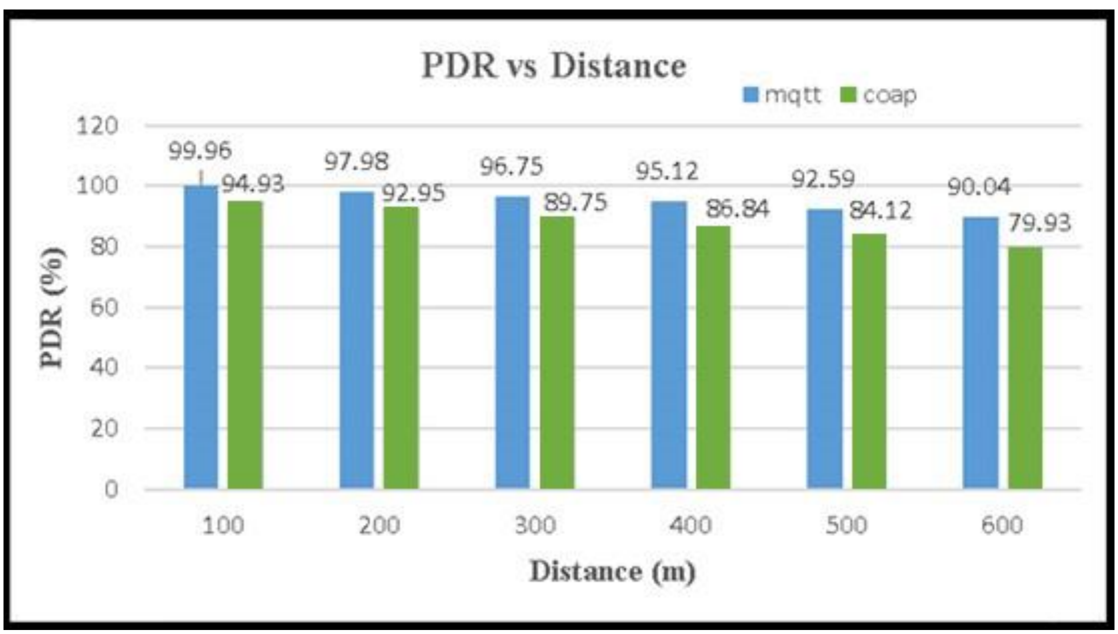

Figure 4.5: Packet Delivery Rate (%) vs Distance (m)

Figure 4.5 shows the calculated PDR for both protocols. From the graph, the readings of MQTT are quite impressive as it stays above 96%, even for 300 meters distance between TX and RX. But for 600 meters it is just reaching 90%. On the other hand, for CoAP, the PDR falls below 90% starting from a 300m distance. Also, for 600 meters it drops below 80%.

An explanation for this situation is that the TCP is a connection-oriented protocol with an extensive error checking mechanism, Sequencing of data, flow control, and acknowledgment of data received. So MQTT is more likely to have a higher number of successfully received data packets. On the other hand, CoAP uses UDP, which goes not guarantee the delivery of data.

4.2.4 Bandwidth vs Number of Packets Transmitted

Figure 4.6 shows the bandwidth consumption measurements with the increasing number of packets.

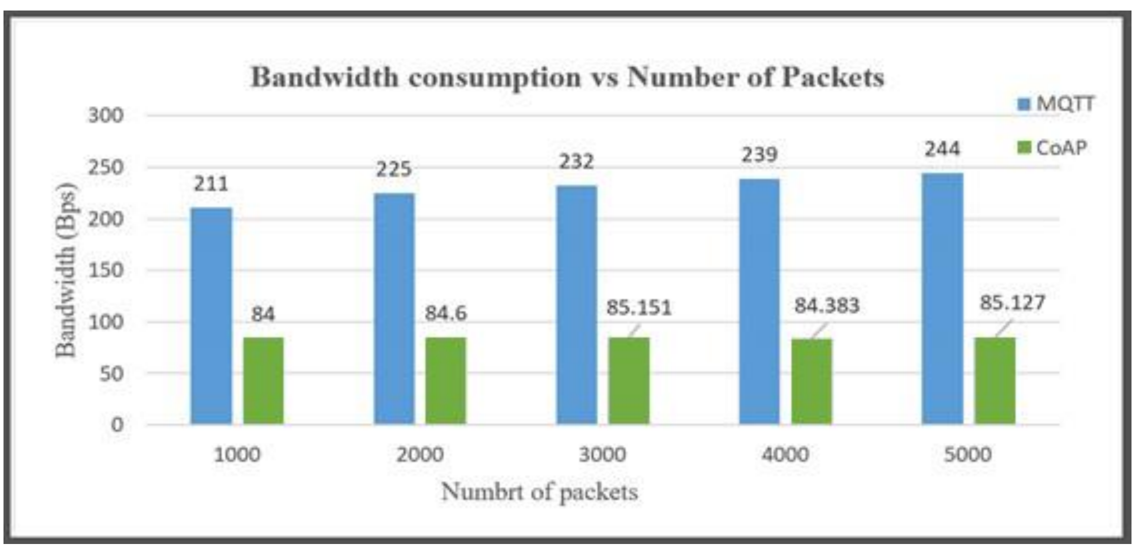

Figure 4.6: Bandwidth Consumption vs Number of Packets Transmitted

In this test, bandwidth consumption is examined for 1000, 2000, 3000, 4000, and 5000 packets transmission for each protocol individually.

In contrast to CoAP, the average bandwidth consumption of MQTT is higher. It is always above 210 Bits/s for all the number of packets transmitted. CoAP does not consume additional bandwidth because it has a smaller packet size than MQTT and involves no retransmission feature. In the case of MQTT, due to the nature of TCP handshakes, acknowledgment and keep alive features it uses far more bandwidth.

4.3 Customized Networking Protocol (CNP) Analysis

In the previous chapter, the architecture of the Customized Networking Protocol has been discussed. Now packet length and transmission time are observed for the new protocol.

4.3.1 Packet Length and Transmission Time

The packets length of Customized Networking Protocol is shown in Figure 4.7 as well as the comparison of the obtained sizes with the other two protocols, MQTT and CoAP.

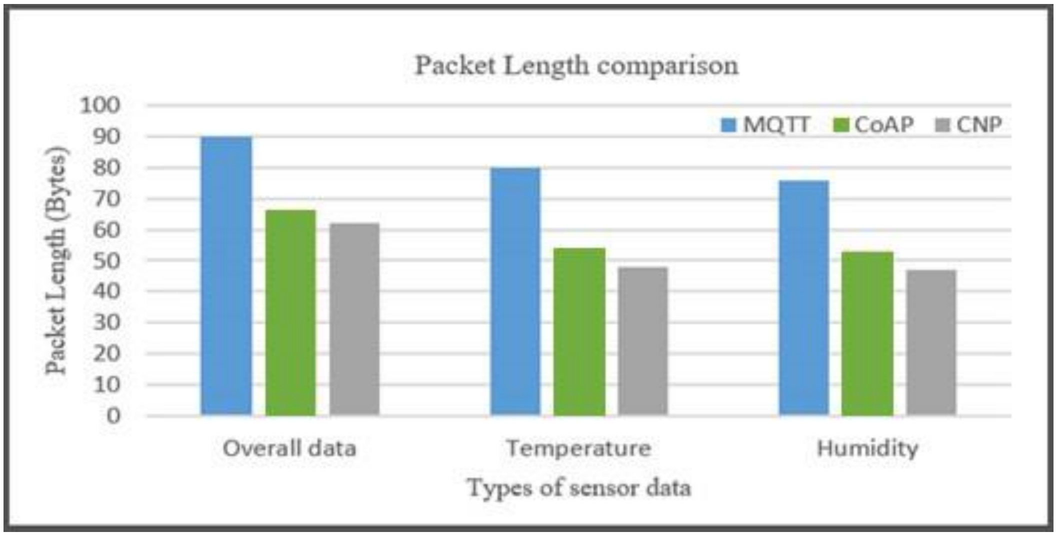

Figure 4.7: Packet Length of 3 Protocols

Figure 4.8 shows the graph of 500 samples of data transmission for Customized Networking Protocol (CNP).

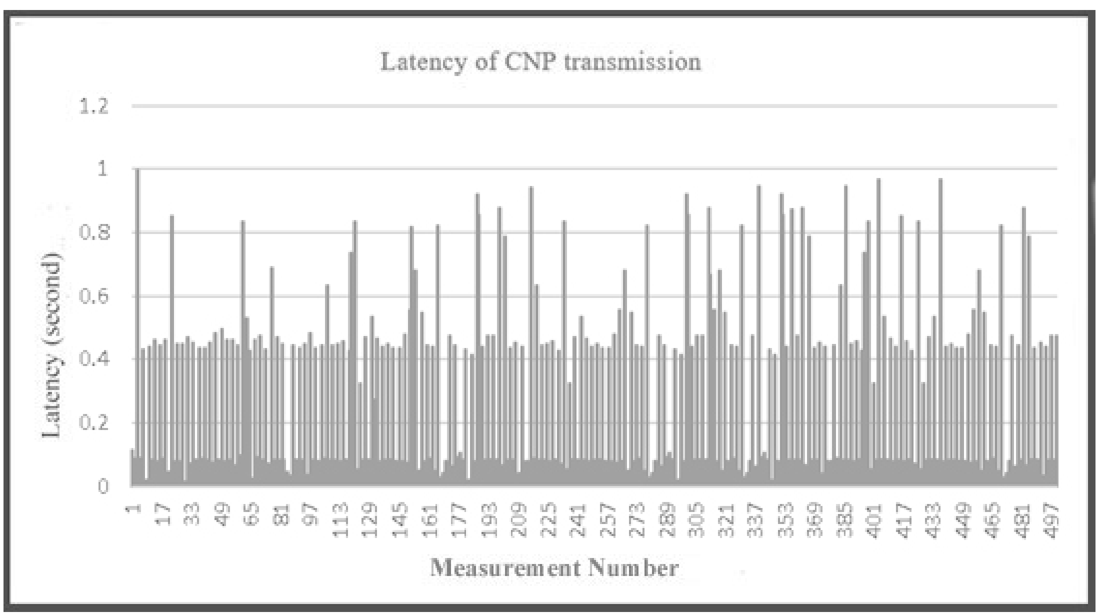

Figure 4.8: Transmission Time of CNP

The average transmission time of each packet of CPN is calculated as well as for MQTT and CoAP. Figure 4.9 shows the comparison.

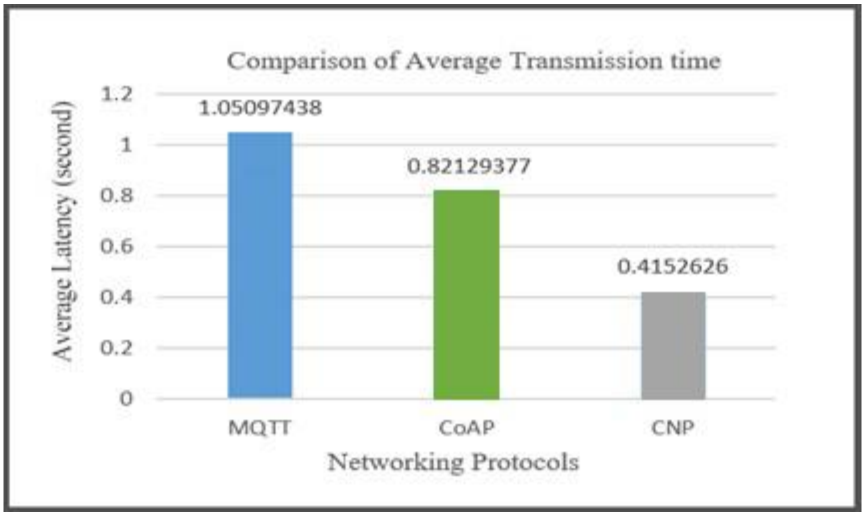

Figure 4.9: Average Transmission Time of Packets for 3 Protocols

4.3.2 Protocol Analysis and Discussion

Table 4.3 shows the analysis for the new protocol with the previous two protocols.

Table 4.3: Analysis of the 3 protocols

Description	**MQTT**	**CoAP**	**CNP**
Packet length	Higher than others	Lower length	Similar to CoAP
Average latency	Higher than others	Medium	Lower than others

Discussion of the obtained outcome:

- As because MQTT uses TCP in its transport layer, it has a higher length of headers, therefore higher in the length of packets. But CoAP has the advantage of using UDP, which is less complex with a smaller size header. The newly designed protocol CNP also uses UDP, so it has smaller packets than MQTT and is similar to CoAP.

- As discussed earlier that the TCP is a connection-oriented protocol with an extensive error checking mechanism, it requires acknowledgment and has the feature of the sliding window, so MQTT has higher latency than CoAP, which does not require establishing a connection. Rather CoAP uses the request-response method to transmit the packets, therefore it is quicker than MQTT.
- But in the case of CoAP, it receives a request for each packet from the server, and then it starts transmitting. Also, for sending each requested packet, every time CoAP needs to decide different routes inside the network from sender to receiver. But the new protocol CNP sends all the packets gradually whenever the server sends its availability to the client and also, using the same route inside the network for sending all the packets. So, it is faster than CoAP.
- In this case, CNP is acting similar to MQTT, that it sends the packets continuously one after another, but as it does not wait for any acknowledgment after every packet being sent like MQTT, it is much faster than MQTT as well.

4.4 Summary

The communication scheme prototype shows different aspects of the two IoT protocols, MQTT and CoAP. The performance investigation provides accurate measurements for the selected outdoor testing site. Data packets of MQTT transmission with higher overhead causing the packet length to become larger than CoAP transmission. But MQTT provides a better packet delivery rate. CoAP is ahead of MQTT by consuming less bandwidth. The improved new protocol provides better packet length and latency than the existing protocols. More future work needs to be applied to the new protocol. The tests offer a better understanding of the performance of the IoT protocols.

CHAPTER-5
CONCLUSION

5.1 Conclusion

The focus of this thesis is to minimize the damages from natural disasters like wildfires with a feasible and robust procedure. So far there have been many attempts to develop a low-cost method for observing the conditions of the high-risk zones. This thesis aims to integrate the new technology of IoT with the consumption of MQTT and CoAP. A functional and consistent communication scheme prototype for environmental monitoring system with the involvement of a UAV was deployed and IoT communication protocols were applied and tested with the experimental setup. The two nodes of the arrangement were able to communicate with each other using the most widely used IoT protocols, MQTT and CoAP. The protocols allow the proposed prototype to consume very little power and it requires a quite simple power source like a battery. The two nodes communicate through special circumstances. For which a testbed was developed in a real site to be able to compute all the applied measurements to evaluate the performance of the used protocols. A specific number of packets was sent from the ground node and the successfully were collected, stored on the server, and monitored at the UAV side.

The IoT protocols preserve their features which lead them to provide different results. MQTT contains a higher length of headers and TCP handshakes, forcing this to have a higher length of data packets and bandwidth consumption. CoAP has its feature of connectionless communication policy causing it to have a decent average latency of 0.8212 seconds. But the connection-oriented MQTT protocol has a magnificent Packet Delivery Rate of over 96.75% even for a 300-meter distance between two nodes. The outcomes of the experiment collectively prove that the IoT protocols are performing appropriately with the experimental setup of the UAV monitoring system.

5.2 Future Works

In this experiment, a new networking protocol has been proposed and tested. The new Customized Networking Protocol (CNP) demonstrates some better execution than the existing protocols MQTT and CoAP. But it requires more analysis and performance checking. Also, more features could be added to the proposed experimental design. For further studies and future developments, the ideas of improvements are described below:

1. Adding more ground nodes to the system to cover a wide range of areas for monitoring. As the nodes would provide the specific address of the location with each packet transmitted to the drone, it will be possible to realize the affected area.
2. Adding more digital and analog sensors to the ground node to measure the verity of air quality information.
3. Introducing some other IoT protocols to this experiment to compare among the different protocols.
4. To make the system user-friendly a GUI might be developed.
5. Testing some other performance for the new networking protocol developed, such as PDR, bandwidth consumption, etc.
6. Some more security features might be introduced to the newly proposed protocol.

BIBLIOGRAPHY

[1] Wikipedia, *Air pollution*, Accessed 17-07-2021 at https://en.wikipedia.org/wiki/Air_pollution#Natural_sources, 2021.

[2] Stoodnt, *IoT Statistics for 2020*, Accessed 17-07-2021 at https://www.stoodnt.com/blog/iot-statistics-for-2020/, 2020.

[3] Y. Chen, T. Kunz, "Performance evaluation of IoT protocols under a constrained wireless access network," in *2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*, Cairo, Egypt, 2016, pp. 1-7.

[4] M. I. Yamin, S. Kuswadi, S. Sukaridhoto, "Real Performance Evaluation on MQTT and COAP Protocol in Ubiquitous Network Robot Platform (UNR-PF) for Disaster Multi-robot Communication," *EMITTER International Journal of Engineering Technology*, vol. 6, pp. 369-385, Dec 2018.

[5] A. Larmo, A. Ratilainen, J. Saarinen, "Impact of CoAP and MQTT on NB-IoT System Performance," in *Global Internet of Things Summit (GIoTS)*, Bilbao, Spain, 2018, pp. 4–7.

[6] D. B. Ansari, A. U. Rehman, R. A. Mughal, "Internet of Things (IoT) Protocols: A Brief Exploration of MQTT and CoAP," *International Journal of Computer Applications*, vol. 179, pp. 9-14, Mar 2018.

[7] M. I. Urkia, A. Orive, A. Urbieta, "Analysis of CoAP Implementations for Industrial Internet of Things: A Survey," in *7th International Conference on Sustainable Energy Information Technology*, Madeira, Portugal, 2017, pp. 188-195.

[8] C. Gündoğan, P. Kietzmann, M. Lenders, H. Petersen, T. C. Schmidt, M. Wählisch, "NDN, CoAP, and MQTT: A Comparative Measurement Study in the IoT," in *Proceedings of the 5th ACM Conference on Information-Centric Networking*, Boston, MA, USA, 2018, pp. 159-171.

[9] D. Thangavel, X. Ma, A. Valera, H. X. Tan, C. K. Tan, "Performance Evaluation of MQTT and CoAP via a Common Middleware," in *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Singapore, 2014, pp. 1-6.

[10] L. Durkop, B. Czybik, J. Jasperneite, "Performance Evaluation of M2M Protocols Over Cellular Networks in a Lab Environment," in *2015 18th International Conference on Intelligence in Next Generation Networks*, Paris, France, 2015, pp 17-19.

[11] Global News, *2 huge wildfires in B.C. this summer caused more than @127 million in damage*, Accessed 17-07-2021 at https://globalnews.ca/news/3771758/2-huge-wildfires-in-b-c-this-summer-caused-more-than-127-million-in-damage/, 2017.

[12] CBC News, *Fort McMurray wildfire costliest insured disaster in Canadian history*, Accessed 17-07-2021 at http://www.cbc.ca/news/canada/edmonton/fortmcmurray-wildfire-costliest-insured-disaster-in-canadian-history-at-nearly-3-6b-1.3668602, 2020.

[13] Wikipedia, *2019 California wildfires*, Accessed 17-07-2021 at https://en.wikipedia.org/wiki/2019_California_wildfires, 2021.

[14] AVSYSTEM, *IoT Standards and Protocols Guide—Protocols of the Internet of Things*, Accessed 17-07-2021 at https://www.avsystem.com/blog/iot-protocols-and-standards/, 2020.

[15] National Geographic, *Climate 101: Wildfires*, Accessed 18-07-2921 at https://www.nationalgeographic.com/environment/natural-disasters/wildfires/, 2019.

[16] Conservation in a Changing Climate, *Increasing Wildfires*, Accessed 17-07-2021 at https://climatechange.lta.org/climate-impacts/increasing-wildfires/, 2021.

[17] iPoint, *Wireless Networking (Wi-Fi) – Advantages and Disadvantages to wireless networking*, Accessed 18-07-2021 at https://ipoint-tech.com/wireless-networking-wi-fi-advantages-and-disadvantages-to-wireless-networking/, 2021.

[18] Techwalla, *The Disadvantages of Bluetooth Technology*, Accessed 18-07-2021 at https://www.techwalla.com/articles/the-disadvantages-of-bluetooth-technology, 2021

[19] RF Wireless World, *Advantages of Cellular Network | disadvantages of Cellular Network*, Accessed 18-07-2021 at https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Cellular-Network.html#:~:text=Drawbacks%20or%20disadvantages%20of%20Cellular,affected%20by%20multipath%20signal%20loss., 2021.

[20] Wikipedia, *Internet of Things*, Accessed 18-07-2021 at https://en.wikipedia.org/wiki/Internet_of_things, 2021.

[21] Appinventiv, *8 Statistics That Prove IoT will become Massive from 2018*, Accessed 17-07-2021 at https://appinventiv.com/blog/8-statistics-prove-iot-will-become-massive-2018/, 2020.

[22] Sam Solutions, *Internet of Things (IoT) Protocols and Connectivity Options: An Overview*, Accessed 17-07-2021 at https://www.sam-solutions.com/blog/internet-of-things-iot-protocols-and-connectivity-options-an-overview/, 2018.

[23] Kellton Tech, *Untangling the potentials of IoT Protocols and standards*, Accessed 17-07-2021 at https://www.kelltontech.com/kellton-tech-blog/internet-of-things-protocols-standards, 2019.

[24] Eclipse Foundation, *MQTT and CoAP, IoT Protocols*, Accessed 18-07-2021 at https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php, 2014.

[25] Adafruit learning system, *Adafruit Feather M0 Radio with LoRa Radio Module*, Accessed 17-07-2021 at https://cdn-learn.adafruit.com/downloads/pdf/adafruit-feather-

m0-radio-with-lora-radio-module.pdf, 2021.

[26] Raspberry Pi Foundation, *Raspberry Pi 3 Model B+*, Accessed 17-07-2021 at https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/, 2021.

[27] IBM Developer, *Getting to know MQTT*, Accessed 18-07-2021 at https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/, 2020.

[28] M. Anusha, E. S. Babu, L. S. M. Reddy, A. V. Krishna, "Performance Analysis of Data Protocols of Internet of Things: A Qualitative Review," *International Journal of Pure and Applied Mathematics*, vol 115, page 37-46, 2017.

[29] IoT by HVM, *COAP vs MQTT | Difference between COAP and MQTT protocols*, Accessed 18-07-2021 at https://iotbyhvm.ooo/coap-vs-mqtt/, 2019.

[30] Postscapes, *IoT Standards and Protocols*, Accessed 17-07-2021 at https://www.postscapes.com/internet-of-things-protocols/, 2020.

[31] Security Compass, *XMPP: Swiss Army Knife for Internet of Things (IoT)*, Accessed 18-07-2021 at https://resources.securitycompass.com/blog/xmpp-swiss-army-knife-for-internet-of-things-iot-2, 2016.

[32] IBM Developer, *Understanding quality of service for Web services*, Accessed 18-07-2021 at https://www.ibm.com/developerworks/library/ws-quality/index.html, 2021.

[33] J. Esquiagola, L. Costa, P. Calcina, G. Fedrecheski, M. Zuffo, "Performance Testing of an Internet of Things Platforms," in *2nd International Conference on Internet of Things, Big Data and Security*, Porto, Portugal, 2017, pp. 309-314.

[34] T. Moraes, B. Nogueira, V. Lira, E. Tavares, "Performance Comparison of IoT

Communication Protocols," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC),* Bari, Italy, 2019, pp. 3249-3254.

[35] F. Ouakasse, S. Rakrak, "A Comparative Study Application Layer Protocols via. Performances Evaluation," *Journal of Engineering and Applied Sciences*, vol 15, page 6053-6061, Sep 2018.

[36] Study tonight, *Comparison of OSI and TCP/IP Reference Model*, Accessed 17-08-2021 at https://www.studytonight.com/computer-networks/comparison-osi-tcp-model, 2021.

[37] BytesofGigabytes, *MQTT Packet Structure,* Accessed 17-07-2021 at https://bytesofgigabytes.com/mqtt/mqtt-protocol-packet-structure/, 2019.

[38] Wikipedia, *Constrained Application Protocol*, Accessed 18-07-2021 at https://en.wikipedia.org/wiki/Constrained_Application_Protocol, 2021.

[39] CoRE Working Group, *Constrained Application Protocol (CoAP),* Accessed 17-07-2021 at https://tools.ietf.org/id/draft-ietf-core-coap-09.html#:~:text=CoAP%20defines%20four%20types%20of,to%20the%20request%2Fresponse%20interactions., 2012.

[40] Components101, *DHT11–Temperature and Humidity Sensor*, Accessed 17-07-2021 at https://components101.com/sensors/dht11-temperature-sensor, 2018.

[41] Mosquitto, *Eclipse Mosquitto,* Accessed 18-07-2021 at https://mosquitto.org/, 2021.

[42] Libcoap, *C-Implementation of CoAP*, Accessed 18-07-2021 at https://libcoap.net/, 2021.

VITA AUCTORIS

NAME:	Nishat Sultana
PLACE OF BIRTH:	Dhaka, Bangladesh
YEAR OF BIRTH:	1990
EDUCATION:	American International University Bangladesh, B.Sc., Dhaka, Bangladesh, 2013
	University of Windsor, M.A.Sc., Windsor, ON, 2021