การปรับเปลี่ยนกลไกการทำงานของโพรโทคอลเอ็มคิวทีทีเพื่อค้นหาเส้นทางสำรอง

MODIFICATION OF MQTT PROTOCOL TO FIND ALTERNATE ROUTE

วิทยานิพนธ์นี้สำหรับการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2560
KMITL-2017-EN-M-070-021

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การปรับเปลี่ยนกลไกการทำงานของโพรโทคอลเอ็มคิวทีทีเพื่อค้นหาเส้นทางสำรอง

MODIFICATION OF MQTT PROTOCOL TO FIND ALTERNATE ROUTE

ชลทวี วงศ์พร้อมรัตน์

CHONTAWEE WONGPROMRAT

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สรพม.
เลขทะเบียน 148792
วัน.เดือน.ปี. 23 พ.ย. 2560

b.00266914
.....

วิทยานิพนธ์นี้สำหรับการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2560

KMITL-2017-EN-M-070-021

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODIFICATION OF MQTT PROTOCOL TO FIND ALTERNATE ROUTE

CHONTAWEE WONGPROMRAT

A THESIS SUBMITTED IN FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN COMPUTER ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2017

KMITL-2017-EN-M-070-021

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2017

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การปรับเปลี่ยนกลไกการทำงานของโพรโทคอลเอ็มคิวทีทีเพื่อค้นหาเส้นทางสำรอง
Thesis Title Modification of MQTT Protocol to Find Alternate Route
นักศึกษา นายชลทวี วงศ์พร้อมรัตน์
รหัสประจำตัว 58601091
ปริญญา วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์ ดร.วัชระ ฉัตรวิริยะ
หมายเลขวิทยานิพนธ์ KMITL-2017-EN-M-070-021

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ดร.กมล	เขมะรังษี	
ผศ.ดร.ศักดิ์ชัย	ทิพย์จักษุรัตน์	
ดร.อำนาจ	ขาวเน	
ผศ.ดร.สุรินทร์	กิตติธรกุล	
ดร.วัชระ	ฉัตรวิริยะ	

วัน / เดือน/ ปี ที่สอบ วันอังคารที่ 21 มีนาคม พ.ศ. 2560 เวลา 13.00-15.00 น.
สถานที่สอบ ณ อาคาร A ชั้น 5 ห้องประชุม 1

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว

(รองศาสตราจารย์ ดร. คมสัน มาลีสี)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 21 มีนาคม พ.ศ. 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การปรับเปลี่ยนกลไกการทำงานของโพรโทคอลเอ็มคิวทีทีเพื่อค้นหาเส้นทางสำรอง
นักศึกษา	นายชลทวี วงศ์พร้อมรัตน์
รหัสประจำตัว	58601091
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2560
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ดร.วัชระ ฉัตรวิริยะ

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ นำเสนอถึงวิธีการค้นหาเส้นทางสำรองสำหรับการส่งข้อมูลด้วยโพรโทคอลเอ็มคิวทีทีเมื่อการส่งข้อมูลระหว่างพับบลิชเชอร์และโบรกเกอร์ไม่สามารถใช้งานผ่านเครือข่ายเดิมที่มีปัญหาได้ โดยใช้วิธีการส่งข้อมูลไปยังอุปกรณ์รอบข้างซึ่งอยู่ในเครือข่ายอื่น และให้อุปกรณ์นั้นทำหน้าที่ส่งต่อข้อมูลไปยังโบรกเกอร์แทน จากการทดลองที่ทำการปิดการเชื่อมต่อที่บางเครือข่ายอินเทอร์เฟซของโบรกเกอร์เป็นระยะเวลา 10 20 และ 30 วินาที พบว่าหลังจากที่วิธีการที่นำเสนอทดลองส่งข้อมูลด้วยโพรโทคอลเอ็มคิวทีทีด้วยตัวเองแล้ว หากไม่สามารถส่งข้อมูลได้ วิธีการที่นำเสนอสามารถที่จะค้นหาเส้นทางผ่านทางอุปกรณ์รอบข้างและส่งข้อมูลไปยังโบรกเกอร์ผ่านทางเครือข่ายอื่นได้โดยไม่ต้องรอให้เครือข่ายหลักที่ถูกตัดขาดไปกลับมาเป็นปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis	Modification of MQTT Protocol to Find Alternate Route
Student	Mr.Chontawee Wongpromrat
Student ID	58601091
Degree	Master of Engineering
Program	Computer Engineering
Year	2017
Thesis Advisor	Dr.Watchara Chatwiriya

ABSTRACT

This thesis proposed a solution to find the alternate route for MQTT protocol when publisher cannot send message to broker through the broken network connection. The proposed solution will send messages to other nearby devices which are in another network and forward it to the broker. From the experimental section, we deactivate some network interface of broker for 10, 20 and 30 seconds. For the result of an experiment, proposed solution can send messaget to the broker using alternate route from nearby devices without waiting time for reconnection of broken network connection.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความกรุณาจากหลายท่าน

ขอขอบคุณ ดร.วัชระ ฉัตรวิริยะ อาจารย์ที่ปรึกษาที่คอยดูแล ให้คำแนะนำ ให้คำปรึกษาสำหรับการความรู้และการแก้ไขปัญหาระหว่างการดำเนินงานวิจัย

ขอขอบคุณสมาชิกในห้องปฏิบัติการสมองกลฝังตัว (Embedded System Laboratory) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ช่วยให้คำแนะนำในระหว่างการศึกษาในระดับปริญญาโท ทั้งด้านกระบวนการในการศึกษาและปัญหางานวิจัย

และสุดท้ายขอขอบคุณครอบครัวทั้ง บิดา มารดา และพี่สาวสำหรับกำลังใจ และคำแนะนำ สำหรับการดำเนินชีวิต

สำหรับประโยชน์อันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ถ่ายทอดความรู้และประสบการณ์ที่ดีให้แก่ข้าพเจ้า

ชลทวี วงศ์พร้อมรัตน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ..... I

ABSTRACT II

กิตติกรรมประกาศ..... III

สารบัญ..... IV

สารบัญตาราง VII

สารบัญรูป..... VIII

บทที่ 1 ที่มาและความสำคัญ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 1

1.3 ปัญหาความสำคัญของวิทยานิพนธ์ 2

1.4 ขอบเขตงานวิจัย 2

1.5 องค์ประกอบของวิทยานิพนธ์ 2

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 3

2.1 อินเทอร์เน็ตของสรรพสิ่ง 3

2.2 ลักษณะการส่งข้อมูลผ่านเอ็มคิวทีทีโพรโทคอล 3

2.2.1 เอ็มคิวทีทีโพรโทคอล 4

2.2.2 การเปรียบเทียบโพรโทคอลเอ็มคิวทีทีกับเอชทีทีพี 7

2.2.3 คุณภาพการบริการ (QoS : Quality of Service) 9

2.3 Open-Source สำหรับการพัฒนาเอ็มคิวทีทีโพรโทคอล 17

2.3.1 Paho 17

2.3.2 Mosquitto 17

2.4 การสื่อสารภายในเครือข่ายเซ็นเซอร์ไร้สาย 18

2.4.1 การสื่อสารในรูปแบบซิงเกิล-ฮ็อป 18

2.4.2 การสื่อสารในรูปแบบมัลติ-ฮ็อป 19

2.5 งานวิจัยที่เกี่ยวข้อง 19

บทที่ 3 การดำเนินงานวิจัย 29

3.1 สมมุติฐาน 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2 แนวคิดการแก้ปัญหา 30
3.2.1 แนวคิดการส่งต่อข้อมูลไปยังอุปกรณ์รอบข้าง 30
3.2.2 แนวคิดการส่งต่อข้อมูลแบบมัลติ-ฮ็อป 31
3.3 การออกแบบและจำลองระบบสำหรับการทดลอง 31
3.4 การทำงานของวิธีการที่นำเสนอ 32
3.5 ฟังก์ชันการส่งและรับข้อมูลด้วยเออาร์พีและเอชทีทีพี 36
3.5.1 ฟังก์ชันสำหรับการค้นหาอุปกรณ์รอบข้าง 36
3.5.2 ฟังก์ชันสำหรับการส่งข้อมูลไปยังอุปกรณ์รอบข้าง 36
3.5.3 ฟังก์ชันสำหรับการจดจำการส่งข้อมูลผ่านอุปกรณ์รอบข้าง 36
3.6 การทดลอง 37
3.7 การวัดผล 40
3.7.1 ผลความล่าช้าของการส่งข้อมูล 40
3.7.2 ผลความเร็วในการตอบสนองการส่งข้อมูล 42

บทที่ 4 ผลและการวิเคราะห์ผลการวิจัย 45
4.1 ผลแสดงความล่าช้าการส่งข้อมูล 45
4.1.1 ผลแสดงความล่าช้าการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 10 วินาที 46
4.1.2 ผลแสดงความล่าช้าการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 20 วินาที 50
4.1.3 ผลแสดงความล่าช้าการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 30 วินาที 54
4.2 ผลแสดงความเร็วในการตอบสนองของข้อมูล 57
4.2.1 ผลแสดงความเร็วในการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 10 วินาที 58
4.2.2 ผลแสดงความเร็วในการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 20 วินาที 62
4.2.3 ผลแสดงความเร็วในการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 30 วินาที 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.3 การวิเคราะห์และเปรียบเทียบผลความล่าช้าการส่งข้อมูลและความเร็วในการตอบสนองของข้อมูลในวิธีการต่าง ๆ 69

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 74

5.1 สรุปผลการวิจัย 74

5.2 ข้อเสนอแนะ 75

เอกสารอ้างอิง 76

ภาคผนวก 79

ประวัติผู้เขียน 93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 ข้อเปรียบเทียบโพรโทคอลเอ็มคิวทีทีและเอชทีทีพี 8

2.2 กระแสข้อมูลของคิวโอเอสศูนย์ 10

2.3 กระแสข้อมูลของคิวโอเอสหนึ่ง 11

2.4 กระแสข้อมูลของคิวโอเอสหนึ่ง 13

2.5 ค่าเฉลี่ยของ Delay สำหรับอัตราการสูญหายของข้อมูลที่ระดับต่าง ๆ 22

3.1 รายละเอียดของ Virtual Machine สำหรับการทดลอง 37

3.2 รายละเอียดของซอฟแวร์ที่ใช้และค่าที่กำหนดสำหรับการทดลอง 38

3.3 อัตราการส่งข้อมูลและ Ping Delay ของแต่ Virtual Machine 39

4.1 ผลความล่าช้าของการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 10 วินาที 49

4.2 ผลความล่าช้าของการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 20 วินาที 53

4.3 ผลความล่าช้าของการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 30 วินาที 57

4.4 ผลแสดงความเร็วในการตอบสนองการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 10 วินาที 61

4.5 ผลแสดงความเร็วในการตอบสนองการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 20 วินาที 65

4.6 ผลแสดงความเร็วในการตอบสนองการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 30 วินาที 69

4.7 ผลความล่าช้าของการส่งข้อมูลโดยเฉลี่ย 70

4.8 ผลความการตอบสนองการส่งข้อมูลโดยเฉลี่ย 70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 เอ็มคิวทีทีเมื่อเปรียบเทียบกับโพรโทคอลอื่นในโพรโทคอลสแต็ค 4
2.2 ตัวอย่างการเชื่อมต่ออุปกรณ์ที่ใช้งานเอ็มคิวทีทีโพรโทคอล 5
2.3 ตัวอย่างระบบที่ใช้งานเอ็มคิวทีทีโพรโทคอล 6
2.4 โครงสร้างของเอ็มคิวทีทีแพ็คเกจ 6
2.5 รูปแบบของ Fixed Header 7
2.6 การส่งข้อมูลด้วยคุณภาพการบริการในคิวโอเอสระดับศูนย์ 10
2.7 การส่งข้อมูลด้วยคุณภาพการบริการในคิวโอเอสระดับหนึ่ง 11
2.8 การส่งข้อมูลด้วยคุณภาพการบริการในคิวโอเอสระดับสอง 12
2.9 การพับบลิชข้อมูลด้วยคิวโอเอสหนึ่งและซับสไครบด้วยคิวโอเอสศูนย์ 15
2.10 การพับบลิชและซับสไครบด้วยคิวโอเอสที่มากกว่าหรือเทียบเท่าหนึ่ง 16
2.11 การสื่อสารในรูปแบบซิงเกิล-ฮ็อป 18
2.12 การสื่อสารในรูปแบบมัลติ-ฮ็อป 19
2.13 การจำลองโหนดที่สามารถเคลื่อนที่ได้ภายในอาคาร 20
2.14 Cumulative distribution function ของค่าความเหลื่อมล้ำของเวลาที่มากที่สุดในระหว่าง การส่งข้อมูลระยะเวลา 10 มิลลิวินาที โดยใช้เอเอ็มคิวพี (ซ้าย) และเอ็มคิวทีที (ขวา) 21
2.15 Cumulative distribution function ของ Delay สำหรับการสูญหายของข้อมูล 22
2.16 การเปรียบเทียบระหว่างค่าเฉลี่ยการส่งข้อมูลต่อข้อความกับอัตราการสูญหายของข้อความ 23
2.17 องค์ประกอบการเชื่อมต่อของเครือข่ายแบบมีสายและไร้สาย 24
2.18 ผล End-to-end Delay ภายในเครือข่ายมีสาย 24
2.19 ผลการสูญหายของข้อมูลภายในเครือข่ายมีสาย 25
2.20 ผล End-to-end Delay ภายในเครือข่ายไร้สาย 26
2.21 ผลการสูญหายของข้อมูลภายในเครือข่ายไร้สาย 26
2.22 สถาปัตยกรรมที่นำเสนอ 27
2.23 การใช้พลังงาน 27
3.1 สมมุติฐานของระบบตัวอย่าง 29
3.2 ตัวอย่างระบบสำหรับการส่งข้อมูลไปยังอุปกรณ์รอบข้าง 32
3.3 Flow Chart แสดงการลำดับขั้นตอนการทำงานของวิธีการที่นำเสนอ 33
3.4 ตัวอย่างการส่งข้อมูลด้วยวิธีการที่นำเสนอ 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.5 รูปแบบการเชื่อมต่อของอุปกรณ์ที่ใช้สำหรับการทดลอง 40

3.6 ช่วงระยะเวลาผลความล่าช้าของการส่งข้อมูลที่ของกระบวนการคิวโอเอส 1 ที่พับบลิชเชอร์ 41

3.7 ช่วงระยะเวลาผลความล่าช้าของการส่งข้อมูลที่ของกระบวนการคิวโอเอส 2 ที่พับบลิชเชอร์ 41

3.8 ช่วงระยะเวลาผลความล่าช้าของการส่งข้อมูลที่ของวิธีการที่นำเสนอ 42

3.9 ช่วงระยะเวลาผลการตอบสนองการส่งข้อมูลที่ของกระบวนการคิวโอเอส 1 ที่พับบลิชเชอร์ 43

3.10 ช่วงระยะเวลาผลการตอบสนองการส่งข้อมูลที่ของกระบวนการคิวโอเอส 2 ที่พับบลิชเชอร์ 43

3.11 ช่วงระยะเวลาผลการตอบสนองการส่งข้อมูลที่ของวิธีการที่นำเสนอ 44

4.1 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 10 วินาที 46

4.2 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 10 วินาที 47

4.3 ความถี่ของความล่าช้าของการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 10 วินาที 48

4.4 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 20 วินาที 50

4.5 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 20 วินาที 51

4.6 ความถี่ของความล่าช้าของการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 20 วินาที 52

4.7 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 30 วินาที 54

4.8 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 30 วินาที 55

4.9 ความถี่ของความล่าช้าของการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 30 วินาที 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.10 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 10 วินาที 58

4.11 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 10 วินาที 59

4.12 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 10 วินาที 60

4.13 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 20 วินาที 62

4.14 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 20 วินาที 63

4.15 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 20 วินาที 64

4.16 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 30 วินาที 66

4.17 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 30 วินาที 67

4.18 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 30 วินาที 68

4.19 ช่วงระยะเวลารอก่อนการส่งข้อมูลซ้ำ 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
ที่มาและความสำคัญ

ในบทนี้จะเป็นการกล่าวถึงที่มาและความสำคัญของงานวิจัย โดยแบ่งออกเป็น 5 หัวข้อ คือ 1) ความเป็นมาและความสำคัญของปัญหา 2) ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 3) ปัญหาความสำคัญของวิทยานิพนธ์ 4) ขอบเขตของงานวิจัย และ 5) องค์ประกอบของวิทยานิพนธ์ โดยมีรายละเอียดดังต่อไปนี้

1.1 ความเป็นมาและความสำคัญของปัญหา

อินเทอร์เน็ตของสรรพสิ่งหรือ IoT (Internet of Things) [1] คือระบบที่อุปกรณ์สามารถเชื่อมต่อหากันได้ผ่านโพรโทคอลการสื่อสาร โดยอุปกรณ์ภายในระบบจะทำการแลกเปลี่ยนข้อมูลและทำงานร่วมกัน แนวคิดการทำงานระหว่างอุปกรณ์ในรูปแบบนี้ทำให้เกิดเป็นนวัตกรรมและบริการในรูปแบบใหม่ อาทิเช่น ระบบฟาร์มอัจฉริยะที่ช่วยให้เกษตรกรสามารถดูแลและจัดการการเพาะปลูกพืชได้ง่ายและทั่วถึงมากขึ้น ระบบบ้านอัจฉริยะที่ช่วยดูแลบ้านและเพิ่มความสะดวกสบายสำหรับผู้อยู่อาศัย ระบบดูแลสุขภาพอัจฉริยะที่ทำให้สามารถติดตามอาการของผู้ป่วย รวมถึงสามารถแจ้งเตือนเมื่อผู้ป่วยมีอาการผิดปกติด้วยระบบอัตโนมัติ เป็นต้น

หนึ่งในโพรโทคอลที่นิยมใช้สำหรับระบบอินเทอร์เน็ตของสรรพสิ่งนี้คือ เอ็มคิวทีทีโพรโทคอล (MQTT protocol) [1][2][3] เนื่องจากเป็นโพรโทคอลที่เหมาะสำหรับการสื่อสารในระบบที่มีอุปกรณ์จำนวนมาก การรับส่งข้อมูลด้วยโพรโทคอลนี้จำเป็นที่จะต้องระบุคุณภาพการรับส่งข้อมูลหรือ คิวโอเอส (QoS : Quality of Services) [2][4] ในกรณีที่การรับส่งข้อมูลไปยังอุปกรณ์ปลายทางมีปัญหา มาตรการการส่งข้อมูลซ้ำจะเป็นไปตามคิวโอเอสที่ระบุไว้ แต่ไม่ว่าจะเป็นคิวโอเอสที่ระดับใด การส่งข้อมูลซ้ำก็จะเกิดจากอุปกรณ์ส่งตัวเดิมเสมอ กล่าวคือ อุปกรณ์ที่ส่งข้อมูลจะส่งข้อมูลได้ช้าลงหากการเชื่อมต่อขาดหายไปเป็นระยะเวลานาน ทำให้ประสิทธิภาพการส่งข้อมูลของระบบไม่ดีเท่าที่ควร ดังนั้นงานวิจัยนี้จึงมุ่งเน้นไปที่การนำเสนอวิธีการแก้ไขปัญหาการส่งข้อมูลล่าช้าในโพรโทคอลเอ็มคิวทีที

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วัตถุประสงค์ของการศึกษางานวิจัยคือ นำเสนอแนวทางสำหรับการแก้ไขปัญหาการส่งข้อมูลล่าช้าของกระบวนการคิวโอเอสในโพรโทคอลเอ็มคิวทีที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ปัญหาความสำคัญของวิทยานิพนธ์

ปัญหาสำคัญสำหรับงานวิจัยนี้คือ การส่งข้อมูลที่ล่าช้าในกระบวนการคิวโอเอสของเอ็มคิวทีทีโพลโทคอลเมื่อการเชื่อมต่อขาดหายเป็นระยะเวลานาน เนื่องจากกระบวนการคิวโอเอสจำเป็นที่จะต้องส่งข้อมูลซ้ำจากอุปกรณ์ตัวเดิม ดังนั้นอุปกรณ์ส่งตั้งต้นจึงจำเป็นจะต้องรอจนกว่าการเชื่อมต่อกลับมาใช้ได้ปกติถึงจะทำการส่งข้อมูลได้ ซึ่งส่งผลอย่างมากต่อระบบที่ไม่สามารถรอช่วงเวลาที่ไม่แน่นอนเหล่านี้ได้ งานวิจัยนี้จึงมุ่งเน้นไปที่การนำเสนอวิธีการที่ช่วยลดระยะเวลาในการส่งข้อมูลที่อาจใช้เวลานานและไม่แน่นอนของคิวโอเอส

1.4 ขอบเขตงานวิจัย

นำเสนอวิธีแก้ไขปัญหาการส่งข้อมูลล่าช้าอันเกิดจากการเชื่อมต่อขาดหายไปเป็นระยะเวลาที่ยาวนาน ซึ่งเกิดจากคิวโอเอสของเอ็มคิวทีทีโพรโทคอล และเพิ่มประสิทธิภาพวิธีการที่นำเสนอให้ใช้เวลาในการรับส่งข้อมูลในระบบให้ดียิ่งขึ้น

1.5 องค์ประกอบของวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์แบ่งออกเป็น 5 ส่วน ได้แก่

บทที่ 1 ที่มาและความสำคัญ แบ่งออกเป็น 5 หัวข้อ ได้แก่ ความเป็นมาและความสำคัญของปัญหา ความมุ่งหมายและวัตถุประสงค์ของการศึกษา ปัญหาความสำคัญของวิทยานิพนธ์ ขอบเขตของงานวิจัย และองค์ประกอบของวิทยานิพนธ์

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง แบ่งออกเป็น 4 หัวข้อ ได้แก่ ระบบอินเทอร์เน็ตของสรรพสิ่ง ลักษณะการส่งข้อมูลผ่านเอ็มคิวทีทีโพรโทคอล เครือข่ายเซ็นเซอร์ไร้สาย และงานวิจัยที่เกี่ยวข้องกับเอ็มคิวทีทีโพรโทคอล

บทที่ 3 การดำเนินงานวิจัย แบ่งออกเป็น 6 หัวข้อ ได้แก่ สมมุติฐาน แนวคิดการแก้ปัญหา การออกแบบและจำลองระบบสำหรับการทดลอง การทำงานของวิธีการที่นำเสนอ ฟังก์ชันการส่งต่อและรับข้อมูลด้วยเออาร์พีและเอชทีทีพี และการทดลอง

บทที่ 4 ผลและการวิเคราะห์ผลการวิจัย แบ่งออกเป็น 3 หัวข้อ ได้แก่ ผลแสดงความล่าช้าการส่งข้อมูล ผลแสดงความเร็วในการตอบสนองของข้อมูล และการวิเคราะห์และเปรียบเทียบผลความล่าช้าการส่งข้อมูลและความเร็วในการตอบสนองของข้อมูลในวิธีการต่าง ๆ

บทที่ 5 สรุปผลการวิจัยและการเสนอแนะ แบ่งออกเป็น 2 หัวข้อ ได้แก่ สรุปผลการวิจัย และข้อเสนอแนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทนี้จะนำเสนอทฤษฎีและงานวิจัยที่เกี่ยวข้อง โดยแบ่งออกเป็น 4 หัวข้อคือ 1) ระบบอินเทอร์เน็ตของสรรพสิ่ง 2) ลักษณะการส่งข้อมูลผ่านเอ็มคิวทีทีโพรโทคอล 3) เครือข่ายเซ็นเซอร์ไร้สาย และ 4) งานวิจัยที่เกี่ยวข้องกับเอ็มคิวทีทีโพรโทคอล ดังมีรายละเอียดดังนี้

2.1 อินเทอร์เน็ตของสรรพสิ่ง

อินเทอร์เน็ตของสรรพสิ่งถูกกล่าวถึงครั้งแรกในปี ค.ศ. 1999 [1] มันถูกใช้สำหรับอธิบายถึงระบบที่มีการเชื่อมต่ออุปกรณ์จากโลกความเป็นจริงเข้ากับเครือข่ายอินเทอร์เน็ตผ่านทางเซ็นเซอร์ต่าง ๆ โดยในครั้งแรก Kevin Ashton นักบุกเบิกทางด้านเทคโนโลยีชาวอังกฤษได้ยกตัวอย่างเกี่ยวกับระบบที่เป็นอินเทอร์เน็ตของสรรพสิ่งเอาไว้ เช่น ระบบที่ใช้งานแท็คอาร์เอฟไอดี (RFID Tag) ในการนับปริมาณสิ่งของโดยที่ไม่ใช้แรงงานของมนุษย์เข้ามาช่วยเหลือ แต่ในปัจจุบันอินเทอร์เน็ตของสรรพสิ่งมักจะหมายถึงการเชื่อมต่อและการประมวลผล จากอุปกรณ์และเซ็นเซอร์ต่าง ๆ ที่มีการทำงานร่วมกันจนเกิดเป็นระบบ

แนวคิดการเชื่อมต่อคอมพิวเตอร์หรืออุปกรณ์เข้ากับเครือข่ายอินเทอร์เน็ตนั้นไม่ใช่เรื่องที่ใหม่ ในช่วงปี ค.ศ. 1990 การสื่อสารผ่านเครือข่ายไร้สายทำให้เกิดแนวคิดที่เรียกว่าการสื่อสารระหว่างเครื่องจักร หรือ M2M (Machine-to-Machine) ขึ้นมา แนวคิดนี้ได้รับการยอมรับในวงกว้างโดยการนำมาประยุกต์ใช้ในโรงงานอุตสาหกรรมสำหรับการติดตามกระบวนการต่าง ๆ อย่างไรก็ตาม แนวคิดการสื่อสารระหว่างเครื่องจักรนี้มักจะถูกใช้สำหรับเครื่องจักรที่เฉพาะด้าน ทำให้ไม่มีมาตรฐานกลางสำหรับการใช้งานและการเชื่อมต่อไปยังระบบอื่น ในขณะที่แนวคิดอินเทอร์เน็ตของสรรพสิ่งมีพื้นฐานแนวคิดมาจากมาตรฐานกลางและอินเทอร์เน็ตทั่วไปที่ใช้อินเทอร์เน็ตโพรโทคอล (IP : internet protocol) ในการเชื่อมต่อไปยังอุปกรณ์ปลายทาง

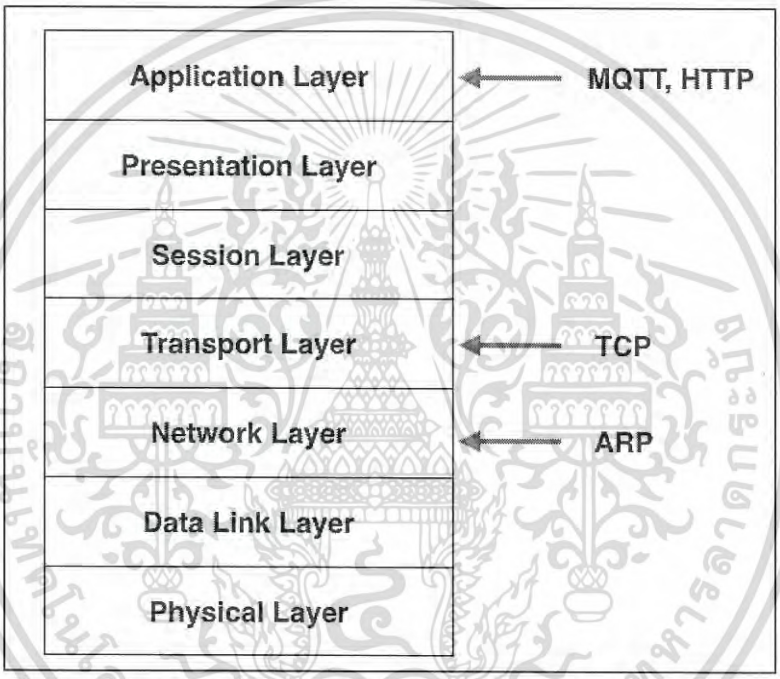
2.2 ลักษณะการส่งข้อมูลผ่านเอ็มคิวทีทีโพรโทคอล

ในปัจจุบันแนวคิดอินเทอร์เน็ตของสรรพสิ่งถูกนำมาประยุกต์ใช้ในระบบต่าง ๆ มากขึ้นกว่าในอดีต ทำให้เกิดโพรโทคอลที่ถูกนำมาใช้กับแนวคิดนี้มากขึ้น เอ็มคิวทีทีโพรโทคอลก็เป็นหนึ่งในโพรโทคอลที่นิยมนำมาใช้งาน เนื่องจากความเหมาะสมในหลาย ๆ ด้าน โดยมีรายะเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 เอ็มคิวทีทีโพรโทคอล

เอ็มคิวทีทีโพรโทคอล (MQTT : Message Queue Telemetry Transport) เป็นโพรโทคอลที่มักถูกนำมาใช้ในระบบอินเทอร์เน็ตของสรรพสิ่ง [2][3][4] โพรโทคอลนี้ถูกออกแบบสำหรับการรับส่งข้อมูลที่มีขนาดเล็กในลักษณะพับบลิช/ซับสไครบ (publish/subscribe) ซึ่งมักจะถูกใช้ในกรณีที่แบนด์วิดธ์มีปริมาณที่จำกัด อย่างเช่น การส่งข้อมูลจากเซ็นเซอร์ไปยังโบรกเกอร์ผ่านช่องทางดาวเทียม หรือการส่งข้อมูลในระบบควบคุมการทำงานภายในบ้านและอุปกรณ์ขนาดเล็กเป็นต้น โดยปกติ โพรโทคอลนี้จะใช้งานได้ดีในแอพพลิเคชั่นบนอุปกรณ์พกพาเพราะว่ามีขนาดที่เล็ก (Header ของเอ็มคิวทีทีโพรโทคอลมีขนาดเพียง 2 ไบท์) และใช้พลังงานน้อย



รูปที่ 2.1 เอ็มคิวทีทีเมื่อเปรียบเทียบกับโพรโทคอลอื่นในโพรโทคอลสแต็ค

เอ็มคิวทีทีเป็นโพรโทคอลที่อยู่ในระดับชั้นของ Application Layer ตามลักษณะการแบ่งของโพรโทคอลสแต็คแบบ OSI Model เช่นเดียวกับเอชทีทีพีโพรโทคอล (HTTP) ดังแสดงในรูปที่ 2.1 โดยในชั้น transport layer ของโพรโทคอลเอ็มคิวทีทีจะใช้โพรโทคอลทีซีพี (TCP) ช่วยในการรับส่งข้อมูลในชั้นนี้

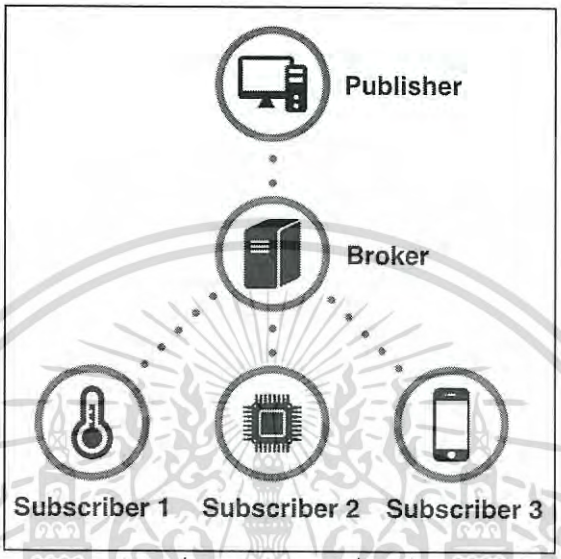
ระบบที่รับส่งข้อมูลด้วยเอ็มคิวทีทีโพรโทคอลจะประกอบไปด้วย 3 ส่วนหลักได้แก่ พับบลิชเชอร์ (publisher) ซับสไครบเบอร์ (subscriber) และโบรกเกอร์ (broker) และ โดยทั้ง 3 ส่วนก็จะทำหน้าที่แตกต่างกันดังนี้

1) พับบลิชเชอร์ ทำหน้าที่ส่งข้อมูลไปยังโบรกเกอร์ โดยข้อมูลที่ส่งในแต่ละครั้งจะประกอบไปด้วย 2 ส่วนคือ หัวข้อ (topic) และข้อความ (message)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ซับสไครบเบอร์ จะทำการเชื่อมต่ออยู่กับโบรกเกอร์ตลอดเวลา โดยทุกครั้งที่ทำการเชื่อมต่อ มันจะระบุหัวข้อที่ต้องการรับข้อมูลจากโบรกเกอร์

3) โบรกเกอร์ ทำหน้าที่เป็นตัวกลางในการรับส่งข้อมูล โดยจะรับข้อมูลจากพับบลิชเชอร์ และส่งต่อไปยังซับสไครบเบอร์ โดยอ้างอิงจากหัวข้อที่พับบลิชเชอร์ส่งมาให้



รูปที่ 2.2 ตัวอย่างการเชื่อมต่ออุปกรณ์ที่ใช้งานเอ็มคิวทีทีโพรโทคอล

รูปที่ 2.2 แสดงถึงตัวอย่างการเชื่อมต่ออุปกรณ์ที่ใช้งานเอ็มคิวทีทีโพรโทคอล โดยการเชื่อมต่อถูกแบ่งออกเป็น 2 ส่วนหลักคือ การเชื่อมต่อระหว่างพับบลิชเชอร์กับโบรกเกอร์ และการเชื่อมต่อระหว่างโบรกเกอร์กับซับสไครบเบอร์

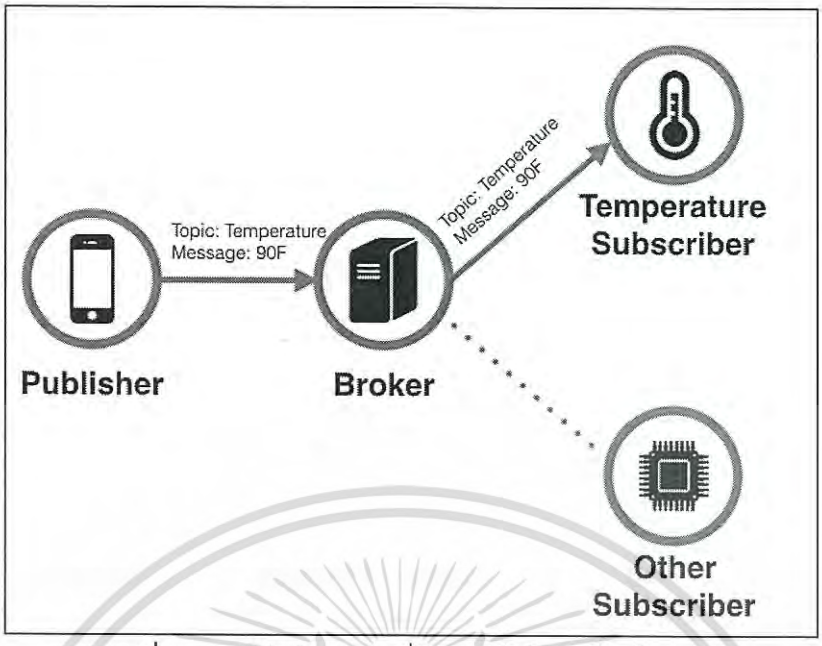
การส่งข้อมูลด้วยโพรโทคอลนี้ สามารถอธิบายได้ดังนี้

1) พับบลิชเชอร์ทำการส่งข้อความพร้อมระบุหัวข้อไปยังโบรกเกอร์

2) เมื่อโบรกเกอร์ได้รับข้อมูลจากพับบลิชเชอร์ มันจะส่งข้อมูลนั้นต่อไปยังซับสไครบเบอร์โดยอ้างอิงจากหัวข้อของข้อความ

3) ซับสไครบเบอร์จะได้รับข้อมูลก็ต่อเมื่อทำการเชื่อมต่อกับโบรกเกอร์ด้วยหัวข้อเดียวกับหัวข้อที่พับบลิชเชอร์ส่งมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ตัวอย่างระบบที่ใช้งานเอ็มคิวทีทีโพรโทคอล

รูปที่ 2.3 แสดงถึงตัวอย่างการส่งข้อมูลด้วยเอ็มคิวทีทีโพรโทคอล จะเห็นว่าเมื่อพับบลิชเชอร์ส่งข้อความที่ระบุหัวข้อเกี่ยวกับอุณหภูมิไปยังโบรกเกอร์ โบรกเกอร์ก็จะทำการส่งต่อข้อมูลไปยังซับสไครบเบอร์ที่เชื่อมต่อด้วยหัวข้ออุณหภูมิเหมือนกัน ในขณะที่ซับสไครบเบอร์ที่เชื่อมต่อด้วยหัวข้ออื่นจะไม่ได้รับข้อความที่พับบลิชเชอร์ส่งมา โดยในรูปนี้จะพูดถึงมุมมองการสั่งการอุปกรณ์จากพับบลิชเชอร์เพื่อปรับเปลี่ยนอุณหภูมิของซับสไครบเบอร์ อย่างเช่น หากอุปกรณ์เป็นโทรศัพท์มือถืออุปกรณ์จะทำการส่งค่าอุณหภูมิที่ต้องการเพื่อให้อุปกรณ์อย่างแอร์ทำการปรับเปลี่ยนอุณหภูมิ แต่ถ้าแอร์เป็นพับบลิชเชอร์จะกลายเป็นว่าแอร์จะทำการส่งข้อมูลอุณหภูมิ ณ ปัจจุบันมาให้มือถือซึ่งเป็นซับสไครบเบอร์สำหรับการตรวจสอบสถานะ เป็นต้น

Fixed Header, present in all MQTT Control Packets
Variable Header, present in some MQTT Control Packets
Payload, present in some MQTT Control Packets

รูปที่ 2.4 โครงสร้างของเอ็มคิวทีทีแพ็คเกจ

การส่งข้อมูลด้วยเอ็มคิวทีทีโพรโทคอลจะมีโครงสร้างของแพ็คเกจที่ทำการส่งข้อมูลดังรูปที่ 2.4 โดยจะแบ่งออกเป็น 3 ระดับขึ้นอยู่กับประเภทของแพ็คเกจที่ทำการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

รูปที่ 2.5 รูปแบบของ Fixed Header

รูปที่ 2.5 แสดงถึงรูปแบบของ Fixed Header ที่มีอยู่ในทุกแพ็คเกจของเอ็มคิวทีทีโพรโทคอล โดยจะมีขนาดอยู่ที่ 2 ไบท์ ซึ่งจะมีการระบุประเภทของแพ็คเกจ ค่าต่าง ๆ ของแพ็คเกจประเภทนั้น และจำนวนไบท์ของ Variable Header และ Payload

รูปแบบของแพ็คเกจบางประเภทจะมีส่วนของ Variable Header ประกอบด้วย โดยจะอยู่ระหว่าง Fixed Header และ Payload ซึ่งเนื้อหาข้างในจะมีความแตกต่างกันแล้วแต่รูปแบบของแพ็คเกจ ประเภทของแพ็คเกจที่มี Variable Header เป็นส่วนประกอบจะได้แก่ PUBLISH (เมื่อ QoS มากกว่า 0) PUBACK PUBREC PUBREL PUBCOMP SUBSCRIBE SUBACK UNSUBSCRIBE และ UNSUBACK

เช่นเดียวกับ Variable Header แพ็คเกจบางประเภทจะมี Payload เป็นส่วนสุดท้ายของแพ็คเกจ เนื้อหาข้างในจะขึ้นอยู่กับรูปแบบของแพ็คเกจ โดยรูปแบบของแพ็คเกจที่ต้องมี Payload เป็นส่วนสุดท้ายของแพ็คเกจ ได้แก่ CONNECT SUBSCRIBE SUBACK UNSUBSCRIBE และแพ็คเกจที่มี Payload เป็นส่วนเสริม ได้แก่ PUBLISH

Variable Header และ Payload นั้นปกติแล้วจะมีขนาดที่ไม่แน่นอนขึ้นอยู่กับประเภทของแพ็คเกจที่ส่งข้อมูล แต่ความยาวสูงสุดของทั้งสองส่วนนี้จะต้องมีความยาวไม่เกิน 256 เมกะไบท์

2.2.2 การเปรียบเทียบโพรโทคอลเอ็มคิวทีทีกับเอชทีทีพี

ด้วยความที่เอ็มคิวทีทีเป็นโพรโทคอลที่มักถูกใช้ในระบบอินเทอร์เน็ตของสรรพสิ่ง จึงมีการเปรียบเทียบกับโพรโทคอลเอชทีทีพี (HTTP : Hypertext Transfer Protocol) ซึ่งเป็นหนึ่งในโพรโทคอลมาตรฐานที่นิยมใช้กันอย่างแพร่หลายในปัจจุบันจนเกิดการประยุกต์ใช้ในหลากหลายระบบไม่เว้นแม้แต่ระบบอินเทอร์เน็ตของสรรพสิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ข้อเปรียบเทียบโพรโทคอลเอ็มคิวทีทีและเอชทีทีพี

	เอ็มคิวทีที	**เอชทีทีพี**
การออกแบบ	Data Centric	Document Centric
รูปแบบการส่งข้อมูล	Publish-Subscribe	Request-Response
ความซับซ้อน	พื้นฐาน	ซับซ้อนมาก
ขนาดข้อความ	ขนาดเล็ก	ขนาดใหญ่
ระดับการบริการ	มี Quality of Service 3 ระดับ	ทุก ๆ ข้อความจะมีระดับการบริการที่เหมือนกันหมด
Libraries	Libraries สำหรับภาษา C (30 KB) และ Java (100 KB)	ขึ้นอยู่กับแอพพลิเคชั่น
การกระจายข้อมูล	1 ต่อ 0, 1 ต่อ 1 และ 1 ต่อ n	1 ต่อ 1 เท่านั้น

ตารางที่ 2.1 [19] แสดงถึงการเปรียบเทียบโพรโทคอลเอ็มคิวทีทีกับเอชทีทีพี โดยมีรายละเอียดดังนี้

2.2.2.2 แนวทางการออกแบบ (Design orientation)

เอ็มคิวทีทีถูกออกแบบให้มีการส่งข้อมูลในรูปแบบไบท์อาเรย์ (Byte Array) ในขณะที่เอชทีทีพีสนับสนุนมาตรฐาน MIME เพื่อกำหนดเนื้อหาภายในแพ็คเกจ

2.2.2.3 รูปแบบการส่งข้อมูล (Pattern)

เอ็มคิวทีทีใช้รูปแบบการส่งข้อมูลแบบพับบลิช-ซับสไครบซึ่งไม่จำเป็นจะต้องรู้ถึง address ของอุปกรณ์ปลายทาง ในขณะที่เอชทีทีพีใช้รูปแบบการส่งข้อมูลแบบ ร้องขอ-ตอบรับ (request-response) ที่เครื่องลูกข่ายจำเป็นจะต้องรู้ address ของอุปกรณ์ที่ทำการเชื่อมต่อ

2.2.2.4 ความซับซ้อน (Complexity)

เอ็มคิวทีทีมีรูปแบบของข้อความที่จำกัดเพียงแค่ CONNECT PUBLISH SUBSCRIBE UNSUBSCRIBE และ DISCONNECT เท่านั้น แต่เอชทีทีพีสามารถมีรูปแบบของข้อความและผลลัพธ์ตอบกลับที่มากกว่า เช่น GET POST PUT DELETE HEAD TRACE CONNECT เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.5 ขนาดของข้อความ (Message size)

เอ็มคิวทีทีมีขนาดที่เล็ก โดยมี Header เพียงแค่ 2 ไบท์และข้อมูลที่จำเป็นเท่านั้น ในขณะที่เอชทีทีพีใช้รูปแบบข้อความที่สามารถขยายขนาดของ Header และข้อความของแพ็คเกจได้ โดยส่วนใหญ่แล้วแพ็คเกจของเอชทีทีพีจะจัดให้อยู่ในรูปแบบที่มนุษย์สามารถอ่านได้ จึงเป็นเหตุให้ขนาดของข้อความในเอชทีทีพีโพรโทคอลมีขนาดที่ใหญ่

2.2.2.6 คุณภาพการให้บริการ (Quality of Services levels)

เอ็มคิวทีทีมีคิวโอเอส 3 ระดับสำหรับการรับส่งข้อมูล ในขณะที่เอชทีทีพีจะไม่มีการส่งข้อมูลซ้ำเมื่อข้อมูลเกิดการสูญหายหรือคิวโอเอส ถ้านักพัฒนาต้องการการการันตีการส่งข้อมูล นักพัฒนาจำเป็นจะต้องพัฒนาขึ้นมาเอง

2.2.2.7 Libraries เพิ่มเติม (Extra libraries)

ด้วยความที่เอ็มคิวทีทีสามารถทำงานได้ดีในสภาพแวดล้อมที่อุปกรณ์มีหน่วยความจำที่จำกัด Library สำหรับการพัฒนาจึงมีขนาดที่เล็ก ในขณะที่เอชทีทีพีไม่จำเป็นต้องพึ่ง Library อื่น แต่ถ้าใช้ SOAP หรือเว็บเซอร์วิสแบบ RESTful ก็จำเป็นที่จะต้องเพิ่ม Library สำหรับการอ่าน JSON หรือ XML เข้าไปในระบบด้วย

2.2.2.8 การกระจายข้อมูล (Data distribution)

เอ็มคิวทีทีมีกระบวนการในการกระจายข้อมูลในตัวอยู่แล้ว โดยรองรับทั้งการส่งข้อมูลแบบหนึ่งต่อศูนย์ (การส่งข้อมูลจากพับบลิชเชอร์ไปยังโบรกเกอร์ โดยที่ยังไม่มีการเชื่อมต่อระหว่างซับสไครบเบอร์และโบรกเกอร์) หนึ่งต่อหนึ่ง และหนึ่งต่อหลายอุปกรณ์ ในขณะที่เอชทีทีพีมีการรับส่งข้อมูลในรูปแบบเดียวคือจุดต่อจุด หากต้องการให้เอชทีทีทำการส่งข้อมูลในรูปแบบอื่น นักพัฒนาจำเป็นที่จะต้องพัฒนากลไกการส่งข้อมูลเอง

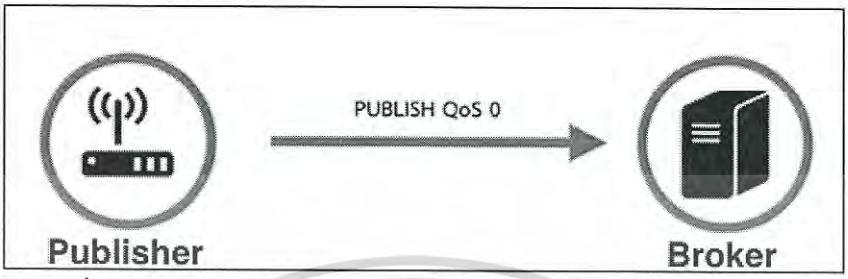
2.2.3 คุณภาพการบริการ (QoS : Quality of Service)

โดยปกติแล้วคุณภาพการบริการนั้นจะหมายถึง ความสามารถในการจัดการเครือข่ายให้มีประสิทธิภาพที่ดีขึ้น ไม่ว่าจะเป็นการจัดการลำดับความสำคัญของข้อมูล การควบคุม Delay ของเครือข่าย เป็นต้น [21] แต่ในโพรโทคอลเอ็มคิวทีทีนี้จะหมายถึงรูปแบบของการส่งข้อมูลระหว่างผู้ส่งกับผู้รับว่าจะให้มีการส่งในลักษณะใด การส่งข้อมูลด้วยโพรโทคอลนี้จะเป็นไปตามระดับคุณภาพการให้บริการที่ถูกกำหนดไว้ โดยกลไกนี้จะเป็นตัวบ่งบอกถึงวิธีการในการจัดการปัญหาเมื่อการส่งข้อมูลจากอุปกรณ์ที่ส่งข้อมูลไปยังอุปกรณ์ที่รับข้อมูลเกิดการสูญหาย (จากพับบลิชเชอร์ไปยังโบรกเกอร์ หรือจากโบรกเกอร์ไปซับสไครบเบอร์) หากคุณภาพการบริการอยู่ในระดับที่ต่ำ การส่งข้อมูลซ้ำเมื่อเกิดปัญหาการสูญหายของข้อมูลก็จะไม่เกิดขึ้น แต่ในทางตรงกันข้าม หากคุณภาพการบริการนี้อยู่ในระดับที่สูง โอกาสการส่งข้อมูลสำเร็จก็จะมากตามไปด้วย ซึ่งจะเห็นได้ว่าคุณภาพการบริการในเอ็มคิวทีทีโพรโทคอลนั้นจะมีนิยามที่แตกต่างจากนิยามคุณภาพการบริการโดยปกติ โดยคุณภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การให้บริการสำหรับโพรโทคอลนี้จะมีทั้งหมด 3 ระดับด้วยกันคือ คิวโอเอสศูนย์ (QoS 0) คิวโอเอสหนึ่ง (QoS 1) และคิวโอเอสสอง (QoS 2) [2][4][5]

2.2.3.1 คิวโอเอสศูนย์



รูปที่ 2.6 การส่งข้อมูลด้วยคุณภาพการบริการในคิวโอเอสระดับศูนย์

จากรูปที่ 2.7 แสดงถึงการส่งข้อมูลด้วยคุณภาพการบริการในระดับคิวโอเอสศูนย์ ซึ่งจะเห็นได้ว่าการส่งข้อมูลจากอุปกรณ์ที่ส่งข้อมูลไปยังอุปกรณ์ที่รับข้อมูลจะเป็นการส่งเพียงครั้งเดียวและไม่มีการตอบกลับผล ดังนั้นหากข้อมูลเกิดการสูญหาย จะไม่เกิดการส่งข้อมูลซ้ำของข้อมูล

ตารางที่ 2.2 กระแสข้อมูลของคิวโอเอสศูนย์

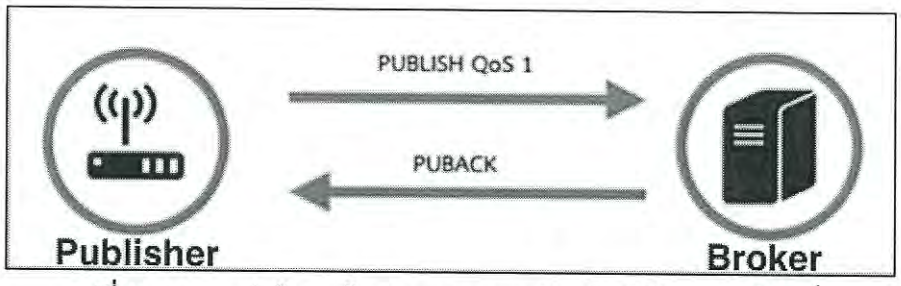
Sender Action	Control Packet	Receiver Action
Publish QoS 0, DUP=0		
	---------->	
		Deliver Application Message to appropriate on ward recipient(s)

จากตารางที่ 2.2 [2] แสดงถึงกระแสข้อมูลที่เกิดจากการส่งข้อมูลในคิวโอเอสระดับศูนย์ ซึ่งจะมีลักษณะการส่งข้อมูลที่เหมือนกับรูปที่ 2.3 โดยหากการผู้ส่งเป็นพับบลิชเชอร์ ผู้รับจะเป็นโบรกเกอร์ และหากผู้ส่งเป็นโบรกเกอร์ ผู้รับก็จะเป็นซับสไครเบอร์ ข้อดีของกระบวนการนี้ คือสามารถส่งข้อมูลได้รวดเร็วกว่าคิวโอเอสระดับอื่น เนื่องจากไม่ต้องรอข้อความตอบกลับ แต่ข้อเสียคือจะไม่สามารถส่งข้อมูลซ้ำได้ หากข้อมูลเกิดการสูญหายระหว่างทาง

แพ็คเกจที่ส่งข้อมูลด้วยคิวโอเอสศูนย์จะมีการระบุระดับของคิวโอเอสและ DUP Flag ซึ่งเป็นค่าที่บอกว่าแพ็คเกจนี้คือแพ็คเกจที่เกิดจากการส่งซ้ำหรือไม่ ดังนั้นที่คิวโอเอสที่ระดับนี้จะมีการตั้งค่า QoS=0 และ DUP=0 เสมอ เนื่องจากไม่เกิดการส่งซ้ำของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.2 คิวโอเอสหนึ่ง



รูปที่ 2.7 การส่งข้อมูลด้วยคุณภาพการบริการในคิวโอเอสระดับหนึ่ง

จากรูปที่ 2.8 แสดงถึงการส่งข้อมูลด้วยคุณภาพการบริการในคิวโอเอสระดับหนึ่ง โดยจะเห็นได้ว่าเมื่ออุปกรณ์ที่ส่งข้อมูลทำการส่งข้อมูลไปยังอุปกรณ์ที่รับข้อมูลแล้ว จะเกิดการรอข้อความตอบกลับ หากไม่ได้รับข้อความตอบกลับนี้ อุปกรณ์ที่ส่งข้อมูลจะทำการส่งข้อมูลซ้ำกลับไปอุปกรณ์ที่รับข้อมูลอีกครั้ง

ตารางที่ 2.3 กระแสข้อมูลของคิวโอเอสหนึ่ง

Sender Action	Control Packet	Receiver Action
Store message		
Publish QoS 1, DUP=0, <Packet Identifier>		
	---------->	
		Initiate onward delivery of the Application Message
		Send PUBACK <Packet Identifier>
	<----------	
Discard message		

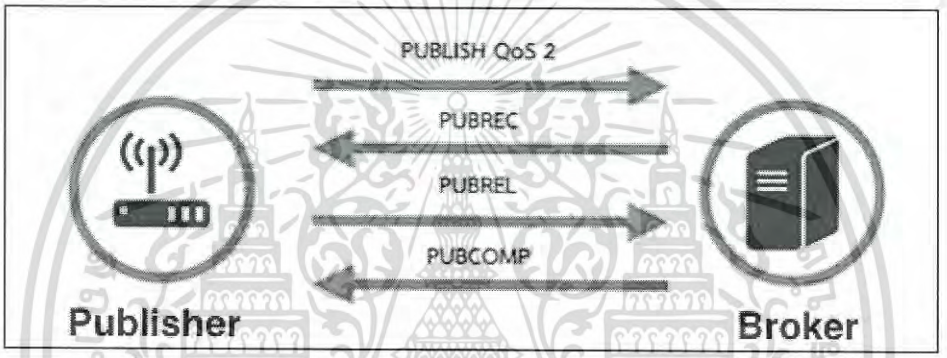
ตารางที่ 2.3 แสดงถึงกระแสข้อมูลที่เกิดจากการส่งข้อมูลที่ระดับคิวโอเอสหนึ่ง โดยหากการผู้ส่งเป็นพับบลิชเชอร์ ผู้รับจะเป็นโบรกเกอร์ และหากผู้ส่งเป็นโบรกเกอร์ ผู้รับก็จะเป็นซับสไครเบอร์ จะเห็นได้ว่าเมื่ออุปกรณ์ที่ส่งข้อมูลทำการส่งข้อมูลไปยังอุปกรณ์ที่รับข้อมูล จะเกิดการรอผลลัพธ์ตอบกลับจากโบรกเกอร์เพื่อเป็นการยืนยันว่าข้อมูลถึงที่หมาย เป็นผลให้อุปกรณ์ที่ส่งข้อมูลสามารถรับรู้ถึงสถานะของข้อมูลได้ แต่ถ้าหากข้อความที่ตอบกลับมานี้เกิดการสูญหายโดยที่ข้อมูลส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงอุปกรณ์ที่รับข้อมูลแล้ว ก็จะกลายเป็นว่าเกิดการส่งซ้ำของข้อมูลขึ้น ซึ่งจะทำให้อุปกรณ์ที่รับข้อมูลได้รับข้อมูลซ้ำสองรอบ แพ็คเกจที่ส่งข้อมูลด้วยคิวโอเอสหนึ่งจะมีการระบุระดับของคิวโอเอส DUP Flag และลำดับหมายเลขของแพ็คเกจ (<Packet Identifier>) โดยหากอุปกรณ์ที่ส่งข้อมูลไม่ได้รับข้อความตอบกลับ (PUBACK) อุปกรณ์ที่ส่งข้อมูลจะส่งข้อมูลกลับไปยังโบรกเกอร์อีกครั้งโดยตั้งค่า DUP=1

การส่งข้อมูลในคิวโอเอสที่ระดับนี้อาจจะมีระยะเวลาที่ยาวนานกว่าคิวโอเอสศูนย์ เนื่องจากอุปกรณ์ที่ส่งข้อมูลจำเป็นที่จะต้องรอผลลัพธ์ตอบกลับก่อนที่จะส่งข้อมูลชุดต่อไป ซึ่งถ้าหากข้อมูลเกิดการสูญหาย ก็จำเป็นที่จะต้องใช้เวลาเพิ่มสำหรับการส่งข้อมูลซ้ำ

2.2.3.3 คิวโอเอสสอง



รูปที่ 2.8 การส่งข้อมูลด้วยคุณภาพการบริการในคิวโอเอสระดับสอง

จากรูปที่ 2.9 แสดงถึงการส่งข้อมูลด้วยคุณภาพการบริการในคิวโอเอสระดับสอง โดยจะเห็นว่าอุปกรณ์ที่ส่งข้อมูลจะใช้หลักการส่งข้อมูลแบบการตอบกลับ 4 ทิศทาง (4-way handshake) เพื่อป้องกันปัญหาการสูญหายของข้อความตอบกลับที่เกิดขึ้นในคิวโอเอสหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 กระแสข้อมูลของคิวโอเอสหนึ่ง

Sender Action	Control Packet	Receiver Action
Store message		
Publish QoS 2, DUP=0, <Packet Identifier>		
	---------->	
		Method A, Store message or Method B, Store <Packet Identifier> then Initiate onward delivery of the application message
		PUBREC <Packet Identifier>
	<----------	
Discard message, Store PUBREC received <Packet Identifier>		
PUBREL <Packet Identifier>		
	---------->	
		Method A, Initiate onward delivery of the application message then discard message or Method B, Discard <Packet Identifier>
	<----------	
Discard stored state		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

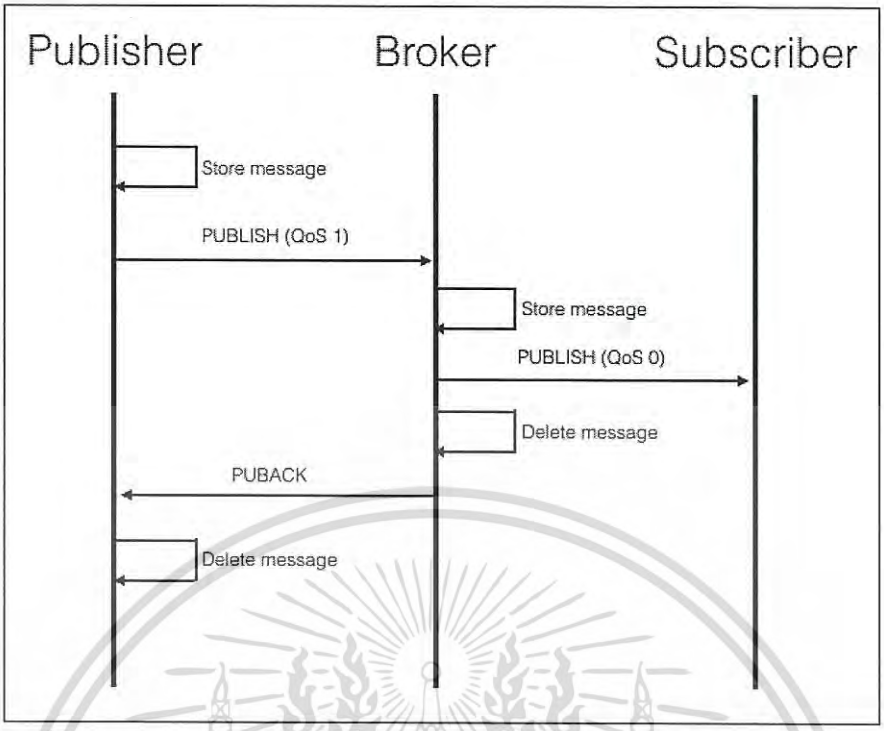
ตารางที่ 2.4 แสดงถึงกระแสข้อมูลของคิวโอเอสสอง โดยหากการผู้ส่งเป็นพับบลิชเชอร์ ผู้รับจะเป็นโบรกเกอร์ และหากผู้ส่งเป็นโบรกเกอร์ ผู้รับก็จะเป็นซับสไครเบอร์ จะเห็นได้ว่าเมื่ออุปกรณ์ที่ส่งข้อมูลทำการส่งข้อมูลไปยังอุปกรณ์ที่รับข้อมูล จะเกิดการรอผลลัพธ์ตอบรับ ก่อนที่จะส่งข้อความตอบกลับไปยังอุปกรณ์ที่รับข้อมูลอีกหนึ่งครั้งเพื่อป้องกันการส่งข้อมูลซ้ำ เมื่ออุปกรณ์ที่รับข้อมูลได้รับข้อความตอบกลับจากอุปกรณ์ที่ส่งข้อมูล อุปกรณ์ที่รับข้อมูลก็จะส่งข้อความกลับไปอีกหนึ่งครั้ง เพื่อเป็นการบอกอุปกรณ์ที่ส่งข้อมูลว่าได้รับข้อความตอบกลับแล้ว และไม่เกิดการสูญหายของข้อมูลขึ้น การส่งข้อมูลแบบนี้จึงเป็นการช่วยป้องกันการส่งข้อมูลซ้ำและยังช่วยให้โอกาสการส่งข้อมูลสำเร็จมีมากขึ้นด้วย แพ็คเกจที่ส่งข้อมูลด้วยคิวโอเอสสองจะมีการระบุระดับของคิวโอเอส DUP Flag และลำดับหมายเลขของแพ็คเกจ เช่นเดียวกับการส่งข้อมูลด้วยคิวโอเอสหนึ่ง เมื่อเกิดปัญหาระหว่างการส่งข้อมูล อุปกรณ์ที่ส่งข้อมูลจะส่งข้อมูลซ้ำโดยการระบุ DUP=1 ไปยังโบรกเกอร์

แม้ว่าการส่งข้อมูลแบบการตอบกลับ 4 ทิศทางนี้จะสามารถช่วยในการแก้ปัญหาการส่งข้อมูลซ้ำที่เกิดจากการสูญหายของข้อความตอบกลับ แต่การรับส่งข้อมูลระหว่างอุปกรณ์ที่ส่งข้อมูลกับอุปกรณ์ที่รับข้อมูลก็เกิดขึ้นหลายครั้ง การส่งข้อมูลที่คิวโอเอสระดับนี้จึงมีโอกาสที่จะส่งข้อมูลได้ล่าช้ากว่าการส่งข้อมูลด้วยคิวโอเอสที่ระดับอื่น

2.2.3.4 การส่งข้อมูลด้วยคิวโอเอสของเอ็มคิวทีทีโพรโทคอล

การส่งข้อมูลด้วยคิวโอเอสของเอ็มคิวทีทีโพรโทคอลจะถูกแบ่งออกเป็นสองส่วนหลักคือ การส่งข้อมูลจากพับบลิชเชอร์ไปยังโบรกเกอร์ และจากโบรกเกอร์ไปยังซับสไครบเบอร์ โดยการส่งข้อมูลสองส่วนนี้ไม่จำเป็นที่จะต้องใช้คิวโอเอสที่ระดับเดียวกันเสมอไป กล่าวคือ คิวโอเอสที่ถูกระบุที่พับบลิชเชอร์ก่อนการส่งข้อมูลจะเป็นการใช้งานเพียงแค่พับบลิชเชอร์กับโบรกเกอร์เท่านั้น ในขณะที่คิวโอเอสของซับสไครบเบอร์จะถูกระบุไว้ขณะที่ซับสไครบเบอร์ทำการซับสไครบหัวข้อมายังโบรกเกอร์ ซึ่งคิวโอเอสของซับสไครบเบอร์นั้นจะเป็นคิวโอเอสที่มากที่สุดที่โบรกเกอร์สามารถส่งข้อมูลมายังซับสไครบเบอร์ได้

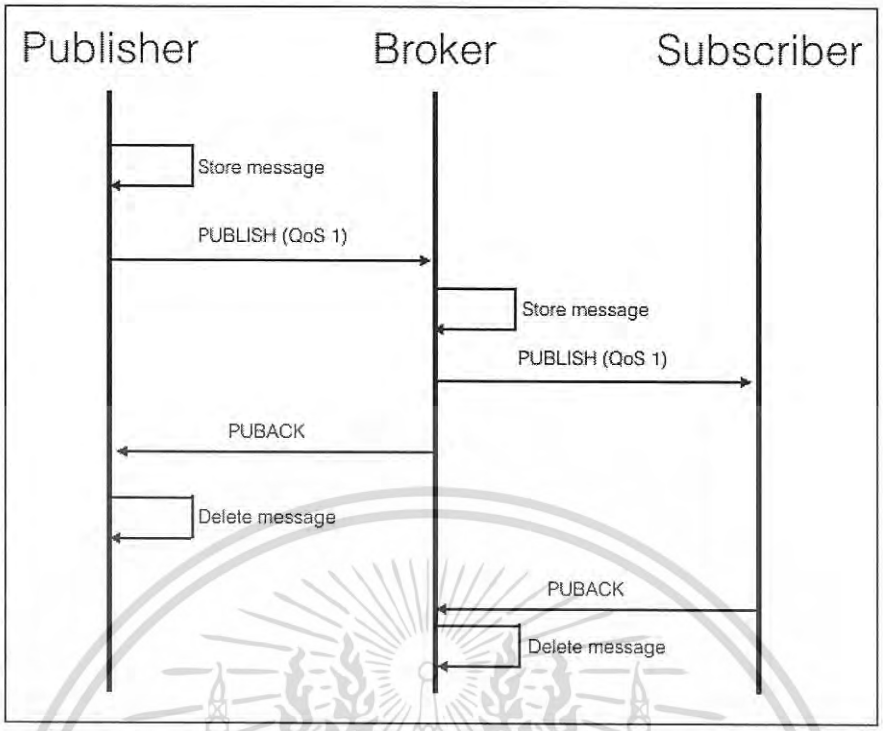
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การพับบลิชข้อมูลด้วยคิวโอเอสหนึ่งและซับสไครบด้วยคิวโอเอสศูนย์

รูปที่ 2.9 แสดงถึงการพับบลิชข้อมูลด้วยคิวโอเอสหนึ่งและซับสไครบด้วยคิวโอเอสศูนย์ จะเห็นได้ว่า แม้พับบลิชเชอร์จะระบุคิวโอเอสของการส่งข้อมูลไว้เป็นระดับที่หนึ่ง แต่การส่งข้อมูลจากพับบลิชเชอร์ไปยังซับสไครบเบอร์จะไม่ใช่การส่งข้อมูลแบบคิวโอเอสหนึ่งตลอดเส้นทาง เนื่องจากซับสไครบเบอร์ทำการซับสไครบเบอร์โบรกเกอร์ไว้ที่คิวโอเอสศูนย์ การส่งข้อมูลจากโบรกเกอร์ไปยังซับสไครบเบอร์จึงเป็นการส่งในรูปแบบของคิวโอเอสศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 การพับบลิชและซับสไครบด้วยคิวโอเอสที่มากกว่าหรือเทียบเท่าหนึ่ง

รูปที่ 2.10 แสดงถึงการรับพับบลิชข้อมูลด้วยคิวโอเอสหนึ่งและซับสไครบด้วยคิวโอเอสที่มากกว่าหรือเทียบเท่าหนึ่ง ซึ่งการส่งในรูปแบบนี้จะเป็นการส่งด้วยคิวโอเอสหนึ่งตลอดเส้นทางตั้งแต่พับบลิชเชอร์ไปยังโบรกเกอร์ และจากโบรกเกอร์ไปยังซับสไครบเบอร์ ถึงแม้ว่าซับสไครบเบอร์จะทำการซับสไครบด้วยคิวโอเอสสอง แต่การส่งข้อมูลจากโบรกเกอร์ไปยังซับสไครบเบอร์ก็จะเป็นไปตามคิวโอเอสที่พับบลิชเชอร์ระบุไว้

กล่าวโดยสรุปคือ คิวโอเอสของซับสไครบเบอร์จะเป็นตัวกำหนดว่าโบรกเกอร์สามารถส่งข้อมูลด้วยคิวโอเอสสูงที่สุดได้ที่ระดับใด หากพับบลิชเชอร์ทำการส่งข้อมูลด้วยคิวโอเอสที่สูงกว่าที่ซับสไครเบอร์ระบุไว้ โบรกเกอร์จะทำการส่งต่อข้อมูลด้วยคิวโอเอสที่สูงที่สุดตามที่ซับสไครบเบอร์ระบุไว้ แต่หากพับบลิชเชอร์ทำการส่งข้อมูลด้วยคิวโอเอสระดับที่เทียบเท่าหรือน้อยกว่าคิวโอเอสที่ซับสไครบเบอร์ระบุไว้ โบรกเกอร์จะทำการส่งต่อข้อมูลตามระดับคิวโอเอสที่พับบลิชเชอร์เป็นคนระบุไปยังซับสไครบเบอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.3 Open-Source สำหรับการพัฒนาเอ็มคิวทีทีโพรโทคอล

2.3.1 Paho

เป็นหนึ่งโครงการ Open-Source ของโครงการ Eclipse [17] ที่นำโพรโทคอลการส่งข้อมูลที่เป็นมาตรฐานมาประยุกต์ใช้งานในระบบและแอพพลิเคชั่นที่มีลักษณะการทำงานในรูปแบบของการสื่อสารระหว่างเครื่องจักรและระบบอินเทอร์เน็ตของสรรพสิ่ง

เอ็มคิวทีทีเป็นโพรโทคอลที่ถูกบรรจุลงในโครงการ Paho เพื่อการประยุกต์ใช้งานในแพลตฟอร์มอุปกรณ์สมองกลฝังตัว โดยมีบริษัท IBM และชุมชนนักพัฒนาช่วยในการสนับสนุนโครงการอย่างต่อเนื่อง จนในปัจจุบัน Paho กลายเป็น Library สำหรับเอ็มคิวทีทีโพรโทคอลที่สามารถนำไปประยุกต์ใช้ได้ถึง 9 รูปแบบในภาษาต่าง ๆ อันได้แก่

1) C for Posix and Windows
2) C++
3) Embedded C/C++
4) Java
5) Android Service
6) JavaScript
7) Python
8) GO
9) C# .Net and WinRT

ในปัจจุบัน Paho จะมีเวอร์ชั่นล่าสุดอยู่ที่ 1.1 โดยในงานวิทยานิพนธ์นี้จะทำการสร้างลูกข่ายของเอ็มคิวทีทีด้วย Paho ในภาษา Python เวอร์ชั่น 2.7 (สนับสนุน Python เวอร์ชั่น 2.6 2.7 และ 3 ขึ้นไป) ผู้ใช้สามารถที่จะติดตั้งผ่าน Python Package (PyPi) หลังจากนั้นจะสามารถเรียกใช้งานฟังก์ชั่นต่าง ๆ ของ Paho ในฐานะของ Python Library ได้ โดยในการส่งข้อมูลนั้นจะมีรูปแบบเดียวกับมาตรฐานของเอ็มคิวทีที (Frame Format เดียวกับการส่งข้อมูลด้วยโพรโทคอลเอ็มคิวทีที)

2.3.2 Mosquitto

เป็นหนึ่งในโครงการ Open-Source ของโครงการ Eclipse [18] ที่สร้างเครื่องแม่ข่ายสำหรับเอ็มคิวทีทีโพรโทคอลที่เหมาะสมกับสถานการณ์อุปกรณ์และเซ็นเซอร์ต่าง ๆ นอกจากนั้น Mosquitto ยังสามารถทำตัวเองให้เป็นสะพานเพื่อเชื่อมต่อกับเครื่องแม่ข่ายที่รองรับเอ็มคิวทีทีโพรโทคอลอื่น เพื่อให้เกิดการส่งข้อมูลสถานที่หนึ่งไปยังอีกสถานที่หนึ่งได้

ในปัจจุบัน Mosquitto จะมีเวอร์ชั่นล่าสุดอยู่ที่ 1.3.4-2 ซึ่งผู้ใช้สามารถติดตั้งผ่าน Linux Package หลังจากนั้น Mosquitto จะเริ่มต้นการทำงานเป็นโบรกเกอร์อัตโนมัติ โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

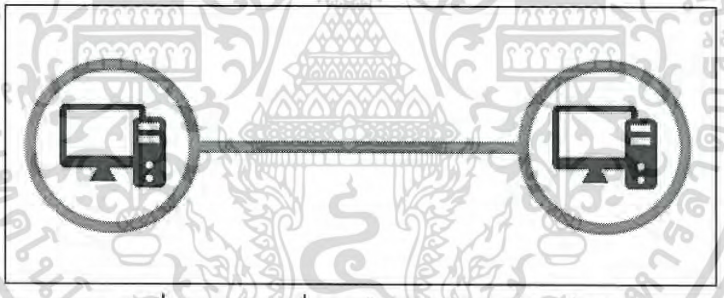
148792

วิทยานิพนธ์นี้จะใช้ค่าตั้งต้นที่ติดตั้งมาพร้อม Package ของ Mosquitto ในการวิจัย เมื่อ Mosquitto ได้รับข้อมูลเอ็มคิวทีที จะเกิดการส่งต่อข้อมูลไปยังซับสไครบเบอร์ทุกตัวที่ทำการเชื่อมต่ออยู่ ณ ขณะนั้นด้วยรูปแบบการส่งข้อมูลตามมาตรฐานของเอ็มคิวทีที (Frame Format เดียวกับการส่งข้อมูลด้วยโพรโทคอลเอ็มคิวทีที)

2.4 การสื่อสารภายในเครือข่ายเซ็นเซอร์ไร้สาย

การสื่อสารภายในเครือข่ายเซ็นเซอร์ไร้สาย (Wireless Sensor Network) [6] คือ การนำอุปกรณ์เซ็นเซอร์หรือโหนดจำนวนมากมาเชื่อมต่อกันเป็นระบบเพื่อให้อุปกรณ์เหล่านั้นเกิดการทำงานและตรวจสอบสถานะร่วมกัน ซึ่งจะสามารถทดแทนข้อจำกัดทางด้านทรัพยากรบางอย่างของแต่ละโหนดได้ เช่น ข้อจำกัดด้านพลังงาน การประมวลผล และพื้นที่หน่วยความจำ เป็นต้น การสื่อสารภายในเครือข่ายนี้มักจะถูกแบ่งเป็น 2 ประเภทคือ การสื่อสารในรูปแบบซิงเกิล-ฮ็อป (Single-Hop Communication) และการสื่อสารในรูปแบบมัลติ-ฮ็อป (Multi-Hop Communication) โดยมีรายละเอียดดังนี้

2.4.1 การสื่อสารในรูปแบบซิงเกิล-ฮ็อป

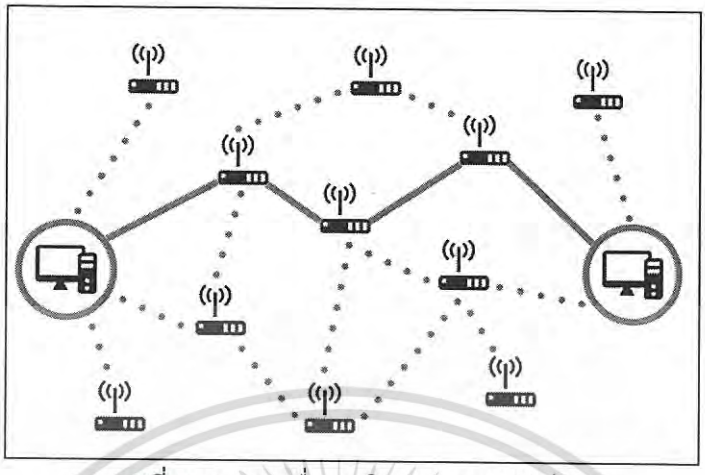


รูปที่ 2.11 การสื่อสารในรูปแบบซิงเกิล-ฮ็อป

การสื่อสารในรูปแบบซิงเกิล-ฮ็อป เป็นการสื่อสารที่ข้อมูลจะถูกส่งจากอุปกรณ์ต้นทางไปยังอุปกรณ์ปลายทางได้โดยตรงโดยไม่ผ่านอุปกรณ์หรือโหนดอื่น ดังนั้นการสื่อสารในรูปแบบซิงเกิล-ฮ็อปจึงมักจะถูกนำไปใช้ในระบบที่อุปกรณ์มีระยะทางไม่ห่างกันมากนัก อย่างเช่นการสื่อสารระหว่างเครื่องจักร เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 การสื่อสารในรูปแบบมัลติ-ฮ็อป



รูปที่ 2.12 การสื่อสารในรูปแบบมัลติ-ฮ็อป

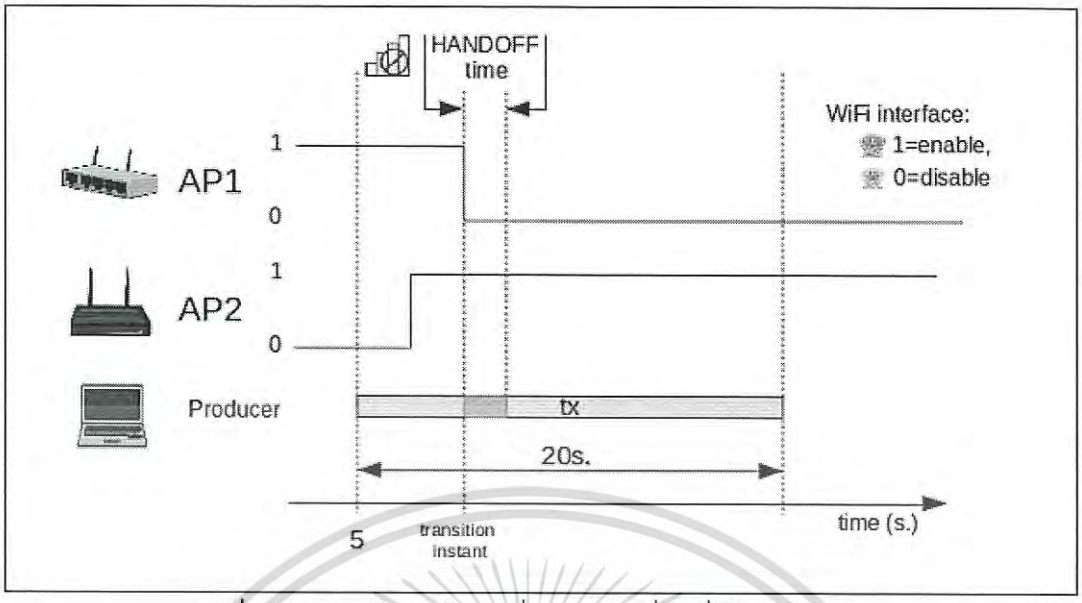
การสื่อสารในรูปแบบมัลติ-ฮ็อป เป็นการสื่อสารที่ข้อมูลจะถูกส่งจากต้นทางไปยังปลายทางโดยอาศัยการเชื่อมต่อผ่านทางโหนดอื่น ดังนั้นการสื่อสารในรูปแบบนี้จึงมักจะต้องมีจำนวนของอุปกรณ์หรือโหนดสำหรับการส่งต่อข้อมูลที่มากกว่า 2 อุปกรณ์ขึ้นไป ข้อดีของการสื่อสารแบบนี้คือ การส่งข้อมูลจากต้นทางไปยังปลายทางสามารถทำจากระยะไกลได้ เนื่องจากอุปกรณ์ที่ส่งและอุปกรณ์ที่รับข้อมูลไม่จำเป็นที่จะต้องอยู่ในสถานที่ที่ใกล้กัน ขอเพียงแค่มีเส้นทางที่สามารถเชื่อมต่อถึงกันได้ก็พอ

2.5 งานวิจัยที่เกี่ยวข้อง

ในปัจจุบันมีงานวิจัยหลายชิ้นที่เกี่ยวข้องกับการนำเอ็มคิวทีทีโพรโทคอลมาวัดผลและเปรียบเทียบ โดยมีรายละเอียดดังนี้

Luzuriaga และคณะ [3] ได้ศึกษาและทดลองเกี่ยวกับพฤติกรรมการส่งข้อมูลของเอเอ็มคิวพีโพรโทคอล (AMQP protocol) และเอ็มคิวทีทีโพรโทคอล ในสภาพแวดล้อมที่การเชื่อมต่อไม่มีเสถียรภาพ เช่น เครือข่ายบนยานพาหนะ เป็นต้น ข้อแตกต่างของโพรโทคอลทั้งสองนี้คือ เอเอ็มคิวพีโพรโทคอลจะรองรับสถานการณ์ที่หลากหลายโดยใช้รูปแบบการส่งทั้งแบบจุดต่อจุด (point-to-point) และรูปแบบการเก็บและส่งต่อข้อมูล (store-and-forward) ในขณะที่เอ็มคิวทีทีโพรโทคอลใช้หลักการส่งโดยอ้างอิงจากหัวข้อของข้อความและไม่ใช้ความน่าจะเป็นช่วยในการเก็บและส่งต่อข้อมูล งานวิจัยนี้ได้ทำการจำลองภาระงานที่มีปริมาณมากเพื่อทดสอบประสิทธิภาพของโพรโทคอล เช่น ความหลากหลายของความเหลื่อมล้ำของเวลา (jitter) และการสูญหายของข้อมูล (information loss)

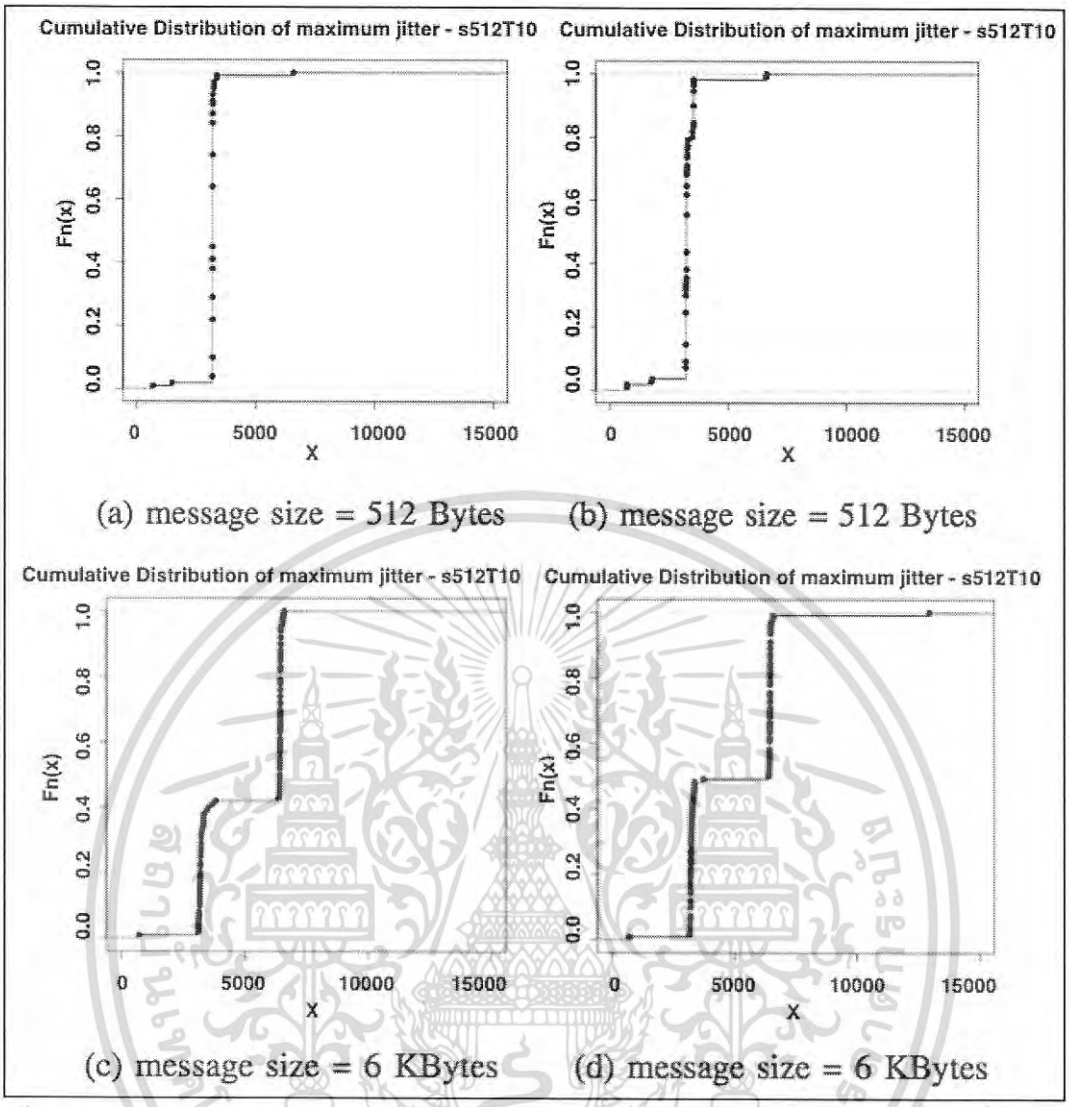
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 การจำลองโหนดที่สามารถเคลื่อนที่ได้ภายในอาคาร

การทดลองสำหรับงานวิจัยนี้จะทำการทดสอบผลเป็นเวลา 20 วินาที โดยจะมีการตั้งค่าโปรแกรมสำหรับเปิดปิดการทำงานของเราเตอร์และการทำงานร่วมกันของอุปกรณ์ทั้งหมดในระบบตามรูปที่ 2.13 [3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 Cumulative distribution function ของค่าความเหลื่อมล้ำของเวลาที่มากที่สุดในระหว่างการส่งข้อมูลระยะเวลา 10 มิลลิวินาที โดยใช้เอเอ็มคิวพี (ซ้าย) และเอ็มคิวทีที (ขวา)

รูปที่ 2.14 [3] แสดงถึง distribution function ของค่าความเหลื่อมล้ำของเวลาโดยใช้ระยะเวลาการส่งข้อมูลที่ 10 มิลลิวินาที และข้อความมีขนาดที่ 512 ไบท์ และ 6 กิโลไบท์ เมื่อขนาดของข้อความอยู่ที่ 512 ไบท์ ค่าความเหลื่อมล้ำของเวลาของทั้งสองโพรโทคอลจะมีค่าอยู่ที่ประมาณ 3.3 วินาที โดยจะมีบางครั้งที่ค่าความเหลื่อมล้ำของเวลาสูงถึง 7 วินาที เมื่อขนาดของข้อความมากขึ้น ค่าความเหลื่อมล้ำของเวลาในครึ่งแรกยังคงอยู่ในช่วงประมาณ 3.3 วินาที ในขณะที่ครึ่งหลังค่าความเหลื่อมล้ำของเวลากลับเพิ่มเป็นสองเท่า ซึ่งเป็นผลจากการแบ่งข้อมูลออกเป็นส่วน (fragmentation ของ payload) ของทั้งสองโพรโทคอล

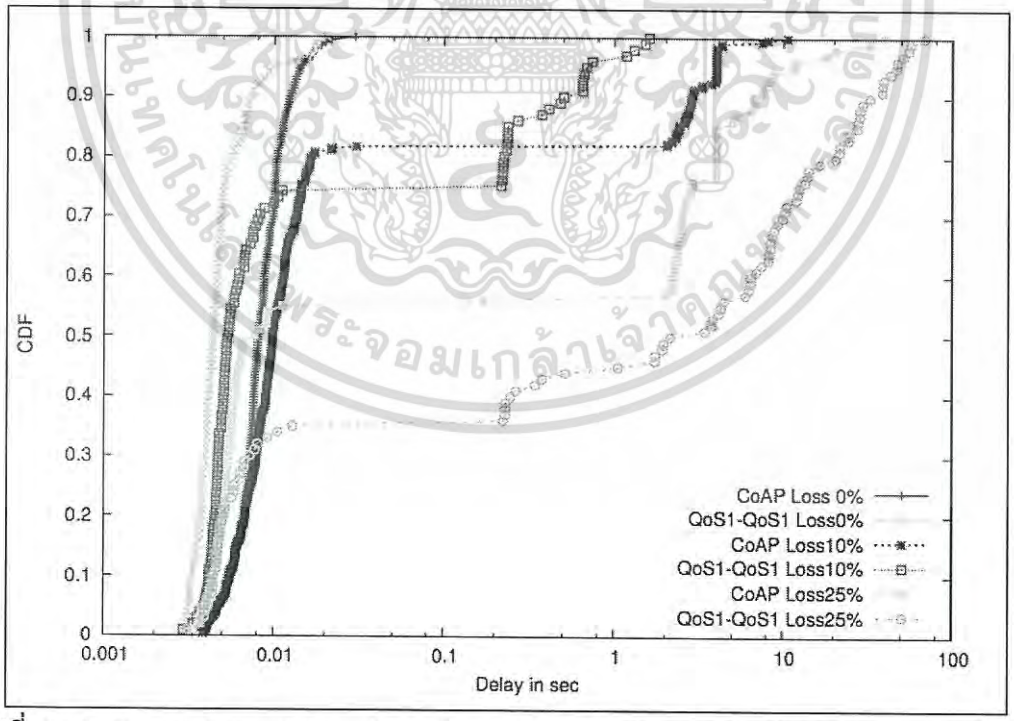
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อปริมาณการส่งข้อมูลสูงขึ้น เอเอ็มคิวพีโพรโทคอลจะส่งข้อมูลในลักษณะของ LIFO (Last-Input First-Output) ซึ่งเป็นผลจากการสลับที่ของข้อมูล ในขณะที่เหตุการณ์นี้ไม่เกิดขึ้นกับการส่งข้อมูลในเอ็มคิวทีทีโพรโทคอล

Thangavel และคณะ [7] ได้นำเสนองานวิจัยที่เกี่ยวข้องกับการออกแบบและสร้าง Middleware ที่สนับสนุนโพรโทคอลเอ็มคิวทีทีและซีโอเอพี (CoAP protocol) สำหรับการทดลอง ได้ทำการศึกษาประสิทธิภาพของทั้งสองโพรโทคอลในเรื่องของ end-to-end delay และปริมาณการใช้แบนด์วิดธ์ (bandwidth consumption)

ตารางที่ 2.5 ค่าเฉลี่ยของ Delay สำหรับอัตราการสูญหายของข้อมูลที่ระดับต่าง ๆ

	Loss rate 0%	Loss rate 5%	Loss rate 10%	Loss rate 15%	Loss rate 20%	Loss rate 25%
MQTT (QoS1-QoS1)	0.005289s	0.022609s	0.144252s	0.450860s	0.871025s	10.33348s
CoAP	0.008847s	0.552006s	0.642688s	1.026007s	1.886399s	2.824115s

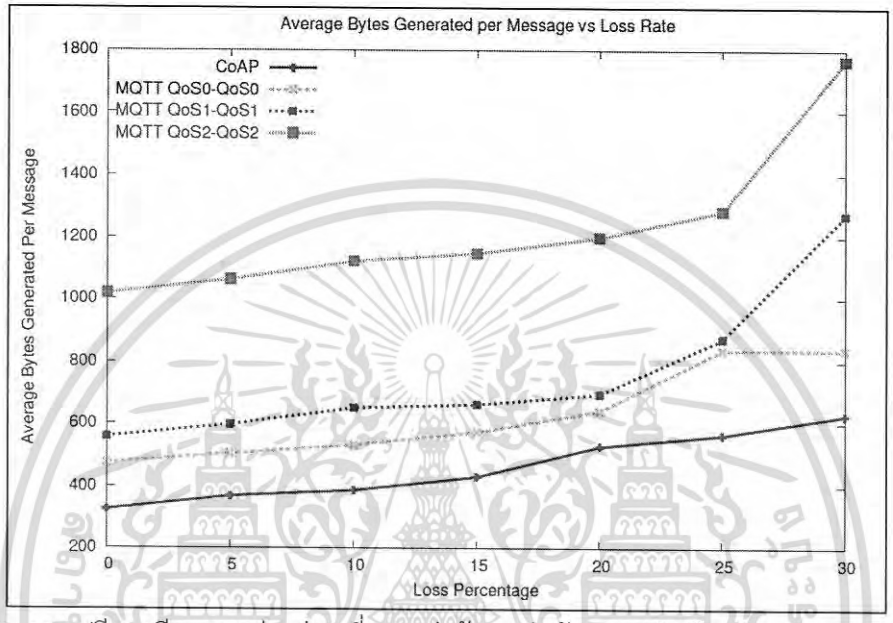


รูปที่ 2.15 Cumulative distribution function ของ Delay สำหรับการสูญหายของข้อมูล

ตาราง 2.5 [7] แสดงถึงค่าเฉลี่ยของ delay สำหรับการสูญหายของข้อมูล ในขณะที่รูปที่ 2.15 [7] แสดงถึง cumulative distribution function ของ delay ผลลัพธ์ของทั้งคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาผลประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงให้เห็นว่า delay ของเอ็มคิวทีทีจะมีค่าที่น้อยกว่าหากอัตราการสูญหายของข้อมูลมีปริมาณที่น้อย หากอัตราการสูญหายของข้อมูลมากขึ้น ซีโอเอพีจะได้ผลลัพธ์ที่ดีกว่า ซึ่งเป็นผลมาจากขนาด overheads ของทีซีพี (TCP) (โพรโทคอลที่ชั้น transport layer ของเอ็มคิวทีทีคือทีซีพี) ในกรณีที่ส่งข้อมูลซ้ำ จะมีขนาดที่มากกว่าเมื่อเปรียบเทียบกับขนาด overhead ของยูดีพี (UDP) ซึ่งเป็นโพรโทคอลที่ถูกใช้ใน transport layer ของซีโอเอพี เมื่ออัตราการสูญหายของข้อมูลสูงขึ้น

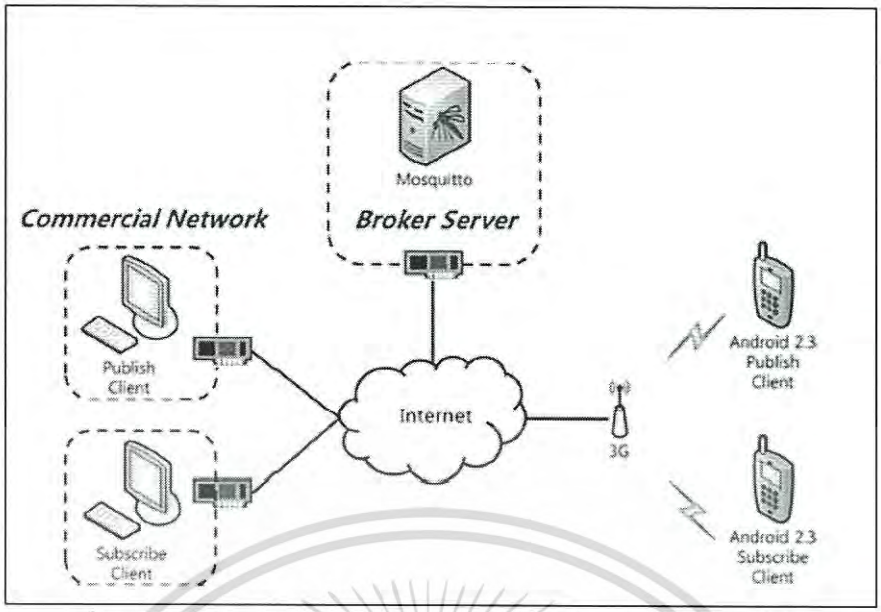


รูปที่ 2.16 การเปรียบเทียบระหว่างค่าเฉลี่ยการส่งข้อมูลต่อข้อความกับอัตราการสูญหายของข้อความ

รูปที่ 2.16 [7] แสดงถึงการเปรียบเทียบระหว่างค่าเฉลี่ยการส่งข้อมูลต่อข้อความกับอัตราการสูญหายของข้อมูล ข้อความของการส่งข้อมูลแบบคิวโอเอส 2 จะใช้แบนด์วิดธ์ที่มากกว่าหากเปรียบเทียบกับคิวโอเอส 0 และคิวโอเอส 1 เมื่ออัตราการสูญหายของข้อมูลมีปริมาณน้อย เนื่องจากกลไกการรับส่งสี่ทิศทาง (4-way handshake) ในคิวโอเอส 2 ในขณะที่คิวโอเอส 0 ในขณะที่คิวโอเอส 0 จะมีการรับส่งข้อมูลที่น้อยที่สุดเนื่องจากไม่ต้องรอผลตอบกลับจากเครื่องแม่ข่าย

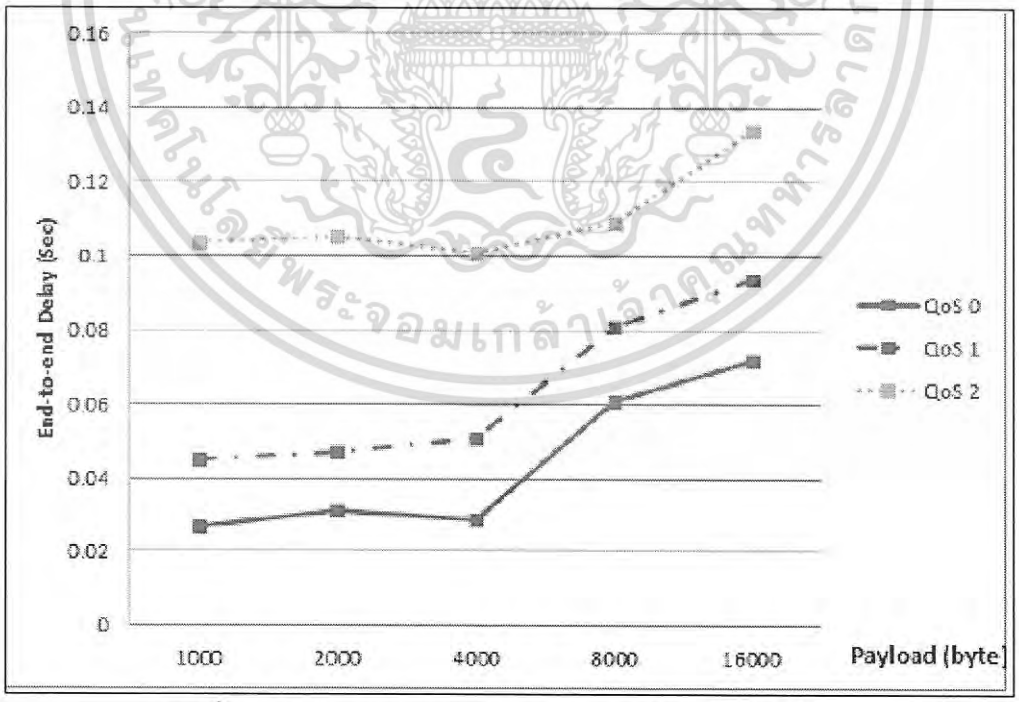
Lee S. และคณะ [8] ได้นำเสนองานวิจัยที่วิเคราะห์เกี่ยวกับ end-to-end delay และการสูญหายของข้อมูล จากการส่งข้อมูลที่ขนาดต่าง ๆ ด้วยกระบวนการคิวโอเอสทั้ง 3 ระดับของเอ็มคิวทีทีโพรโทคอลผ่านเครือข่ายมีสายและไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



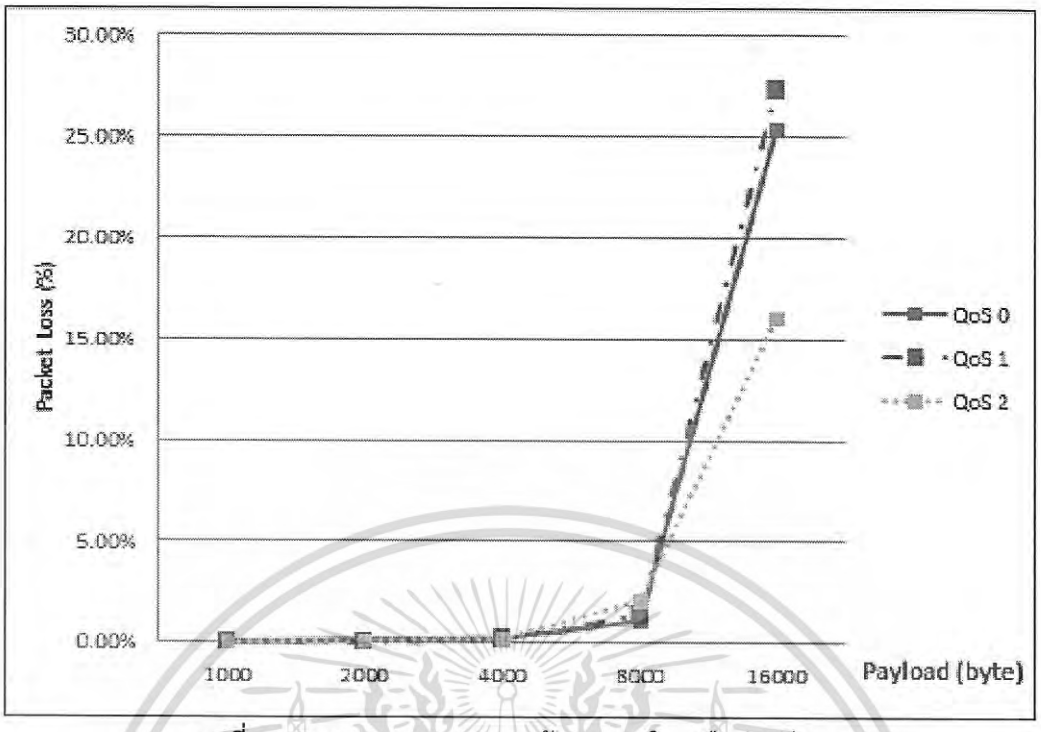
รูปที่ 2.17 องค์ประกอบการเชื่อมต่อของเครือข่ายแบบมีสายและไร้สาย

รูปที่ 2.17 [8] แสดงถึงองค์ประกอบการเชื่อมต่อของเครือข่ายแบบมีสายและไร้สาย โดยมีการตั้งโบรกเกอร์ไว้ภายนอกเครือข่ายเพื่อจำลองสภาพแวดล้อมที่ใกล้เคียงความเป็นจริง ในขณะที่การส่งข้อมูลแบบไร้สายจะใช้โทรศัพท์มือถือระบบปฏิบัติการ Android 2.3 ซึ่งเชื่อมต่อผ่านเครือข่าย 3G เพื่อเชื่อมต่อไปยังโบรกเกอร์



รูปที่ 2.18 ผล End-to-end Delay ภายในเครือข่ายมีสาย

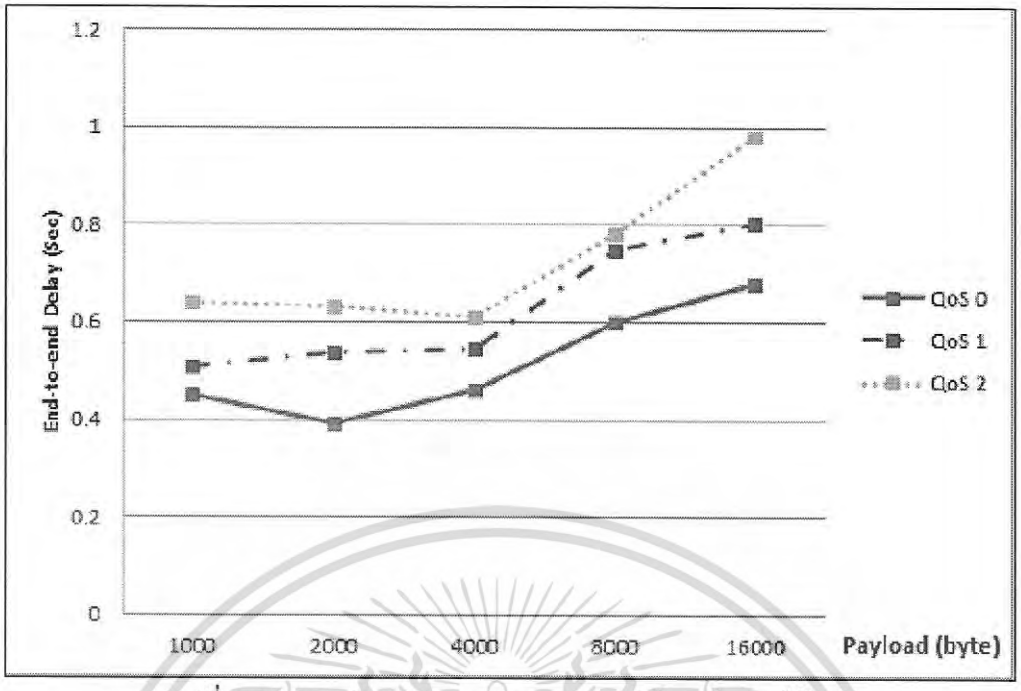
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



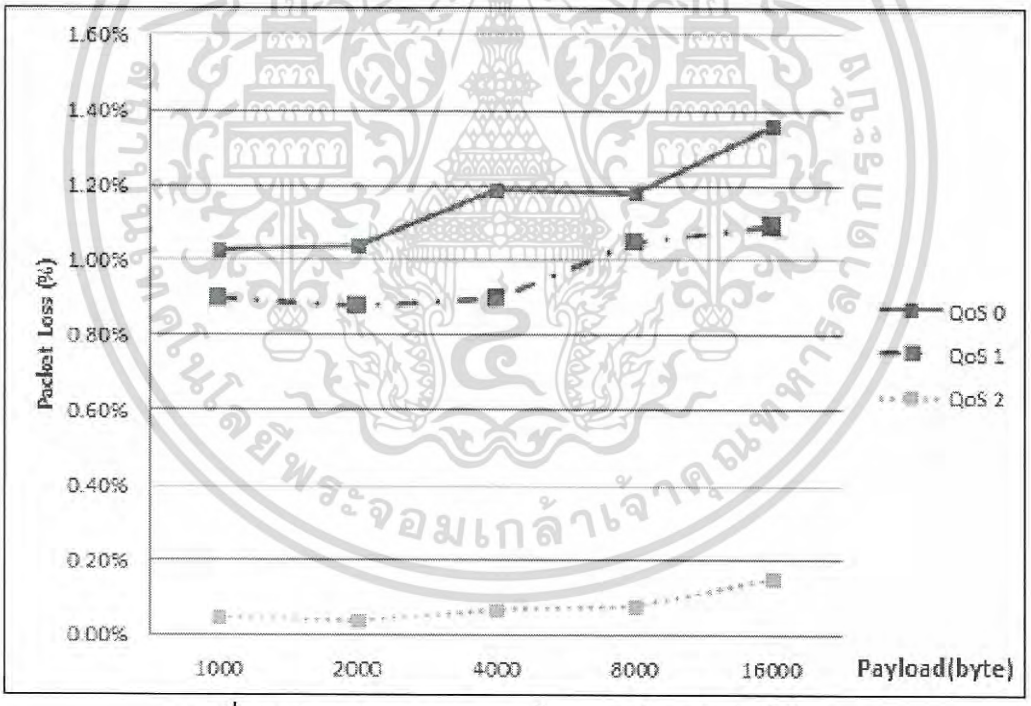
รูปที่ 2.19 ผลการสูญหายของข้อมูลภายในเครือข่ายมีสาย

รูปที่ 2.18 [8] แสดงถึง end-to-end delay ภายในเครือข่ายมีสาย โดยเมื่อขนาดของข้อมูลมีค่าที่มากกว่า 4,000 ไบท์ ปริมาณแพ็คเกจของข้อมูลที่ถูกแบ่งจะมีมากขึ้น เป็นผลให้ค่า delay ของกระบวนการคิวโอเอสทุกระดับจะมีค่าที่สูงขึ้น ในขณะที่รูปที่ 2.19 [8] แสดงถึงผลการสูญหายของข้อมูลภายในเครือข่ายมีสาย โดยสามารถสรุปได้ว่า เมื่อขนาดของข้อมูลมีค่าที่มากขึ้น คิวโอเอสสองสามารถช่วยลดการสูญหายของข้อมูลได้มากกว่าคิวโอเอส 1 และคิวโอเอส 2 อันเป็นผลมาจากการใช้การรับส่งแบบสี่ทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 ผล End-to-end Delay ภายในเครือข่ายไร้สาย

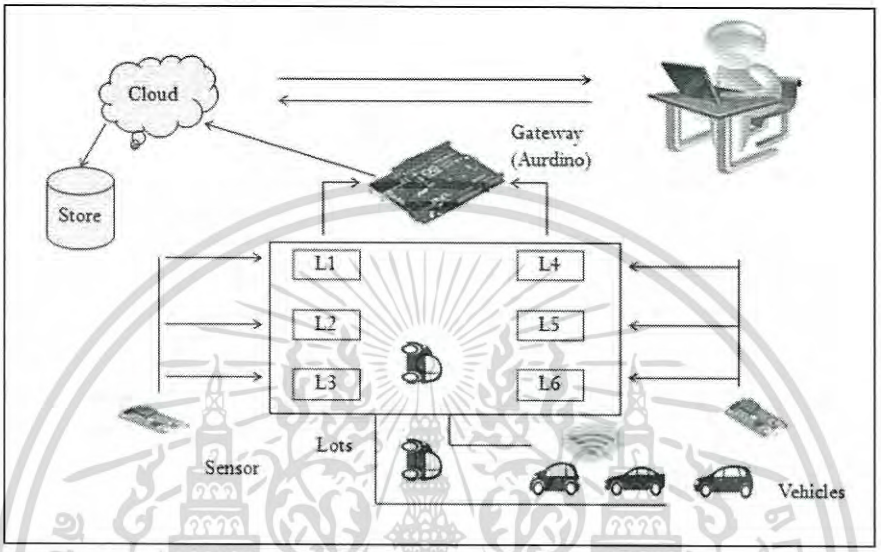


รูปที่ 2.21 ผลการสูญหายของข้อมูลภายในเครือข่ายไร้สาย

รูปที่ 2.20 [8] แสดงถึงผล end-to-end delay ภายในเครือข่ายไร้สาย โดยจะเห็นได้ว่าผลที่ได้จากเครือข่ายไร้สายจะมีค่าที่มากกว่าผลที่เกิดขึ้นในเครือข่ายมีสาย ในขณะที่รูปที่ 2.21 [8] แสดงถึงผลการสูญหายของข้อมูลภายในเครือข่ายไร้สาย ถึงแม้ว่าผลการสูญหายของข้อมูลจะค่อนข้างคงที่กว่าเมื่อเปรียบเทียบกับเครือข่ายที่มีสาย แต่ผลลัพธ์ของคิวโอเอส 2 ก็ยังคงมีค่าการสูญหายของข้อมูลที่น้อยกว่าคิวโอเอส 0 และคิวโอเอส 1

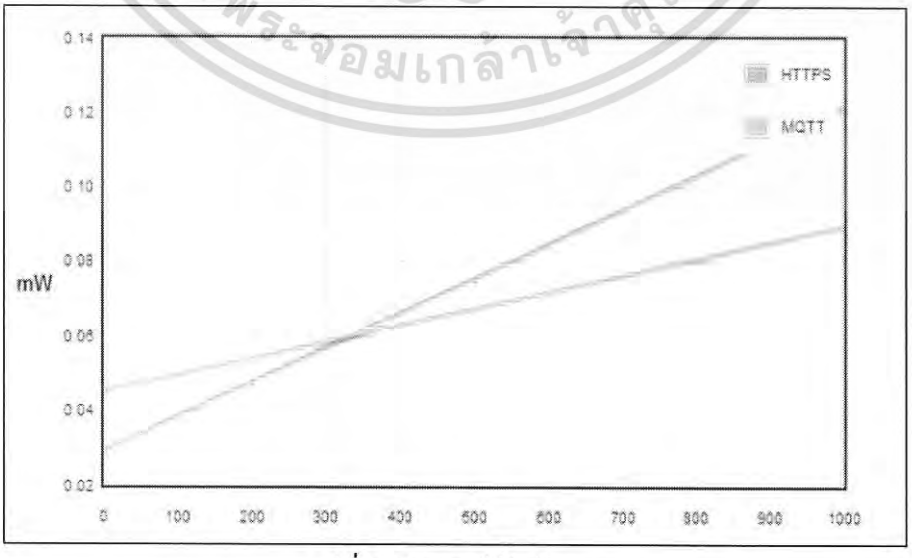
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dhar P. และคณะ [20] ได้นำเสนอแนวคิดการออกแบบเซอร์วิสบนเครื่องแม่ข่ายแบบกลุ่มเมฆสำหรับระบบจอดรถยนต์ด้วยวิธีการสื่อสารแบบเอ็มคิวทีทีโพรโทคอล และทำการเปรียบเทียบกับระบบเดิมที่ใช้เอชทีทีพีโพรโทคอลเป็นพื้นฐานที่ไม่สามารถตอบสนองความต้องการทุกอย่าง (ความสามารถในการขยายระบบ ประสิทธิภาพ คุณภาพการให้บริการ) ได้ในเวลาเดียวกัน รวมถึงมีการพบปัญหาการรับส่งข้อมูลที่ล่าช้าอีกด้วย



รูปที่ 2.22 สถาปัตยกรรมที่นำเสนอ

รูปที่ 2.22 [20] แสดงถึงสถาปัตยกรรมของระบบที่นำเสนอ เมื่อรถยนต์เข้าสู่พื้นที่ที่จอดรถ ผู้ใช้จะทำการจอดรถตามได้จองไว้ผ่านทางระบบ เซ็นเซอร์จะทำการอัพเดทข้อมูลไปยังเครื่องแม่ข่ายแบบกลุ่มเมฆผ่านทางเกตเวย์ว่าพื้นที่ดังกล่าวมีรถยนต์เข้ามาจอดแล้ว ซึ่งจะทำให้ผู้ใช้งานที่คอยดูสถานะที่จอดรถผ่านทางเครื่องแม่ข่ายแบบกลุ่มเมฆทราบได้ว่าที่จอดรถนั้นว่างหรือไม่



รูปที่ 2.23 การใช้พลังงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการเปรียบเทียบระบบที่นำเสนอที่ใช้เอ็มคิวทีทีกับระบบเดิมที่ใช้เอชทีทีพี พบว่าขนาดของข้อมูลที่ส่งด้วยเอชทีทีพีจะมีขนาดที่มากกว่าเอ็มคิวทีที เป็นผลให้ latency ที่เกิดขึ้นมีค่าที่มากกว่า ในขณะที่การใช้พลังงานของเอชทีทีพีมีปริมาณมากกว่าเอ็มคิวทีทีเนื่องจากต้องสร้างและคงสถานะการเชื่อมต่อไว้ในขณะที่เกิดการรับส่งข้อมูล ดังที่แสดงให้เห็นในรูปที่ 2.23 [20]

งานวิจัยข้างต้นแสดงให้เห็นถึงรูปแบบ การทดลองและการวัดผลสำหรับเอ็มคิวทีทีโพรโทคอล ซึ่งสามารถนำมาประยุกต์ใช้ได้ในเรื่องของการเตรียมการทดลองและวิธีการวัดผลของ delay เพื่อเปรียบเทียบกระบวนการสำหรับการทดลองได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

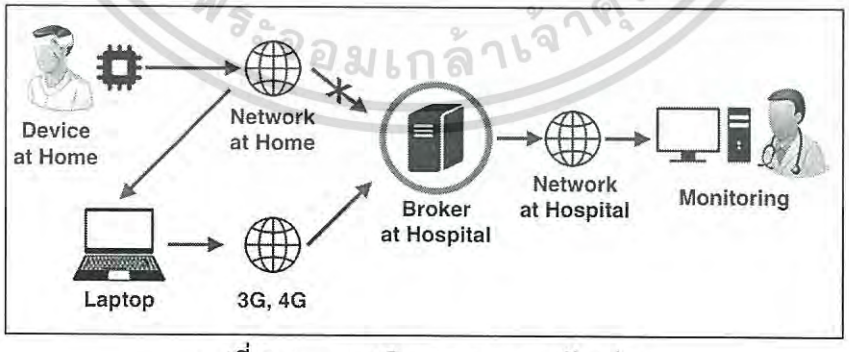
บทที่ 3
การดำเนินงานวิจัย

ในบทนี้จะเป็นการกล่าวถึงขั้นตอนการดำเนินงานวิจัย โดยจะถูกแบ่งออกเป็น 6 หัวข้อ ดังนี้ 1) สมมุติฐาน 2) แนวคิดการแก้ปัญหา 3) การออกแบบและจำลองระบบสำหรับการทดลอง 4) การทำงานของวิธีการที่นำเสนอ 5) ฟังก์ชันการส่งต่อและรับข้อมูลด้วยเออาร์พีและเอชทีทีพี และ 6) การทดลอง โดยมีรายละเอียดดังนี้

3.1 สมมุติฐาน

ระบบที่นำเอ็มคิวทีทีโพรโทคอลไปประยุกต์ใช้งานส่วนใหญ่แล้วมักจะเป็นระบบที่มีเครือข่ายและอุปกรณ์เป็นจำนวนมาก ดังนั้นสมมุติฐานสำหรับงานวิจัยนี้จึงมีรายละเอียดดังนี้

1. มีอุปกรณ์อื่นที่อยู่ในเครือข่ายเดียวกับอุปกรณ์ส่งข้อมูลอย่างน้อย 1 อุปกรณ์
2. มีเครือข่ายในระบบมากกว่าหนึ่งเครือข่าย และโบรกเกอร์มีการเชื่อมต่อไปยังอุปกรณ์อื่นมากกว่าหนึ่งเครือข่าย
3. ปัญหาการส่งข้อมูลเกิดขึ้นที่อินเทอร์เฟซของโบรกเกอร์ โดยมีอย่างน้อยหนึ่งอินเทอร์เฟซที่สามารถใช้งานได้ เพื่อให้มีอุปกรณ์ภายในระบบบางอุปกรณ์ยังสามารถส่งข้อมูลไปยังโบรกเกอร์ผ่านทางเครือข่ายที่ใช้งานได้
4. อุปกรณ์มีอินเทอร์เฟซสำหรับการเชื่อมต่อเครือข่ายมากกว่าหนึ่งอินเทอร์เฟซ โดยแต่ละอินเทอร์เฟซจะทำการเชื่อมต่ออยู่กับเครือข่ายคนละเครือข่ายบนอุปกรณ์ตัวเดียวกัน เพื่อให้เกิดการส่งต่อข้อมูลจากเครือข่ายหนึ่งไปยังอีกเครือข่ายหนึ่งผ่านทางอุปกรณ์ได้



รูปที่ 3.1 สมมุติฐานของระบบตัวอย่าง

ยกตัวอย่างเช่น ระบบตรวจสอบสถานะผิดปกติของผู้ป่วยภายในบ้าน เมื่ออุปกรณ์ตรวจสอบสถานะที่ผิดปกติแต่ไม่สามารถส่งข้อมูลผ่านทางเครือข่ายอินเทอร์เน็ตปกติภายในบ้านได้ (Network at Home) ผ่านทางอินเทอร์เฟซ Ethernet Module หรือ WiFi Module ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ อุปกรณ์นั้นจะทำการส่งต่อข้อมูลไปยังโทรศัพท์มือถือหรือคอมพิวเตอร์โน๊คบุ๊คที่เชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ตเดียวกัน ก่อนที่จะทดลองส่งข้อมูลผ่านทางเครือข่ายอื่นที่อาจจะมีปัญหาน้อยกว่าหรือไม่มีปัญหาเลยเช่น โทรศัพท์มือถือ (4G, 3G) เป็นต้น เพื่อส่งต่อข้อมูลไปยังเครื่องแม่ข่ายที่อาจจะอยู่ที่โรงพยาบาลเพื่อให้ผู้เชี่ยวชาญตรวจสอบอาการได้อย่างทันท่วงที ตามรูปที่ 3.1

3.2 แนวคิดการแก้ปัญหา

ในส่วนนี้จะเป็นการกล่าวถึงวิธีการแก้ปัญหาที่นำเสนอในวิทยานิพนธ์ฉบับนี้ โดยแบ่งเป็นแนวคิดการส่งต่อข้อมูลไปยังอุปกรณ์รอบข้าง และแนวคิดการส่งต่อข้อมูลแบบมัลติ-ฮ็อป

3.2.1 แนวคิดการส่งต่อข้อมูลไปยังอุปกรณ์รอบข้าง

หนึ่งในแนวคิดที่ช่วยแก้ปัญหาการเชื่อมต่อที่ขาดหายไปเป็นระยะเวลาที่ยาวนาน คือการส่งต่อข้อมูลไปยังอุปกรณ์รอบข้างในเครือข่ายเดียวกันหรือเครือข่ายใกล้เคียง เพื่อให้อุปกรณ์รอบข้างเหล่านั้นทำหน้าที่ส่งข้อมูลไปยังโบรกเกอร์ผ่านทางเครือข่ายอื่น ซึ่งหากอุปกรณ์รอบข้างมีอินเทอร์เฟซสำหรับการเชื่อมต่อเครือข่ายที่มากกว่าหนึ่งอินเทอร์เฟซแล้ว ก็มีโอกาสที่การส่งข้อมูลผ่านทางเครือข่ายอื่นที่ไม่มีปัญหาหรือมีปัญหาน้อยกว่าจะสามารถลดระยะเวลาที่จำเป็นจะต้องรอให้การเชื่อมต่อกลับมาเป็นปกติก่อนการส่งข้อมูลซ้ำตามกระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอล ดังนั้นหากพับบลิชเชอร์ไม่สามารถส่งข้อมูลด้วยเอ็มคิวทีทีตามปกติ แต่สามารถเชื่อมต่อไปยังอุปกรณ์รอบข้างอื่นได้ ก็มีโอกาสที่การส่งข้อมูลจะสำเร็จก่อนที่การเชื่อมต่อจะกลับมาใช้ได้ตามปกติ

การใช้วิธีการนี้จำเป็นจะต้องพัฒนาฟังก์ชันเพิ่มเติมจากเอ็มคิวทีทีโพรโทคอล เนื่องจากปัญหาหลัก 2 ประการคือ

1. ต้องค้นหาอุปกรณ์รอบข้างก่อนการส่งข้อมูล เนื่องจากเอ็มคิวทีทีโพรโทคอลไม่มีฟังก์ชันการค้นหาอุปกรณ์รอบข้าง แต่จะใช้การระบุ IP address โดยตรงสำหรับการส่งข้อมูลไปยังโบรกเกอร์ ดังนั้นในวิธีการนี้จะใช้เออาร์พี (ARP : address Resolution Protocol) ซึ่งเป็นโพรโทคอลที่ใช้สำหรับการจับคู่ IP address กับ physical address ช่วยในการค้นหาอุปกรณ์รอบข้าง (อุปกรณ์รอบข้างในที่นี้คือ Virtual Machine) โดยใช้คำสั่ง Arp-scan [23] ในการส่งเออาร์พีแพ็คเกจไปยัง IP address ทั้งหมดในแต่ละอินเทอร์เฟส เพื่อเป็นการตรวจสอบว่ามีอุปกรณ์รอบข้างใดบ้างที่สามารถติดต่อได้ นอกจากนั้นการใช้คำสั่ง Arp-scan ยังช่วยให้การจำลองระบบสำหรับการทดลองทำได้ง่ายขึ้นอีกด้วย เนื่องจากเป็นเครื่องมือที่สามารถติดตั้งและใช้งานได้ง่ายในระบบปฏิบัติการของอุปกรณ์ที่ใช้สำหรับการทดลอง

2. ต้องใช้โพรโทคอลอื่นในการส่งต่อข้อมูลจากพับบลิชเชอร์ตั้งต้นไปยังอุปกรณ์รอบข้าง เนื่องจากเอ็มคิวทีทีโพรโทคอลจำเป็นที่จะต้องใช้โบรกเกอร์เป็นตัวกลางในการส่งข้อมูลจากพับบลิชเชอร์ไปยังซับสไครบเบอร์ แต่การส่งต่อข้อมูลจากพับบลิชเชอร์ไปยังอุปกรณ์รอบข้างนั้นไม่มีโบรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกอร์เป็นตัวกลาง ดังนั้นในวิธีการนี้จึงจะใช้เอชทีทีพีช่วยในการส่งข้อมูลจากพับบลิชเชอร์ตั้งต้นไปยังอุปกรณ์รอบข้างแทน

3.2.2 แนวคิดการส่งต่อข้อมูลแบบมัลติ-ฮ็อป

โดยปกติแล้วอินเทอร์เน็ตของสรรพสิ่งมักจะมีอุปกรณ์และเครือข่ายที่เชื่อมต่อกันอยู่เป็นจำนวนมาก การส่งต่อข้อมูลไปยังอุปกรณ์ภายในเครือข่ายเดียวกันจึงอาจจะทำให้โอกาสส่งข้อมูลสำเร็จไม่มากเท่าที่ควร เนื่องจากมีความเป็นไปได้ว่าปัญหาการเชื่อมต่ออาจจะมีผลกระทบทั้งเครือข่าย ดังนั้นการส่งต่อข้อมูลไปยังอุปกรณ์ที่อยู่ภายนอกเครือข่ายเพื่อให้อุปกรณ์เหล่านั้นทำหน้าที่ส่งต่อข้อมูลผ่านทางเครือข่ายอื่นอาจจะช่วยเพิ่มโอกาสในการส่งข้อมูลให้มากยิ่งขึ้น

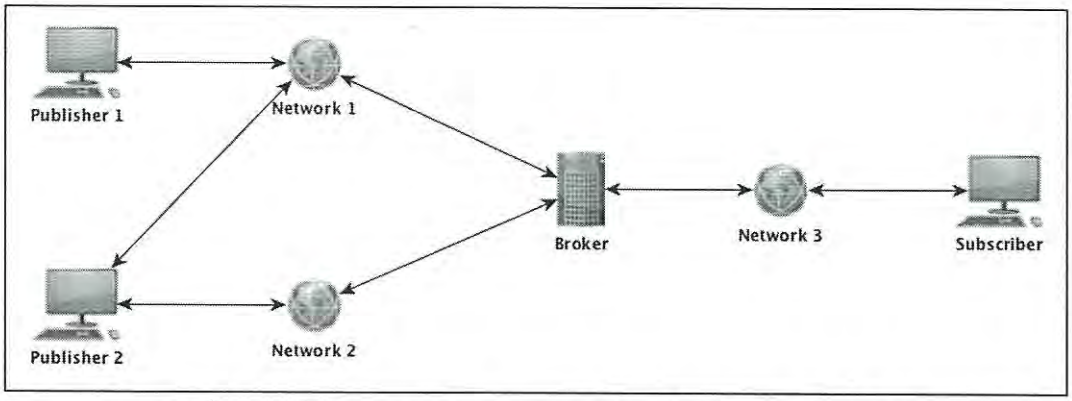
เพื่อเพิ่มโอกาสในการส่งข้อมูล งานวิจัยนี้จึงสร้างการส่งต่อข้อมูลแบบมัลติ-ฮ็อปขึ้น โดยการใช้ IP address ที่ได้จากการค้นหาอุปกรณ์รอบข้าง (Virtual Machine อื่น) ด้วยเออาร์พีในขั้นตอนก่อนหน้า แล้วทำการส่งข้อมูลไปยังอุปกรณ์รอบข้างเรื่อย ๆ จนกว่าจะมีอุปกรณ์ตัวใดตัวหนึ่งสามารถส่งข้อมูลแทนได้ โดยหนึ่งในปัญหาสำหรับการส่งข้อมูลแบบนี้คือ จะทำอย่างไรให้การส่งต่อข้อมูลระหว่างเครือข่ายไม่เกิดการส่งข้อมูลซ้ำไปยังอุปกรณ์กลุ่มเดิม เนื่องจากการส่งข้อมูลซ้ำไปยังอุปกรณ์เดิมที่ไม่สามารถส่งข้อมูลแทนได้จะส่งผลให้ระยะเวลาในการส่งข้อมูลมากขึ้น

การตรวจสอบเครือข่ายก่อนการส่งข้อมูลจึงเป็นวิธีการหนึ่งที่ช่วยแก้ปัญหาการส่งข้อมูลซ้ำ โดยตรวจสอบจาก network address ของอุปกรณ์ที่จะส่งข้อมูลไป หาก network address นี้อยู่ในรายการเครือข่ายที่เคยส่งแล้ว อุปกรณ์ส่งข้อมูลจะข้ามเครือข่ายนั้นไป และทำการส่งข้อมูลไปยังเครือข่ายอื่นแทน โดยรายการเครือข่ายที่เคยทดลองส่งนี้จะเกิดจากการที่อุปกรณ์ที่รับข้อมูลเข้ามาในเครือข่ายอันดับแรกทำการส่งข้อมูลไปยังอุปกรณ์ทั้งหมดภายในเครือข่าย แต่ไม่มีอุปกรณ์ใดเลยที่สามารถส่งข้อมูลแทนได้ เป็นผลให้อุปกรณ์ที่รับข้อมูลเข้ามาในเครือข่ายเป็นอันดับแรกส่ง network address กลับไปเพื่อเป็นการบอกว่าเครือข่ายดังกล่าวไม่มีอุปกรณ์ใดเลยที่สามารถส่งข้อมูลแทนได้

3.3 การออกแบบและจำลองระบบสำหรับการทดลอง

ในส่วนของการออกแบบและจำลองระบบสำหรับการทดลอง จะมีการออกแบบให้ระบบมีหลายเครือข่าย โดยในแต่ละเครือข่ายจะมีพับบลิชเชอร์อย่างน้อย 1 ตัวสำหรับส่งข้อมูลออกไปยังเครือข่ายอื่น เพื่อทำให้เกิดการส่งข้อมูลไปยังโบรกเกอร์จากเครือข่ายอื่นแทนเครือข่ายที่มีปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



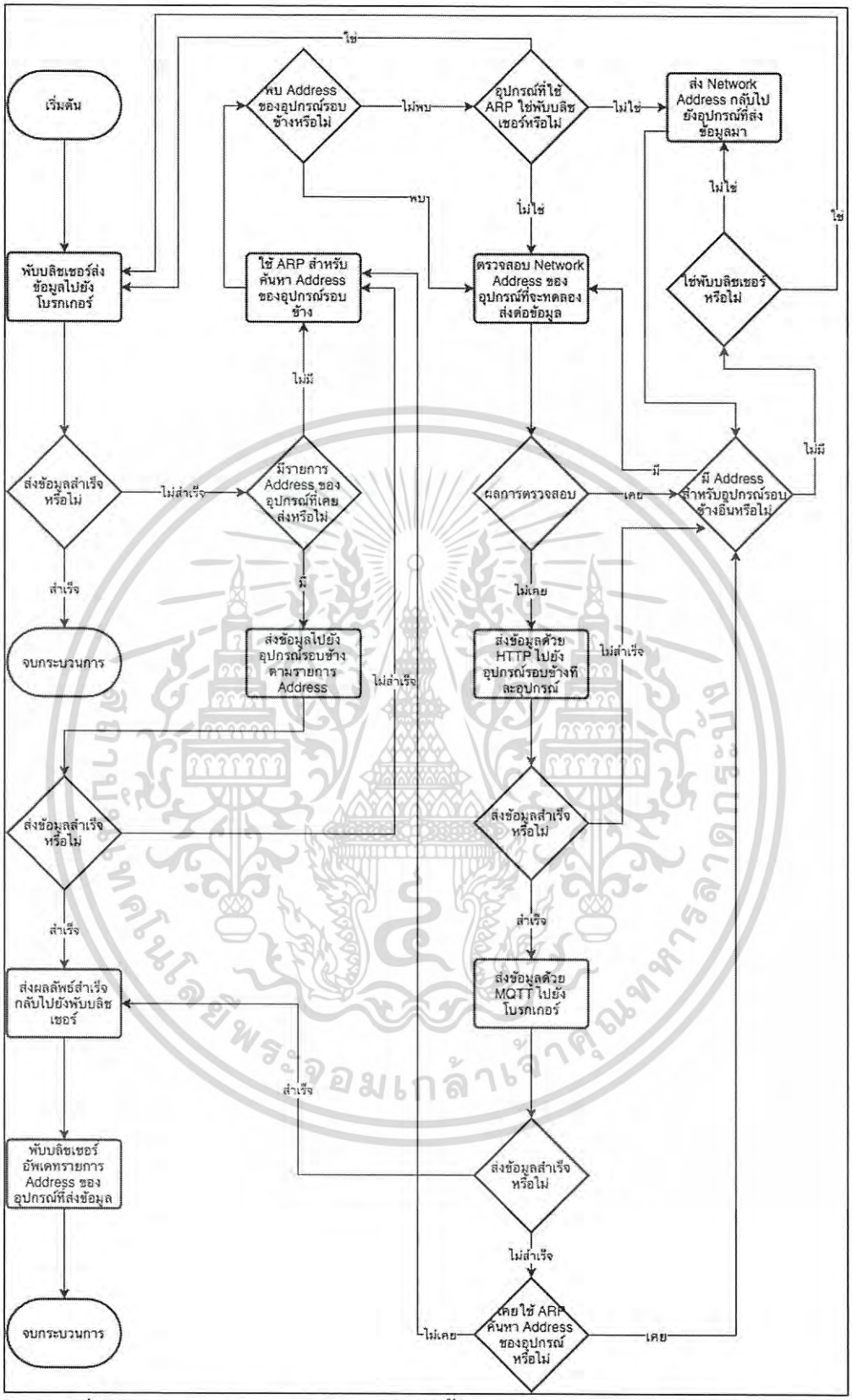
รูปที่ 3.2 ตัวอย่างระบบสำหรับการส่งข้อมูลไปยังอุปกรณ์รอบข้าง

จากรูปที่ 3.2 แสดงถึงตัวอย่างระบบสำหรับการส่งข้อมูลไปยังอุปกรณ์รอบข้าง ซึ่งหากการเชื่อมต่อระหว่างพับบลิชเชอร์ 1 กับโบรกเกอร์เกิดปัญหาจนไม่สามารถส่งข้อมูลได้ พับบลิชเชอร์ 1 จะเปลี่ยนเส้นทางการส่งข้อมูลไปยังพับบลิชเชอร์ 2 ผ่านทางเครือข่ายที่ 1 แทน ซึ่งอาจจะไม่มีปัญหาหรือมีปัญหาน้อยกว่าเครือข่ายอื่น

3.4 การทำงานของวิธีการที่นำเสนอ

เมื่อการเชื่อมต่อจากพับบลิชเชอร์ไปยังโบรกเกอร์เกิดปัญหาขึ้น โดยปกติแล้วคิวโอเอสของเอ็มคิวทีทีโพรโทคอลจะทำการส่งข้อมูลซ้ำกลับไปยังโบรกเกอร์ตามระดับคิวโอเอสที่เลือก แต่วิธีการที่นำเสนอนี้จะตรวจสอบข้อความการส่งข้อมูลผิดพลาดที่เกิดขึ้นและเปลี่ยนวิธีการส่งซ้ำของข้อมูลให้ทำการทดลองส่งไปยังอุปกรณ์รอบข้างแทน และหากการส่งต่อข้อมูลจากอุปกรณ์รอบข้างไปยังโบรกเกอร์ไม่สามารถส่งได้สำเร็จ อุปกรณ์รอบข้างนั้นก็จะส่งข้อมูลไปยังอุปกรณ์รอบข้างตัวอื่นอีกโดยจะทำการตรวจสอบก่อนว่า เครือข่ายที่ส่งไปนั้น เคยส่งไปแล้วหรือไม่ เพื่อป้องกันการส่งข้อมูลซ้ำไปยังเครือข่ายเดิมซึ่งจะส่งผลต่อระยะเวลาในการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 Flow Chart แสดงการลำดับขั้นตอนการทำงานของวิธีการที่นำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการดังกล่าวสามารถอธิบายได้ด้วยแผนภาพ Flow Chart ตามรูปที่ 3.3 โดยลักษณะการทำงานจะเรียงลำดับตามขั้นตอนดังต่อไปนี้

1. พับบลิชเชอร์ตั้งต้นจะทำการส่งข้อมูลไปยังโบรกเกอร์ด้วยเอ็มคิวทีทีโพรโทคอลตามปกติ หากสามารถส่งได้สำเร็จ กระบวนการทั้งหมดจะจบลง แต่ถ้าไม่สำเร็จ ระบบจะเริ่มต้นกระบวนการทำตามวิธีที่นำเสนอในขั้นตอนต่อไป

2. พับบลิชเชอร์ตั้งต้นตรวจสอบว่า เคยมีรายการ IP address ที่เคยส่งข้อมูลผ่านอุปกรณ์รอบข้างสำเร็จเก็บไว้อยู่หรือไม่ ถ้ามีพับบลิชเชอร์จะทำการทดลองส่งข้อมูลไปตามรายการ IP addressที่เก็บไว้ แต่ถ้าไม่มี ระบบจะทำตามขั้นตอนที่ 4 ต่อไป

3. หากการทดลองส่งข้อมูลตามรายการ IP address ที่เก็บไว้สามารถส่งได้ อุปกรณ์ปลายทางจะส่งผลลัพธ์ว่าสำเร็จกลับไปยังพับบลิชเชอร์ส่งตั้งต้นและจบกระบวนการลง แต่ถ้าไม่สำเร็จระบบจะดำเนินการตามขั้นตอนต่อไป

4. พับบลิชเชอร์ตั้งต้นทำการค้นหา IP address ของอุปกรณ์รอบข้างด้วยเออาร์พีโพรโทคอลและเก็บไว้ในรายการ IP address

5. ทำการตรวจสอบว่าเครือข่ายที่กำลังจะส่งข้อมูลไปยังอุปกรณ์รอบข้างนั้น เคยมีการทดลองส่งไปแล้วหรือไม่ ถ้าเคยมีการทดลองส่งแต่ไม่สามารถส่งข้อมูลได้ ระบบจะข้ามการส่งข้อมูลไปยังเครือข่ายที่เคยทดลองส่งนั้นไป แต่ถ้าไม่เคย ระบบจะดำเนินการตามขั้นตอนต่อไป

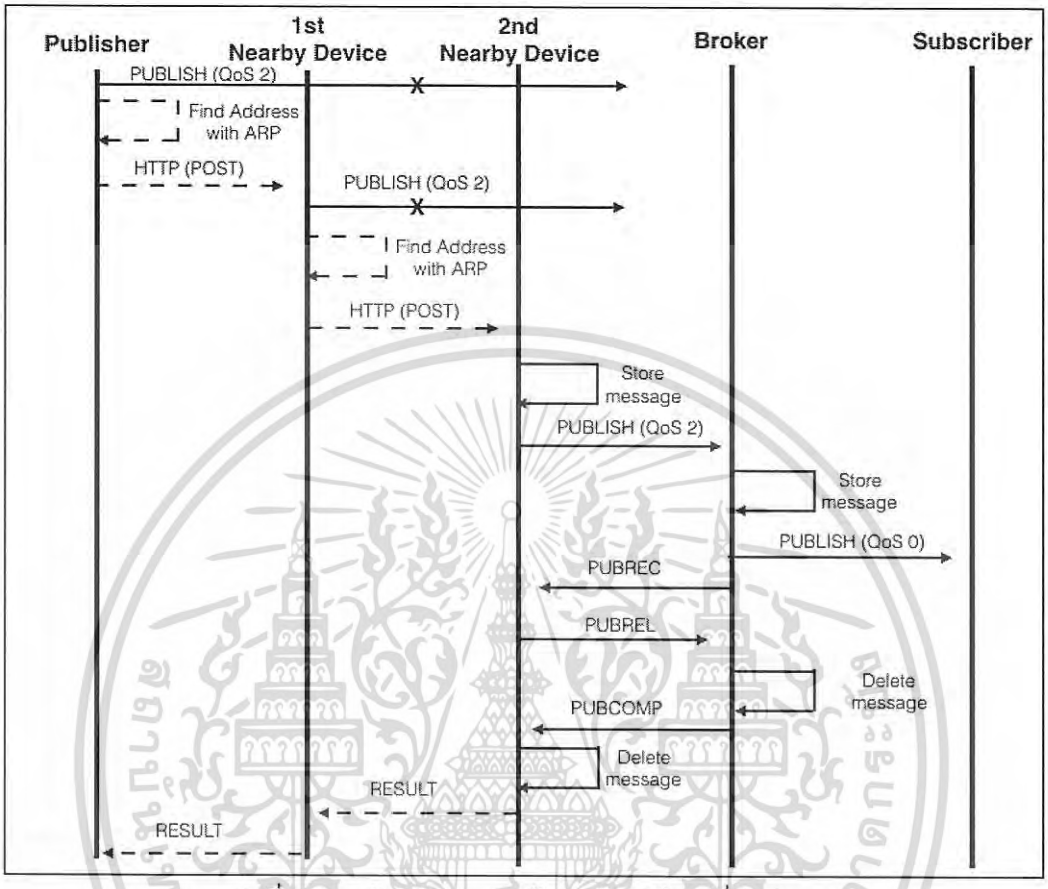
6. ทำการส่งข้อมูลไปยังอุปกรณ์รอบข้างตามรายการ IP address ทีละเครื่องด้วยเอชทีทีพีโพรโทคอล หากส่งข้อมูลสำเร็จ อุปกรณ์รอบข้างนั้นจะทำการส่งผลลัพธ์ว่าสำเร็จตอบกลับไปยังพับบลิชเชอร์ตั้งต้น และจบกระบวนการ แต่ถ้าไม่สำเร็จอุปกรณ์ส่งรอบข้างนั้นจะทำกระบวนการที่ 4 และ 5 เพื่อส่งข้อมูลไปยังอุปกรณ์รอบข้างอื่น

7. อุปกรณ์ต่าง ๆ จะทำกระบวนการที่ 4 และ 5 ซ้ำไปเรื่อย ๆ จนกว่าจะมีอุปกรณ์ที่ส่งข้อมูลแทนสำเร็จ หรือภายในเครือข่ายนั้นไม่มีอุปกรณ์ใดเลยที่สามารถส่งข้อมูลแทนได้ หากไม่มีอุปกรณ์ที่สามารถส่งได้ อุปกรณ์ตัวแรกภายในเครือข่ายที่รับข้อมูลเข้ามา จะส่งข้อมูลหมายเลขประจำเครือข่ายกลับไปเพื่อเป็นการบอกอุปกรณ์ต้นทางว่า ไม่มีอุปกรณ์ใดภายในเครือข่ายนี้ที่สามารถส่งข้อมูลแทนได้ เพื่อป้องกันการส่งข้อมูลซ้ำอันเกิดจากการส่งต่อข้อมูลของอุปกรณ์รอบข้างในครั้งต่อไป

8. หากการส่งข้อมูลผ่านทางอุปกรณ์รอบข้างสามารถส่งได้สำเร็จ และเมื่อผลลัพธ์นี้ถูกส่งกลับมายังพับบลิชเชอร์ตั้งต้นแล้ว พับบลิชเชอร์ตั้งต้นจะทำการเก็บเส้นทางตาม IP address ของอุปกรณ์ที่สามารถส่งแทนได้ไว้ในรายการ เมื่อพับบลิชเชอร์ส่งตั้งต้นไม่สามารถส่งได้อีกในครั้งต่อไป พับบลิชเชอร์ก็จะทำการส่งข้อมูลไปตามเส้นทางอุปกรณ์ที่เก็บไว้ก่อน เพื่อลดระยะเวลาในการค้นหาอุปกรณ์และทดลองส่งข้อมูลไปยังอุปกรณ์รอบข้างที่ไม่สามารถส่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. หากไม่มีอุปกรณ์ใดเลยที่สามารถส่งแทนได้ กระบวนการทั้งหมดจะกลับไปเริ่มที่ขั้นตอนที่ 1 และเริ่มต้นการทำกระบวนการส่งข้อมูลผ่านอุปกรณ์รอบข้างนี้ใหม่อีกครั้ง



รูปที่ 3.4 ตัวอย่างการส่งข้อมูลด้วยวิธีการที่นำเสนอ

รูปที่ 3.4 แสดงถึงตัวอย่างการส่งข้อมูลผ่านทางอุปกรณ์รอบข้างจำนวน 2 อุปกรณ์ ด้วยวิธีการที่นำเสนอ เส้นประแสดงถึงกระบวนการการส่งข้อมูลด้วยเอ็มคิวทีทีโพรโทคอล เส้นทึบแสดงถึงกระบวนการที่เพิ่มเติมจากเอ็มคิวทีทีโพรโทคอลเดิม ซึ่งประกอบไปด้วยการค้นหาอุปกรณ์รอบข้างด้วยเออาร์พี และส่งข้อมูลไปยังอุปกรณ์รอบข้างด้วยเอชทีทีพี หากพับบลิชเชอร์ไม่สามารถส่งข้อมูลด้วยเอ็มคิวทีทีไปยังโบรกเกอร์ได้ กระบวนการค้นหา Address ของอุปกรณ์รอบข้างจะเกิดขึ้น ก่อนที่จะดำเนินการส่งข้อมูลด้วยเอชทีทีพีไปยังอุปกรณ์รอบข้างตัวที่หนึ่ง เพื่อที่จะให้อุปกรณ์รอบข้างตัวที่หนึ่งทำการทดลองส่งข้อมูลด้วยเอ็มคิวทีทีโพรโทคอลไปยังพับโบรกเกอร์แทน แต่ถ้าไม่สามารถส่งข้อมูลแทนได้ กระบวนการค้นหา Address ของอุปกรณ์รอบข้างจะเกิดขึ้นอีกครั้ง ก่อนที่จะเกิดการส่งต่อข้อมูลไปยังอุปกรณ์รอบข้างตัวที่สองเพื่อให้เกิดการส่งต่อข้อมูลไปยังโบรกเกอร์ด้วยเอ็มคิวทีทีโพรโทคอล หากอุปกรณ์ตัวที่สองสามารถส่งข้อมูลแทนได้ตามกระบวนการของเอ็มคิวทีที ผลลัพธ์จะถูกส่งกลับมายังอุปกรณ์ตัวที่หนึ่งผ่านช่องทางเอชทีทีพีโพรโทคอล และผลลัพธ์จากอุปกรณ์ตัวที่หนึ่งก็จะถูกส่งกลับไปยังพับบลิชเชอร์ตัวแรก เป็นอันจบกระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ฟังก์ชันการส่งและรับข้อมูลด้วยเออาร์พีและเอชทีทีพี

เนื่องจากเอ็มคิวทีทีโพรโทคอลมีข้อจำกัดในหลายด้านสำหรับการพัฒนาเพื่อให้เป็นระบบตามที่ได้ออกแบบไว้ จึงจำเป็นที่จะต้องพัฒนาฟังก์ชันเพิ่มเติมสำหรับการรับส่งข้อมูล เพื่อให้สามารถทำงานร่วมกับโพรโทคอลเอ็มคิวทีทีด้วยภาษา Python โดยหาก Paho ซึ่งเป็น Library สำหรับการรับส่งข้อมูลสำหรับเอ็มคิวทีทีไม่สามารถส่งข้อมูลได้ จะมีการส่งข้อความผิดพลาดเกิดขึ้น ซึ่งสามารถเขียนฟังก์ชันเพิ่มเติมหลังจากการตรวจสอบข้อผิดพลาดแล้วได้นั่นเอง

ฟังก์ชันที่พัฒนาเพิ่มเติมสามารถแบ่งตามลักษณะการทำงานได้ดังนี้

3.5.1 ฟังก์ชันสำหรับการค้นหาอุปกรณ์รอบข้าง

ในการส่งข้อมูลไปยังอุปกรณ์รอบข้างนั้นจำเป็นที่จะต้องรู้หมายเลขประจำเครื่องของอุปกรณ์ปลายทางที่จะส่งข้อมูลไป แต่เนื่องจากเอ็มคิวทีทีโพรโทคอลไม่สามารถค้นหาอุปกรณ์รอบข้างได้ด้วยตัวเอง จึงจำเป็นต้องเขียนฟังก์ชันรองรับการทำงานในส่วนนี้

หนึ่งในวิธีการที่สามารถช่วยในการค้นหาอุปกรณ์รอบข้างคือการใช้เออาร์พี (ARP) ซึ่งเป็นโพรโทคอลที่ใช้สำหรับการจับคู่ IP address กับ physical address ในการค้นหาอุปกรณ์รอบข้าง โดยเมื่อค้นหา IP address ทั้งหมดแล้ว พับบลิชเชอร์ตั้งต้นจะทำการเก็บผลลัพธ์ที่ได้ไว้ในรายการก่อนที่จะนำรายการนี้มาใช้ในการส่งข้อมูลไปยังอุปกรณ์รอบข้างทีละอุปกรณ์

3.5.2 ฟังก์ชันสำหรับการส่งข้อมูลไปยังอุปกรณ์รอบข้าง

การส่งข้อมูลไปยังอุปกรณ์รอบข้างก็เป็นหนึ่งในกระบวนการที่ไม่สามารถใช้เอ็มคิวทีทีโพรโทคอลช่วยได้ เนื่องจากการส่งข้อมูลจากพับบลิชเชอร์ตั้งต้นไปยังอุปกรณ์ที่อยู่รอบข้างไม่มีโบรกเกอร์คั่นกลางสำหรับการรับและกระจายข้อมูล

เอชทีทีพีเป็นโพรโทคอลที่ถูกใช้สำหรับการส่งข้อมูลไปยังอุปกรณ์รอบข้างแทนเอ็มคิวทีทีโพรโทคอลที่ไม่สามารถใช้ได้ ก่อนการส่งข้อมูล อุปกรณ์ที่ทำการส่งจะตรวจสอบเครือข่ายของอุปกรณ์ปลายทางก่อนว่าเคยมีอุปกรณ์ที่ทดลองส่งข้อมูลไปแล้วหรือไม่ เพื่อป้องกันการส่งข้อมูลซ้ำในเครือข่ายที่ไม่สามารถส่งได้ และถ้าอุปกรณ์รอบข้างใดที่ไม่สามารถส่งข้อมูลไปยังโบรกเกอร์แทนได้ พับบลิชเชอร์ตั้งต้นก็จะทำการส่งข้อมูลไปยังอุปกรณ์รอบข้างตัวอื่นแทน และทำกระบวนการนี้ซ้ำจนกว่าจะส่งข้อมูลไปยังโบรกเกอร์ได้สำเร็จ หรือจนกว่าจะไม่มีอุปกรณ์ใดเลยที่สามารถส่งข้อมูลแทนได้ โดยการส่งข้อมูลด้วยเอชทีทีพีนี้จะสร้างแพ็คเกจจาก Library ของ Python เพื่อส่งข้อมูลไปยังเว็บเซิฟเวอร์ของอุปกรณ์รอบข้าง

3.5.3 ฟังก์ชันสำหรับการจดจำการส่งข้อมูลผ่านอุปกรณ์รอบข้าง

การจดจำเส้นทางสำหรับการส่งข้อมูล เป็นฟังก์ชันที่ช่วยสำหรับการลดระยะเวลาการส่งข้อมูลผ่านอุปกรณ์รอบข้าง โดยหากอุปกรณ์รอบข้างใดก็ตามสามารถที่จะส่งข้อมูลไปยังโบรกเกอร์แทนอุปกรณ์ส่งตั้งต้นได้สำเร็จ อุปกรณ์รอบข้างนั้นจะทำการส่งผลลัพธ์กลับไปเรื่อย ๆ จนถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ส่งตั้งต้น เมื่ออุปกรณ์ส่งตั้งต้นได้รับผลลัพธ์แล้วก็จะเก็บหมายเลขประจำเครื่องของแต่ละอุปกรณ์ที่สามารถส่งได้ไว้เป็นเส้นทางสำหรับการส่งข้อมูลผ่านทางอุปกรณ์รอบข้างในครั้งต่อไป เพื่อลดกระบวนการในการค้นหาอุปกรณ์รอบข้างและทดลองส่งข้อมูลไปยังอุปกรณ์รอบข้างตัวอื่นที่ไม่สามารถส่งข้อมูลแทนได้

3.6 การทดลอง

สำหรับการทดลอง ได้มีการสร้างระบบต้นแบบสำหรับการวัดผลด้วย Virtual Machine โดยมีรายละเอียดดังตารางที่ 3.1 3.2 และ 3.3

ตารางที่ 3.1 รายละเอียดของ Virtual Machine สำหรับการทดลอง

หัวข้อ	รายละเอียด
Virtual Machine	Oracle VirtualBox version 5.0
Operating System	Debian version 8.5
Processor	1 core
Memory	2024 MB
Storage	8 GHz
Virtual Networking	Intel PRO/1000 MT Desktop (82545EM)
Networking Mode	NAT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 รายละเอียดของซอฟแวร์ที่ใช้และค่าที่กำหนดสำหรับการทดลอง

หัวข้อ	รายละเอียดและค่าที่กำหนด
ซอฟแวร์สำหรับพับบลิชเชอร์และซับสไครบเบอร์ (สำหรับพับบลิชเชอร์และซับสไครบเบอร์)	Paho Version 1.1
ซอฟแวร์สำหรับโบรกเกอร์ (สำหรับโบรกเกอร์)	Mosquitto Version 1.3.4-2
ซอฟแวร์สำหรับเอชทีทีพีเซิฟเวอร์ (สำหรับพับบลิชเชอร์)	Apache2 Version 2.4.10-10+deb8u4
ภาษาที่ใช้สำหรับการพัฒนาเครื่องลูกข่ายที่ใช้เอ็มคิวทีที (สำหรับพับบลิชเชอร์และซับสไครเบอร์)	Python 2.7
เครื่องมือสำหรับวัดประสิทธิภาพทางเครือข่าย (สำหรับทุก Virtual Machine)	iperf Version 2.0.5
เครื่องมือสำหรับการค้นหา IP address ของอุปกรณ์รอบข้าง (สำหรับพับบลิชเชอร์และซับสไครเบอร์)	arp-scan Version 1.8.1
จำนวนพับบลิชเชอร์	3 ตัว
จำนวนโบรกเกอร์	1 ตัว
จำนวนซับสไครบเบอร์	1 ตัว
ขนาดของข้อความ	1,000 ไบท์
Timeout ของการส่งข้อมูลด้วยเอ็มคิวทีที (keep-alive ของ MQTT)	5 วินาที
Timeout ของการส่งข้อมูลด้วยเอชทีทีพี (keep-alive ของ HTTP)	5 วินาที
กระบวนการที่ใช้ในการทดลอง	1. คิวโอเอส 1 ที่พับบลิชเชอร์ 2. คิวโอเอส 2 ที่พับบลิชเชอร์ 3. วิธีการที่นำเสนอ (คิวโอเอส 2 ที่พับบลิชเชอร์)
ระดับคิวโอเอสของซับสไครบเบอร์	คิวโอเอส 0
ระยะเวลาที่ตัดการเชื่อมต่อ	10 20 และ 30 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 (ต่อ)

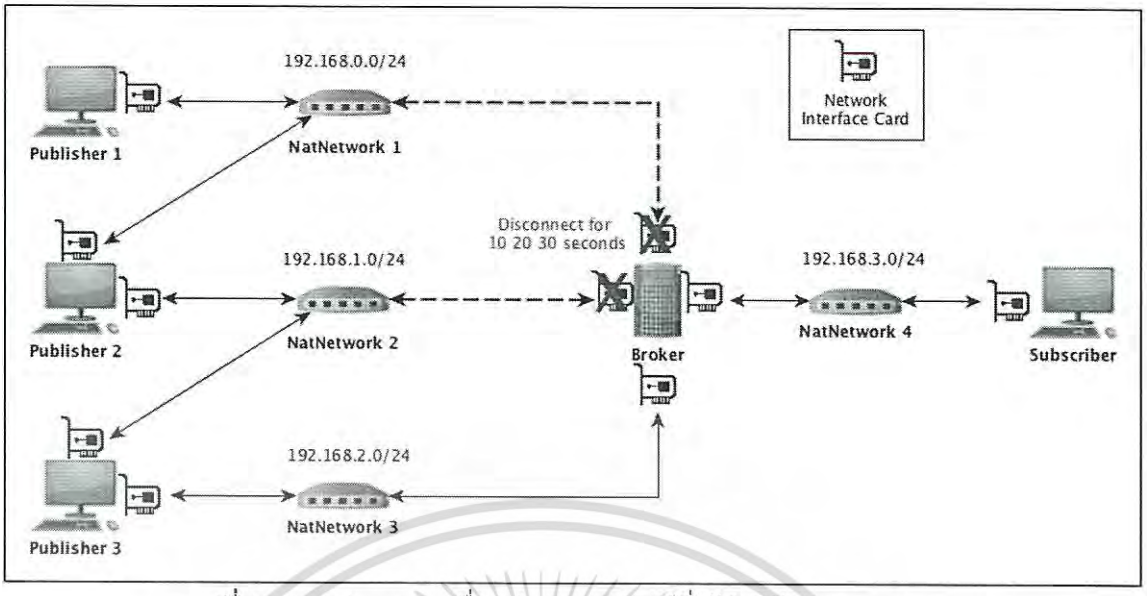
หัวข้อ	รายละเอียดและค่าที่กำหนด
จำนวนครั้งที่ทำการทดลอง	100 ครั้งต่อกระบวนการที่ใช้ในการทดลองต่อระยะเวลาที่ตัดการเชื่อมต่อ
การวัดผล	ผลความล่าช้าของการส่งข้อมูล (Delay Time) และผลความเร็วในการตอบสนองการส่งข้อมูล (Response Time)

ตารางที่ 3.3 อัตราการส่งข้อมูลและ Ping Delay ของแต่ Virtual Machine

การเชื่อมต่อ	อัตราการส่งข้อมูล (Mbps)	Ping Delay (ms)
พับบลิชเชอร์ 1 – โบรเกอร์	955	1.235
พับบลิชเชอร์ 2 – โบรเกอร์	876	1.171
พับบลิชเชอร์ 3 – โบรเกอร์	954	0.622
ซับสไครบเบอร์ – โบรเกอร์	962	0.862
พับบลิชเชอร์ 1 - พับบลิชเชอร์ 2	846	0.821
พับบลิชเชอร์ 2 - พับบลิชเชอร์ 3	949	0.757

ตารางที่ 3.1 แสดงถึงรายละเอียดของ Virtual Machine สำหรับการทดลอง โดยจะมีการจำลองทั้งหมด 5 ตัวด้วยกัน ซึ่งแบ่งเป็น พับบลิชเชอร์ 3 ตัว โบรกเกอร์ 1 ตัว และซับสไครบเบอร์ 1 ตัว ซึ่งจะมีการจำลอง Virtual Networking ในรูปแบบของ Intel PRO/1000 MT Desktop โดยมีอัตราการสูงข้อมูลสูงสุดอยู่ที่ 1Gbps แต่จากการใช้ iPerf ที่เป็นเครื่องมือสำหรับการวัดผลประสิทธิภาพทางเครือข่ายแล้วพบว่าอัตราการส่งข้อมูลและค่า Ping Delay จะได้ผลตามตารางที่ 3.3 ซึ่งอาจจะเป็นผลมาจาก Scheduling ของ Virtual Machine ที่ทำให้ค่าดังกล่าวไม่คงที่ [22]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 รูปแบบการเชื่อมต่อของอุปกรณ์ที่ใช้สำหรับการทดลอง

รูปแบบการเชื่อมต่อจะมีลักษณะเป็นดังรูปที่ 3.5 โดยการตัดการเชื่อมต่อจะเกิดขึ้นที่อินเทอร์เฟสของโบรกเกอร์ที่เชื่อมต่อกับ NatNetwork 1 และอินเทอร์เฟสของโบรกเกอร์ที่เชื่อมต่อกับNatNetwork 2 เพื่อเป็นการจำลองการเกิดปัญหาของภายในเครือข่ายตามระยะเวลาที่กำหนด และก่อนการทดลอง Virtual Machine ทั้งหมดจะทำการเชื่อมต่อกับเครื่องแม่ข่าย NTP ที่อยู่ในเครื่อง Host เพื่อเป็นการตั้งค่าเวลาของ Virtual Machine ทั้งหมดให้มีค่าตรงกัน โดยที่ Virtual Machine แต่ละคู่จะมีค่าเฉลี่ยของ Ping Delay น้อยกว่า 1 มิลลิวินาที (วัดค่า Ping Delay ระหว่างพับบลิชเชอร์แต่ละคู่ พับบลิชเชอร์ไปยังโบรกเกอร์ และโบรกเกอร์ไปยังซับสไครบเบอร์) ซึ่งการเชื่อมต่อในระบบทั้งหมดจะเกิดขึ้นในโหมด NAT เพื่อให้การเชื่อมต่อภายใน Virtual Machine ทั้งหมดเกิดขึ้นภายในเครื่อง Host โดยไม่ส่งผลถึง Physical Network Adapter ของเครื่อง Host จริง ซึ่งอาจจะส่งผลต่อผลลัพธ์ต่อการทดลอง เนื่องจากข้อมูลจะถูกส่งออกไปยังเราเตอร์ที่อยู่นอกเครื่อง Host นั่นเอง

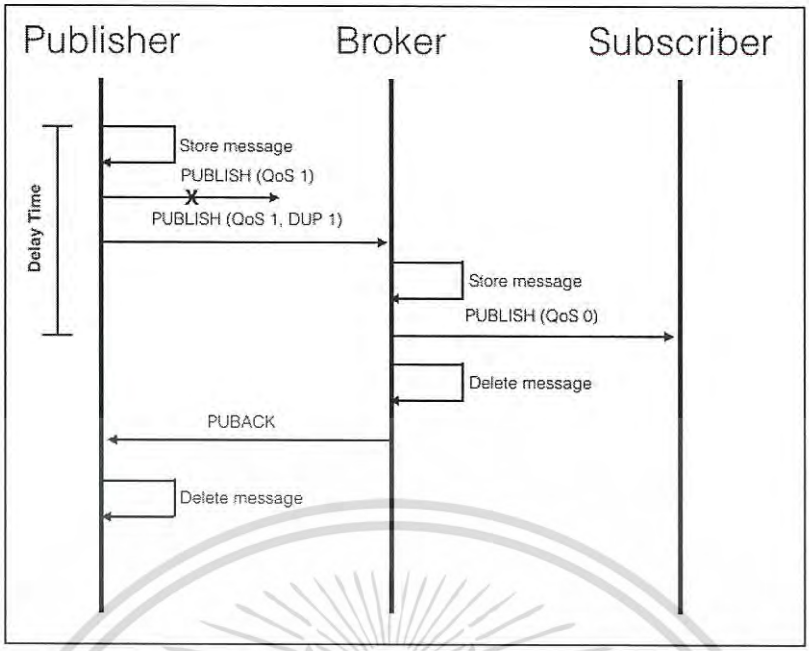
ระยะเวลา keep-alive สำหรับการทดลองนี้ถูกกำหนดไว้ที่ 5 วินาที เพื่อให้การทดลองนั้นเกิดความรวดเร็วขึ้นเนื่องจากต้องทำการทดลองซ้ำจำนวน 100 ครั้ง หากใช้ค่าตั้งต้นที่ library กำหนดไว้ (60 วินาที) อาจจะทำให้ใช้ระเวลาในการทดลองมากจนเกินไป

3.7 การวัดผล

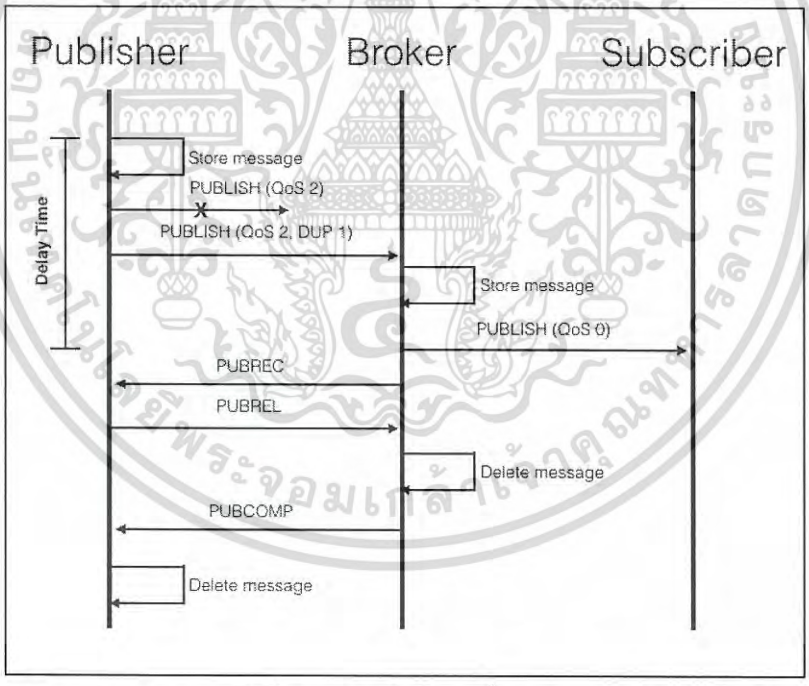
3.7.1 ผลความล่าช้าของการส่งข้อมูล

ผลความล่าช้าของการส่งข้อมูล (Delay Time) จะวัดจากเวลา ณ ขณะที่ซับสไครบเบอร์ได้รับข้อมูล หักลบกับเวลา ณ ขณะที่พับบลิชเชอร์เริ่มต้นส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

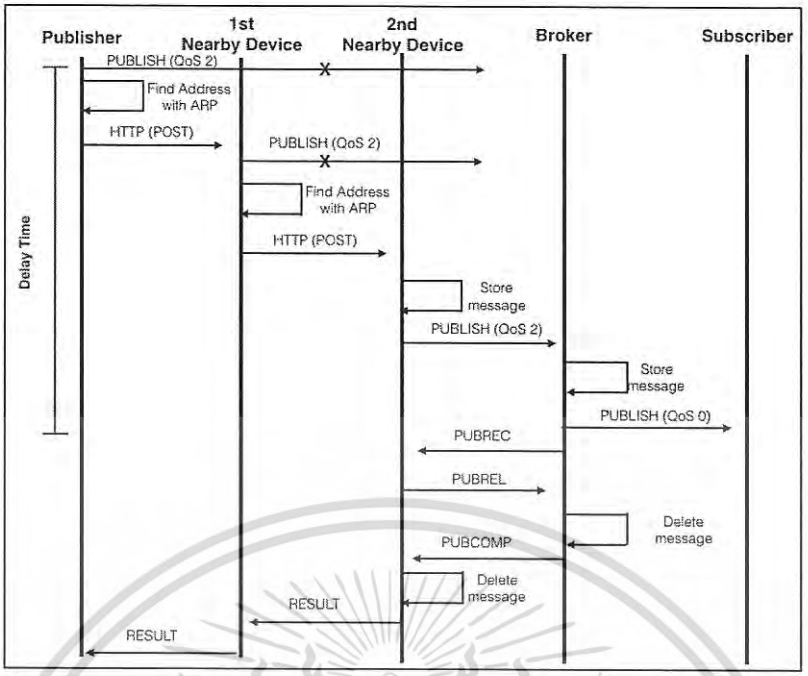


รูปที่ 3.6 ช่วงระยะเวลาผลความล่าช้าของการส่งข้อมูลที่ของกระบวนการคิวโอเอส 1 ที่พับบลิชเชอร์



รูปที่ 3.7 ช่วงระยะเวลาผลความล่าช้าของการส่งข้อมูลที่ของกระบวนการคิวโอเอส 2 ที่พับบลิชเชอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



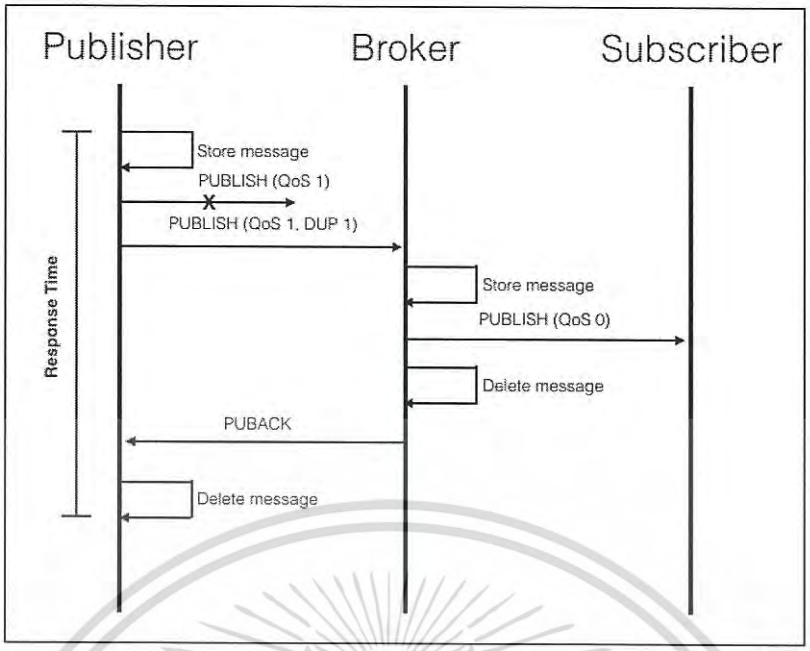
รูปที่ 3.8 ช่วงระยะเวลาผลความล่าช้าของการส่งข้อมูลที่ของวิธีการที่นำเสนอ

รูปที่ 3.6 3.7 และ 3.8 แสดงถึงผลความล่าช้าของการส่งข้อมูลของกระบวนการคิวโอเอส 1 คิวโอเอส 2 และวิธีการที่นำเสนอ โดยจะเห็นได้ว่าผลความล่าช้าของกระบวนการคิวโอเอส 1 คิวโอเอส 2 และวิธีการที่นำเสนอจะเริ่มวัดตั้งแต่การทดลองส่งจากพับบลิชเชอร์ครั้งแรกตลอดจนการส่งข้อมูลไปยังอุปกรณ์รอบข้างจนกระทั่งข้อมูลถึงซับสไครบเบอร์

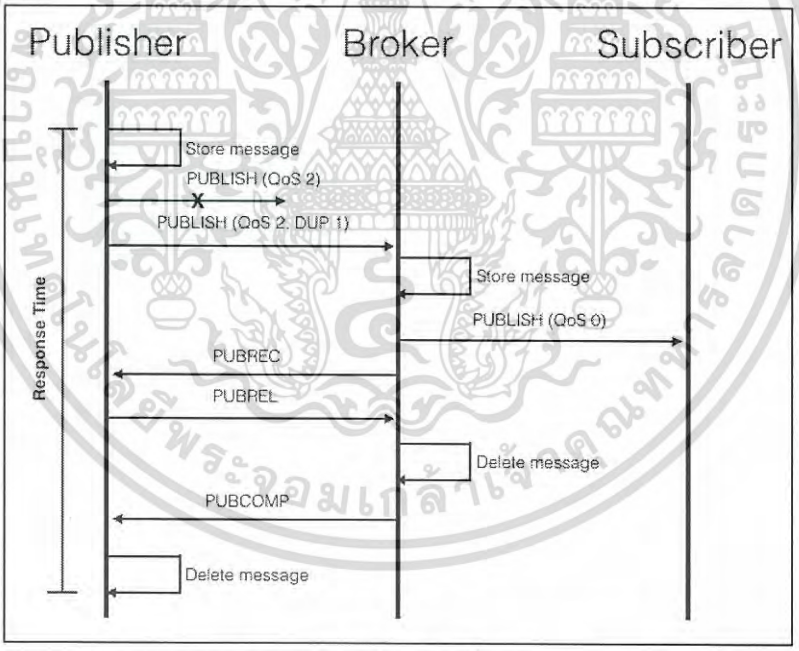
3.7.2 ผลความเร็วในการตอบสนองการส่งข้อมูล

ผลความเร็วในการตอบสนองการส่งข้อมูล (Response Time) จะวัดจากเวลา ณ ขณะที่พับบลิชเชอร์ได้รับผลลัพธ์ตอบกลับ หักลบกับเวลา ณ ขณะที่พับบลิชเชอร์เริ่มต้นส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

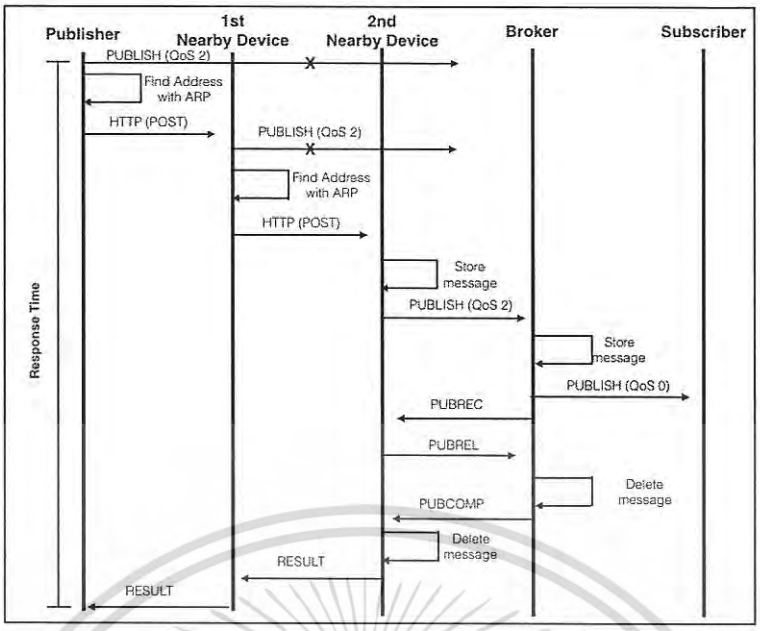


รูปที่ 3.9 ช่วงระยะเวลาผลการตอบสนองการส่งข้อมูลที่ของกระบวนการคิวโอเอส 1 ที่พับบลิชเชอร์



รูปที่ 3.10 ช่วงระยะเวลาผลการตอบสนองการส่งข้อมูลที่ของกระบวนการคิวโอเอส 2 ที่พับบลิชเชอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 ช่วงระยะเวลาผลการตอบสนองการส่งข้อมูลที่ของวิธีการที่นำเสนอ

จากรูปที่ 3.9 3.10 และ 3.11 แสดงถึงช่วงระยะเวลาผลการตอบสนองการส่งข้อมูลของกระบวนการคิวโอเอส 1 คิวโอเอส 2 และวิธีการที่นำเสนอ โดยจะเห็นได้ว่าผลการตอบสนองการส่งข้อมูลของกระบวนการคิวโอเอส 1 และคิวโอเอส 2 จะเริ่มวัดตั้งแต่การทดลองส่งข้อมูลจนกระทั่งได้รับผลตอบรับกลับจากโบรกเกอร์ ในขณะที่วิธีการที่นำเสนอจะเริ่มวัดตั้งแต่การทดลองส่งข้อมูลไปยังโบรกเกอร์จนกระทั่งได้รับผลตอบรับกลับจากอุปกรณ์รอบข้างที่สามารถส่งข้อมูลแทนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลและการวิเคราะห์ผลการวิจัย

ในบทนี้จะแบ่งออกเป็น 3 หัวข้อ ได้แก่ 1) ผลแสดงความล่าช้าการส่งข้อมูล 2) ผลแสดงความเร็วในการตอบสนองของข้อมูล และ 3) การวิเคราะห์และเปรียบเทียบผลความล่าช้าการส่งข้อมูลและความเร็วในการตอบสนองของข้อมูลในวิธีการต่าง ๆ โดยมีรายละเอียดดังนี้

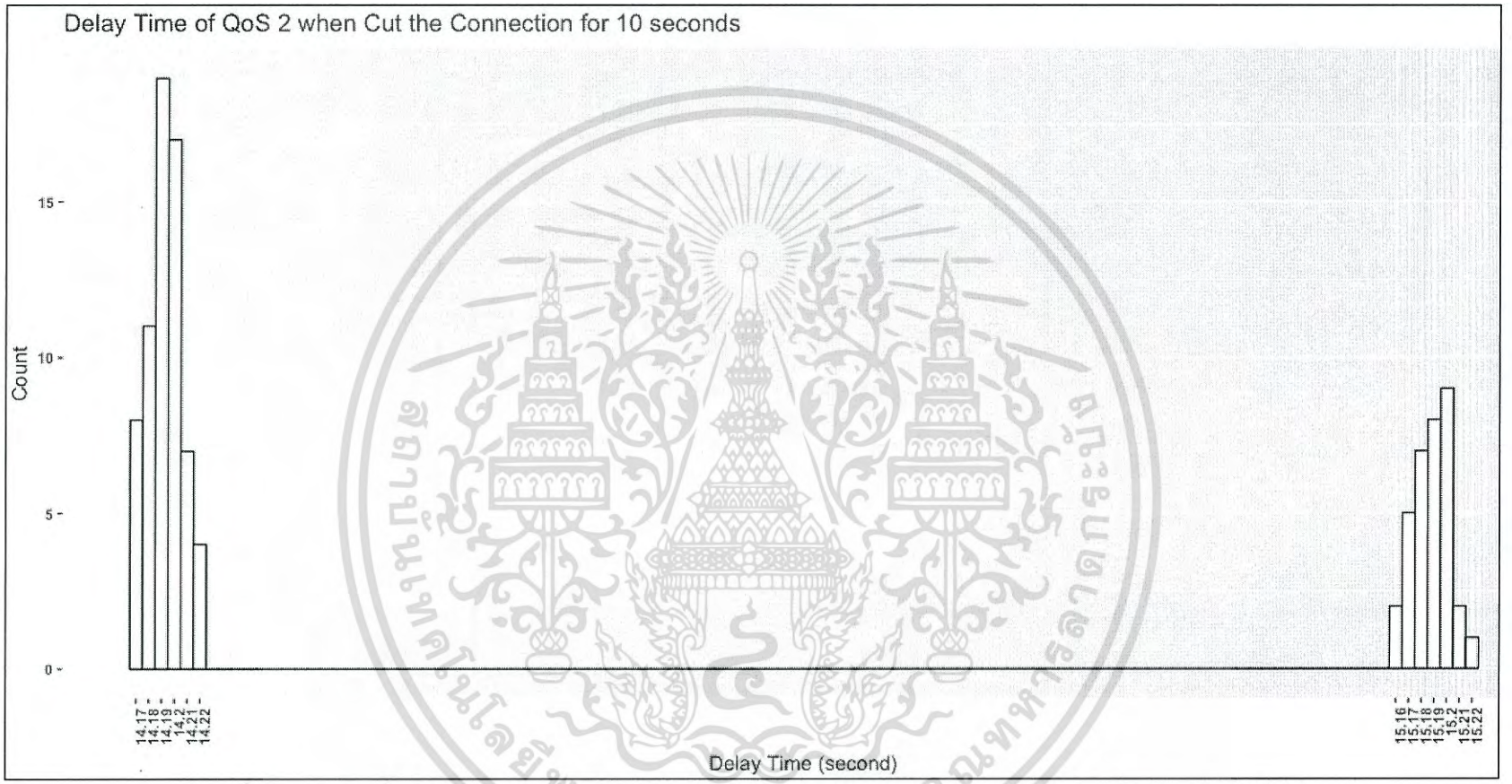
4.1 ผลแสดงความล่าช้าการส่งข้อมูล

จากการทดลองที่กล่าวไว้ในบทที่ 3 ผลการทดลองในส่วนนี้จะนำเสนอในเรื่องของความล่าช้าในการส่งข้อมูลจากการทดลองเป็นจำนวน 100 ครั้ง โดยแบ่งตามระยะเวลาที่ตัดการเชื่อมต่อระหว่างโบรกเกอร์กับพับบลิชเชอร์ตั้งต้นเป็นระยะเวลา 10 20 และ 30 วินาที โดยมีรายละเอียดผลลัพธ์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 ผลแสดงความล่าช้าการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 10 วินาที

รูปที่ 4.1 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 10 วินาที



รูปที่ 4.2 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 10 วินาที

รูปที่ 4.3 ความถี่ของความล่าช้าของการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 10 วินาที

รูปที่ 4.1 4.2 และ 4.3 แสดงถึงความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 1 คิวโอเอส 2 และวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 10 วินาที ตามลำดับ โดยจะเห็นได้ว่าความล่าช้าของกระบวนการคิวโอเอส 1 และคิวโอเอส 2 มีระยะเวลาอยู่ในช่วง 13.2-15.2 วินาที ในขณะที่ความล่าช้าของวิธีการที่นำเสนอส่วนใหญ่อยู่ในช่วง 6.7-6.9 วินาที แต่มีค่าที่มากที่สุดอยู่ที่ 21.6 วินาที ซึ่งค่าที่มากที่สุดนี้เกิดขึ้นแค่เพียงครั้งเดียวในระหว่างที่ทำการทดลองเท่านั้น

ตารางที่ 4.1 ผลความล่าช้าของการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 10 วินาที

	ความล่าช้าของการส่งข้อมูล (วินาที)		
	คิวโอเอส 1	คิวโอเอส 2	วิธีการที่นำเสนอ
ค่าเฉลี่ย	13.64	14.53	6.95
ค่ามากที่สุด	14.34	15.22	21.65
ค่าน้อยที่สุด	13.29	14.17	6.74
ค่าเบี่ยงเบนมาตรฐาน	0.47	0.47	1.48

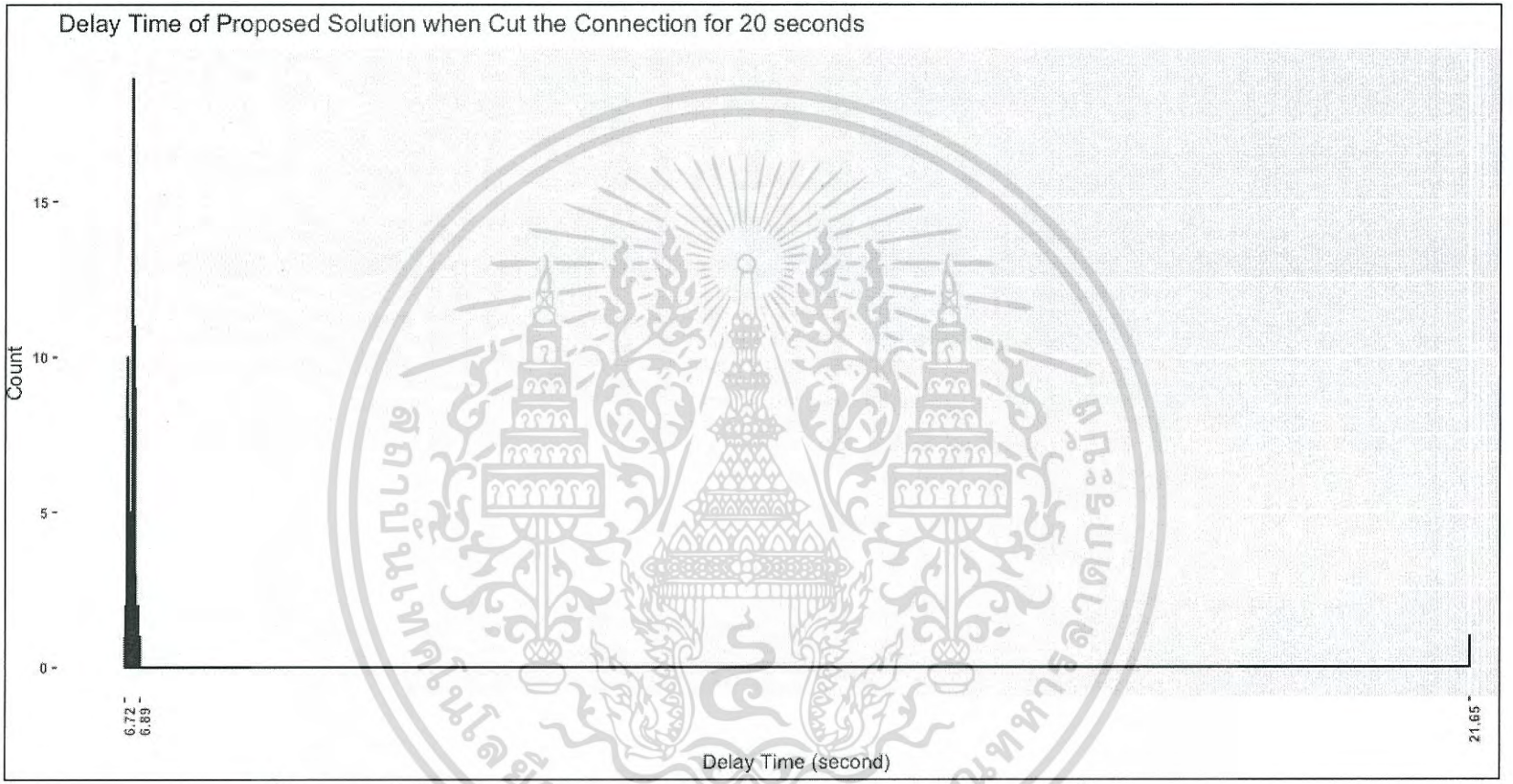
ผลลัพธ์ที่ได้จากตารางที่ 4.1 แสดงให้เห็นว่า ค่าเฉลี่ยความล่าช้าของวิธีการที่นำเสนอและความล่าช้าการส่งข้อมูลที่น้อยที่สุดของวิธีการที่นำเสนอจะใช้เวลาน้อยกว่ากระบวนการคิวโอเอส 1 และคิวโอเอส 2 ของเอ็มคิวทีทีโพรโทคอล เนื่องจากวิธีการที่นำเสนอไม่จำเป็นที่จะต้องรอให้การเชื่อมต่อกลับมาใช้งานได้ปกติ แต่จะใช้วิธีการทดลองส่งข้อมูลไปยังอุปกรณ์รอบข้างผ่านทางเครือข่ายอื่น เพื่อให้อุปกรณ์เหล่านั้นทำการส่งข้อมูลต่อไปยังโบรกเกอร์แทน แต่ในทางกลับกันค่าที่มากที่สุดของความล่าช้าการส่งข้อมูลด้วยวิธีการที่นำเสนอก็มีค่ามากที่สุดด้วยเมื่อเทียบกับกระบวนการคิวโอเอส ซึ่งค่าดังกล่าวจะเกิดจากครั้งแรกที่ทำการทดลองเนื่องจากการทดลองส่งข้อมูลไปยังอุปกรณ์รอบข้างในครั้งแรกนั้นยังไม่มีการจดจำเส้นทางของอุปกรณ์ที่เคยทดลองส่งได้สำเร็จ จึงจำเป็นที่จะต้องทดลองส่งข้อมูลไปยังอุปกรณ์รอบข้างทีละอุปกรณ์ในเครือข่าย ในขณะที่การส่งข้อมูลไปยังอุปกรณ์รอบข้างในครั้งต่อไป หลังจากทดลองส่งข้อมูลด้วยตัวเองแล้วพบว่าไม่สามารถส่งข้อมูลได้ พับบลิชเชอร์จะทำการส่งข้อมูลไปยังอุปกรณ์รอบข้างที่เคยทดลองส่งสำเร็จในครั้งก่อนหน้าเลยโดยที่ไม่ต้องทดลองส่งข้อมูลไปยังอุปกรณ์อื่นที่ไม่สามารถส่งข้อมูลแทนได้ ด้วยเหตุนี้จึงทำให้ค่าเบี่ยงเบนมาตรฐานของวิธีการที่นำเสนอมีค่ามากกว่าคิวโอเอส 1 และคิวโอเอส 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลแสดงความล่าช้าการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 20 วินาที

รูปที่ 4.4 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 20 วินาที

รูปที่ 4.5 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 20 วินาที



รูปที่ 4.6 ความถี่ของความล่าช้าของการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 20 วินาที

รูปที่ 4.4 4.5 และ 4.6 แสดงถึงความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 1 คิวโอเอส 2 และวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 20 วินาที ตามลำดับ โดยจะเห็นได้ว่าความล่าช้าของกระบวนการคิวโอเอส 1 และคิวโอเอส 2 มีระยะเวลาอยู่ในช่วง 23.3-25.3 วินาที ในขณะที่ความล่าช้าของวิธีการที่นำเสนอส่วนใหญ่อยู่ในช่วง 6.7-6.8 วินาที แต่มีค่าที่มากที่สุดอยู่ที่ 21.6 วินาที ซึ่งค่าที่มากที่สุดนี้เกิดขึ้นแค่เพียงครั้งเดียวในระหว่างที่ทำการทดลองเท่านั้น

ตารางที่ 4.2 ผลความล่าช้าของการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 20 วินาที

	ความล่าช้าของการส่งข้อมูล (วินาที)		
	คิวโอเอส 1	คิวโอเอส 2	วิธีการที่นำเสนอ
ค่าเฉลี่ย	23.66	24.76	6.94
ค่ามากที่สุด	24.53	25.34	21.65
ค่าน้อยที่สุด	23.31	24.15	6.72
ค่าเบี่ยงเบนมาตรฐาน	0.45	0.50	1.48

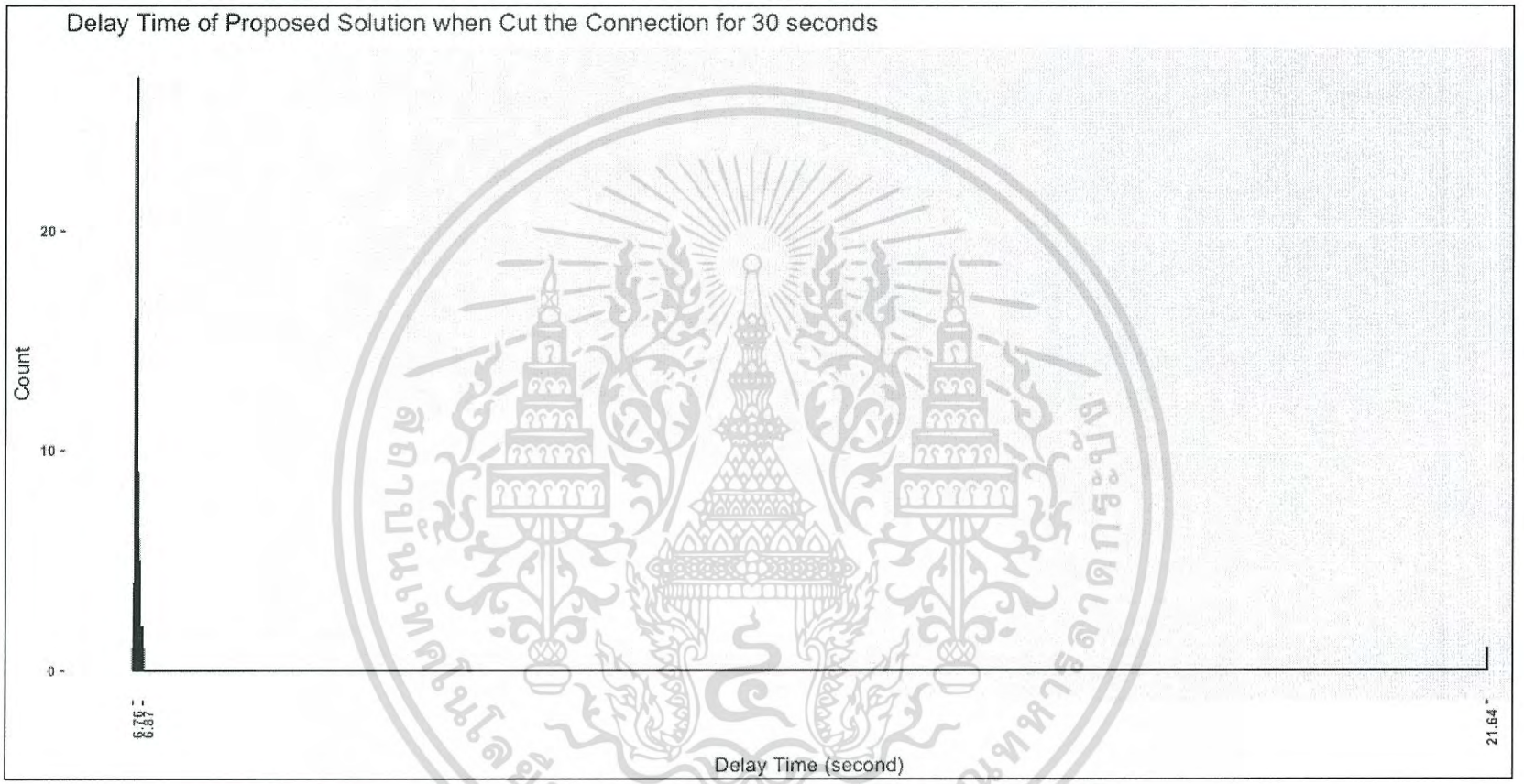
ผลลัพธ์ที่ได้จากตารางที่ 4.2 จะเห็นได้ว่าทั้งค่าเฉลี่ย ค่าที่มากที่สุด และค่าที่น้อยที่สุดของความล่าช้าการส่งข้อมูลในวิธีการที่นำเสนอจะใช้ระยะเวลาที่น้อยกว่ากระบวนการคิวโอเอส 1 และคิวโอเอส 2 ของเอ็มคิวทีทีโพรโทคอลทั้งหมด เนื่องจากกระบวนการคิวโอเอส 1 และ คิวโอเอส 2 จำเป็นที่จะต้องรอให้การเชื่อมต่อกลับมาใช้งานได้ปกติก่อนจึงจะทำการส่งข้อมูลซ้ำได้ ในขณะที่วิธีการที่นำเสนอได้ผลลัพธ์ใกล้เคียงกับผลความล่าช้าของการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 10 วินาที เพราะใช้วิธีการส่งข้อมูลผ่านทางอุปกรณ์รอบข้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ผลแสดงความล่าช้าการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 30 วินาที

รูปที่ 4.7 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 30 วินาที

รูปที่ 4.8 ความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 30 วินาที



รูปที่ 4.9 ความถี่ของความล่าช้าของการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 30 วินาที

รูปที่ 4.7 4.8 และ 4.9 แสดงถึงความถี่ของความล่าช้าของการส่งข้อมูลในกระบวนการคิวโอเอส 1 คิวโอเอส 2 และวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 30 วินาที ตามลำดับ โดยจะเห็นได้ว่าความล่าช้าของกระบวนการคิวโอเอส 1 และคิวโอเอส 2 มีระยะเวลาอยู่ในช่วง 35.3-37.2 วินาที ในขณะที่ความล่าช้าของวิธีการที่นำเสนอส่วนใหญ่อยู่ในช่วง 6.7-6.8 วินาที แต่มีค่าที่มากที่สุดอยู่ที่ 21.6 วินาที ซึ่งค่าที่มากที่สุดนี้เกิดขึ้นแค่เพียงครั้งเดียวในระหว่างที่ทำการทดลองเท่านั้น

ตารางที่ 4.3 ผลความล่าช้าของการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 30 วินาที

	ความล่าช้าของการส่งข้อมูล (วินาที)		
	คิวโอเอส 1	คิวโอเอส 2	วิธีการที่นำเสนอ
ค่าเฉลี่ย	35.75	36.67	6.95
ค่ามากที่สุด	36.46	37.26	21.64
ค่าน้อยที่สุด	35.39	36.18	6.76
ค่าเบี่ยงเบนมาตรฐาน	0.46	0.50	1.48

ตารางที่ 4.3 จะมีผลลัพธ์ที่คล้ายคลึงกับตารางที่ 4.2 คือวิธีการที่นำเสนอจะได้ผลลัพธ์ที่ดีกว่ากระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอลในทุก ๆ ผลลัพธ์ทั้งค่าเฉลี่ย ค่าที่มากที่สุด และค่าที่น้อยที่สุดของความล่าช้าในการส่งข้อมูล

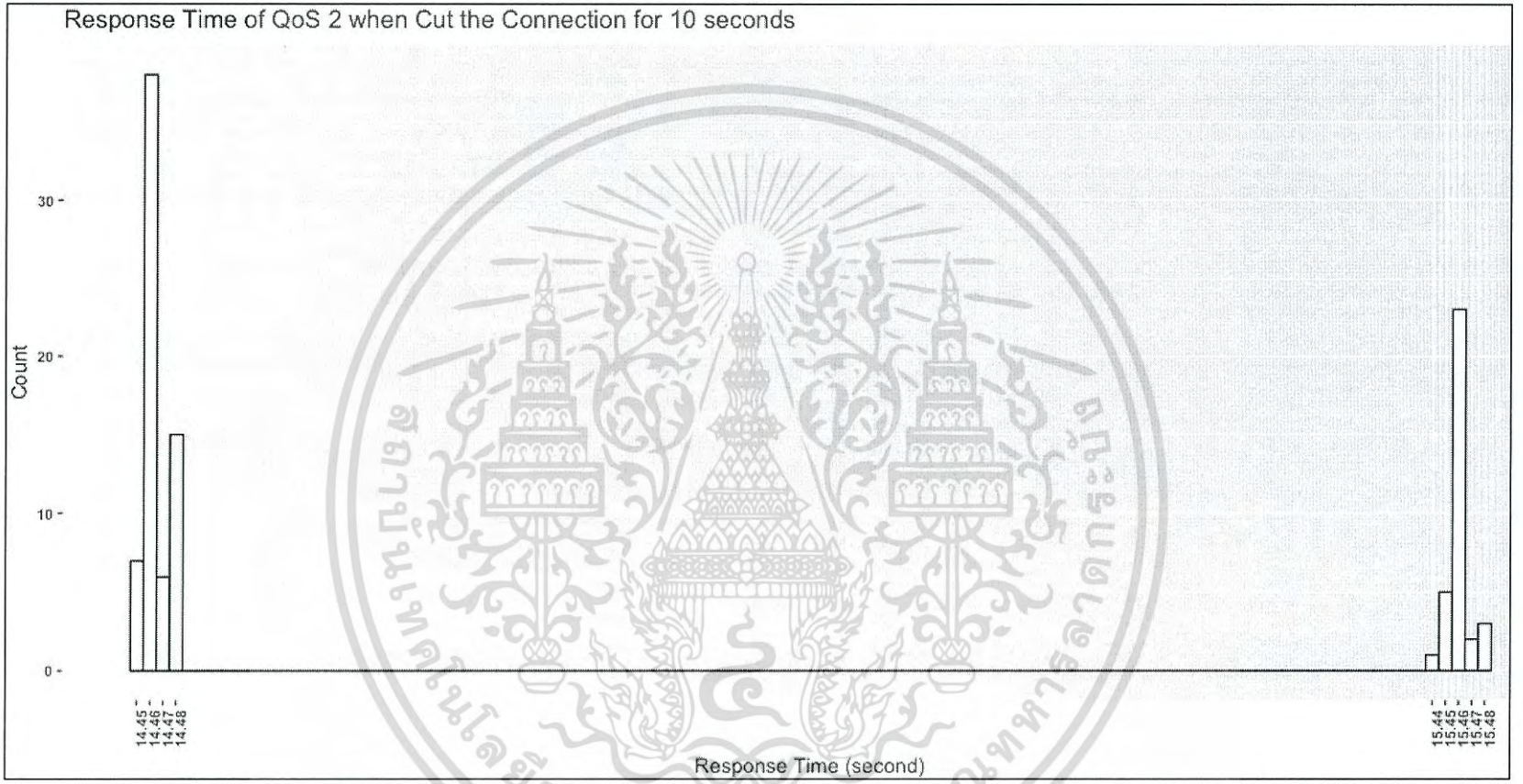
4.2 ผลแสดงความเร็วในการตอบสนองของข้อมูล

เช่นเดียวกับผลแสดงความล่าช้าการส่งข้อมูล ความเร็วในการตอบสนองของข้อมูลก็จะถูกแบ่งตามระยะเวลาที่การเชื่อมต่อระหว่างพับบลิชเชอร์ตั้งต้นกับโบรกเกอร์ถูกตัดขาด โดยมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 ผลแสดงความเร็วในการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 10 วินาที

รูปที่ 4.10 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 10 วินาที



รูปที่ 4.11 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 10 วินาที

รูปที่ 4.12 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 10 วินาที

รูปที่ 4.10 4.11 และ 4.12 แสดงถึงความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 1 คิวโอเอส 2 และวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 10 วินาที ตามลำดับ โดยจะเห็นได้ว่าความเร็วการตอบสนองของกระบวนการคิวโอเอส 1 และคิวโอเอส 2 มีระยะเวลาอยู่ในช่วง 13.5-15.4 วินาที ในขณะที่ความเร็วการตอบสนองของวิธีการที่นำเสนอส่วนใหญ่อยู่ในช่วง 6.7-6.9 วินาที แต่มีค่าที่มากที่สุดอยู่ที่ 21.6 วินาที ซึ่งค่าที่มากที่สุดนี้เกิดขึ้นแค่เพียงครั้งเดียวในระหว่างที่ทำการทดลองเท่านั้น

ตารางที่ 4.4 ผลแสดงความเร็วในการตอบสนองการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 10 วินาที

	ความเร็วการตอบสนองการส่งข้อมูล (วินาที)		
	คิวโอเอส 1	คิวโอเอส 2	วิธีการที่นำเสนอ
ค่าเฉลี่ย	13.89	14.80	6.97
ค่ามากที่สุด	14.58	15.48	21.65
ค่าน้อยที่สุด	13.54	14.45	6.75
ค่าเบี่ยงเบนมาตรฐาน	0.47	0.47	1.48

ตารางที่ 4.4 แสดงถึงความเร็วในการตอบสนองเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 10 วินาที ซึ่งจากผลลัพธ์นี้แสดงให้เห็นว่า ค่าเฉลี่ยและค่าที่น้อยที่สุดของความเร็วการตอบสนองในวิธีการที่นำเสนอจะใช้ระยะเวลาที่น้อยกว่ากระบวนการคิวโอเอส 1 และคิวโอเอส 2 ของเอ็มคิวทีทีโพรโทคอล ในทางกลับกัน ค่าที่มากที่สุดของความเร็วการตอบสนองการส่งข้อมูลในวิธีการที่นำเสนอก็จะใช้ระยะเวลาที่มากด้วยเช่นกัน ผลลัพธ์นี้จะมีลักษณะที่คล้ายกับผลความล่าช้าการส่งข้อมูลในตารางที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลแสดงความเร็วในการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 20 วินาที

รูปที่ 4.13 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 20 วินาที

รูปที่ 4.14 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 20 วินาที

รูปที่ 4.15 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 20 วินาที

รูปที่ 4.13 4.14 และ 4.15 แสดงถึงความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 1 คิวโอเอส 2 และวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 20 วินาที ตามลำดับ โดยจะเห็นได้ว่าความเร็วการตอบสนองของกระบวนการคิวโอเอส 1 และคิวโอเอส 2 มีระยะเวลาอยู่ในช่วง 25-25.6 วินาที ในขณะที่ความเร็วการตอบสนองของวิธีการที่นำเสนอส่วนใหญ่อยู่ในช่วง 6.7-6.9 วินาที แต่มีค่าที่มากที่สุดอยู่ที่ 21.6 วินาที ซึ่งค่าที่มากที่สุดนี้เกิดขึ้นแค่เพียงครั้งเดียวในระหว่างที่ทำการทดลองเท่านั้น

ตารางที่ 4.5 ผลแสดงความเร็วในการตอบสนองการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 20 วินาที

	ความเร็วการตอบสนองการส่งข้อมูล (วินาที)		
	คิวโอเอส 1	คิวโอเอส 2	วิธีการที่นำเสนอ
ค่าเฉลี่ย	23.84	25.05	6.96
ค่ามากที่สุด	24.75	25.63	21.67
ค่าน้อยที่สุด	23.54	24.44	6.74
ค่าเบี่ยงเบนมาตรฐาน	0.45	0.50	1.48

ผลลัพธ์ที่ได้จากตารางที่ 4.5 จะเห็นได้ว่าทั้งค่าเฉลี่ย ค่าที่มากที่สุด และค่าที่น้อยที่สุดของความเร็วการตอบสนองในวิธีการที่นำเสนอจะใช้ระยะเวลาที่น้อยกว่าในทุก ๆ รอบการทดลอง เมื่อเทียบกับกระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ผลแสดงความเร็วในการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 30 วินาที

รูปที่ 4.16 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 1 เมื่อตัดการเชื่อมต่อ 30 วินาที

รูปที่ 4.17 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 2 เมื่อตัดการเชื่อมต่อ 30 วินาที

รูปที่ 4.18 ความถี่ของความเร็วการตอบสนองการส่งข้อมูลในวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 30 วินาที

รูปที่ 4.16 4.17 และ 4.18 แสดงถึงความเร็วการตอบสนองการส่งข้อมูลในกระบวนการคิวโอเอส 1 คิวโอเอส 2 และวิธีการที่นำเสนอ เมื่อตัดการเชื่อมต่อ 30 วินาที ตามลำดับ โดยจะเห็นได้ว่าความเร็วการตอบสนองของกระบวนการคิวโอเอส 1 และคิวโอเอส 2 มีระยะเวลาอยู่ในช่วง 35.5-37.5 วินาที ในขณะที่ความเร็วการตอบสนองของวิธีการที่นำเสนอส่วนใหญ่อยู่ในช่วง 6.8-6.9 วินาที แต่มีค่าที่มากที่สุดอยู่ที่ 21.6 วินาที ซึ่งค่าที่มากที่สุดนี้เกิดขึ้นแค่เพียงครั้งเดียวในระหว่างที่ทำการทดลองเท่านั้น

ตารางที่ 4.6 จะมีผลลัพธ์ที่คล้ายคลึงกับตารางที่ 4.5 คือวิธีการที่นำเสนอจะได้ผลลัพธ์ที่ดีกว่ากระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอลในทุก ๆ ผลลัพธ์ ทั้งค่าเฉลี่ย ค่าที่มากที่สุด และค่าที่น้อยที่สุดของความเร็วการตอบสนอง

ตารางที่ 4.6 ผลแสดงความเร็วในการตอบสนองการส่งข้อมูลเมื่อตัดการเชื่อมต่อเป็นระยะเวลา 30 วินาที

	ความเร็วการตอบสนองการส่งข้อมูล (วินาที)		
	คิวโอเอส 1	คิวโอเอส 2	วิธีการที่นำเสนอ
ค่าเฉลี่ย	35.95	36.97	6.98
ค่ามากที่สุด	36.65	37.57	21.67
ค่าน้อยที่สุด	35.59	36.49	6.80
ค่าเบี่ยงเบนมาตรฐาน	0.46	0.50	1.48

4.3 การวิเคราะห์และเปรียบเทียบผลความล่าช้าการส่งข้อมูลและความเร็วในการตอบสนองของข้อมูลในวิธีการต่าง ๆ

ในหัวข้อนี้เป็นการนำผลลัพธ์ความล่าช้าการส่งข้อมูลและความเร็วในการตอบสนองของข้อมูลที่เกิดจากการทดลองในส่วนก่อนหน้ามาสรุป วิเคราะห์ และเปรียบเทียบผลลัพธ์ของแต่ละวิธีการทดลองโดยแบ่งตามระยะเวลาที่การเชื่อมต่อถูกตัดขาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 ผลความล่าช้าของการส่งข้อมูลโดยเฉลี่ย

ระยะเวลาที่ตัดการเชื่อมต่อ (วินาที)	ความล่าช้าของการส่งข้อมูล (วินาที)		
	คิวโอเอส 1	คิวโอเอส 2	วิธีการที่นำเสนอ
10	13.64	14.53	6.95
20	23.66	24.76	6.94
30	35.75	36.67	6.95

ตารางที่ 4.7 แสดงถึงความล่าช้าของการส่งข้อมูลโดยเฉลี่ย คอลัมน์แรกแสดงถึงระยะเวลาที่การเชื่อมต่อถูกตัดขาดเป็นระยะเวลา 10 20 และ 30 วินาทีตามลำดับ ในขณะที่ความล่าช้าของแต่ละวิธีการจะ แสดงในคอลัมน์ที่ 2 3 และ 4 โดยที่คิวโอเอส 1 อยู่ในคอลัมน์ที่ 2 คิวโอเอส 2 อยู่ในคอลัมน์ที่ 3 และวิธีการที่นำเสนออยู่ในคอลัมน์ที่ 4 เพื่อแสดงให้เห็นถึงภาพรวมและการเปรียบเทียบที่ชัดเจนมากยิ่งขึ้น

ตารางที่ 4.8 ผลความการตอบสนองการส่งข้อมูลโดยเฉลี่ย

ระยะเวลาที่ตัดการเชื่อมต่อ (วินาที)	ความเร็วการตอบสนองการส่งข้อมูล (วินาที)		
	คิวโอเอส 1	คิวโอเอส 2	วิธีการที่นำเสนอ
10	13.89	14.80	6.97
20	23.84	25.05	6.96
30	35.95	36.98	6.98

ตารางที่ 4.8 แสดงถึงความเร็วการตอบสนองของการส่งข้อมูลโดยเฉลี่ย คอลัมน์แรกแสดงระยะเวลาที่การเชื่อมต่อถูกตัดขาดเป็นระยะเวลา 10 20 และ 30 วินาทีตามลำดับ ในขณะที่ความล่าช้าของแต่ละวิธีการจะแสดงในคอลัมน์ที่ 2 3 และ 4 โดยที่คิวโอเอส 1 อยู่ในคอลัมน์ที่ 2 คิวโอเอส 2 อยู่ในคอลัมน์ที่ 3 และวิธีการที่นำเสนออยู่ในคอลัมน์ที่ 4 เพื่อแสดงให้เห็นถึงภาพรวมและ การเปรียบเทียบที่ชัดเจนมากยิ่งขึ้น

ตารางที่ 4.7 และ 4.8 เป็นผลลัพธ์โดยเฉลี่ยความล่าช้าของการส่งข้อมูลและความเร็วในการส่งข้อมูลจากการทดลองตามลำดับ โดยผลลัพธ์จากทั้งสองตารางมีแนวโน้มไปในทิศทางเดียวกันคือ วิธีการที่นำเสนอจะได้ผลลัพธ์ที่รวดเร็วที่สุด คิวโอเอส 1 ของเอ็มคิวทีทีโพรโทคอลได้ผลลัพธ์เป็นลำดับที่สอง และคิวโอเอส 2 ของเอ็มคิวทีทีโพรโทคอลใช้ระยะเวลามากที่สุดตามลำดับ

หากพิจารณาที่คิวโอเอส 1 และคิวโอเอส 2 ของเอ็มคิวทีทีโพรคอล จะสังเกตได้ว่าไม่ว่าจะเป็นความล่าช้าหรือความเร็วการตอบสนองของการส่งข้อมูล ล้วนมีผลลัพธ์ที่มากกว่าระยะเวลาที่ การเชื่อมต่อถูกตัดขาดทั้งสิ้น จากการวิเคราะห์ผลลัพธ์ทำให้ทราบว่า การส่งข้อมูลซ้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปยังอุปกรณ์ตัวเดิมในกระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอลจำเป็นที่จะต้องรอการเชื่อมต่อให้กลับมาใช้ได้ปกติก่อนจึงจะทำการส่งข้อมูลไปยังอุปกรณ์ปลายทางได้ ดังนั้นผลลัพธ์ความล่าช้าและความเร็วในการตอบสนองของการส่งข้อมูลในกระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอลจึงแปรผันตามระยะเวลาที่การเชื่อมต่อถูกตัดขาดไป กล่าวคือ หากปัญหาการเชื่อมต่อมีระยะเวลาที่ยาวนาน ระยะเวลาในการส่งข้อมูลซ้ำก็จะยาวนานตามไปด้วย

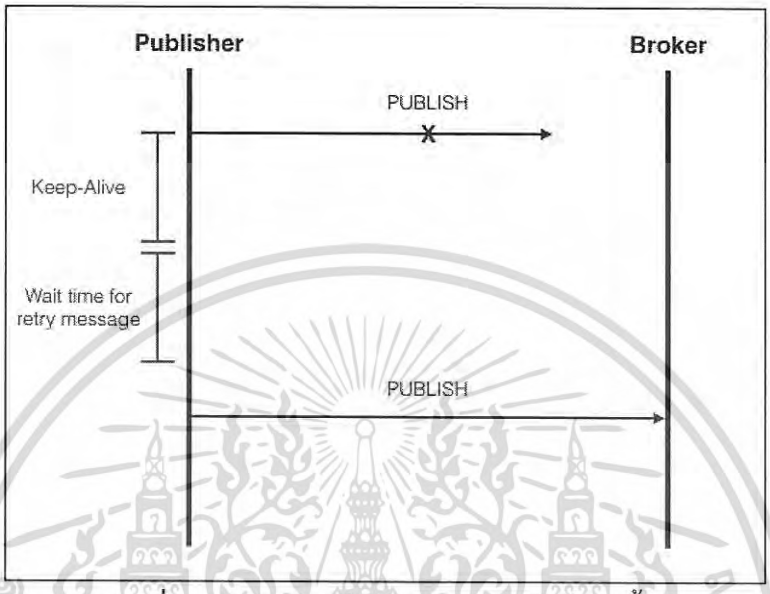
ในขณะที่ผลลัพธ์ความล่าช้าและความเร็วในการตอบสนองของการส่งข้อมูลในกระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอลจำเป็นที่จะต้องรอการเชื่อมต่อให้กลับมาใช้ได้ใหม่ วิธีการที่นำเสนอกลับแสดงถึงผลลัพธ์ที่รวดเร็วกว่าอย่างเห็นได้ชัด และยังใช้ระยะเวลาในการส่งข้อมูลน้อยกว่าระยะเวลาที่การเชื่อมต่อถูกตัดขาด เนื่องจากวิธีการที่นำเสนอไม่จำเป็นที่จะต้องรอการเชื่อมต่อที่มีปัญหาให้กลับมาใช้ได้ใหม่เหมือนกับกระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอลแต่ใช้วิธีการส่งข้อมูลไปยังอุปกรณ์รอบข้างแทน นอกจากนั้นการจดจำเส้นทางของอุปกรณ์ที่สามารถส่งข้อมูลได้ก็ช่วยลดระยะเวลาในค้นหาอุปกรณ์รอบข้างและการทดลองส่งข้อมูลไปยังอุปกรณ์อื่นโดยไม่จำเป็นได้ แต่ถ้าไม่มีการจดจำเส้นทางมาก่อนหน้า ก็มีโอกาสที่กระบวนการคิวโอเอส 1 และ 2 ของเอ็มคิวทีทีโพรโทคอลจะได้ผลลัพธ์ที่ดีกว่าหากปัญหาการเชื่อมต่อมีระยะเวลาที่สั้น

นอกจากวิธีการที่นำเสนอนี้จะใช้ระยะเวลาที่น้อยกว่ากระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอลแล้ว ผลลัพธ์ของเวลาที่ได้ยังมีความเสถียรมากกว่ากระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอลเมื่อเทียบกับระยะเวลาที่การเชื่อมต่อถูกตัดขาดในแต่ละครั้ง กล่าวคือ แม้ระยะเวลาที่ใช้ในการตัดการเชื่อมต่อจะมากขึ้น (10 20 และ 30 วินาทีตามลำดับ) แต่วิธีการที่นำเสนอจะใช้ระยะเวลาที่ใกล้เคียงเวลาเดิมเสมอ ในขณะที่กระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอลใช้ระยะเวลาแปรผันตามระยะเวลาที่การเชื่อมต่อถูกตัดขาด เนื่องจากระยะเวลาในการส่งข้อมูลของวิธีการที่นำเสนอจะขึ้นอยู่กับจำนวนอุปกรณ์รอบข้าง จำนวนเครือข่ายที่เชื่อมต่อในระบบ และจำนวนอุปกรณ์ที่สามารถส่งต่อข้อมูลแทนได้ หากยิ่งมีจำนวนอุปกรณ์รอบข้างกับเครือข่ายที่เชื่อมต่ออยู่ในระบบมากเท่าไร พับบลิชเชอร์ตั้งต้นก็จำเป็นที่จะต้องทดลองส่งข้อมูลไปยังอุปกรณ์และเครือข่ายเหล่านั้นมากยิ่งขึ้น ซึ่งถ้าหากอุปกรณ์ที่สามารถส่งต่อได้นี้มีน้อย ก็จะยิ่งทำให้ระยะเวลาที่ใช้ในการทดลองส่งนี้มีค่ามากตามไปด้วย แต่ถ้าอุปกรณ์ที่สามารถส่งต่อแทนได้มีมาก ก็จะเป็นการช่วยเพิ่มโอกาสในการส่งข้อมูลให้มากยิ่งขึ้น นอกจากนั้น หากการทดลองส่งข้อมูลผ่านทางอุปกรณ์รอบข้างประสบความสำเร็จ จะส่งผลให้การส่งข้อมูลผ่านทางอุปกรณ์รอบข้างในครั้งต่อไปอาจใช้ระยะเวลาที่สั้นลง เนื่องจากไม่จำเป็นที่จะต้องค้นหาอุปกรณ์รอบข้างและทดลองส่งข้อมูลไปยังอุปกรณ์ที่ไม่สามารถส่งแทนได้ ซึ่งถือเป็นการลดกระบวนการการทำงานในวิธีการที่นำเสนอ เป็นผลให้ระยะเวลาที่ใช้สั้นลง

ผลลัพธ์ที่ได้จากการทดลองนี้จะพบว่าการแกว่งของข้อมูลเกิดขึ้นทั้งในคิวโอเอส 1 คิวโอเอส 2 และวิธีการที่นำเสนอ จากการวิเคราะห์ผลลัพธ์นี้น่าจะมีสาเหตุมาจาก Virtual Machine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Scheduling เนื่องจากการส่งข้อมูลแพ็คเกจคิวมีความเป็นไปได้ที่จะไม่ตรงกับ time slot ของการส่งข้อมูล [22] จึงเป็นเหตุให้มีผลลัพธ์ที่ได้ไม่คงที่และเกิดการกระจายตัวของข้อมูลในแต่ละวิธีการ ประกอบกับจำนวนผลการทดลองที่อาจจะน้อยเกินไป จึงทำให้การกระจายตัวของข้อมูลไม่อยู่ในรูปแบบปกติ (ลักษณะระฆังคว่ำ)



รูปที่ 4.19 ช่วงระยะเวลารอก่อนการส่งข้อมูลซ้ำ

หากพิจารณาที่หน่วยของผลลัพธ์ที่ได้จากการทดลองจะพบว่าหน่วยเป็นวินาที เนื่องจากหลังจากการส่งข้อมูลครั้งแรกไปยังโบรกเกอร์ คิวโอเอส 1 และคิวโอเอส 2 จะทำการรอให้ผลตอบกลับจากโบรกเกอร์ ซึ่งหากการเชื่อมต่อถูกตัดขาดไป ผลตอบกลับจะไม่มีทางกลับมาถึงพับบลิชเชอร์แน่นอน ซึ่งคิวโอเอส 1 และคิวโอเอส 2 จะรอผลตอบกลับนี้จนกระทั่งหมดช่วงระยะเวลา keep-alive แล้วจึงรอช่วงระยะเวลาก่อนที่จะทำการส่งข้อมูลกลับไปยังโบรกเกอร์ตามกระบวนการคิวโอเอสที่ระบุ (สามารถตั้งค่าได้จาก message_retry_set() ซึ่งค่าตั้งต้นมีค่าเท่ากับ 5 วินาที) (รูปที่ 4.19) ดังนั้นการส่งข้อมูลด้วยคิวโอเอส 1 และคิวโอเอส 2 ส่วนใหญ่จึงถูกใช้ไปกับการรอให้ช่วงระยะเวลาของ keep-alive และช่วงระยะเวลาที่รอก่อนการส่งข้อมูลซ้ำ จึงมีความเป็นไปได้ที่หลังจากหมดช่วงระยะเวลาที่ตัดการเชื่อมต่อ 10 20 และ 30 วินาทีไปแล้ว การส่งข้อมูลจากพับบลิชเชอร์ยังคงอยู่ระหว่างการรอระยะเวลาสำหรับการส่งข้อมูลใหม่ ในขณะที่วิธีการที่นำเสนอจะทำการรอเวลา keep-alive ของการทดลองส่งข้อมูลด้วยเอ็มคิวทีทีโพรโทคอล ถ้าหมดช่วงระยะเวลานี้ ก็จะเป็นการเริ่มทำตามวิธีการที่นำเสนอ ซึ่งสำหรับการทดลองนี้ วิธีการที่นำเสนอจำเป็นที่จะต้องรอช่วงระยะเวลา keep-alive ของการทดลองส่งข้อมูลที่พับบลิชเชอร์และอุปกรณ์ข้างเคียงตัวที่หนึ่ง ด้วยเหตุนี้จึงเป็นเหตุผลให้การส่งข้อมูลด้วยวิธีการที่นำเสนอในครั้งแรกใช้ระยะเวลาที่ยาวนาน เนื่องจากจำเป็นที่จะต้องค้นหาอุปกรณ์รอบข้างและทดลองส่งข้อมูลไปยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์รอบข้างทีละอุปกรณ์พร้อมรอให้หมดช่วงเวลา keep-alive หรือรอผลลัพธ์ตอบกลับจากโบรกเกอร์ก่อนที่ส่งผลลัพธ์กลับไปยังพับบลิชเชอร์ตัวแรกที่ทำการส่งข้อมูล ก่อนที่การส่งข้อมูลในครั้งต่อไปจะส่งข้อมูลไปตามรายการอุปกรณ์รอบข้างที่เคยทดลองส่งในครั้งก่อนได้สำเร็จ จึงเป็นผลให้สามารถลดระยะเวลาสำหรับการรอช่วงเวลา keep-alive ของอุปกรณ์ที่ไม่สามารถส่งข้อมูลได้ กล่าวคือ หากการเชื่อมต่อเป็นปกติ การส่งข้อมูลด้วยโพรโทคอลเอ็มคิวทีทีจะมีระยะเวลาที่สั้น แต่ในกรณีที่ไม่สามารถส่งข้อมูลได้ พับบลิชเชอร์จะทำการรอผลตอบกลับจากโบรกเกอร์จนกว่าจะหมดระยะเวลาการเชื่อมต่อ และรอต่อไปอีก 5 วินาทีจึงจะทำการส่งข้อมูลซ้ำไปยังโบรกเกอร์ ดังนั้นระยะเวลาที่ได้จากการทดลองนี้ที่มากถึงระดับวินาทีจึงน่าจะมีสาเหตุมาจากการที่ต้องรอให้หมดช่วงระยะเวลาดังกล่าว

จากผลลัพธ์ข้างต้นจะพบว่า วิธีการที่นำเสนอจะใช้ระยะเวลาในการส่งข้อมูลอยู่ที่ประมาณ 6 วินาที ซึ่งระยะเวลาทั้งหมดนี้จะถูกแบ่งเป็นระยะเวลาที่ใช้ในการรอการเชื่อมต่อถูกตัดขาดในช่วงทดลองการส่งข้อมูลครั้งแรกจากพับบลิชเชอร์ตั้งต้น 5 วินาที และกระบวนการที่ใช้ในการค้นหาอุปกรณ์รอบข้าง ส่งต่อข้อมูลด้วยเอชทีทีพีไปยังอุปกรณ์รอบข้างสองอุปกรณ์ ตลอดจนการส่งข้อมูลด้วยเอ็มคิวทีทีจากอุปกรณ์รอบข้างไปยังโบรกเกอร์ผ่านทางเครือข่ายอื่นและรอให้เกิดการส่งผลลัพธ์กลับมายังพับบลิชเชอร์ตั้งต้นประมาณ 1 วินาที

ในกรณีที่การเชื่อมต่อสามารถใช้งานได้ปกติ (ไม่เกิดการสูญหายของข้อมูล) วิธีการที่นำเสนอจะใช้ผลลัพธ์ที่ใกล้เคียงกับกระบวนการคิวโอเอส 2 เนื่องจากขั้นตอนแรกของวิธีการที่นำเสนอคือการทดลองส่งข้อมูลจากพับบลิชเชอร์ไปยังโบรกเกอร์ด้วยคิวโอเอส 2 ของเอ็มคิวทีทีโพรโทคอล ดังนั้นหากเปรียบเทียบกับคิวโอเอส 1 แล้ว ระยะเวลาที่ใช้ในการส่งข้อมูลอาจจะมากกว่าเล็กน้อยตามลักษณะการส่งข้อมูลของคิวโอเอส 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ในบทนี้แบ่งออกเป็น 2 หัวข้อคือ สรุปผลการวิจัย และข้อเสนอแนะ โดยมีรายละเอียดดังต่อไปนี้

5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอแนวคิดในการแก้ไขปัญหาการส่งข้อมูลล่าช้าของกระบวนการคิวโอเอสในเอ็มคิวทีทีโพรโทคอล อันเกิดจากปัญหาการเชื่อมต่อที่เกิดขึ้นเป็นระยะเวลานาน ทำให้การส่งข้อมูลซ้ำที่เกิดจากอุปกรณ์ส่งตัวเดิม ไม่สามารถส่งข้อมูลได้ตามระยะเวลาของปัญหาที่เกิดขึ้น โดยวิธีการที่นำเสนอนี้จะใช้แนวคิดของการส่งข้อมูลไปยังอุปกรณ์รอบข้าง เพื่อเปลี่ยนเส้นทางการส่งข้อมูลไปยังช่องทางอื่นแทนช่องทางการเชื่อมต่อเดิมที่เกิดปัญหา

การส่งต่อข้อมูลไปยังอุปกรณ์รอบข้างด้วยวิธีการนี้ไม่จำกัดเฉพาะอุปกรณ์ที่เชื่อมอยู่กับพับบลิชเชอร์โดยตรงเท่านั้น แต่รวมถึงอุปกรณ์ที่อยู่ในเครือข่ายอื่นด้วย ทำให้โอกาสการส่งข้อมูลมีมากขึ้นตามจำนวนอุปกรณ์ที่มีในระบบ แต่ก็จะทำให้การทดลองส่งข้อมูลเกิดความล่าช้าตามจำนวนอุปกรณ์ที่มีอยู่ในระบบเช่นเดียวกัน ดังนั้นวิธีการนี้จึงมีการจดจำเส้นการส่งข้อมูลที่เคยส่งได้สำเร็จสำหรับใช้ในการส่งข้อมูลผ่านทางอุปกรณ์รอบข้างในครั้งต่อไปเพื่อลดระยะเวลาในการค้นหาอุปกรณ์รอบข้างและการทดลองส่งข้อมูล

ผลลัพธ์จากการทดลอง สามารถสรุปได้ว่า วิธีการที่นำเสนอจะรับส่งข้อมูลได้ช้ากว่าหากระยะเวลาที่เกิดปัญหาการเชื่อมต่อมีระยะเวลาสั้น แต่หากปัญหาระยะเวลาการเชื่อมต่อเกิดขึ้นเป็นระยะเวลานาน วิธีการที่นำเสนอจะได้ผลลัพธ์ที่รวดเร็วและคงที่กว่าคิวโอเอสของเอ็มคิวทีทีโพรโทคอล นอกจากนั้นหลังจากการส่งข้อมูลครั้งแรกของวิธีการที่นำเสนอ ระยะเวลาในการส่งข้อมูลจะใช้เวลาที่น้อยลงอย่างมาก เนื่องจากมีการจดจำเส้นทางสำหรับการส่งข้อมูลไว้อยู่แล้ว จึงทำให้ไม่จำเป็นที่จะต้องค้นหาอุปกรณ์รอบข้างหรือทดลองส่งข้อมูลไปยังอุปกรณ์อื่นที่ไม่สามารถส่งได้อีก อย่างไรก็ตาม ความเร็วในการทดลองส่งข้อมูลของวิธีการที่นำเสนอนั้น ก็ขึ้นอยู่กับจำนวนอุปกรณ์ภายในเครือข่าย จำนวนอุปกรณ์ที่สามารถส่งข้อมูลแทนได้ และจำนวนเครือข่ายทั้งหมดในระบบ กล่าวคือ ยิ่งมีปัจจัยเหล่านี้มากเท่าไหร่ ก็จะทำให้ระยะเวลาในการส่งข้อมูลมากขึ้น แต่ก็มีโอกาสที่จะส่งข้อมูลได้สำเร็จมากขึ้นเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อเสนอแนะ

ในการทดลองของงานวิจัยนี้ยังมีสิ่งที่น่าสนใจอยู่อีกหลายส่วนเพื่อเพิ่มประสิทธิภาพในการส่งข้อมูลให้ดียิ่งขึ้นได้ โดยมีรายละเอียดดังนี้

1. การส่งต่อข้อมูลไปยังอุปกรณ์รอบข้างในขณะนี้ยังเป็นการส่งทีละอุปกรณ์ ถ้าสามารถปรับให้เป็นการส่งแบบหลายอุปกรณ์พร้อมกันได้ ก็จะสามารถลดระยะเวลาที่ทดลองส่งข้อมูลได้มากยิ่งขึ้น แต่ก็จะต้องแก้ไขปัญหาในส่วนของการรับข้อมูลซ้ำของโบรกเกอร์ด้วย

2. การทดลองยังอยู่ในสถานการณ์ที่จำกัด จะทำอย่างไรให้วิธีการนี้สามารถนำไปประยุกต์ใช้ในระบบที่มีการเปลี่ยนแปลงของสภาพเครือข่ายและจำนวนอุปกรณ์ตลอดเวลา

3. การค้นหาอุปกรณ์รอบข้างขณะนี้ใช้เออาร์พีในการค้นหา IP address ทั้งหมดในเครือข่ายซึ่งทำให้ใช้เวลาค่อนข้างมากในการค้นหา การเปลี่ยนไปใช้ routing protocol เพื่อทำให้เกิดการสร้าง routing table ของเส้นทางการส่งข้อมูลระหว่างอุปกรณ์ อาจจะช่วยให้กระบวนการค้นหา IP address ของระบบดำเนินการได้รวดเร็วยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Internet Society. 2015. "**The Internet of Things: An Overview**". [White Papers]. Available: https://www.internetsociety.org/sites/default/files/ISOC-IoT-Overview-20151221-en.pdf. [Accessed 16 January 2017].

[2] OASIS. 2014. "**MQTT v3.1.1**". [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html. [Accessed 16 January 2017].

[3] Luzuriaga J.E., Perez M., Boronat P., Cano J.C., Calafate C. and Manzoni P. "A Comparative Evaluation of AMQP and MQTT Protocols over Unstable and Mobile Networks". **12th Consumer Communications and Networking Conference**. 2015.

[4] Aziz B. "A Formal Model and Analysis of the MQ Telemetry Transport Protocol". **9th International Conference on Availability, Reliability and Security**. 2014. pp.59-68.

[5] Hunkeler U., Truong H.L. and Standford-Clark A. "MQTT-S - a Publish/ Subscribe Protocol for Wireless Sensor Networks". **3rd Communication System Software and Middleware and Workshops**. 2008. pp.791-198

[6] คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์. "Architectures and Applications for Wireless Sensor Networks". [Online] . https://www.cpe.ku.ac.th/~cpj/204525/slides/03-Network.pptx. [Accessed 18 January 2017].

[7] Thangavel D., Ma X., Valera A. and Tan C.K.-Y. "Performance Evaluation of MQTT and CoAP via a Common Middleware". **9th International Conference on Intelligent Sensor, Sensor Networks and Information Processing**. 2014. pp.1-6.

[8] Lee S., Kim H. Hong D.K. and Ju H. "Correlation Analysis of MQTT Loss and Delay According to QoS Level". **2013 International Conference Information Networking**. 2013. pp.714-717.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[9] Davis E.G., Calveras A. and Demirkol I. "Improving Packet Delivery Performance of Publish/ Subscribe Protocols in Wireless Sensor Networks". 2013. pp.648-680.

[10] Jain K., Padhye J., Padmanabhan V.N. and Qiu L. "Impact of Interference on Multi-hop Wireless Network Performance". **The 9th Annual International Conference on Mobile Computing and Networking**. 2003.

[11] Gerla M., Tang K. and Bagrodia R. "TCP Performance in Wireless Multi-hop Networks". **Second IEEE Workshop on Mobile Computing Systems and Applications**. 1999.

[12] Collina Matteo., Corrazza G.E. and Vanelli-Coralli A. "Introduction the QEST Broker: Scaling the IoT Bridging MQTT and REST". **23rd Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications**. 2012.

[13] Grupta V., Goldman R. and Udupi P. "A Network Architecture for the Web of Things". **2nd International Workshop on Web of Things**. article. 3, 2011.

[14] Zhang J.C., Cheng C. and Ji Y. "Architecture Design for Social Web of Things". **1st International Workshop on Context Discovery and Data Mining**. article. 3, 2012.

[15] Mineraud J., Mazhelis O., Su X. and Tarkoma S. "A Gap Analysis of Internet-of-Things Platforms". Journal of Computer Communication. vol. 89, 2016. pp.5-16.

[16] ORACLE. 2017. "**Virtual networking**". [Online]. Available: https://www.virtualbox.org/manual/ch06.html. [Accessed 22 February 2017].

[17] Eclipse Foundation. 2017. "**Eclipse Paho**". [Online]. Available: https://projects.eclipse.org/projects/iot.paho. [Accessed 22 February 2017].

[18] Eclipse Foundation. 2017. "**Eclipse Mosquitto**". [Online]. Available: https://projects.eclipse.org/projects/technology.mosquitto. [Accessed 22 February 2017].

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[19] IBM. "Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry". September 2012. pp.7-8, 212-217.

[20] Dhar P. and Gupta P. "Intelligent Parking Cloud Services based on IoT using MQTT Protocol". **2016 International Conference on Automatic Control and Dynamic Optimization Techniques.** 2016. pp.30-34.

[21] Cisco. 2012. "**Quality of Service Networking**". [Online]. Available: http://docwiki.cisco.com/wiki/Quality_of_Service_Networking. [Accessed 13 April 2017].

[22] Whiteager J., Schneider F. and Teixeira Renata. "Explaining Packet Delays Under Virtualization". **ACM SIGCOMM Computer Communication Review.** vol. 41. issue 1. January 2011. p.38-44

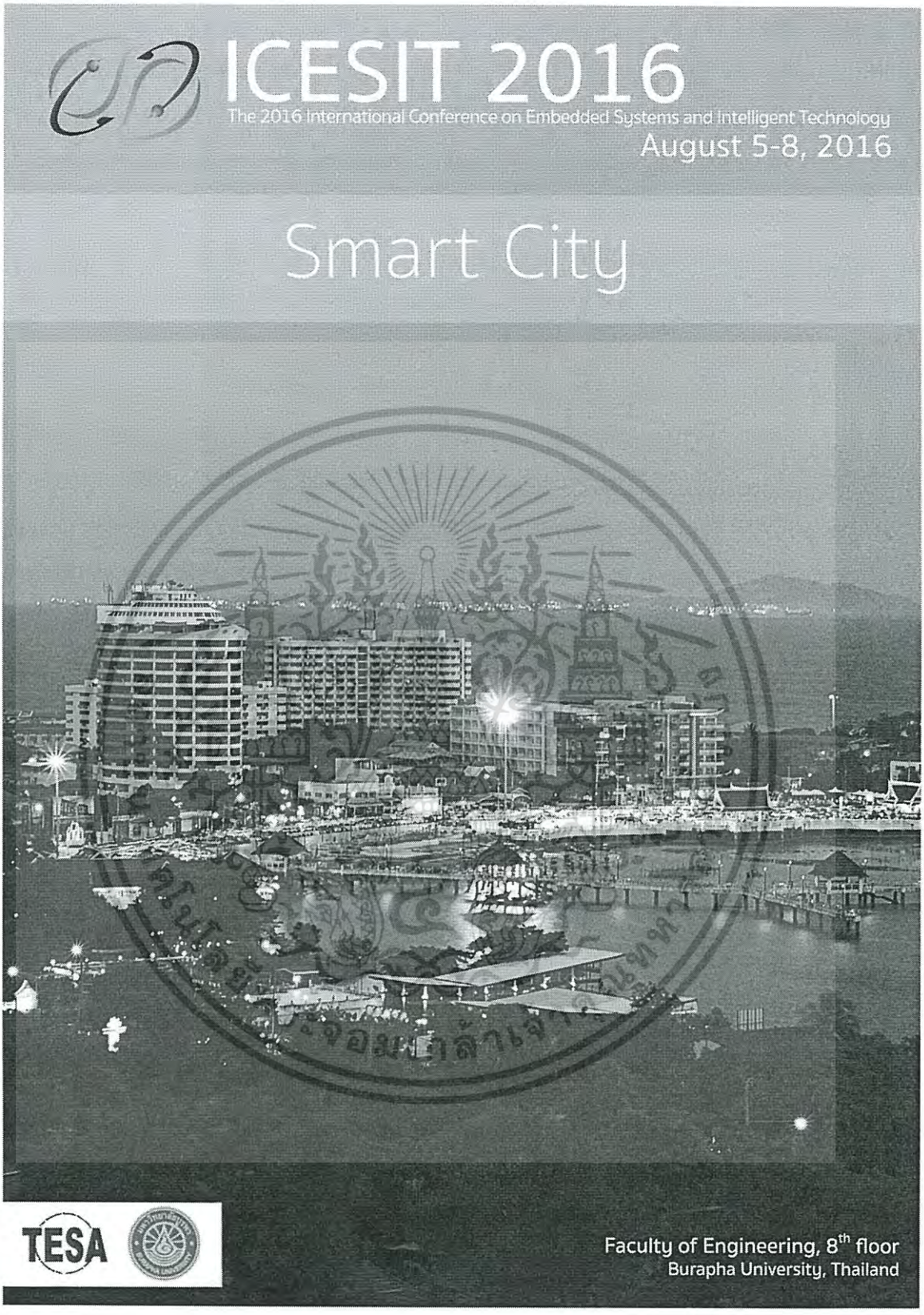
[23] NTA. 2011. "**Arp-scan User Guide**". [Online]. Available: http://www.nta-monitor.com/wiki/index.php/Arp-scan_User_Guide. [Accessed 18 April 2017].

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Relayed Message Sending for MQTT Protocol

Chontawee Wongpromrat
Department of Computer Engineering
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
58601091@kmitl.ac.th

Watchara Chatwiriya
Department of Computer Engineering
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
watchara.ch@kmitl.ac.th

***Abstract*— Nowadays, many protocols are used in Internet of Things (IoT) system. Message Queue Telemetry Transport (MQTT) is one of adopted the communication between devices and server in IoT ecosystem. To solve the "repeatedly sending the data to the broker" problem, MQTT use the Quality of Service (QoS). However, in some critical system, it is important to resend the data again without or with the shortest waiting time. This paper will propose concept and solution for solving this problem for removing or decrease the uncertain time that have to wait in QoS process.**

Keywords*— *IoT, MQTT, HTTP, ARP

I. Introduction

IoT is more practical in many application areas because an increasing number of smart devices and ease of Internet accessibility. Even though the concept of IoT is defined, but the communication protocols to be used with are still in the developing and competing to be standard [12].

One of the proposed protocols is MQTT, a lightweight protocol with many advantages [1]. MQTT use "publish-subscribe" pattern in which the server (This server is called a broker) uses with message intermediate [1,2]. In this pattern, a broker will receive the data from devices, and forward to connected clients, as listed in topic of devices. The broker will forward data to that client only if the client subscribes common topics.

In MQTT standard [1], QoS is defined as the service level agreed between sender and receiver for message sending, with the guarantees of delivering messages. Actually, the sender has to keep sending until it gets the response, to meet QoS conditions. With restrained bandwidth and noisy environment, the waiting time for this pattern is unpredictable and ineffective.

In this paper, we proposed a solution to solve this problem, assumed there is nearby communication devices or other communication channel. HTTP and ARP Protocol can apply by sending data to nearby devices and retransmit data to a broker instead of ordinary sender using.

The rest of the paper is organized as follows. Section 2 describes background knowledge. Section 3 discusses about the concept of the proposed solution. Section 4 explains about implementation. Section 5 describes the experiment. Lastly, section 6 concludes the work.

II. Background

IoT communication protocols are developed to meet various requirements such as support heterogeneous hardware, capable of data management, support application developers, extend the different platforms for the formation of ecosystems and the availability of dedicated marketplaces to the IoT [11].

One of the popular protocols is MQTT. The communication model in MQTT can be described as sending messages between brokers and senders, with QoS mechanism to prevent disruption in the transmission. The QoS is described as three levels as the following [13]:

- QoS level 0 sends the message only once following the message distribution flow, and does not check whether the message arrived to its destination. Therefore, in case of sizable messages. It is possible that the message will be lost when any kind of loss comes in the way.
- QoS Level 1 sends the message at least once and checks the delivery status of the message by using the status check message, PUBACK. However, when PUBACK is lost, it is possible that the server will send the same message twice, since it has no confirmation of the message being delivered.
- QoS level 2 passes the message through exactly once utilizing the 4-way handshake. It is not possible to have a message loss in this level, but due to the complicated process of 4-way handshake, it is possible to have relatively longer end-to-end delays.

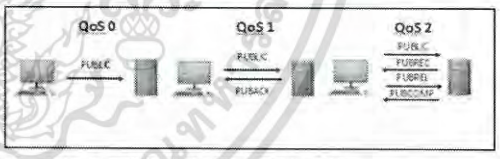


Fig. 1. Packet transmission method about QoS level [13]

We note that the QoS process needs to send the message repeated in some situation, if the connection is unstable, it will difficult to predict the communication time and can will consider as a serious factor in critical systems.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Moreover, the system cannot use only MQTT protocol to solve this problem because the limitation of the protocol. So we have to use another protocol for sending data to nearby devices. We will use response time to measure performance as suggested in [9].

III. Concept

In the previous section, we describe a basic QoS process in MQTT protocol. In terms of performance, there is a research [13] which analysis about a delay and message loss problem in the QoS process of MQTT protocol but proposed no solution.

Assume that there are other communication channels beside the problematic channel, we can propose the solution with the following scenario:

From Fig. 2, if Sender 1 cannot send message to the broker because of the connection problem, it usually uses a QoS process for sending messages again and again. If the waiting time is more than the threshold, Sender 1 will forward messages to other sender via the alternative channel, assumed with more reliable communication with the client, can successfully transmit the message.

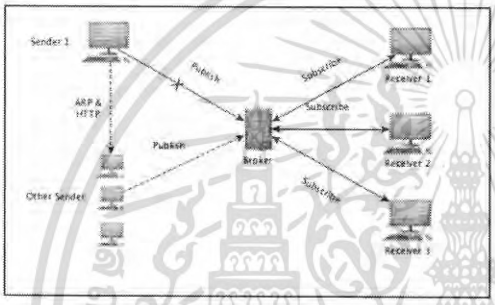


Fig. 2. Proposed concept of solution

However, in this scenario, we need to find the nearby devices. Then the Address Translation Protocol or ARP protocol can be useful [3,4].

In order to forward messages to nearby devices. we apply HTTP protocol, because it uses a lot in present day. Moreover, the HTTP can embed in embedded devices. This approach can be called Web of Things or WoT [5,6].

The modified scenario can be described as the following steps (as Fig. 3 and Fig. 4):

- Primary sender (we call the ordinary sender as primary sender) will send messages to the broker. If it cannot do this process, it will find nearby devices and keep device addresses to list. On the other hand, primary sender will finish the process.
- Primary sender will forward messages to each nearby device from address list respectively.
- When nearby devices receive HTTP messages, it will forward the message to the broker and wait for the result until time is out. But if it cannot receive forwarded the message, primary sender will skip this nearby device and forward the message to the others.
- If the nearby device can forward the message, it will return success result to the primary sender before primary sender will finish the process. On the other hand, primary sender will forward the message to other nearby devices.

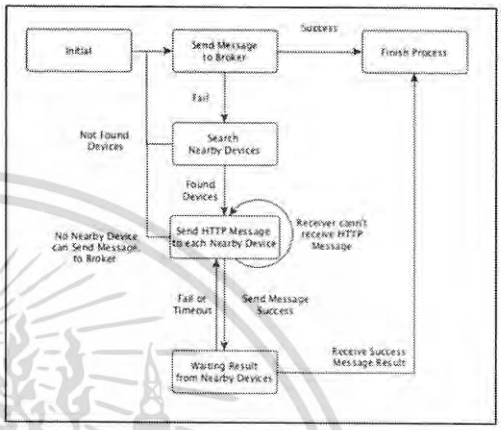


Fig. 3. Proposed state diagram of primary sender

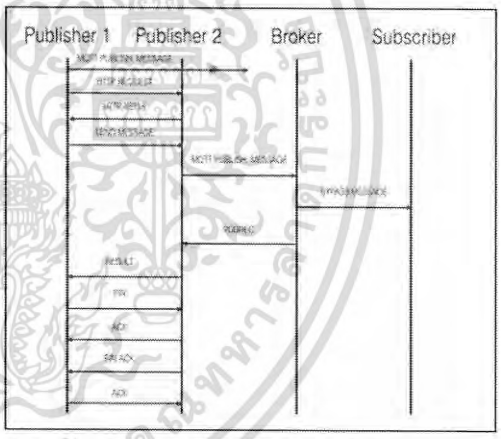


Fig. 4. Proposed sequence diagram of solution mechanism

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Moreover, the system cannot use only MQTT protocol to solve this problem because the limitation of the protocol. So we have to use another protocol for sending data to nearby devices. We will use response time to measure performance as suggested in [9].

III. Concept

In the previous section, we describe a basic QoS process in MQTT protocol. In terms of performance, there is a research [13] which analysis about a delay and message loss problem in the QoS process of MQTT protocol but proposed no solution.

Assume that there are other communication channels beside the problematic channel, we can propose the solution with the following scenario:

From Fig. 2, if Sender 1 cannot send message to the broker because of the connection problem, it usually uses a QoS process for sending messages again and again. If the waiting time is more than the threshold, Sender 1 will forward messages to other sender via the alternative channel, assumed with more reliable communication with the client, can successfully transmit the message.

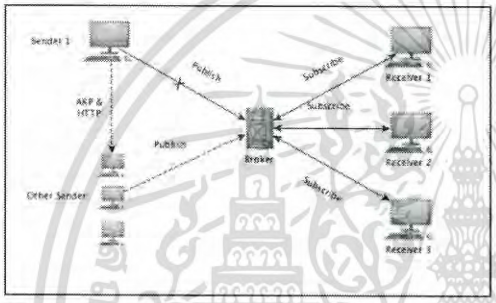


Fig. 2. Proposed concept of solution

However, in this scenario, we need to find the nearby devices. Then the Address Translation Protocol or ARP protocol can be useful [3,4].

In order to forward messages to nearby devices. we apply HTTP protocol, because it uses a lot in present day. Moreover, the HTTP can embed in embedded devices. This approach can be called Web of Things or WoT [5,6].

The modified scenario can be described as the following steps (as Fig. 3 and Fig. 4):

- Primary sender (we call the ordinary sender as primary sender) will send messages to the broker. If it cannot do this process, it will find nearby devices and keep device addresses to list. On the other hand, primary sender will finish the process.
- Primary sender will forward messages to each nearby device from address list respectively.
- When nearby devices receive HTTP messages, it will forward the message to the broker and wait for the result until time is out. But if it cannot receive forwarded the message, primary sender will skip this nearby device and forward the message to the others.
- If the nearby device can forward the message, it will return success result to the primary sender before primary sender will finish the process. On the other hand, primary sender will forward the message to other nearby devices.

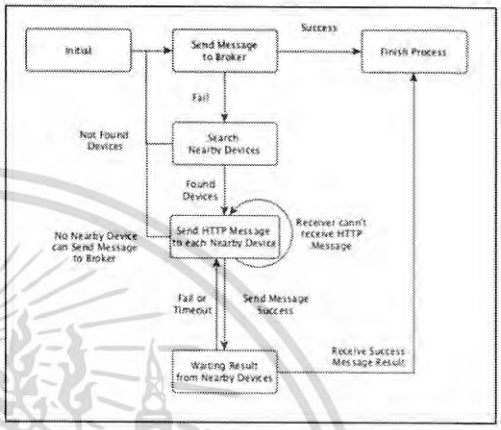


Fig. 3. Proposed state diagram of primary sender

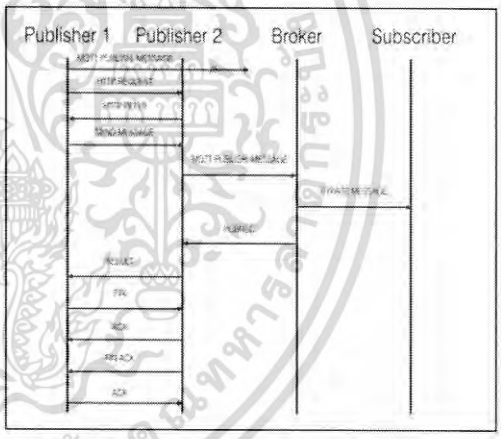


Fig. 4. Proposed sequence diagram of solution mechanism

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IV. Implementation

In this section, we illustrate the network diagram of system implementation as Fig. 5. If one of the network is not working, primary sender can transmit messages to nearby devices which will forward the messages to the broker server.

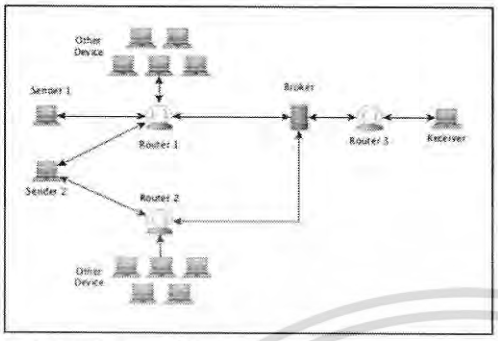


Fig. 5. Network diagram of this experiment

In the first process, the primary sender tries to send message to the broker. If it cannot send this message, it will find nearby devices for forwarding the message (pseudocode shows in Fig. 6).

```
# First Process – Try to send message to the broker
result = mqtt.send(server.ip, topic, data)
if result == 'Success':
    stopAllProcess()
else:
    findNearbyDevice()
```

Fig. 6. Show pseudocode about condition of MQTT sending process

In finding nearby devices process, primary sender will use the ARP protocol for asking IP Address of nearby devices and store it to array list (pseudocode shows in Fig. 7).

```
# Second Process – Get IP of nearby devices from ARP
allNearbyDeviceIp = ARP()
```

Fig. 7. Showpseudocode about store IP Address to array list

In forwarding message process, primary sender will forward data to each nearby device using the HTTP protocol and wait for receiving a result from that nearby device. If that nearby device can forward this message, primary sender will finish all processes. On the other hand, primary sender will forward message to the others. But if there are no nearby devices can forward the message, primary sender will back to the first process again (pseudocode shows in Fig. 8).

```
# Third Process – Forward message to each nearby device and wait for the result
for eachNearbyDeviceIp in allNearbyDeviceIp:
    result = http.send(NearbyDeviceIp[i], topic, data)
    if result == 'Success':
        stopAllProcess()
backToFirstProcess()
```

Fig. 8. Show pseudocode about HTTP transmitting process

When a nearby device receives an HTTP message from primary sender, it will forward that message to a broker server. A result from this process will return to primary sender (pseudocode shows in Fig. 9).

```
# Nearby Device Process – Receive HTTP message and forward it to the broker
message = http.receive();
topic = message[0]; # Receive topic of MQTT message
data = message[1]; # Receive data of MQTT message
result = mqtt.send(server.ip, topic, data);
if result == 'Success':
    returnSuccessResultToPrimarySender()
else:
    returnFailResultToPrimarySender()
```

Fig. 9. Show pseudocode about data retransmitting process of nearby devices

V. Experiment

In this section, we show the experimental setup and the result. By we set up the system as the following:

- Open sources which we choose in this experiment are Mosquitto as a broker server and Paho as clients.
- There are 1 broker, 1 receiver, 1 primary sender, only 1 nearby device which can forward messages to the broker and 10 other nearby devices in the network.
- The network topology is same as Fig. 5.
- Size of payload is 1,000 bytes. (It is minimal payload in the experiment of S. Lee, H. Kim, D.K. Hong and H. Ju's research [13])
- For HTTP connection, we set the threshold time to 10 seconds. (Recommended default timeout in experiment of [14])

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- For QoS 1, QoS 2 levels and the proposed solution, we repeat the measurement for 10 times.

We measured transmission time by creating network congestion between primary sender and broker for 10, 20 and 30 seconds. We have compared proposed solution with transmission time obtained by each QoS level. We found that transmission time from proposed solution is less than every QoS level when delay is longer than 10 seconds (as shown in Fig. 10, 11 and 12).

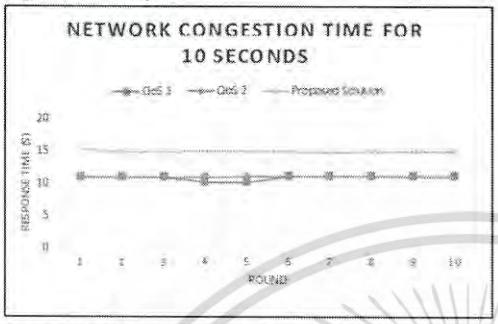


Fig. 10. Result when network congestion has occurred for 10 seconds

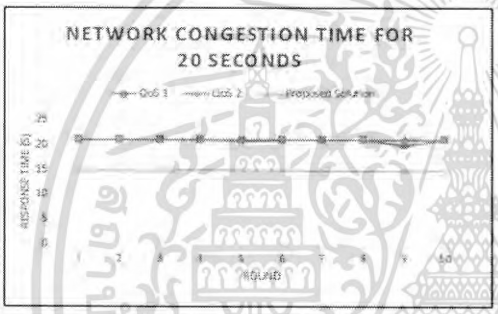


Fig. 11. Result when when network congestion has occurred for 20 seconds

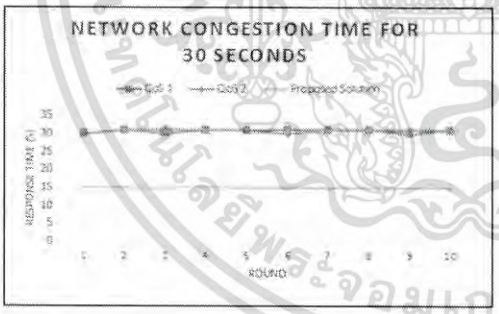


Fig. 12. Result when when network congestion has occurred for 30 seconds

Table 1 summarizes the average transmission time and quantity of forwarded devices. Analysis results in our experiments show that transmission time and quantity of forward devices has a strong positive relationship for the proposed solution.

TABLE I. SHOW THE AVERAGE RESULT OF EXPERIMENT

Network congestion time (second)	Average response time (second)			Average quantity of forwarded devices (unit)
	QoS 1	*QoS 2*	*Proposed Solution*	
10	10.81	11.01	14.88	7.2
20	20.91	21.01	14.83	7.0
30	30.91	30.81	14.93	7.4

In the result, the proposed solution has stable response time than the QoS mechanism because it does not use congestion channel for sending messages. But it is necessary to use more process than QoS mechanism (find nearby devices by ARP, forward message to each nearby device by HTTP and wait for the result from each nearby device). So these processes affect response time in the proposed solution, whereas response time in QoS mechanism depends on network congestion time.

VI. CONCLUSION

In this paper, we propose the solution for solving delay and message loss problem under QoS process in MQTT protocol. Experimental results suggest that this solution can reduce transmission time when the system has a long delay. This transmission time is significantly associated with quantity of forwarded devices. If we allow nearby devices to forward message, we will be able to build a more effective system. Regarding future studies, we will experiment under various priorities of message for performing the experiment in a more realistic scenario. Also, we will analyze launch time of new added message under different payload.

REFERENCES

[1] Advancing open standard for the information society, https://www.oasisopen.org/

[2] U. Hunkeler, H.L. Truong and A. Standford-Clark, "MQTT-S - A publish/subscribe protocol for Wireless Sensor Networks," 3rd Communication Systems Software and Middleware and Workshops, Bangalore, IEEE, 2008, pp. 791-198.

[3] RFC Editor, https://www.rfc-editor.org/

[4] Y. Qin, Y. Zhou, and B. Liu, "A High Performance ARP Lookup System for Gigabit Ethernet," 2nd Communications and Networking in China, Shanghai, IEEE, 2007 pp. 82-86.

[5] V. Gupta, R. Goldman, and P. Udupi, "A Network Architecture for the Web of Things," 2nd International Workshop on Web of Things, San Francisco, ACM, 2011, article. 3.

[6]] C. Zhang, C. Cheng, and Y. Ji, "Architecture Design for Social Web of Things," 1st International Workshop on Context Discovery and Data Mining, Beijing, ACM, 2012 article. 3.

[7] B. Aziz, "A Formal Model and Analysis of the MQ Telemetry Transport Protocol," 9th International Conference on Availability, Reliability and Security, Fribourg, IEEE, 2014 pp. 59-68.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[8] A.J. Coddington, "Message Queuing for Network Monitoring," Technical Report, The University of Waikato, 2012

[9] E.G Davis, A. Calveras, and I. Demirkol, "Improving Packet Delivery Performance of Publish/Subscribe Protocols in Wireless Sensor Networks," Barcelona, Sensor, 2013, pp. 648-680.

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

[11] J. Mineraud, O. Mazhelis, X. Su, ans S. Tarkoma, "A gap analysis of Internet-of-Things platforms," Preprint, arXiv, 2015

[12] A.P. Castellani, M. Bui, P. Casari, M. Rossi, Z. Shelby and M. Zorzi, "Architecture and protocols for the Internet of Things: A case study," IEEE International Conference on Pervasive Computing and Communication, Mannheim, IEEE, 2010, pp. 678-683

[13] S. Lee, H. Kim, D.K. Hong and H. Ju, "Correlation Analysis of MQTT Loss and Delay According to QoS Level," 2013 International Conference Information Networking, Bangkok, IEEE, 2013, pp. 714-717

[14] B. Liu, G. Abdulla, T. Johnson and E.A. Fox, "Web Response Time and Proxy Caching," Technical Report, Virginia Polytechnic and State University, 200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Proceeding of 2016 International Computer Science and Engineering Conference (ICSEC 2016)

การส่งต่อข้อมูลแบบมัลติ-ฮ็อปสำหรับเอ็มคิวทีทีโพรโทคอล
Multi-Hop Relayed Message Sending for MQTT Protocol

ชลทวี วงศ์พร้อมรัตน์[1], วัชระ ฉัตรวิริยะ[2]
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพฯ
E-mail: 58601091@kmitl.ac.th[1], watchara.ch@kmitl.ac.th[2]

บทคัดย่อ

เอ็มคิวทีทีโพรโทคอล (MQTT Protocol) เป็นหนึ่งในโพรโทคอลที่ถูกนำมาใช้งานอย่างกว้างขวางในระบบอินเทอร์เน็ตออฟธิงส์ (Internet of Things) หรือไอโอที (IoT) จากงานวิจัยของเราที่ผ่านมาพบว่าเอ็มคิวทีทีนั้นมีปัญหาในกระบวนการของการการันตีการให้บริการ (Quality of Services) หรือคิวโอเอส (QoS) เราได้นำเสนอวิธีการส่งข้อมูลไปยังอุปกรณ์ข้างเคียงเพื่อลดปัญหาการเชื่อมต่อที่ขาดหายไปเป็นระยะเวลายาวนาน โดยงานวิจัยนี้จะทำการพัฒนาแนวคิดดังกล่าวให้มีประสิทธิภาพในการส่งข้อมูลไปยังอุปกรณ์ข้างเคียงได้ดียิ่งขึ้น โดยการเปลี่ยนรูปแบบการส่งข้อมูลแบบซิงเกิล-ฮ็อป (Single-Hop) เป็นแบบมัลติ-ฮ็อป (Multi-Hop) แทน เพื่อส่งข้อมูลไปยังอุปกรณ์ที่อยู่ในเครือข่ายอื่นที่การส่งข้อมูลแบบซิงเกิล-ฮ็อปไม่สามารถส่งได้ โดยระยะเวลาที่ใช้จะน้อยกว่ากระบวนการคิวโอเอสหากปัญหาการเชื่อมต่อมีระยะเวลาที่ยาวนาน

คำสำคัญ: มัลติ-ฮ็อป, เครือข่ายคอมพิวเตอร์, ไอโอที, เอ็มคิวทีที

Abstract

MQTT is one of many protocol, which be used in Internet of Things system. From our previous research, there is problem about Quality of Service mechanism in this protocol. We purposed solution by using nearby devices for decreasing response time in network problem. For this research, we improve our solution in this protocol for increasing send message chance by changing single-hop communication pattern to multi-hop for sending data to devices in other network which single-hop pattern cannot send data to them. This solution uses less time than processing time in QoS mechanism when the connection problem has a long time.

Keywords: Multi-Hop, Computer Network, IoT, MQTT

1. บทนำ

อินเทอร์เน็ตออฟธิงส์เป็นระบบที่ทำงานกับอุปกรณ์จำนวนมากเพื่อรับส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตโดยอาศัยโพรโทคอลที่มีอยู่มากมายในปัจจุบัน หนึ่งในโพรโทคอลที่ได้รับความนิยมในระบบไอโอทีคือเอ็มคิวทีทีเนื่องจากเป็นโพรโทคอลที่มีขนาดเล็ก จึงเหมาะสมกับระบบไอโอทีที่อุปกรณ์มีประสิทธิภาพในการประมวลผลไม่มาก และด้วยขนาดที่เล็กจึงทำให้สามารถลดขนาดแบนด์วิดท์ (Bandwidth) ในการรับส่งแต่ละครั้งได้

หนึ่งในปัญหาของเอ็มคิวทีทีคือกระบวนการคิวโอเอสจะมีระยะเวลาการส่งซ้ำที่ไม่แน่นอน หากการเชื่อมต่อไปยังโบรกเกอร์ (Broker Server) เกิดปัญหาเป็นระยะเวลานาน ระยะเวลาของการส่งข้อมูลด้วยกระบวนการคิวโอเอสก็จะยาวนานตามไปด้วย เนื่องจากกระบวนการนี้จำเป็นที่จะต้องส่งข้อมูลซ้ำ จากงานวิจัยที่ผ่านมาของเรา เราได้นำเสนอวิธีการแก้ปัญหาโดยการส่งข้อมูลไปยังอุปกรณ์รอบข้าง เพื่อให้อุปกรณ์รอบข้างเหล่านั้นทำหน้าที่ส่งข้อมูลแทน

ในการส่งข้อมูล อุปกรณ์ดั้งต้นจะส่งข้อมูลไปยังอุปกรณ์รอบข้างแบบซิงเกิล-ฮ็อปเท่านั้น ซึ่งจะมีผลให้หากอุปกรณ์รอบข้างเหล่านั้นยังมีอุปกรณ์รอบข้างหรือเครือข่ายอื่นที่เชื่อมต่ออยู่อีก มันจะไม่สามารถส่งต่อข้อมูลเพิ่มเติมอีกได้ ดังนั้นในงานวิจัยนี้ เราจะทำการเพิ่มประสิทธิภาพของการส่งข้อมูลไปยังเครือข่ายของอุปกรณ์ข้างเคียง เพื่อเพิ่มโอกาสในการส่งข้อมูลไปยังอุปกรณ์อื่นที่อุปกรณ์ส่งดั้งต้นไม่สามารถเชื่อมต่อได้

ในงานวิจัยชิ้นนี้จะถูกกล่าวตามลำดับต่อไปนี้ บทที่ 2 อธิบายถึงความรู้พื้นฐานและงานวิจัยที่เกี่ยวข้อง บทที่ 3 กล่าวถึงแนวคิดของงานวิจัย บทที่ 4 อธิบายแนวทางการพัฒนาระบบ บทที่ 5 กล่าวถึงการทดลองและผลการทดลอง และบทที่ 6 สรุปผล

2. งานวิจัยที่เกี่ยวข้อง

การรับส่งข้อมูลด้วยเอ็มคิวทีทีโพรโทคอลถูกแบ่งออกเป็น 3 ส่วนหลักคือ ตัวส่ง (Publisher) โบรกเกอร์ (Broker) และตัวรับ (Receiver) เมื่อทำการส่งข้อมูล อุปกรณ์ที่ส่งข้อมูลต้นทางจะส่งข้อความ (Message) พร้อมระบุหัวข้อ (Topic) ของข้อความนั้นแล้วส่งไปยังโบรกเกอร์ เมื่อโบรกเกอร์ได้รับข้อความ มันจะทำการส่งต่อข้อมูลไปยังอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับปลายทางทุกตัวที่เชื่อมต่อกับมันโดยอ้างอิงจากหัวข้อของข้อความ หากเป็นหัวข้อเดียวกัน อุปกรณ์รับปลายทางก็จะได้รับข้อความจากโบรกเกอร์ แต่ในทางกลับกัน ก็จะไม่ได้รับข้อความ [4]

หนึ่งในกระบวนการหลักที่สำคัญของเอ็มคิวทีทีคือคิวโอเอสซึ่งเป็นกระบวนการที่ช่วยในเรื่องการรับประกันการรับส่งข้อมูล โดยกระบวนการนี้ถูกแบ่งออกเป็น 3 ระดับ [1-2] ได้แก่ 1. คิวโอเอสศูนย์ (QoS0) เป็นการส่งข้อมูลจากอุปกรณ์ส่งต้นทางเพียงครั้งเดียว และไม่รอผลลัพธ์กลับ ดังนั้นหากข้อมูลมีการสูญหายก็จะไม่มีการส่งข้อมูลซ้ำเกิดขึ้น 2. คิวโอเอสหนึ่ง (QoS1) เป็นการส่งข้อมูลที่รอผลตอบรับกลับจากโบรกเกอร์หนึ่งครั้ง ซึ่งเป็นการช่วยการันตีการส่งข้อมูล แต่ถ้าหากข้อความที่ตอบกลับจากโบรกเกอร์เกิดการสูญหาย ก็มีโอกาสที่อุปกรณ์ตั้งต้นจะส่งข้อมูลซ้ำได้ 3. คิวโอเอสสอง (QoS2) เป็นการรับส่งข้อมูลที่ปลอดภัยที่สุดสำหรับเอ็มคิวทีทีโพรโทคอล โดยอาศัยหลักการ 4-เวย์แฮนด์เชค (4-way handshake) แต่ก็มีโอกาสที่การส่งข้อมูลจะช้าเมื่อเทียบกับระดับอื่น

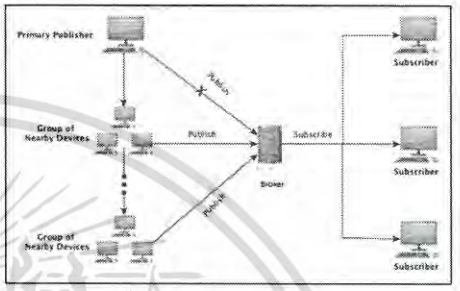
หนึ่งในวิธีการที่ใช้สำหรับการแก้ปัญหาการส่งข้อมูลล่าช้าที่เกิดจากปัญหาการเชื่อมต่อในคิวโอเอส [1] คือ การส่งข้อมูลออกไปยังอุปกรณ์รอบข้าง เพื่อให้อุปกรณ์รอบข้างทดลองส่งข้อมูลไปยังโบรกเกอร์แทน โดยอาศัยอาร์พโพรโทคอล (ARP) และ เอชทีทีพีโพรโทคอล (HTTP) ช่วยในการสื่อสารไปยังอุปกรณ์รอบข้าง ซึ่งจากงานวิจัยที่ผ่านมา การส่งข้อมูลจะใช้รูปแบบซิงเกิล-ฮ็อป กล่าวคือ อุปกรณ์ตั้งต้นจะส่งข้อมูลไปยังอุปกรณ์รอบข้างเพียงหนึ่งครั้งเท่านั้น และอุปกรณ์รอบข้างจะไม่สามารถส่งข้อมูลไปยังอุปกรณ์ตัวอื่นเพื่อให้ส่งข้อมูลแทนได้อีก ซึ่งถ้ามีอุปกรณ์อื่นที่เชื่อมต่ออยู่กับอุปกรณ์รอบข้าง แต่ไม่ได้เชื่อมต่อกับอุปกรณ์ตั้งต้น อุปกรณ์เหล่านั้นจะไม่ได้รับข้อมูลเพื่อทดลองส่งข้อมูลไปยังโบรกเกอร์ ทำให้โอกาสที่จะส่งต่อข้อมูลไปยังโบรกเกอร์ลดลง

งานวิจัยชิ้นนี้จึงมุ่งเน้นไปที่การเพิ่มประสิทธิภาพในการส่งข้อมูลแบบมัลติ-ฮ็อปเพื่อเพิ่มโอกาสในการส่งข้อมูลไปยังเครือข่ายและอุปกรณ์รอบข้างอื่นที่อุปกรณ์ตั้งต้นไม่สามารถเชื่อมต่อได้

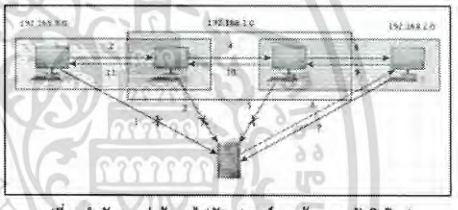
3. แนวคิด

ในบทนี้จะเป็นการนำเสนอแนวคิดในการส่งข้อมูลแบบมัลติ-ฮ็อปไปยังอุปกรณ์ข้างเคียง หนึ่งในปัญหาที่เราพบคือ จะทำอย่างไรให้การส่งข้อมูลแบบมัลติ-ฮ็อปนี้ ไม่ส่งข้อมูลซ้ำไปยังอุปกรณ์กลุ่มเดิม เนื่องจากการส่งข้อมูลซ้ำไปยังอุปกรณ์เดิมที่ไม่สามารถส่งข้อมูลแทนได้นั้น มีผลทำให้ระยะเวลาในการส่งข้อมูลมากขึ้น

หนึ่งในวิธีการที่สามารถใช้แก้ปัญหานี้ได้คือ ใช้วิธีการตรวจสอบก่อนการส่งข้อมูลว่า อุปกรณ์ที่กำลังจะส่งไปนั้น เคยถูกส่งข้อมูลให้แล้วหรือยัง โดยตรวจสอบจากหมายเลขเครือข่าย (Network IP) โดยทุกครั้งที่มีการส่งข้อมูลไปยังอุปกรณ์รอบข้าง อุปกรณ์ตั้งต้นจะทำการส่งข้อมูลของหมายเลขเครือข่ายที่เคยทดลองส่งข้อมูลแล้วไปพร้อมกัน เพื่อให้อุปกรณ์ภายในเครือข่ายนั้นทำการตรวจสอบก่อนการส่งต่อไปยังอุปกรณ์ในเครือข่ายอื่น หากอุปกรณ์ที่จะส่งไปนั้นอยู่ในเครือข่ายที่อยู่ในรายการหมายเลขเครือข่ายที่เคยถูกส่งแล้ว มันก็จะข้ามการส่งข้อมูลไปยังอุปกรณ์นั้นไป และหากอุปกรณ์ทุกตัวภายในเครือข่าย ไม่สามารถส่งข้อมูลไปยังโบรกเกอร์ได้เลย อุปกรณ์แรกที่รับข้อมูลเข้ามาภายในเครือข่ายนั้น ก็จะทำการส่งหมายเลขเครือข่ายกลับไป เพื่อเป็นการบ่งบอกว่า เครือข่ายนี้ไม่สามารถส่งข้อมูลแทนได้ แต่ในทางกลับกัน หากมีอุปกรณ์ที่สามารถส่งข้อมูลแทนได้ อุปกรณ์แรกที่รับข้อมูลเข้ามาภายในเครือข่ายนั้นก็จะทำการส่งผลลัพธ์ว่าสำเร็จกลับไปยังอุปกรณ์ส่งตั้งต้นแทน



รูปที่ 1 แนวคิดของระบบการส่งต่อข้อมูล



รูปที่ 2 ลำดับการส่งข้อมูลไปยังอุปกรณ์รอบข้างแบบมัลติ-ฮ็อป

วิธีการที่นำเสนอจะเป็นไปตามขั้นตอนดังนี้

1.อุปกรณ์ส่งตั้งต้นส่งข้อมูลไปยังโบรกเกอร์ หากไม่สามารถส่งข้อมูลได้อุปกรณ์ส่งตั้งต้นจะทำการค้นหาอุปกรณ์รอบข้างด้วยอาร์พโพรโทคอล

2.อุปกรณ์ส่งตั้งต้นจะส่งข้อมูลไปยังอุปกรณ์รอบข้างผ่านเอชทีทีพีโพรโทคอลทีละอุปกรณ์ตามหมายเลขอุปกรณ์ที่ได้จากขั้นตอนที่ 1

3.อุปกรณ์รอบข้างจะส่งข้อมูลไปยังโบรกเกอร์แทนอุปกรณ์ส่งตั้งต้น หากส่งสำเร็จ มันจะทำการส่งผลลัพธ์ว่าสำเร็จกลับไปยังอุปกรณ์ส่งตั้งต้นและจบกระบวนการลง แต่หากไม่สำเร็จ มันจะทำการค้นหาอุปกรณ์รอบข้าง พร้อมส่งข้อมูลรวมถึงหมายเลขเครือข่ายไปยังอุปกรณ์รอบข้างที่ยังไม่เคยทดลองส่งข้อมูล

4.อุปกรณ์ที่รับข้อมูลจะทำกระบวนการในข้อที่ 3 ซ้ำไปเรื่อยๆ จนกว่าจะไม่มีอุปกรณ์ใหม่ในเครือข่ายที่มันสามารถเชื่อมต่อได้ให้ทดลองส่งอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

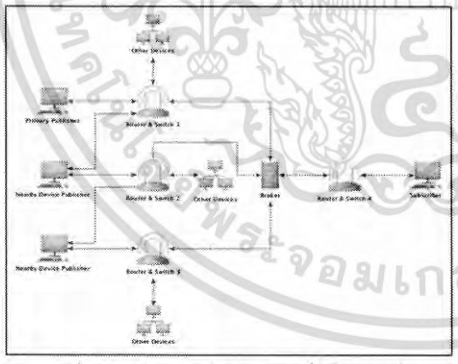
5.อุปกรณ์รอบข้างตัวสุดท้ายที่ทดลองส่งข้อมูลจะทำการส่งผลลัพธ์กลับไปยังอุปกรณ์ที่ส่งข้อมูลให้ ถ้าหากส่งสำเร็จ มันจะทำการส่งผลลัพธ์ว่าสำเร็จไปเรื่อยๆจนกลับไปถึงอุปกรณ์ส่งต้น แต่ถ้าส่งไม่สำเร็จ มันจะทำการส่งหมายเลขเครือข่ายกลับไปแทน

6.หากไม่มีอุปกรณ์ใดเลยสามารถส่งข้อมูลแทนได้อุปกรณ์ส่งต้นจะเริ่มทำกระบวนการทั้งหมดซ้ำอีกครั้ง

รูปที่ 3 แผนภาพสถานะ (State Diagram) ของวิธีการที่นำเสนอ

4. แนวทางการพัฒนาระบบ

เราได้ทำการจำลองระบบขึ้นมาโดยมีระบบเครือข่าย 3 ระบบเชื่อมต่อกันตามรูปที่ 4 โดยทุกครั้งที่อุปกรณ์ใดก็ตามไม่สามารถส่งข้อมูลไปยังโบรกเกอร์ด้วยตัวเองได้ อุปกรณ์นั้นก็จะทำตามกระบวนการที่กล่าวไว้ในบทก่อนหน้า



รูปที่ 4 โครงสร้างระบบเครือข่ายและการเชื่อมต่ออุปกรณ์

รูปที่ 5 เป็นซูโดโค้ด (Pseudocode) สำหรับการส่งและรับข้อมูลไปยังอุปกรณ์รอบข้างด้วยเอชทีทีพีโพรโทคอลซึ่งการส่งนี้จะมีลักษณะเป็นแบบฟังก์ชันรีเคอร์ซีฟ (Recursive Function) ทำให้สามารถส่งข้อมูลไปยังหางอุปกรณ์ข้างเคียงได้เรื่อยๆ โดยมีการตรวจสอบว่าอุปกรณ์ปลายทางที่จะส่งไปนั้น อยู่ในเครือข่ายที่เคยทดลองส่งไปแล้วหรือไม่ ซึ่งหากไม่มีอุปกรณ์ใดเลยในเครือข่ายสามารถส่งข้อมูลได้ อุปกรณ์ตั้งต้นของเครือข่ายก็จะส่งหมายเลขเครือข่ายทั้งหมดที่ทดลองส่งกลับไปยังอุปกรณ์ส่งก่อนหน้าเพื่อไม่ให้เกิดการส่งซ้ำของข้อมูลภายในเครือข่าย

```
def mqtt_publish(broker_ip, topic, data, network_ip):
  publish_result = mqtt.publish(broker_ip, topic, data)
    if publish_result == 'Success':
      return 'Success'
    else:
      nearby_device_ip = findNearbyDevice()
      network_ip.add(local_network_ip)
      for each_ip in nearby_device_ip:
        nearby_network_ip = getNetworkIp(each_ip)
          if nearby_network_ip not in network_ip:
            nearby_result = http.send(each_ip, topic, data,
network_ip)
            if nearby_result == 'Success':
              return 'Success'
      return network_ip

def http_receive(topic, data, network_ip):
  message = http.receive(topic, data, network_ip):
  topic = message[0]
  data = message[1]
  sent_network_ip = message[2]
  send_result = mqtt_publish(broker_ip, topic, data,
network_ip)
  return send_result
```

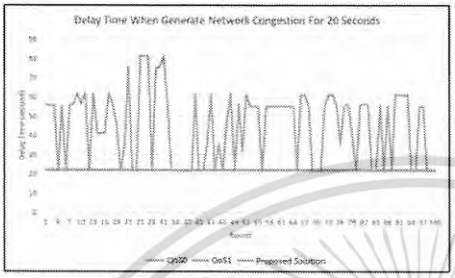
รูปที่ 5 ซูโดโค้ดการรับส่งข้อมูลแบบมัลติ-ฮ็อป

5. การทดลอง

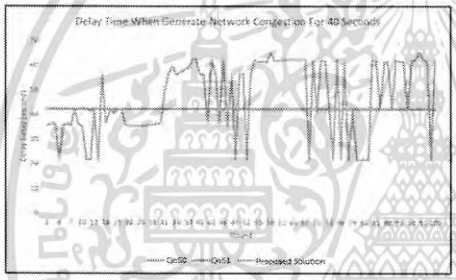
วิธีการทดสอบวิธีการแก้ไขที่เรานำเสนอนั้น เราได้ทำการจำลองระบบตามรูปที่ 4 แล้วทำการสร้างความแออัดภายในเครือข่าย (Network Congestion) เป็นเวลา 20 40 และ 60 วินาที เพื่อเก็บผลลัพธ์ความล่าช้า (Delay Time) (จับเวลาการส่งข้อมูลจากอุปกรณ์ส่งต้นทางจนถึงอุปกรณ์รับปลายทาง) และความเร็วในการตอบสนอง (Response

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

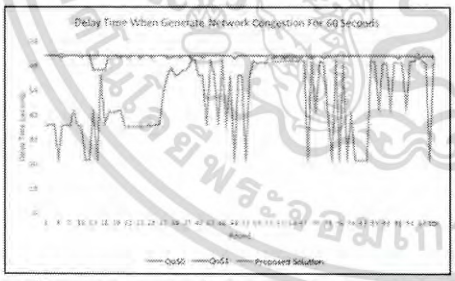
Time) (จับเวลาตั้งแต่การส่งข้อมูลจากอุปกรณ์ส่งต้นทางจนได้รับผลลัพธ์ตอบกลับ) ในการส่งข้อมูลของกระบวนการคิวโอเอสหนึ่ง คิวโอเอสสอง และวิธีการที่นำเสนอ โดยการส่งข้อมูลขนาด 1,000 ไบท์ ไปยังอุปกรณ์รับปลายทาง รูปที่ 6 7 8 เป็นผลลัพธ์ความล่าช้าของการส่งข้อมูล และ รูปที่ 9 10 11 เป็นความเร็วในการตอบสนอง



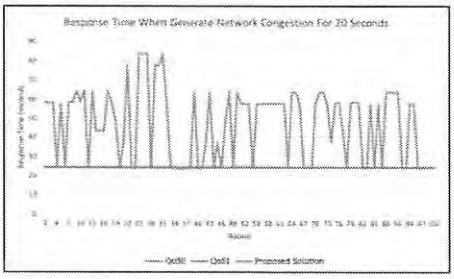
รูปที่ 6 ความล่าช้าของผลลัพธ์เมื่อสร้างความแออัดภายในเครือข่ายเป็นเวลา 20 วินาที



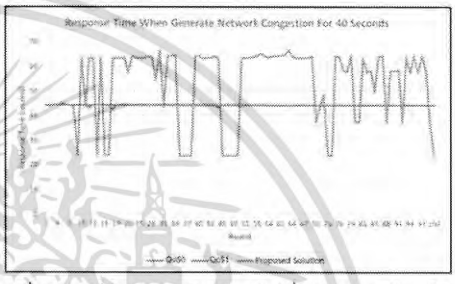
รูปที่ 7 ความล่าช้าของผลลัพธ์เมื่อสร้างความแออัดภายในเครือข่ายเป็นเวลา 40 วินาที



รูปที่ 8 ความล่าช้าของผลลัพธ์เมื่อสร้างความแออัดภายในเครือข่ายเป็นเวลา 60 วินาที



รูปที่ 9 ความเร็วในการตอบสนองของผลลัพธ์เมื่อสร้างความแออัดภายในเครือข่ายเป็นเวลา 20 วินาที



รูปที่ 10 ความเร็วในการตอบสนองของผลลัพธ์เมื่อสร้างความแออัดภายในเครือข่ายเป็นเวลา 40 วินาที

รูปที่ 11 ความเร็วในการตอบสนองของผลลัพธ์เมื่อสร้างความแออัดภายในเครือข่ายเป็นเวลา 60 วินาที

ตารางที่ 1 และ 2 เป็นตารางที่แสดงถึงค่าเฉลี่ยความเร็วในการตอบสนองและความล่าช้าของการผลการทดลองตามลำดับ แม้วิธีการที่นำเสนอนั้น จะใช้ระยะเวลาที่มากกว่าคิวโอเอสหนึ่งและคิวโอเอสสองในช่วงระยะเริ่มต้น แต่หากปัญหาการเชื่อมต่อมีระยะเวลาที่ยาวนาน วิธีการที่นำเสนอจะสามารถช่วยลดระยะเวลาการส่งข้อมูลลงได้ เนื่องจากเป็นการส่งต่อข้อมูลไปยังอุปกรณ์ในเครือข่ายอื่นที่ไม่มีปัญหาการเชื่อมต่อหรือมีปัญหาน้อยกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 ผลลัพธ์เฉลี่ยของความล่าช้า

Network Congestion Time (Second)	Delay Time (second)		
	QoS1	QoS2	Proposed Solution
20	22.05	22.04	45.00
40	42.06	42.04	50.39
60	63.79	63.96	46.76

ตารางที่ 2 ผลลัพธ์เฉลี่ยของความเร็วในการตอบสนอง

Network Congestion Time (Second)	Response Time (second)		
	QoS1	QoS2	Proposed Solution
20	24.02	24.03	47.09
40	44.06	44.06	52.47
60	65.79	65.97	48.82

แม้วิธีการที่นำเสนอจะใช้ระยะเวลาที่เร็วกว่ากระบวนการคิวโอเอสในกรณีที่ปัญหาการเชื่อมต่อเกิดขึ้นเป็นระยะเวลายาวนาน แต่วิธีการนี้ก็จำเป็นที่จะต้องมีองค์ประกอบหลายอย่างมากกว่ากระบวนการคิวโอเอสตามปกติ อาทิ อุปกรณ์ที่รองรับวิธีการส่งต่อนี้ และจำนวนของเครือข่ายและอุปกรณ์ทั้งหมดในระบบ กล่าวคือ หากไม่มีอุปกรณ์ใดเลยที่รองรับการส่งต่อด้วยเอชทีทีพี การส่งต่อข้อมูลก็จะไม่เกิดขึ้น รวมถึงหากจำนวนของเครือข่ายและอุปกรณ์ในระบบมีปริมาณที่มาก เวลาที่ใช้ในการทดลองส่งก็จะมากตามไปด้วย

6. สรุปผลงานวิจัย

ในงานวิจัยชิ้นนี้ ได้นำเสนอถึงวิธีที่ช่วยในการเพิ่มประสิทธิภาพในการส่งต่อข้อมูลไปยังอุปกรณ์รอบข้าง จากการส่งแบบซิงเกิล-ฮ็อปเป็นการส่งแบบมัลติ-ฮ็อปผ่านทางระบบเครือข่ายที่มีการเชื่อมต่อกับอุปกรณ์ภายในระบบ แม้วิธีการส่งแบบมัลติ-ฮ็อปจะใช้ระยะเวลาที่มากกว่ากระบวนการคิวโอเอสของเอ็มคิวทีทีโพรโทคอลในช่วงระยะเวลาเริ่มต้น แต่หากระยะเวลาที่เกิดปัญหายาวนาน วิธีการนี้ก็สามารถช่วยลดระยะเวลาในการส่งข้อมูลได้ โดยขึ้นอยู่กับจำนวนอุปกรณ์และจำนวนเครือข่ายภายในระบบ

เอกสารอ้างอิง

[1] C. Wongpromrat and C. Chatwiriya, "Relayed Message Sending for MQTT Protocol", The 2016 International Conference on Embedded Systems and Intelligent Technology, Chonburi, Thailand, August 5-8, 2016, pp. 42-46.

[2] S. Lee, H. Kim, D.K. Hong and H. Ju., "Correlation Analysis of MQTT Loss and Delay According to QoS Level", 2013 International Conference Information Networking, Bangkok, Thailand, 2013, p. 714-717.

[3] K. Jain, J. Padhye, V.N. Padmanabhan and L. Qiu, "Impact of Interference on Multi-Hop Wireless Network Performance", The 9th Annual International Conference on Mobile Computing and Networking, Sandiego California, USA, September 14-19, 2003.

[4] *MQTT v3.1.1*, ISO/IEC 20922, 2016.

[5] B. Aziz, "A Formal Model and Analysis of the MQ Telemetry Transport Protocol", 9th International Conference on Availability, Reliability and Security, Fribourg, Switzerland, 2014 pp. 59-68.

[6] M.Gerla, K.Tang and R. Bagrodia, "TCP Performance in Wireless Multi-hop Networks", Second IEEE Workshop on Mobile Computing Systems and Applications, Louisiana, USA, 1999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายชลทวี วงศ์พร้อมรัตน์
วัน เดือน ปีเกิด	21 พฤศจิกายน 2534
ที่อยู่	101/281 หมู่ 1 ถนนวัดศรีวารีน้อย ตำบลศีรษะจรเข้น้อย อำเภอบางเสาธง สมุทรปราการ 10540
ประวัติการศึกษา	2556 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ (เกียรตินิยมอันดับ 2) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประสบการณ์การทำงานและผลงานวิจัย

พ.ศ.2556–2557	วิศวกรระบบ ที่บริษัท อีเอสอาร์ไอ (ประเทศไทย) จำกัด
พ.ศ.2559	วิทยากรอบรมหลักสูตร IoT สำหรับนักพัฒนา (IoT for Developer) โดยสถาบันวิทยาการ สวทช. (NSTDA Academy)
พ.ศ. 2559	ชนะเลิศการประกวด Emergency Disaster Mitigation Hackathon (EDM Hackathon) โดยสำนักงานนวัตกรรมแห่งชาติ (NIA)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้