

ชุดวัดประสิทธิภาพโพรโทคอลไอโอที่สำหรับการทดสอบ
โดยใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง

IOT PROTOCOL PERFORMANCE TESTING SUITE FOR REAL EQUIPMENT OR
VIRTUAL EQUIPMENT



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2561
KMITL-2018-EN-M-070-079

ชุดวัดประสิทธิภาพโพรโทคอลไอโอที่สำหรับการทดสอบ
โดยใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง

IOT PROTOCOL PERFORMANCE TESTING SUITE FOR REAL EQUIPMENT OR
VIRTUAL EQUIPMENT



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2561

KMITL-2018-EN-M-070-079

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IOT PROTOCOL PERFORMANCE TESTING SUITE FOR REAL EQUIPMENT OR
VIRTUAL EQUIPMENT



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN COMPUTER ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2018

KMITL-2018-EN-M-070-079

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2018

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ ชุดวัดประสิทธิภาพโพรโทคอลไอโอที่สำหรับการทดสอบโดยใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง

Thesis Title IoT Protocol Performance Testing Suite for Real Equipment or Virtual Equipment

นักศึกษา นายวิปศย์ ปุຍສໍາລີ

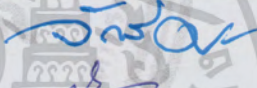
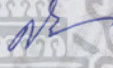

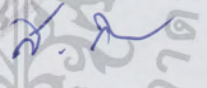
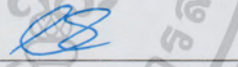
รหัสประจำตัว 56601131

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ดร.วัชระ ฉัตรวิริยะ

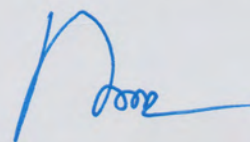
หมายเลขวิทยานิพนธ์ KMITL-2018-EN-M-070-079

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.จักรี	ศรีนนท์ฉัตร	
ผศ.ดร.ศักดิ์ชัย	ทิพย์จักขุรัตน์	
ดร.อำนาจ	ชวเน	
ผศ.ดร.สุรินทร์	กิตติธรรกุล	
ดร.วัชระ	ฉัตรวิริยะ	

วัน / เดือน / ปี ที่สอบ วันจันทร์ที่ 16 กรกฎาคม พ.ศ. 2561 เวลา 09.00-11.00 น.
สถานที่สอบ ณ ห้อง HM-402 อาคารเฉลิมพระเกียรติ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร. คมสัน มาลีสี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ฉบับนี้ คณะวิศวกรรมศาสตร์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
วันที่ 16 กรกฎาคม พ.ศ. 2561

หัวข้อวิทยานิพนธ์	ชุดวัดประสิทธิภาพโพรโตคอลไอโอที่สำหรับการทดสอบโดยใช้ อุปกรณ์จริงหรืออุปกรณ์จำลอง
นักศึกษา	นายวิศิษฐ์ ปุยสำลี
รหัสประจำตัว	56601131
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2561
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ดร.วีชระ ฉัตรวิริยะ

บทคัดย่อ

การทดสอบวัดประสิทธิภาพโพรโตคอลไอโอที่โดยการทดสอบด้วยอุปกรณ์จริงหรืออุปกรณ์จำลอง ยังคงเป็นวิธีที่ได้รับความนิยมในหมู่นักวิจัย เนื่องจากเป็นวิธีที่ได้ทดสอบจริงและให้ผลลัพธ์น่าเชื่อถือ ซึ่งจากการศึกษาวิจัยเกี่ยวกับการทดสอบโพรโตคอลไอโอที่แบบที่ใช้อุปกรณ์จริงหรืออุปกรณ์จำลองพบว่า ผู้วิจัยต้องพัฒนาโปรแกรมเพื่อดำเนินการการทดสอบ เช่น การควบคุมการรับส่งข้อมูล การควบคุมสภาพแวดล้อมการทดสอบ การคำนวณผลลัพธ์ การเก็บบันทึกผล และการแสดงผล ซึ่งยังไม่มีเครื่องมือสำหรับทดสอบแบบสำเร็จรูปหรือพร้อมใช้ วิทยานิพนธ์นี้เล็งเห็นถึงความซับซ้อนและภาระในการเตรียมทดสอบ จึงนำเสนอชุดวัดประสิทธิภาพโพรโตคอลไอโอที่สำหรับการทดสอบโดยใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง โดยชุดทดสอบจะเป็นเครื่องมือสำเร็จรูปสำหรับทดสอบวัดประสิทธิภาพโพรโตคอลไอโอที่โดยมีส่วนควบคุมการทดสอบ การควบคุมการรับส่งข้อมูล การควบคุมสภาพแวดล้อมการทดสอบ การคำนวณผลลัพธ์ การเก็บบันทึกผล และการแสดงผล พร้อมฟังก์ชันดำเนินการทดสอบในเงื่อนไขต่าง ๆ แบบอัตโนมัติ โดยชุดทดสอบจะมีโพรโตคอลไอโอที่ MQTT MQTT-SN CoAP และ AMQP ให้ผู้ทดสอบทำการทดสอบได้โดยไม่ต้องพัฒนาโปรแกรมเพิ่ม ชุดทดสอบจะให้ผลการทดสอบพื้นฐานที่ครบถ้วน อาทิเช่น ค่าเวลาไปกลับ (RTT) อัตราการสูญหายของข้อมูล (Loss rate) ค่าเวลาหน่วงไม่คงที่ (Jitter) ส่วนเกินข้อมูลจากการส่ง (Overhead)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis	IoT Protocol Performance Testing Suite for Real Equipment or Virtual Equipment
Student	Mr.Wiput Puisamlee
Student ID	56601131
Degree	Master of Engineering
Program	Computer Engineering
Year	2018
Thesis Advisor	Dr.Watchara Chatwiriya

ABSTRACT

The Internet of Thing performance testing based on real or virtual equipment is popular solution due to convenience and reliable result. However, from survey, the researchers had to develop the program implementing with process to simulate specific environment, such as packet sending and receiving management, network environment management, result calculating and result display. It might take several times on doing them. To reduce these workloads and make it less complicated, The IoT performance testing suite for real or virtual equipment is, therefore, given in this research. With this testing suite, process management, packet sending and receiving management, network environment management, result calculating result display and automate case process testing are included. All testing processes could be applied without additional programing. And the results could deliver in form of round trip time(RTT), loss rate, Jitter and Overhead.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความกรุณา ช่วยเหลือและการสนับสนุนจากหลายฝ่าย ขอขอบคุณ ดร.วัชร ฉัตรวิริยะ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่คอยดูแล ให้คำปรึกษา และคอยแนะนำสำหรับข้อคิดและความรู้ในการทำงานวิจัย ตลอดจนการดำเนินงานวิจัยและการศึกษาของข้าพเจ้า

ขอขอบคุณห้องปฏิบัติการสมองฝังตัว (Embedded System Laboratory) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้ข้าพเจ้าได้พบกับเพื่อนร่วมงาน ซึ่งได้ร่วมทุกข์ร่วมสุขด้วยกันตลอดการศึกษาในระดับปริญญาโท

และสุดท้ายขอขอบคุณครอบครัวทั้ง บิดา มารดา และญาติมิตร ที่ได้ส่งมอบกำลังใจ คำแนะนำ สำหรับการต่อสู้กับอุปสรรคต่าง ๆ ให้สามารถผ่านพ้นไปได้ด้วยดี และขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้ให้ประสบการณ์ที่ดีแก่ข้าพเจ้าตลอดการศึกษา

วิปศย์ ปุยสำลี

สารบัญ

	หน้า
บทคัดย่อ.....	I
ABSTRACT.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 ที่มาและความสำคัญ.....	1
1.1 ความเป็นมาและความสำคัญของงานวิจัย.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	2
1.3 ปัญหาของงานวิจัย.....	2
1.4 ขอบเขตงานวิจัย.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 อินเทอร์เน็ตของสรรพสิ่ง.....	3
2.2 การส่งข้อมูลด้วยโพรโทคอลไอโอที.....	4
2.3 งานวิจัยที่เกี่ยวข้องกับการวัดประสิทธิภาพโพรโทคอลไอโอที.....	5
2.4 ตัวแปรที่ใช้ในการวัดประสิทธิภาพโพรโทคอลไอโอที.....	8
2.4.1 ตัวแปรตั้งต้น.....	8
2.4.2 ตัวแปรวัดผล.....	12
บทที่ 3 การดำเนินงานวิจัย.....	17
3.1 สมมุติฐานของงานวิจัย.....	17
3.2 ผลการศึกษาการวัดประสิทธิภาพโพรโทคอลไอโอทีที่ใช้อุปกรณ์จริงหรือ อุปกรณ์จำลอง.....	17
3.3 การสนับสนุนของเครื่องมือเพื่อทำการทดสอบ.....	19
3.3.1 การนำโพรโทคอลไอโอทีมาทดสอบ.....	19
3.3.2 เครื่องมือควบคุมการรับส่งข้อมูล.....	20
3.3.3 เครื่องมือควบคุมสภาพแวดล้อม.....	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ IV อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.3.4 เครื่องมือเก็บผล	21
3.3.5 เครื่องมือแสดงผลลัพธ์.....	21
3.4 แนวคิดการแก้ไขปัญหา	23
3.5 การออกแบบระบบ.....	23
3.5.1 การเตรียมการก่อนการทดสอบ	24
3.5.2 การนำโพรโทคอลไอโอที่มาทดสอบ.....	25
3.5.3 การควบคุมการรับส่งข้อมูล	26
3.5.4 แพ็คเก็ต.....	28
3.5.5 การควบคุมสภาพแวดล้อม	28
3.5.6 ตัวรับข้อมูล ประมวลผลและการบันทึกผลลัพธ์	29
3.5.7 การแสดงผลลัพธ์.....	30
3.5.8 ส่วนการนำเข้าโพรโทคอลไอโอที่ด้วยมือ.....	31
3.6 ชุดทดลองที่พัฒนาเสร็จสิ้น	32
3.6.1 การติดตั้ง	32
3.6.2 ตัวอย่างการใช้งานชุดทดสอบวัดประสิทธิภาพโพรโทคอลไอโอที่	33
3.6.3 ส่วนแสดงผลลัพธ์	35
3.7 การทดสอบและอุปกรณ์ที่ใช้ในการทดสอบ	36
บทที่ 4 ผลการทดลอง.....	38
4.1 ผลการทดสอบวัดประสิทธิภาพโพรโทคอลไอโอที่โดยชุดทดสอบที่พัฒนาขึ้น	38
4.1.1 ระยะเวลาส่งข้อมูลไปกลับ.....	38
4.1.2 อัตราการสูญหายของข้อมูล.....	46
4.1.3 ค่าเวลาหน่วงไม่คงที่.....	49
4.1.4 ข้อมูลส่วนเกิน.....	56
4.1.5 การทดสอบชุดทดสอบกับเครือข่ายแบบผสม	59
4.2 เปรียบเทียบการเตรียมการแบบดั้งเดิมกับใช้ชุดทดสอบไอโอที่โพรโทคอล	61
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	63
5.1 สรุปผลการวิจัย	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ V ึ่งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
5.2 ข้อเสนอแนะในการพัฒนาต่อยอด	63
เอกสารอ้างอิง	65
ภาคผนวก.....	68
ประวัติผู้เขียน.....	73



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และแจ้งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 รายชื่ออุปกรณ์จริงที่ใช้ในการทดสอบ.....	36
3.2 รายชื่ออุปกรณ์จำลองที่ใช้ในการทดสอบ.....	37
4.1 ค่าตัวแปรตั้งต้นในการทดสอบที่ 1.....	38
4.2 ค่าตัวแปรตั้งต้นในการทดสอบที่ 2.....	40
4.3 ค่าตัวแปรตั้งต้นในการทดสอบที่ 3.....	42
4.4 ค่าตัวแปรตั้งต้นในการทดสอบที่ 4.....	43
4.5 ค่าตัวแปรตั้งต้นในการทดสอบที่ 5.....	46
4.6 ค่าตัวแปรตั้งต้นในการทดสอบที่ 6.....	48
4.7 ค่าตัวแปรตั้งต้นในการทดสอบที่ 7.....	50
4.8 ค่าตัวแปรตั้งต้นในการทดสอบที่ 8.....	53
4.9 ค่าตัวแปรตั้งต้นในการทดสอบที่ 9.....	56
4.10 ค่าตัวแปรตั้งต้นในการทดสอบที่ 10.....	58
4.11 เปรียบเทียบการเตรียมการแบบดั้งเดิมกับใช้ชุดทดสอบโพรโทคอลไอโอที.....	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และ VII อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างการประยุกต์ใช้เทคโนโลยี Internet of Things ในแขนงต่างๆ	3
2.2 แสดงระดับคุณภาพบริการของโพรโทคอล MQTT.....	5
2.3 แสดงโครงสร้างเครือข่ายที่ใช้ในงานวิจัยของ Shinho Lee และคณะ	5
2.4 อุปกรณ์จริงและอุปกรณ์จำลองที่ใช้ในงานของ Dinesh Thangavel และคณะ	6
2.5 โครงสร้างการเชื่อมต่อเครือข่ายในการทดสอบของ Dinesh Thangavel และคณะ.....	6
2.6 อุปกรณ์จริงที่ใช้ทดสอบในงานของ Yuang Chen และ Thomas Kunz.....	7
2.7 อุปกรณ์จริงที่ใช้ในงานวิจัยของ Jorge E. Luzuriaga และคณะ	8
2.8 ความสัมพันธ์ระหว่างข้อมูลส่วนเกิน (Overhear) กับขนาดของแพ็คเก็ต.....	9
2.9 ความสัมพันธ์ระหว่าง Jitter กับความถี่ในการส่งข้อมูล.....	9
2.10 ความสัมพันธ์อัตราการสูญหายข้อมูลกับระบบ QoS ของ MQTT.....	10
2.11 อัตราการสูญหายข้อมูลของโพรโทคอล MQTT DDS CoAP และ UDP เมื่อกำหนดอัตราการสูญหายข้อมูลเครือข่าย.....	10
2.12 ค่าเวลาแฝงในการส่งข้อมูลของโพรโทคอล MQTT DDS CaAP และ UDP เมื่อกำหนดค่าเวลาแฝงระบบ.....	11
2.13 ความสัมพันธ์ระหว่างเวลาในการส่งข้อมูลของโพรโทคอล MQTT DDS CaAP และ UDP ในสภาพแวดล้อมเครือข่ายที่จำกัดปริมาณการรับ/ส่งข้อมูล.....	11
2.14 Round Trip Time	12
2.15 ความสัมพันธ์ของ RTT เมื่อระดับ QoS ต่างกันของโพรโทคอล MQTT	12
2.16 ความสัมพันธ์ระหว่างอัตราการสูญหายข้อมูลกับระดับ QoS ของโพรโทคอล MQTT	13
2.17 ความสัมพันธ์ของค่าเวลาหน่วงไม่คงที่กับความถี่ในการส่งข้อมูล.....	14
2.18 ผลลัพธ์ค่าเวลาหน่วงไม่คงที่ในการทดลองของ Jorge E. Luzuriaga	15
2.19 ความสัมพันธ์ระหว่าง Overhead กับขนาดของแพ็คเก็ตในการทดสอบของ Dinesh Thangavel และคณะ.....	16
3.1 Diagram model การวัดประสิทธิภาพ	18
3.2 การเชื่อมต่อ WANem.....	20
3.3 ส่วนตั้งค่าของ WANem	21
3.4 ตัวอย่างการแสดงผลลัพธ์แบบตาราง	21
3.5 ตัวอย่างการแสดงผลลัพธ์แบบแผนภูมิแบบเส้น	22
3.6 ตัวอย่างการแสดงผลลัพธ์แบบแผนภูมิแบบฮิสโตแกรมสะสม	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ VIII อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.7 ตัวอย่างการแสดงผลล์พ์แบบแผนภูมิแบบแท่ง	23
3.8 ไดอะแกรม ส่วนต่างๆ ของการออกแบบชุดทดสอบ	24
3.9 ลักษณะโครงสร้างเครือข่ายสำหรับชุดทดสอบ	24
3.10 โครงสร้างเครือข่ายแบบต่างๆ ที่ชุดทดสอบรองรับ	25
3.11 โพล์ชาร์ทการทำงานการควบคุมการรับส่งข้อมูล	27
3.12 แสดงการทำงานของเตรียมแพ็คเก็ต	28
3.13 โพล์ชาร์ทแสดงการทำงานส่วนควบคุมสภาพแวดล้อม	29
3.14 โพล์ชาร์ทภาครับข้อมูล	30
3.15 ไดอะแกรมการทำงานของส่วนแสดงผลล์พ์	31
3.16 การนำเข้าโปรโตคอลกับส่วนตัวดำเนินการส่งข้อมูล	31
3.17 การนำเข้าโปรโตคอลกับส่วนตัวดำเนินการรับข้อมูล	32
3.18 โครงสร้างการติดตั้งชุดทดสอบ	33
3.19 การเริ่มใช้โปรแกรมแม่ข่าย	34
3.20 การเริ่มใช้โปรแกรมส่วนรับข้อมูล	34
3.21 ตัวอย่างการใช้งานโปรแกรมส่วนส่งข้อมูล	35
3.22 ตัวอย่างโปรแกรมแสดงผลของผลการทดลอง	35
4.1 ระยะเวลาส่งข้อมูลไป-กลับในแต่ละค่าเวลาแฝงจากงานวิจัยเปรียบเทียบ	39
4.2 ระยะเวลาส่งข้อมูลไป-กลับในแต่ละค่าเวลาแฝงจากชุดทดสอบในเครือข่าย แบบสายและไร้สาย	39
4.3 ระยะเวลาส่งข้อมูลไป-กลับเมื่อมีอัตราการสูญหายข้อมูลในเครือข่าย เพิ่มขึ้นจากงานวิจัยเปรียบเทียบ	40
4.4 ระยะเวลาส่งข้อมูลไป-กลับเมื่อมีอัตราการสูญหายข้อมูลในเครือข่าย เพิ่มขึ้นจากชุดทดสอบเครือข่ายแบบสายแบบไร้สาย	41
4.5 ระยะเวลาส่งข้อมูลไป-กลับเมื่อความกว้างในการรับ-ส่งข้อมูลของเครือข่าย แตกต่างกันจากงานวิจัยเปรียบเทียบ	42
4.6 ระยะเวลาส่งข้อมูลไป-กลับเมื่อความกว้างในการรับ-ส่งข้อมูลของเครือข่าย แตกต่างกันจากชุดทดสอบในเครือข่ายแบบสายและไร้สาย	42
4.7 ระยะเวลาส่งข้อมูลไป-กลับพรโทคอล MQTT เมื่อคุณภาพบริการ และขนาดของแพ็คเก็ตที่ต่างกันเครือข่ายแบบสายจากงานวิจัยเปรียบเทียบ	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และ IX อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.8	ระยะเวลาส่งข้อมูลไป-กลับพรโทคอล MQTT เมื่อคุณภาพบริการและขนาดของแพ็คเก็ตที่ต่างกันในการ์ข่ายแบบสายจากชุดทดสอบ 44
4.9	ระยะเวลาส่งข้อมูลไป-กลับพรโทคอล MQTT เมื่อคุณภาพบริการและขนาดของแพ็คเก็ตที่ต่างกันในการ์ข่ายแบบไร้สายจากงานวิจัยเปรียบเทียบ..... 45
4.10	ระยะเวลาส่งข้อมูลไป-กลับพรโทคอล MQTT เมื่อคุณภาพบริการและขนาดของแพ็คเก็ตที่ต่างกันในการ์ข่ายแบบไร้สายจากชุดทดสอบ 45
4.11	อัตราการสูญหายข้อมูลเมื่อมีอัตราการสูญหายข้อมูลของการ์ข่ายต่างๆจากงานวิจัยเปรียบเทียบ 46
4.12	อัตราการสูญหายข้อมูลเมื่อมีอัตราการสูญหายข้อมูลของการ์ข่ายต่างๆจากชุดทดสอบในการ์ข่ายแบบสายและไร้สาย..... 47
4.13	อัตราการสูญหายของข้อมูล เมื่อกำหนดคุณภาพบริการและขนาดของแพ็คเก็ตของพรโทคอล MQTT ในการ์ข่ายแบบไร้สายจากงานวิจัยเปรียบเทียบ..... 48
4.14	อัตราการสูญหายของข้อมูล เมื่อกำหนดคุณภาพบริการและขนาดของแพ็คเก็ตของพรโทคอล MQTT ในการ์ข่ายแบบสายและไร้สายจากชุดทดสอบ..... 48
4.15	วิธีการทดสอบค่าเวลาหน่วงไม่คงที่ของในงานวิจัยของ Jorge E. Luzuriaga และคณะ 49
4.16	ค่าเวลาหน่วงไปคงที่ของพรโทคอล MQTT จากงานวิจัยเปรียบเทียบ..... 50
4.17	ค่าเวลาหน่วงไปคงที่ของพรโทคอล MQTT จากงานชุดทดสอบ..... 51
4.18	ค่าเวลาหน่วงไม่คงที่ของพรโทคอล AMQP จากงานวิจัยเปรียบเทียบ..... 52
4.19	ค่าเวลาหน่วงไม่คงที่ของพรโทคอล AMQP จากชุดทดสอบ 52
4.20	ค่าเวลาหน่วงไม่คงที่ที่มากที่สุดของพรโทคอล MQTT และ AMQP ที่ความถี่ในการส่ง 10 มิลลิวินาที กับขนาดแพ็คเก็ตที่ 512 ไบต์ และ 6K ไบต์ จากงานวิจัยเปรียบเทียบ..... 54
4.21	ผลการทดสอบค่าเวลาหน่วงไม่คงที่ที่มากที่สุดของพรโทคอล MQTT และ AMQP ที่ความถี่ในการส่ง 10 มิลลิวินาที กับขนาดแพ็คเก็ตที่ 512 ไบต์ จากชุดทดสอบ 55
4.22	ผลการทดสอบค่าเวลาหน่วงไม่คงที่ที่มากที่สุดของพรโทคอล MQTT และ AMQP ที่ความถี่ในการส่ง 10 มิลลิวินาที กับขนาดแพ็คเก็ตที่ 6K ไบต์ จากชุดทดสอบ 55
4.23	ข้อมูลส่วนเกินของพรโทคอล CoAP และ MQTT เมื่ออัตราการสูญหายข้อมูลของการ์ข่ายแตกต่างกันจากงานวิจัยเปรียบเทียบ..... 56

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.24 ข้อมูลส่วนเกินของโพรโทคอล CoAP และ MQTT เมื่ออัตราการสูญหายข้อมูลของเครือข่ายแตกต่างกันจากชุดทดสอบในเครือข่ายแบบสายและไร้สาย.....	57
4.25 อัตราส่วนของข้อมูลกับข้อมูลที่ถูกลบของโพรโทคอล CoAP และ MQTT จากงานวิจัยเปรียบเทียบ	58
4.26 อัตราส่วนของข้อมูลกับข้อมูลที่ถูกลบของโพรโทคอล CoAP และ MQTT จากชุดทดสอบในเครือข่ายแบบสายและไร้สาย	59
4.27 การทดสอบในเครือข่ายแบบผสม.....	60



บทที่ 1

บทนำ

บทนี้จะกล่าวส่วนเนื้อหาเริ่มต้นของงานวิจัยนี้ประกอบด้วย 4 หัวข้อคือ ความเป็นมาและความสำคัญของงานวิจัย วัตถุประสงค์ของงานวิจัย ปัญหาของงานวิจัย และขอบเขตงานวิจัย โดยมีรายละเอียดดังต่อไปนี้

1.1 ความเป็นมาและความสำคัญของงานวิจัย

อินเทอร์เน็ตของสรรพสิ่งหรือ Internet of Thing [1][2] เป็นแนวคิดที่ต้องการให้สิ่งต่าง ๆ สามารถติดต่อสื่อสารแลกเปลี่ยนข้อมูลเพื่อนำข้อมูลมาใช้ให้เกิดประโยชน์ เนื่องด้วยความก้าวหน้าของเทคโนโลยีการผลิต อุปกรณ์เครือข่ายจึงมีราคาถูกและมีขนาดเล็กลง ช่วยส่งเสริมให้แนวคิดอินเทอร์เน็ตของสรรพสิ่งพัฒนาไปอย่างรวดเร็ว และถูกนำไปประยุกต์ใช้กับงานในด้านต่าง ๆ มากมาย อาทิเช่น ระบบดูแลผู้ป่วยที่สามารถรายงานสภาพของผู้ป่วยและแจ้งผู้ดูแลได้เมื่อมีเหตุฉุกเฉินอัตโนมัติ ระบบจัดการและติดตามสภาพแวดล้อมโรงเรือนอัตโนมัติ ระบบบ้านอัจฉริยะที่อุปกรณ์ต่าง ๆ ตัวภายในและภายนอกสามารถทำงานร่วมกันได้อัตโนมัติ

อินเทอร์เน็ตของสรรพสิ่งถูกออกแบบให้สามารถใช้ได้กับอุปกรณ์ที่มีทรัพยากรต่ำ โพรโทคอลเครือข่ายแบบเดิมมีขนาดใหญ่ จึงมีการออกแบบโพรโทคอลสำหรับอินเทอร์เน็ตของสรรพสิ่งขึ้นมา โดยเฉพาะ อาทิเช่น MQTT MQTT-SN AMQP CoAP และอื่น ๆ โพรโทคอลเหล่านี้มีส่วนที่แตกต่างกันและความเหมาะสมในการเลือกใช้งานต่างกัน ทำให้มีนักวิจัยมากมายศึกษาเกี่ยวกับการนำโพรโทคอลเหล่านี้ไปประยุกต์ใช้กับงาน ซึ่งแตกต่างกันด้วยโทโปโลยี เครือข่าย และอุปกรณ์ โดยมีการทดสอบต่าง ๆ อาทิเช่น ทดสอบส่งข้อมูลด้วยขนาดข้อมูลที่ขนาดต่างกัน การส่งข้อมูลในสภาพแวดล้อมเครือข่ายแตกต่างกัน หรือส่งข้อมูลในเวลาที่แตกต่างกัน ปัญหาหนึ่งที่เกิดในการทดสอบวัดประสิทธิภาพของโพรโทคอล คือ ไม่มีเครื่องมือที่ช่วยทดสอบ ผู้ทดสอบจะต้องพัฒนาโปรแกรมในการดำเนินการทดสอบด้วยตนเองเช่น การควบคุมการทดสอบ การรับส่งข้อมูล การควบคุมสภาพแวดล้อมเครือข่าย ดังนั้นหากมีเครื่องมือที่สามารถช่วยให้ผู้ทดสอบสามารถทำการทดสอบวัดประสิทธิภาพ โดยที่ผู้ทดสอบไม่จำเป็นต้องพัฒนาโปรแกรมในการทดสอบ พร้อมทั้งสามารถปรับเปลี่ยนสภาพแวดล้อมการทดสอบ คำนวณผลลัพธ์พื้นฐานที่สำคัญ เก็บผลการทดลองและดำเนินการทดสอบโดยอัตโนมัติพร้อมแสดงผลที่ได้ทันทีเมื่อทดสอบเสร็จสิ้น ก็จะเป็นการลดขั้นตอนในการเตรียมการในการทดสอบซึ่งจะช่วยให้ผู้ทดสอบดำเนินการทดสอบได้รวดเร็วและสะดวกยิ่งขึ้น ดังนั้นวิทยานิพนธ์นี้จึงมุ่งเน้นไปที่การเสนอเครื่องมือสำหรับการทดสอบประสิทธิภาพสำหรับโพรโทคอลอินเทอร์เน็ตของสรรพสิ่งแบบอัตโนมัติสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของงานวิจัย

จัดทำเครื่องมือสำหรับทดสอบวัดประสิทธิภาพโพรโทคอลอินเทอร์เน็ตของสรรพสิ่งสำหรับการทดสอบด้วยอุปกรณ์จริงหรืออุปกรณ์จำลอง ที่พร้อมใช้งาน สามารถดำเนินการแบบอัตโนมัติ โดยที่ผู้ทดสอบไม่ต้องพัฒนาโปรแกรมใด ๆ โดยมีโพรโทคอลไอโอที MQTT MQTT-SN CoAP AMQP ให้ผู้ทดสอบสามารถเลือกทดสอบได้ทันที นอกจากนี้ผู้ทดสอบสามารถนำเข้าโพรโทคอลอื่นหรือที่พัฒนาขึ้นมาใช้มาทดสอบ พร้อมแสดงผลพื้นฐานสำคัญการทดสอบได้ทันทีเมื่อทดสอบเสร็จสิ้น

1.3 ปัญหาของงานวิจัย

การทดสอบประสิทธิภาพโพรโทคอลสำหรับอินเทอร์เน็ตของสรรพสิ่งโดยใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง ผู้ทดสอบจำเป็นต้องจัดทำชุดโปรแกรมเพื่อดำเนินการทดสอบด้วยตนเอง ทั้งการรับส่งข้อมูล ควบคุมการเปลี่ยนขนาดของข้อมูล ควบคุมความถี่การส่ง ปรับเปลี่ยนระดับคุณภาพของบริการ ควบคุมสภาพแวดล้อมของเครือข่าย การเก็บผล และการแสดงผล

1.4 ขอบเขตงานวิจัย

พัฒนาชุดทดสอบวัดประสิทธิภาพโพรโทคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลอง และทดสอบโดยใช้เทคโนโลยีเครือข่ายแบบสายหรือไร้สาย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 อินเทอร์เน็ตของสรรพสิ่ง

อินเทอร์เน็ตของสรรพสิ่งหรือ Internet of Thing หรือที่เรียกกันว่าไอโอที ได้ถูกคิดขึ้นโดย Kevin Ashton [2] โดยมีแนวคิดในการท ให้เซ็นเซอร์ต่าง ๆ สามารถเชื่อมต่อสื่อสารแลกเปลี่ยนข้อมูลซึ่งเป็นประโยชน์ ต่อมาเทคโนโลยีการผลิตอุปกรณ์อิเล็กทรอนิกส์พัฒนาขึ้นทำให้หน่วยประมวลผลมีขนาดเล็กและใช้พลังงานต่ำ จนมีการพัฒนาอุปกรณ์เซ็นเซอร์ที่สามารถประมวลผลและสื่อสารได้ด้วยเทคโนโลยีสื่อสารแบบไร้สายต้นทุนต่ำ ที่เรียกกันว่า wireless sensor network (WSN) ทำให้เทคโนโลยีไอโอทีพัฒนาไปอย่างรวดเร็ว โดยคำว่า Thing นั้นหมายถึงทุกสิ่ง เป้าหมายคือนา เซ็นเซอร์ติดไปยังสิ่งต่าง ๆ จากนั้นสร้างโครงสร้างหรือระบบทำให้สิ่งเหล่านั้นสามารถทำงานร่วมกัน ได้ จนกระทั่งมีการใช้คำว่า อัจฉริยะ และนำไปใช้กับระบบต่าง ๆ มากมายและขยายไปยังทุก ๆ แขนง เช่น บ้านอัจฉริยะ รถอัจฉริยะ ฟาร์มอัจฉริยะ รูปที่ 2.1 แสดงที่แขนงต่าง ๆ ที่สามารถนำเอา เทคโนโลยีไอโอทีไปประยุกต์ใช้ได้ [21]



รูปที่ 2.1 ตัวอย่างการประยุกต์ใช้เทคโนโลยี Internet of Things ในแขนงต่างๆ [21]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีอุปกรณ์สื่อสารก็ย่อมต้องมีภาษาที่ใช้ในการสื่อสาร ซึ่งเรียกกันว่าโพรโทคอล ในการสื่อสารเครือข่ายดั้งเดิมจะนิยมใช้โพรโทคอล HTTP FTP หรืออื่น ๆ โพรโทคอลเหล่านี้มักจะมีขนาดใหญ่เนื่องจากออกแบบมาใช้งานกับอุปกรณ์ที่มีหน่วยประมวลผลที่มีความเร็วสูง แต่เนื่องจากอุปกรณ์ internet of thing เป็นอุปกรณ์ที่มีหน่วยประมวลผลความเร็วต่ำ หน่วยความจำน้อย และมีพลังงานจำกัด ดังนั้นอุปกรณ์เหล่านี้ส่วนมากจะออกแบบมาเพื่อใช้เป็นเซ็นเซอร์หรือประมวลผลเพียงเล็กน้อยและส่งข้อมูลไปยังแม่ข่าย จึงมีการสร้างโพรโทคอลสำหรับ internet of thing ซึ่งมีขนาดเล็กกว่าโพรโทคอลเครือข่ายปกติ เช่น MQTT CoAP MQTT-SN AMQP ซึ่งแต่ละตัวมีความแตกต่างกันเช่น ขนาด กลไกการทำงาน และคุณภาพบริการ จึงทำให้มีคุณสมบัติที่เหมาะสม ข้อดีข้อเสียและความได้เปรียบที่ต่างกันไปด้วย จึงทำให้มีการศึกษาวิจัยเปรียบเทียบประสิทธิภาพโพรโทคอลเหล่านี้กันอย่างแพร่หลาย

2.2 การส่งข้อมูลด้วยโพรโทคอลไอโอที

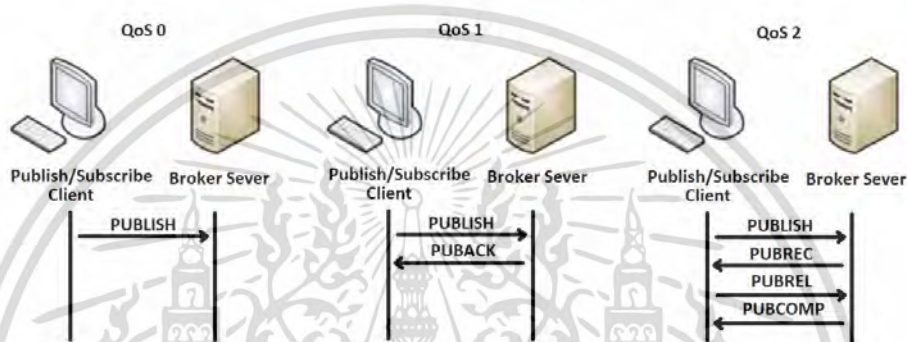
แม้ว่าโพรโทคอลไอโอทีจะถูกจัดให้อยู่คนละกลุ่มกับโพรโทคอลเครือข่ายแบบเก่า แต่ก็ยังจัดเป็นโพรโทคอลประยุกต์ (Application layer) ใน OIS Model เช่นเดียวกัน ดังนั้นโพรโทคอลไอโอทีนอกจากจะแตกต่างกันด้วย ขนาด กลไกในการทำงานและคุณภาพบริการในระดับชั้นโพรโทคอลประยุกต์แล้ว ยังมีความแตกต่างกันที่ แต่ละตัวมีการออกแบบให้เลือกใช้ระดับชั้นขนส่ง (Transport layer) ซึ่งจะมีโพรโทคอลในการส่งข้อมูลอยู่ด้วยกัน 2 ตัวคือ TCP[22] และ UDP[22]

โพรโทคอล TCP (Transmission Control Protocol) เป็นโพรโทคอลในการส่งข้อมูลที่มีเสถียรภาพ มีลักษณะเด่นคือ มีตัวจัดการข้อมูลให้มีความเหมาะสมกับเครือข่ายที่ใช้ มีการสร้างการเชื่อมต่อทุกครั้งก่อนที่จะทำการส่งข้อมูล มีกลไกการตรวจสอบการส่งข้อมูลเมื่อไปถึงปลายทาง (Handshake) และมีการจัดการลำดับในการส่งข้อมูลซึ่งจะทำให้ได้รับข้อมูลตามลำดับที่ถูกอย่างแน่นอน โดยส่วนมากโพรโทคอล TCP จะถูกใช้ในการส่งข้อมูลที่ต้องการความถูกต้องและเป็นล ดับ

โพรโทคอล UDP (User Datagram Protocol) ซึ่งมีลักษณะที่ตรงกันข้ามกับโพรโทคอล TCP คือจะไม่มีส่วนในการจัดการข้อมูล ไม่มีการสร้างการเชื่อมก่อนเริ่มการรับส่งข้อมูล ไม่มีการตรวจสอบการส่งข้อมูลเมื่อข้อมูลไปถึงปลายทาง (Non - Handshake) และไม่มีมีการจัดลำดับการส่งข้อมูล ข้อดีของโพรโทคอล UDP คือสามารถส่งข้อมูลได้รวดเร็วกว่า TCP จึงจะเหมาะสมกับการส่งข้อมูลที่ต้องการความเร็ว หรือความต่อเนื่องเช่นการบริการสื่อออนไลน์ และการส่งข้อมูลแบบไม่ต้องส่งการเชื่อมต่อทำให้สามารถส่งข้อมูลจากตัวส่ง 1 ตัวไปยังตัวรับหลายตัวได้อย่างง่ายดายเช่นการส่งแบบ มัลติแคสต์ (Multicast) หรือบรอดแคสต์ (Broadcast) เป็นต้น

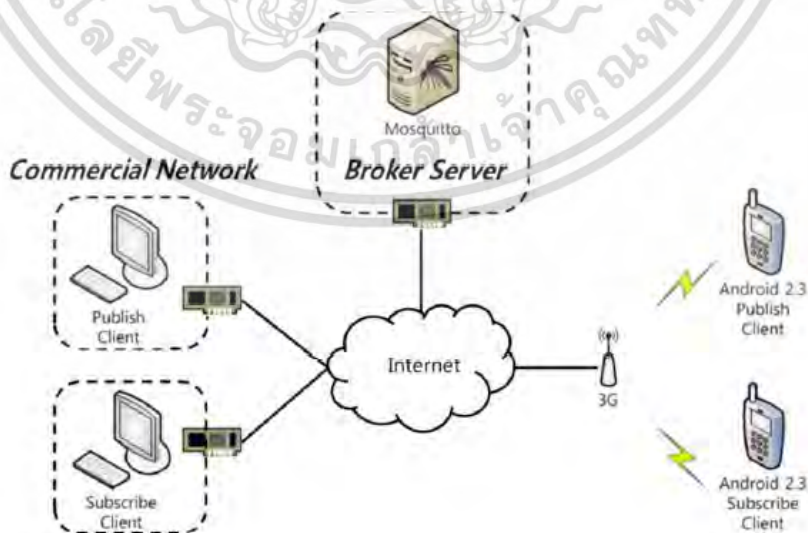
2.3 งานวิจัยที่เกี่ยวกับการวัดประสิทธิภาพโพรโทคอลไอโอที

1. Shinho Lee. และคณะ ได้ศึกษาวิจัย (Correlation Analysis of MQTT Loss and Delay According to QoS Level) [3] ความสัมพันธ์ของค่าแพ็คเก็ตสูญหายและระยะเวลาในการส่งข้อมูลเมื่อเลือกใช้ระดับคุณภาพบริการ (Quality of Service : QoS) ที่ต่างกันของโพรโทคอล MQTT ซึ่งมีด้วยกัน 3 ระดับได้แก่ QoS ระดับ 0 โพรโทคอลจะส่งแพ็คเก็ตโดยไม่คำนึงว่าปลายทางจะได้รับข้อมูลหรือไม่ QoS ระดับ 1 โพรโทคอลจะส่งแพ็คเก็ตพร้อมรอการตอบกลับจาก Broker server เป็นอันทราบว่าได้รับแพ็คเก็ตแล้ว QoS ระดับ 2 โพรโทคอลจะส่งแพ็คเก็ตพร้อมรอการตอบกลับจาก Broker server ถึง 4 ขั้นตอนเพื่อเป็นการยืนยันว่า Broker server ได้รับแพ็คเก็ตแล้ว



รูปที่ 2.2 แสดงระดับคุณภาพบริการของโพรโทคอล MQTT [3]

โดยในการวิจัยใช้อุปกรณ์จริงทั้งหมด พร้อมกันใช้เครือข่ายเชื่อมต่อกันผ่านอินเทอร์เน็ต และมีเทคโนโลยีเครือข่ายทั้งแบบสายและไร้สาย ดังรูปที่ 2.3 [3]

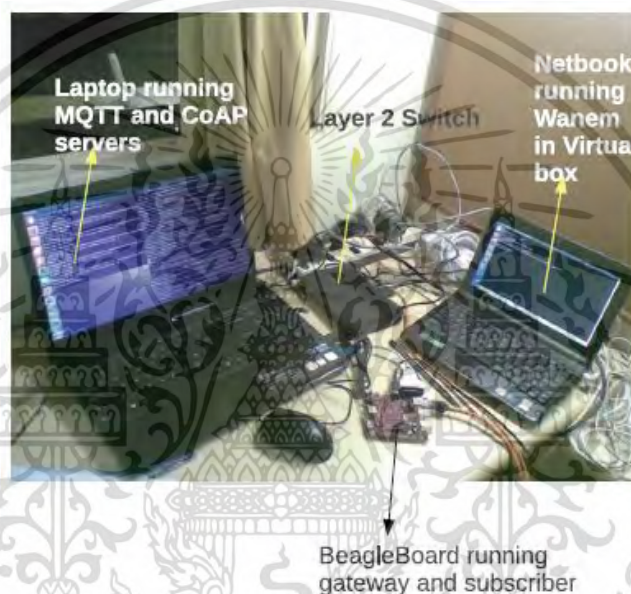


รูปที่ 2.3 แสดงโครงสร้างเครือข่ายที่ใช้ในงานวิจัยของ Shinho Lee และคณะ [3]

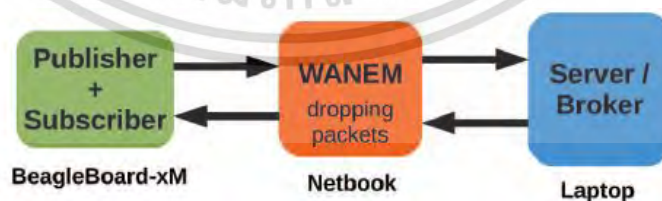
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในงานวิจัย Lee และคณะได้ทำการส่งแพ็คเกจที่มีขนาดตั้งแต่ 1000 ไบต์ ถึง 16000 ไบต์ ที่คุณภาพบริการระดับ 0 ถึง 2 ของโพรโทคอล MQTT ทั้งทดสอบในเครือข่ายแบบสายและไร้สาย ในงานวิจัยไม่ได้ใช้การควบคุมสภาพแวดล้อม แต่ทดสอบในสภาพแวดล้อมอินเทอร์เน็ตจริง

2. Dinesh Thangavel และคณะ (Performance Evaluation of MQTT and CoAP via Common Middleware) [4] ศึกษาวิจัยในการวิจัยประสิทธิภาพของโพรโทคอล MQTT และ CoAP เพื่อพัฒนาการใช้งานแบบโปรแกรมกลางร่วม ในการทดสอบมีการใช้อุปกรณ์จริงและอุปกรณ์จำลองร่วมกัน และใช้ตัวควบคุมสภาพแวดล้อม WANEM ดังรูปที่ 2.4 [4] และ 2.5 [4]



รูปที่ 2.4 อุปกรณ์จริงและอุปกรณ์จำลองที่ใช้ในงานของ Dinesh Thangavel และคณะ [4]

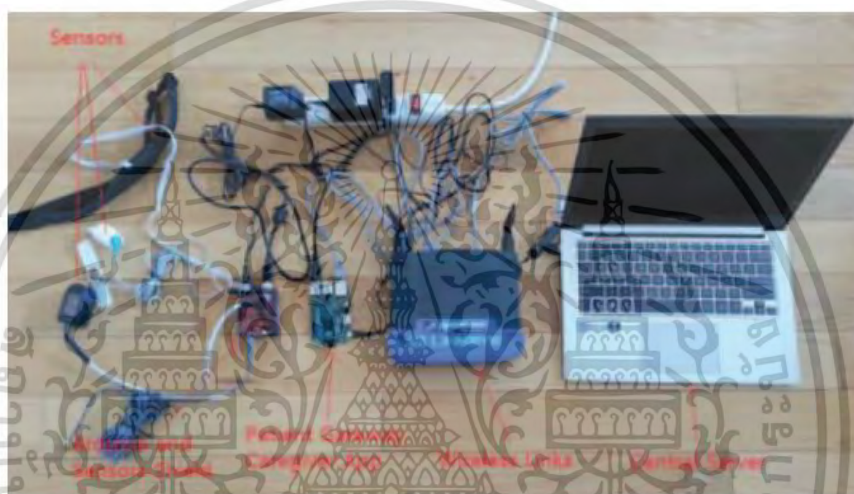


รูปที่ 2.5 โครงสร้างการเชื่อมต่อเครือข่ายในการทดสอบของ Dinesh Thangavel และคณะ [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย Dinesh Thangavel และคณะได้ทำการทดสอบโดยวัดระยะเวลาที่แพ็คเก็ตเดินทางไปกลับ อัตราการสูญหายแพ็คเก็ตเกิดและข้อมูลส่วนเกิน โดยในการทดสอบได้ควบคุมสภาพแวดล้อมอัตราการสูญหายของแพ็คเก็ตในเครือข่าย

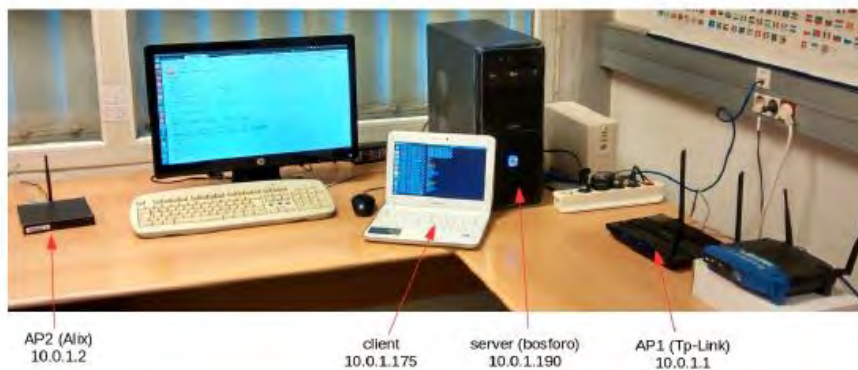
3. Yuang Chen และ Thomas Kunz (Performance Evaluation of IoT Protocols under a Constrained Wireless Access Network) [5] ทดสอบประสิทธิภาพโปรโตคอล MQTT CoAP DDS และโปรโตคอลที่พัฒนาขึ้นเองโดย UDP socket ในสภาพแวดล้อมเครือข่ายที่จำกัดโดยทดสอบกับอุปกรณ์จริงดังรูปที่ 2.6 [5] และใช้ NetEM ในการควบคุมสภาพแวดล้อม



รูปที่ 2.6 อุปกรณ์จริงที่ใช้ทดสอบในงานของ Yuang Chen และ Thomas Kunz [5]

โดยการทดสอบได้ควบคุม อัตราสูญหายของข้อมูล ค่าหน่วงเวลา และแบนด์วิธของเครือข่ายและในการส่งข้อมูลมีการปรับเปลี่ยนขนาดของแพ็คเก็ต ระดับคุณภาพบริการ และความถี่ในการส่งแพ็คเก็ต

4. Jorge E. Luzuriaga และคณะ (A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks) [6] ศึกษาวิจัยเปรียบเทียบโปรโตคอล AMQP และ MQTT ในเครือข่ายอุปกรณ์เคลื่อนที่ที่ไม่เสถียร โดยการทดสอบใช้อุปกรณ์จริงโดยได้ทำการทดสอบค่าเวลาหน่วงไม่คงที่ (Jitter) ด้วยวิธีการสลับจุดรับสัญญาณแบบไร้สายดังโดยใช้อุปกรณ์ดังรูปที่ 2.7 [6] เพื่อหาความถี่และขนาดของข้อมูลที่เหมาะสมในการรับส่งข้อมูล



รูปที่ 2.7 อุปกรณ์จริงที่ใช้ในงานวิจัยของ Jorge E. Luzuriaga และคณะ [6]

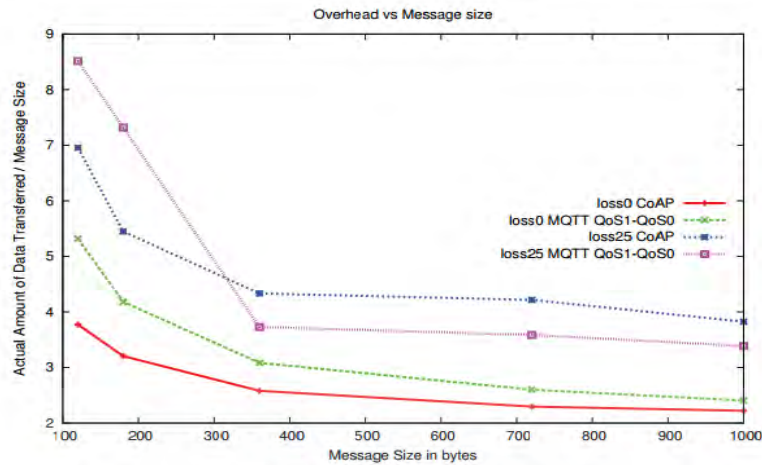
ในการทดสอบมีการปรับค่าความถี่ในการส่งแพ็คเก็ต 10 มิลลิวินาที ถึง 500 มิลลิวินาที และขนาดของแพ็คเก็ต 512 bytes ถึง 6000 bytes เมื่อการเชื่อมต่อเครือข่ายขาดหายและเชื่อมต่อกันอีกครั้ง

2.4 ตัวแปรที่ใช้ในการวัดประสิทธิภาพโพรโทคอลไอโอที

จากการศึกษาวิจัยที่เกี่ยวข้องกับการวัดประสิทธิภาพหรือเปรียบเทียบโพรโทคอลไอโอที ซึ่งแต่ละงานวิจัยมีจุดประสงค์ในการนำเสนอที่แตกต่างกันเช่น การเปรียบเทียบการท างานของโพรโทคอลแต่ละตัว ในสภาพแวดล้อมที่เหมือนกัน หรือ การใช้โพรโทคอลไอโอทีตัวเดียวกันแต่เปรียบเทียบด้วยค่าตัวแปรที่ต่างกัน ซึ่งจากการศึกษาวิจัยพบว่าตัวแปรที่เกี่ยวข้อง แบ่งออกเป็น 2 กลุ่มคือ ตัวแปรตั้งต้น และ ตัวแปรวัดผลดังนี้

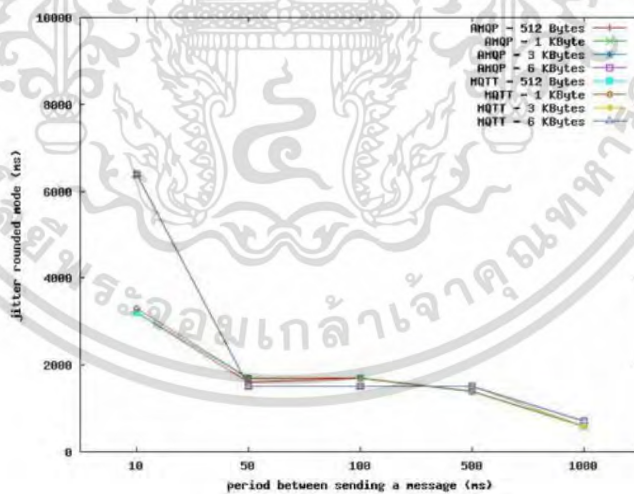
2.4.1 ตัวแปรตั้งต้น

1. ขนาดข้อมูล ขนาดข้อมูลเป็นตัวแปรตั้งต้นที่ผู้วิจัยนิยามมาทดลองหรือทดสอบ เพื่อหาความสัมพันธ์ต่าง ๆ ตัวอย่างเช่นงานวิจัยของ Dinesh Thangavel และคณะ พบว่าโพรโทคอล MQTT และ CoAP ในการส่งข้อมูลมีขนาดเล็กมีขนาดของการเกิดข้อมูลส่วนเกิน (Overhear) [4] ในระบบมากกว่าการส่งข้อมูลขนาดใหญ่ดังแสดงในรูปที่ 2.8 [4]



รูปที่ 2.8 ความสัมพันธ์ระหว่างข้อมูลส่วนเกิน (Overhear) กับขนาดของแพ็คเก็ต [4]

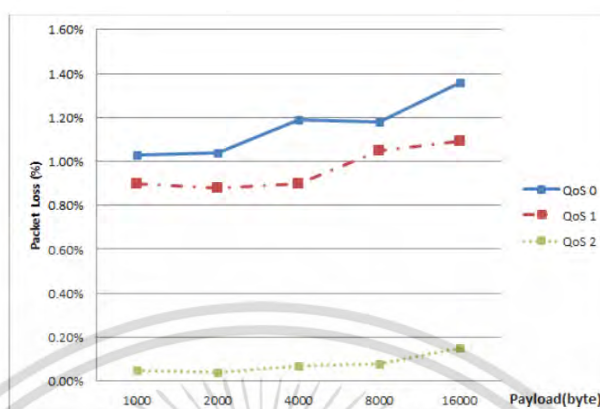
2. ความถี่ในการส่งข้อมูล เช่น 10มิลลิวินาที/แพ็คเก็ต, 50มิลลิวินาที/แพ็คเก็ต, 100 มิลลิวินาที/แพ็คเก็ต การส่งข้อมูลแบบต่อเนื่องการเลือกใช้ความถี่ในการส่งที่เหมาะสมเป็นสิ่งสำคัญเพื่อป้องกันการเกิดข้อมูลสะสมทั้งการรับและส่งข้อมูล ตัวอย่างเช่นงานวิจัยของ Jorge E. Luzuriaga ได้ทดสอบโพรโทคอล MQTT และ AMQP พบว่าในการส่งข้อมูลด้วยความถี่สูงมีผลทำให้ค่าเวลาหน่วงไม่คงที่ (Jitter) [6] สูงขึ้นดังรูปที่ 2.9 [6]



รูปที่ 2.9 ความสัมพันธ์ระหว่าง Jitter กับความถี่ในการส่งข้อมูล [6]

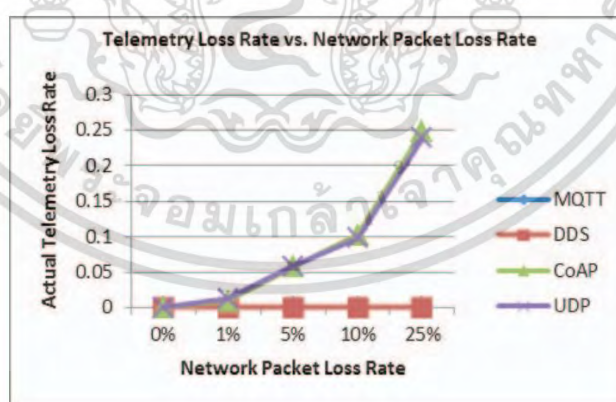
3. คุณภาพบริการ หรือ QoS (Quality of Service) [3] คือการการันตีการรับ-ส่งข้อมูลเพื่อเป็นการการันตีว่าข้อมูลจะถูกส่งไปยังผู้รับ ซึ่งแต่ละโพรโทคอล จะมีระดับให้เลือกใช้แตกต่างกัน เช่น MQTT มีด้วยกัน 3 ระดับได้แก่ ระดับ 0 1 และ 2 CoAP มี 2 ระดับคือ ON กับ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OFF เป็นต้น ตัวอย่างจากงานวิจัยของ Shinho Lee และคณะได้ทดสอบโปรโตคอล MQTT แสดงให้เห็นถึงอัตราของการสูญหายของข้อมูลเมื่อใช้ระดับของ QoS ที่แตกต่างกันดังรูปที่ 2.10 [3]



รูปที่ 2.10 ความสัมพันธ์อัตราการสูญหายข้อมูลกับระบบ QoS ของ MQTT [3]

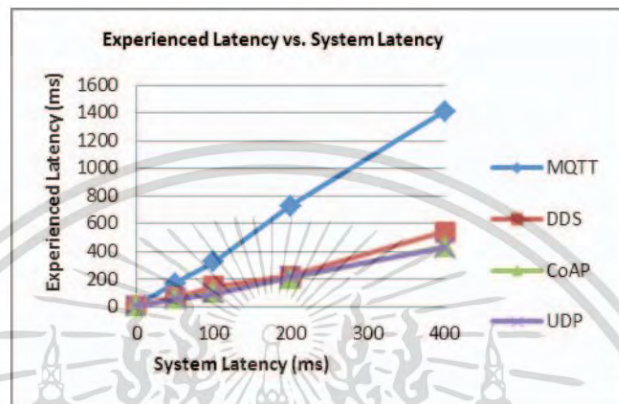
4. อัตราการสูญหายของข้อมูลในเครือข่าย (Loss rate) เป็นการกำหนดให้เกิดอัตราการสูญหายของข้อมูลขึ้นในเครือข่าย เช่น ร้อยละ 5 ร้อยละ 10 หรือ ร้อยละ 20 เป็นต้น ตัวอย่างงานวิจัยของ Yuang Chen และ Thomas Kunz [5] ได้ทดสอบโปรโตคอล MQTT DDS CoAP และ custom UDP โดยการกำหนดอัตราการสูญหายของข้อมูลของระบบจากนั้นดูผลลัพธ์ของอัตราการสูญหายจริงดังรูปที่ 2.11 [5]



รูป 2.11 อัตราการสูญหายข้อมูลของโปรโตคอล MQTT DDS CoAP และ UDP เมื่อกำหนดอัตราสูญหายข้อมูลเครือข่าย [5]

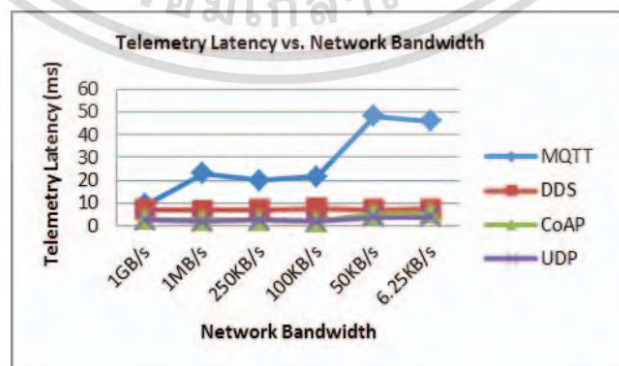
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ค่าเวลาแฝงในการรับ-ส่งข้อมูล หรือที่นิยมเรียกว่าเวลาเทนซี (Latency time) คือ การกำหนดค่าเวลาแฝงในการรับ-ส่งข้อมูล ส่วนมากจะมีหน่วยเป็น มิลลิวินาที (มิลลิวินาที) เช่น 100 มิลลิวินาที 200มิลลิวินาที หรือ 1000มิลลิวินาที ตัวอย่างจากงานวิจัยของ Yuang Chen และ Thomas Kunz [5] ได้ทดสอบโพรโทคอล MQTT DDS CoAP และ custom UDP โดยการกำหนดค่าเวลาแฝงในการรับ-ส่งข้อมูล จากนั้นดูผลลัพธ์ของเวลาที่ใช้ในการส่งจริงดังรูปที่ 2.12[5]



รูปที่ 2.12 ค่าเวลาแฝงในการส่งข้อมูลของโพรโทคอล MQTT DDS CaAP และ UDP เมื่อ กำหนดค่าเวลาแฝงระบบ [5]

6. ปริมาณการรับ-ส่งข้อมูลของการเครือข่าย หรือ Bandwidth คือการกำหนดปริมาณการรับ-ส่งข้อมูลของเครือข่ายอาทิเช่น 100Kbps 1000kbps หรือ 1Gbps ตัวอย่างจากงานวิจัย Yuang Chen และ Thomas Kunz [5] ได้ทดสอบโพรโทคอล MQTT DDS CoAP และ custom UDP โดยการปริมาณการรับ-ส่งข้อมูล จากนั้นดูผลลัพธ์ของเวลาที่ใช้ในการรับ-ส่งข้อมูล ดังรูปที่ 2.13 [5]

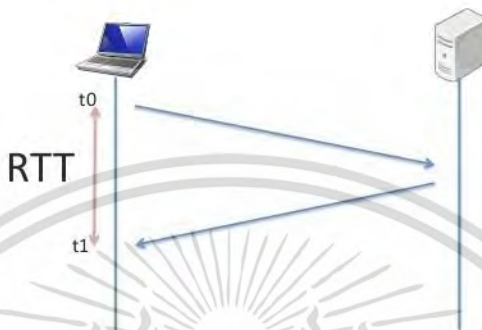


รูป 2.13 ความสัมพันธ์ระหว่างเวลาในการส่งข้อมูลของโพรโทคอล MQTT DDS CaAP และ UDP ในสภาพแวดล้อมเครือข่ายที่จำกัดปริมาณการรับ-ส่งข้อมูล [5]

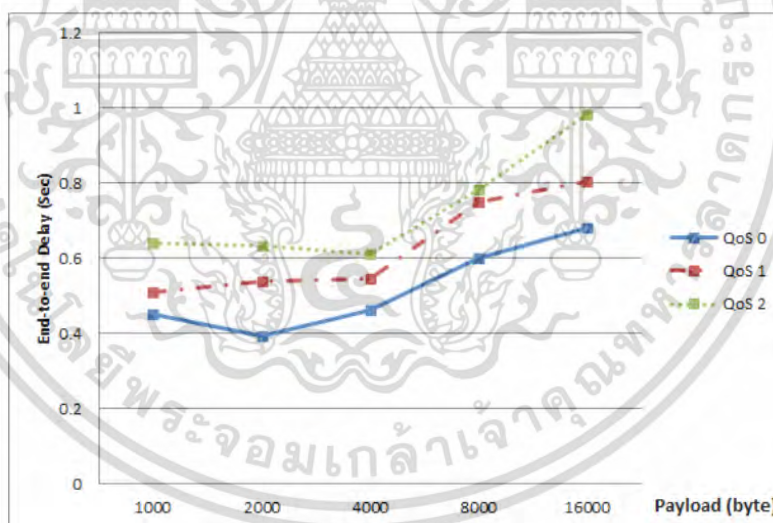
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ตัวแปรวัดผล

1. RTT (Round Trip Time) คือระยะเวลาที่ใช้ในการส่งข้อมูลไป-กลับระหว่างเครื่องลูกข่ายกับเครื่องแม่ข่ายหรือเครื่องส่งกับเครื่องรับ หรือบางงานวิจัยจะเป็นการคำนวณ End-to-End Delay [7] สามารถคำนวณได้จากผลต่างของเวลาที่ได้รับแพ็คเก็ตกลับมาที่ส่งแพ็คเก็ต



รูป 2.14 การส่งข้อมูลไปกลับ (RTT : Round Trip Time)



รูปที่ 2.15 ความสัมพันธ์ของ RTT เมื่อระดับ QoS ต่างกันของโปรโตคอล MQTT [3]

ยกตัวอย่างในงานวิจัยของ Shinho Lee และคณะจากรูปที่ 2.15 [3] ได้ทำการทดลองวัดผลค่า End-to-end Delay (RTT) ของโปรโตคอล MQTT ในแต่ละระดับของ QoS ซึ่งได้ค่า End-to-end Delay แตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. อัตราการสูญหายของข้อมูลคือ ร้อยละของการสูญหายของแพ็คเกจที่คำนวณได้จาก

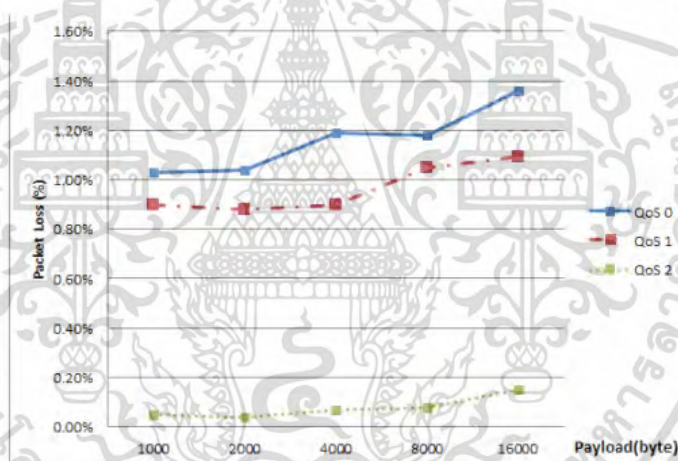
$$L_{rate} = \frac{P_{receive}}{P_{sent}} \times 100 \quad (2.1)$$

L_{rate} = ร้อยละของการสูญหายของแพ็คเกจ

$P_{receive}$ = จำนวนแพ็คเกจทั้งหมดที่รับ

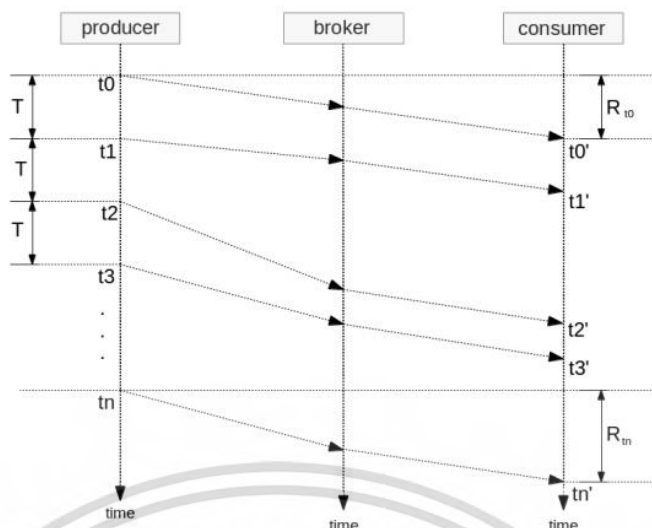
P_{sent} = จำนวนแพ็คเกจทั้งหมดที่ส่ง

ตัวอย่างในงานวิจัยของ Shinho Lee และคณะ [3] ได้ทำการทดลองวัดผลค่าร้อยละของการสูญหายของแพ็คเกจของโปรโตคอล MQTT ในแต่ละระดับของ QoS ซึ่งได้ค่าร้อยละของการสูญหายของแพ็คเกจแตกต่างกันดังรูปที่ 2.16



รูปที่ 2.16 ความสัมพันธ์ระหว่างอัตราการสูญหายข้อมูลกับระดับ QoS ของโปรโตคอล MQTT [3]

3. ค่าเวลาหน่วงที่ไม่คงที่ (Jitter) รูปที่ 2.17 [6] แสดงถึงการข้อมูลจาก producer ผ่าน broker ไปยัง consumer ด้วยความถี่ T producer ทาการส่งข้อมูลออกไป ณ เวลาที่ t_0 t_1 t_2 t_3 ... t_n ตามลำดับ ในขณะที่เดียวกัน ข้อมูลที่มาถึง consumer ณ เวลา t_0' t_1' t_2' t_3' ... t_n' ตามลำดับซึ่งในอุดมคติช่วงเวลาในการมาถึงของข้อมูลจะเท่ากับกับเวลา T



รูปที่ 2.17 ความสัมพันธ์ของค่าเวลาหน่วงไม่คงที่กับความถี่ในการส่งข้อมูล [6]

ค่าเวลาหน่วงไม่คงที่ที่นับบ่งบอกถึงเสถียรภาพโดยรวมของการรับ-ส่งข้อมูลในระบบสามารถมีต้นเหตุมาจากเครือข่าย อุปกรณ์ กระบวนการทำงานของซอฟต์แวร์ หรือกระบวนการรับส่ง ซึ่งค่าเวลาหน่วงไม่คงที่ที่ดีในอุดมคติคือ 0 โดยสามารถคำนวณได้ดังสมการที่ 2.2 [6]

$$J_t = (t_n - t_{n-1}) - T \quad (2.2)$$

J_t = ค่าเวลาหน่วงไม่คงที่, (มิลลิวินาที)

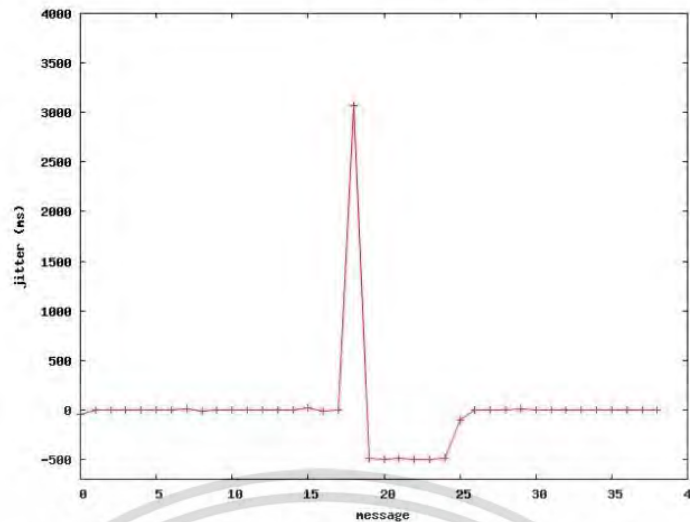
t_n = เวลาที่แพ็คเก็ตปัจจุบันมาถึง, (มิลลิวินาที)

t_{n-1} = เวลาที่แพ็คเก็ตก่อน, (มิลลิวินาที)

T = ความถี่ในการส่ง, (มิลลิวินาที)

ตัวอย่างงานวิจัยของ Jorge E. Luzuriaga และคณะ [6] ได้ทดลองส่งข้อมูลด้วยความถี่ 500 มิลลิวินาที จำนวน 40 แพ็คเก็ต และทำการตัดการเชื่อมต่อของเครือข่ายที่การส่งแพ็คเก็ต 15 เชื่อมต่ออีกครั้งเมื่อเวลาผ่านไป 3 วินาที จากรูปที่ 2.18 [6] จะเห็นได้ว่าแพ็คเก็ตที่ 18 เกิดค่า Jitter สูงถึง 3000 มิลลิวินาที และเมื่อการเชื่อมต่อกลับมาอีกครั้งทำให้แพ็คเก็ตที่ 19 – 24 มีค่า Jitter เป็น -500 มิลลิวินาที เนื่องจากแพ็คเก็ตที่ 19 – 24 มาถึงพร้อมกันจากที่ควรจะมาถึงห่างกันแต่ละแพ็คเก็ต 500 มิลลิวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



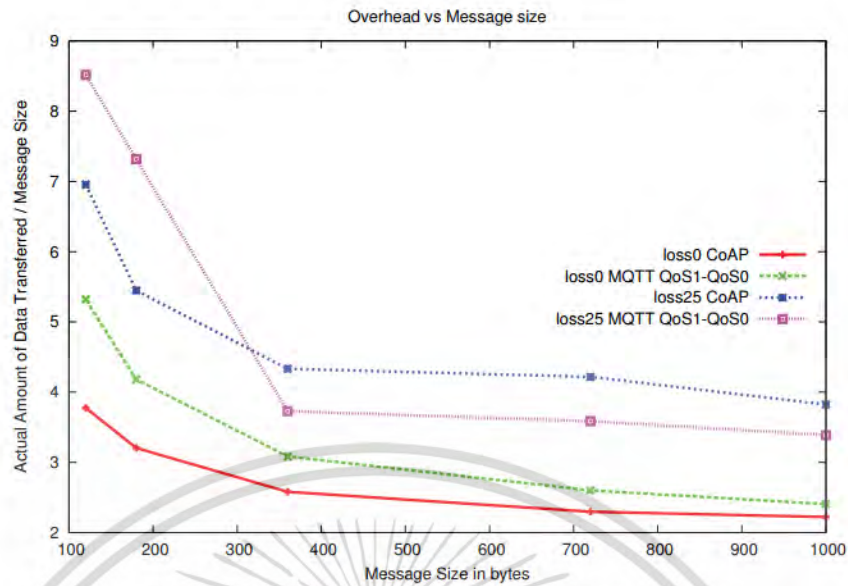
รูปที่ 2.18 ผลลัพธ์ค่าเวลาหน่วงไม่คงที่ในการทดลองของ Jorge E. Luzuriaga [6]

4. ข้อมูลส่วนเกิน (Overhead) คือ ข้อมูลส่วนเกินที่ใช้จริงเมื่อส่งข้อมูลกล่าวคือ ส่วนของข้อมูลที่เกิดขึ้นมาจากข้อมูลที่ต้องการส่ง ซึ่งข้อมูลส่วนเกินนั้นรวมไปถึง Header ของโพรโทคอลด้วย จึงสังเกตได้ว่าโพรโทคอลไอโอทีจะมีการออกแบบ Header ในมีขนาดเล็กเพื่อลดการเกิดข้อมูลส่วนเกินเพื่อลดการใช้ทรัพยากรในการท างาน การค านวณหาข้อมูลส่วนเกินสามารถหาได้จาก

$$Oh = p_{raw} - p \quad (2.3)$$

Oh = ขนาดข้อมูลส่วนเกิน, (ไบต์)
 p_{raw} = ขนาดข้อมูลทั้งหมดที่ได้รับ, (ไบต์)
 p = ขนาดข้อมูลทั้งหมดที่ส่ง, (ไบต์)

ตัวอย่างงานวิจัยของ Dinesh Thangavel และคณะ [4] ได้ทำการทดลองส่งข้อมูลด้วยขนาดต่างกับกับโพรโทคอล MQTT และ CoAP เพื่อวัดผลการเกิดข้อมูลส่วนเกินดังรูปที่ 2.19 [4]



รูปที่ 2.19 ความสัมพันธ์ระหว่าง Overhead กับขนาดของแพ็คเก็ตในการทดสอบของ Dinesh Thangavel และคณะ [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การดำเนินงานวิจัย

ในบทที่ 2 ได้นำเสนองานวิจัยที่เกี่ยวข้องกับการทดสอบวัดประสิทธิภาพโปรแกรมไอโอทีตัวแปรทั้งที่ตั้งต้นที่ใช้ในการควบคุมการทดสอบและตัวแปรสำหรับวัดผล ต่อมาในบทนี้จะกล่าวถึงการดำเนินการวิจัยเพื่อพัฒนาชุดทดสอบวัดประสิทธิภาพโปรแกรมไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลองซึ่งบทนี้จะกล่าวถึง สมมุติฐานของงานวิจัย ผลการศึกษาในรูปแบบวิธีการทดสอบ เครื่องที่ใช้สนับสนุนการทดสอบที่ได้พบจากงานวิจัยที่เกี่ยวข้อง การออกแบบชุดทดสอบวัดประสิทธิภาพโปรแกรมไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลอง ชุดทดสอบที่พัฒนาแล้วเสร็จ และตัวอย่างการทดสอบวัดประสิทธิภาพโปรแกรมไอโอทีที่ใช้ชุดทดสอบที่พัฒนาขึ้น

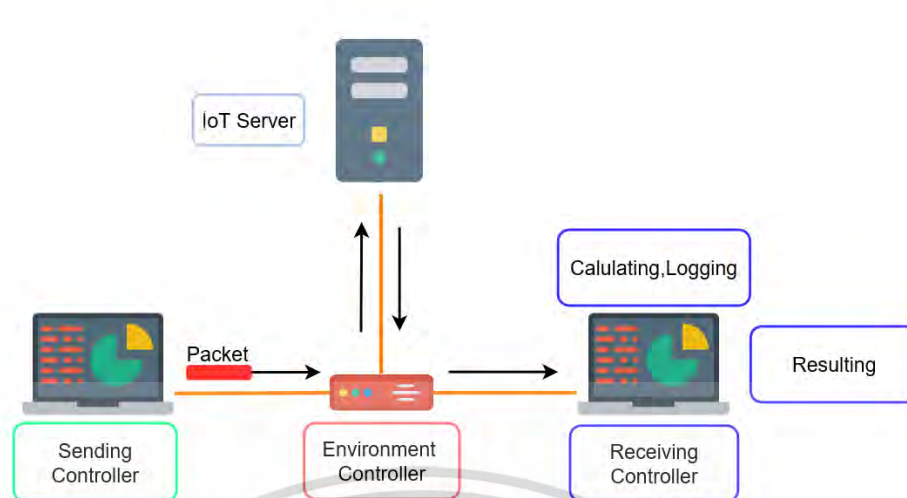
3.1 สมมุติฐานของงานวิจัย

1. ในการทดสอบวัดประสิทธิภาพโปรแกรมไอโอทีที่ใช้อุปกรณ์จริงหรืออุปกรณ์จำลองผู้ทดสอบต้องทำส่วนจัดการ การทดสอบบางส่วนด้วยตนเองเช่น จัดการการรับส่งข้อมูล ควบคุมสภาพแวดล้อม ค่าวนผลลัพธ์ เก็บผลลัพธ์ และการแสดงผลลัพธ์

2. ไม่มีเครื่องมือสำหรับวัดประสิทธิภาพโปรแกรมไอโอทีที่ทดลองโดยใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง ที่สนับสนุนทุกกระบวนการทดสอบแบบสำเร็จรูปพร้อมสามารถทดสอบได้แบบอัตโนมัติ

3.2 ผลการศึกษาการวัดประสิทธิภาพโปรแกรมไอโอทีที่ใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง

จากการศึกษาวิจัยที่เกี่ยวข้อง พบว่าการทดสอบประสิทธิภาพโปรแกรมไอโอทีที่ใช้ อุปกรณ์จริงหรืออุปกรณ์จำลอง มีวิธีการทดสอบโดยการใช้อุปกรณ์อย่างน้อย 3 ส่วน ประกอบด้วย 1.) อุปกรณ์ที่ทำหน้าเป็นตัวทดสอบซึ่งจะทำหน้าที่ ดำเนินการทดสอบ จัดการรับหรือส่งข้อมูล ควบคุมสภาพแวดล้อมเครือข่าย ค่าวนผลลัพธ์ จัดเก็บผลลัพธ์ 2.) อุปกรณ์ที่ทำหน้าที่เป็นแม่ข่าย และ 3.) อุปกรณ์ที่ทำหน้าที่สร้างเครือข่าย ขั้นตอนนี้เรียกว่าการเตรียมการ ถัดมาคือการทดสอบ โดยจากที่พบในงานวิจัยที่เกี่ยวข้องซึ่งผู้ทดสอบใช้การพัฒนาโปรแกรมเพื่อดำเนินการทดสอบ อธิบายได้โดยแผนภาพที่ 3.1



รูปที่ 3.1 Diagram model การวัดประสิทธิภาพ

จากแผนภาพที่ 3.1 แสดงให้เห็นถึงรูปแบบการทดสอบวัดประสิทธิภาพโพรโทคอลไอโอทีที่พบในงานวิจัยที่ผ่านมา ซึ่งจำกการวิเคราะห์และศึกษาสรุปได้ 5 ส่วนดังนี้

1.) อุปกรณ์และเทคโนโลยีเครือข่าย อุปกรณ์ประกอบด้วย 1) อุปกรณ์ลูกข่ายที่ทำหน้าที่รับส่งข้อมูลไปยังอุปกรณ์แม่ข่ายหรืออาจเรียกได้ว่าเป็นอุปกรณ์ทดสอบ (Tester Device) ซึ่งจะดำเนินการทดสอบต่าง ๆ ตลอดจนการคำนวณผลลัพธ์และการบันทึกผล 2) อุปกรณ์แม่ข่าย (Server Device) ในกระบวนการทดสอบ อุปกรณ์แม่ข่ายมีหน้าที่รับข้อมูลจากอุปกรณ์ลูกข่ายและส่งข้อมูลกลับหรือส่งต่อไปยังอุปกรณ์ลูกข่ายเพื่อคำนวณผลลัพธ์ 3) อุปกรณ์เครือข่ายซึ่งจะเป็นเทคโนโลยีเครือข่ายแบบสายหรือไร้สายขึ้นอยู่กับจุดประสงค์ของผู้ทดสอบ โดยที่อุปกรณ์ทั้งหมดสามารถเป็นได้ทั้งอุปกรณ์จริงหรืออุปกรณ์จำลอง

2.) ตัวควบคุมการรับส่งข้อมูล เป็นโปรแกรมที่ถูกพัฒนาขึ้นเพื่อทดสอบตามวัตถุประสงค์ ซึ่งแบ่งออกเป็นการส่งและการรับข้อมูล การส่งข้อมูลจะส่งด้วยโพรโทคอลไอโอทีไปยังอุปกรณ์แม่ข่าย โดยมีกำหนดตัวแปรตั้งต้นเกี่ยวกับการส่งข้อมูล อาทิเช่น โพรโทคอลไอโอทีที่เลือกใช้ กำหนดขนาดของข้อมูลที่ต้องการส่ง กำหนดความถี่ในการส่ง จำนวนครั้งในการส่ง ส่วนการรับข้อมูลซึ่งจะเป็นไอโอทีโพรโทคอลเดียวกัน เมื่อรับข้อมูลจะมีการนำไปคำนวณตัวแปรวัดผลต่าง ๆ อาทิเช่น ค่าเวลาที่แพ็คเก็ตเดินทางไปและกลับ แพ็คเก็ตสูญหาย

3.) แม่ข่ายไอโอที โพรโทคอลไอโอทีแต่ละตัวจะมีระบบจัดการแม่ข่ายที่แตกต่างกันซึ่งเลือกใช้ให้ตรงกับโพรโทคอลไอโอทีที่ใช้ทดสอบ

4.) ตัวควบคุมสภาพแวดล้อม ในการทดสอบโพรโทคอลไอโอที มีการควบคุมสภาพแวดล้อม เพื่อดูผลลัพธ์ต่าง ๆ อาทิเช่น การควบคุมอัตราการสูญหายของข้อมูล (Loss rate) ควบคุมแบนด์วิดท์ (Bandwidth) ของเครือข่าย

5.) ส่วนเก็บบันทึกผลลัพธ์ ในแต่ละงานวิจัยไม่ว่าจะเป็นการทดสอบประสิทธิภาพหรือการเปรียบเทียบไอโอทีโพรโทคอล ผลลัพธ์และการเก็บผลลัพธ์เป็นไปตามที่ผู้ทดสอบกำหนดหรือต้องการ นำเสนอ ซึ่งวิธีการบันทึกผลไม่ถูกระบุแน่ชัดว่าเป็นการเก็บแบบใด ซึ่งขึ้นอยู่กับกับความสะดวกของผู้ทดสอบ อาทิเช่น การเก็บผลในรูปแบบของไฟล์ text ธรรมดา หรืออาจจะบันทึกลงฐานข้อมูล

3.3 การสนับสนุนของเครื่องมือเพื่อทำการทดสอบ

จากหัวข้อที่ผ่านมาได้กล่าวถึงรูปแบบและวิธีการทดสอบวัดประสิทธิภาพของโพรโทคอลไอโอที จากการศึกษาข้างงานวิจัยที่เกี่ยวข้องเป็นที่เรียบร้อยแล้ว ในหัวข้อนี้จะกล่าวถึงเครื่องมือหรือวิธีการที่สนับสนุนการทำงานวิจัยด้านนี้ โดยมีหัวข้อย่อยดังต่อไปนี้ 1.) การนำโพรโทคอลไอโอที มาทดสอบ 2.) เครื่องมือควบคุมการรับส่งข้อมูล 3.) เครื่องมือควบคุมสภาพแวดล้อม 4.) เครื่องมือเก็บผล 5.) เครื่องมือแสดงผล

3.3.1 การนำโพรโทคอลไอโอทีมาทดสอบ

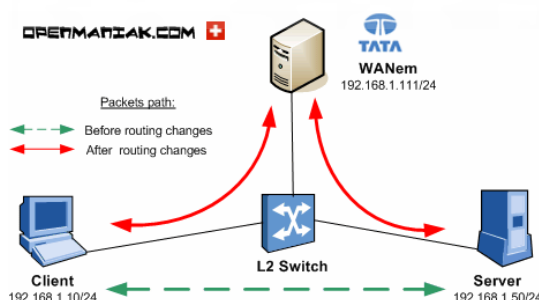
โพรโทคอลไอโอทีในปัจจุบันมีอยู่ด้วยกันหลายตัว ยกตัวอย่างโพรโทคอลที่ได้รับความนิยมเช่น MQTT CoAP AMQP MQTT-SN ซึ่งแต่ละตัวมีความแตกต่างกัน และมีข้อดีข้อเสียในการนำไปประยุกต์ใช้เป็นงานที่ต่างกันออกไป การนำมาใช้หรือนำมาทดสอบนั้น จากการศึกษาข้างงานวิจัยที่เกี่ยวข้องพบว่าส่วนมากโพรโทคอลไอโอทีนั้นมีการสนับสนุน API หรือ Library ให้นักพัฒนาเลือกใช้กันได้หลายภาษา แต่ในปัจจุบันนิยมมากที่สุดคือ ภาษา Python [8] เนื่องจากตัวภาษามีความซับซ้อนน้อย สะดวกต่อการพัฒนา ในการนำโพรโทคอลไอโอทีมาทดสอบซึ่งจะประกอบด้วยส่วนที่เป็นโปรแกรมลูกข่าย อีกส่วนหนึ่งคือโปรแกรมแม่ข่าย โปรแกรมลูกข่ายผู้ทดสอบจำเป็นต้องนำ API หรือ Library ของโพรโทคอลไอโอทีมาพัฒนาโปรแกรมเพื่อดำเนินการทดสอบตามที่ต้องการด้วยตนเอง ส่วนโปรแกรมแม่ข่าย บางโพรโทคอลไอโอทีจะมีโปรแกรมแม่ข่ายที่พร้อมใช้งานไว้ให้เช่น MQTT จะมีโปรแกรม Mosquitto [9] ซึ่งใช้เป็น MQTT Broker สามารถทำงานได้ทันที เพียงแค่เรียกใช้ชุดคำสั่ง แต่บางโพรโทคอลไอโอทีเช่น CoAP จะไม่มีโปรแกรมสำเร็จรูปที่เป็นชุดคำสั่งแต่จะเป็นโปรแกรมที่เป็นภาษา Python เช่น CoAPthon [10] และในภาษาซี ซึ่งผู้ทดสอบจะต้องพัฒนาโปรแกรมขึ้นมาด้วยตนเอง

3.3.2 เครื่องมือควบคุมการรับส่งข้อมูล

จากการศึกษางานวิจัยที่เกี่ยวข้อง การทดสอบโพรโทคอลไอโอทีที่มีการกำหนดตัวแปรตั้งต้นที่เกี่ยวข้องกับการส่งข้อมูลอยู่ด้วยกัน 4 ตัวคือ ขนาดของแพ็คเกจ ความถี่การส่ง บริการคุณภาพของการส่งและจำนวนครั้งในการส่ง การควบคุมการส่งข้อมูลจากการศึกษางานวิจัยที่เกี่ยวข้องผู้วิจัยต้องจัดทำโปรแกรมสำหรับดำเนินการควบคุมการส่งโพรโทคอลไอโอทีที่ทำทดลองนั้นด้วยตัวเอง การส่งส่งแพ็คเกจที่มีการนำข้อมูลอ้างอิงบางอย่างลงไปด้วยเพื่อใช้ในการคำนวณตัวแปรวัดผลอาชีเช่น การใส่หมายเลขแพ็คเกจเพื่อใช้คำนวณ ค่าเวลาในการส่งแพ็คเกจไป-กลับ(RTT) หรือการวัดแพ็คเกจสูญหาย ในส่วนของการรับข้อมูลผู้วิจัยจะต้องพัฒนาโปรแกรมสำหรับดำเนินการเช่นเดียวกัน ซึ่งในส่วนนี้จะรวมถึงการคำนวณผลลัพธ์ของตัวแปรวัดผลต่าง ๆ และการเก็บบันทึกข้อมูลอีกด้วย ยังพบอีกว่า ผู้วิจัยนิยมนำโปรแกรมส่งและโปรแกรมรับข้อมูลดำเนินการไว้ในเครื่องเดียวกัน เพื่อป้องกันปัญหาเวลาของอุปกรณ์ไม่ตรงการ สรุปคือยังไม่มีเครื่องมือใด ๆ สนับสนุนการควบคุมการรับส่งข้อมูลสำหรับทดสอบโพรโทคอลไอโอทีแบบใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง

3.3.3 เครื่องมือควบคุมสภาพแวดล้อม

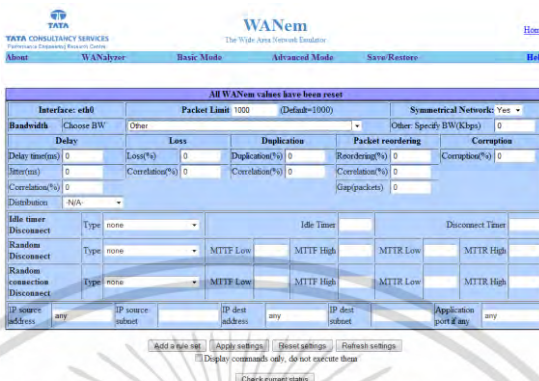
จากการศึกษางานวิจัยที่เกี่ยวข้อง การทดสอบโพรโทคอลไอโอทีจะมีการกำหนดตัวแปรตั้งต้นที่เกี่ยวข้องสภาพแวดล้อมเช่นเดียวกันกับการส่งข้อมูลซึ่งจากการศึกษางานวิจัยที่ผ่านมา มีด้วยกัน 3 ค่าดังนี้ 1) อัตราการสูญหายของข้อมูล 2) ค่าเวลาแฝงในการรับ-ส่งข้อมูล 3) ปริมาณการรับ-ส่งข้อมูลของการ์ดเครือข่าย หรือ Bandwidth ซึ่งการควบคุมตัวแปรเหล่านี้พบการใช้เครื่องมือชื่อ NetEM[12] และ WANem[13] ซึ่ง NetEM เป็นชุดคำสั่งที่ทำงานเป็นระบบปฏิบัติการลินุกซ์ซึ่งมีชุดคำสั่งย่อยที่ควบคุมสภาพแวดล้อมของเครือข่ายเช่น อัตราการสูญหายของข้อมูล ค่าเวลาแฝงในการรับ-ส่งข้อมูล ปริมาณการรับ-ส่งข้อมูลของการ์ดเครือข่าย การสร้างแพ็คเกจซ้ำซ้อน การเปลี่ยนแปลงข้อมูลในแพ็คเกจ NetEM นั้นสามารถติดตั้งไว้กับเครื่องแม่ข่ายหรือลูกข่ายได้โดยไม่ต้องใช้อุปกรณ์เพิ่ม ส่วน WANEM จำเป็นต้องใช้คอมพิวเตอร์เพิ่มจำนวน 1 เครื่องเพื่อทำหน้าที่เป็นเกตเวย์ให้ข้อมูลถูกส่งผ่าน WANEM การเชื่อมต่อในการใช้งานสำหรับ WANem แสดงดังรูปที่ 3.2[13]



รูปที่ 3.2 การเชื่อมต่อ WANem[13]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบของ WANem เป็นลินุกซ์และถูกจัดทำอยู่ในรูปของ image file สามารถทำงานได้ทันทีเมื่อเริ่มต้นทำงานจาก image file โดยที่ตัวเครื่องไม่จำเป็นต้องมีระบบปฏิบัติการ มีส่วนติดต่อกับผู้ทดสอบเป็น Web application ดังรูปที่ 3.3 [13]



รูปที่ 3.3 ส่วนตั้งค่าของ WANem [13]

3.3.4 เครื่องมือเก็บผล

การเก็บผลในระบบการทดลองหรือทดสอบเป็นอีกหนึ่งส่วนที่ผู้ทดสอบจำเป็นต้องสร้างขึ้นเพื่อให้อ่านกับการทดลอง แต่การเก็บผลถูกกล่าวถึงน้อยมาก แต่ในงานวิจัย Correlation analysis of MQTT Loss and Delay According to QoS Level [3] มีการกล่าวถึงการเก็บผลการทดลองอยู่ในรูปของ text file ซึ่งสามารถนำไปวิเคราะห์ต่อไป

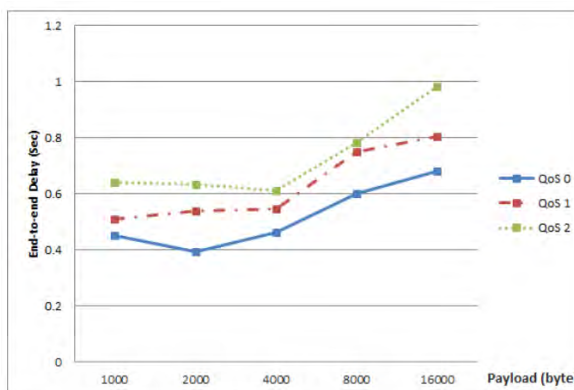
3.3.5 เครื่องมือแสดงผลลัพธ์

การแสดงผลลัพธ์ คือ การนำข้อมูลมาแสดงอยู่ในรูปแบบของ ตาราง หรือ แผนภูมิ ซึ่งจะมีด้วยกันหลายรูปแบบการแสดงผลอย่างเหมาะสมสามารถช่วยให้การวิเคราะห์ผลทำได้สะดวกและรวดเร็วตัวอย่างการแสดงผลลัพธ์ที่พบในงานวิจัยที่เกี่ยวข้อง

	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 3.4 ตัวอย่างการแสดงผลลัพธ์แบบตาราง [4]

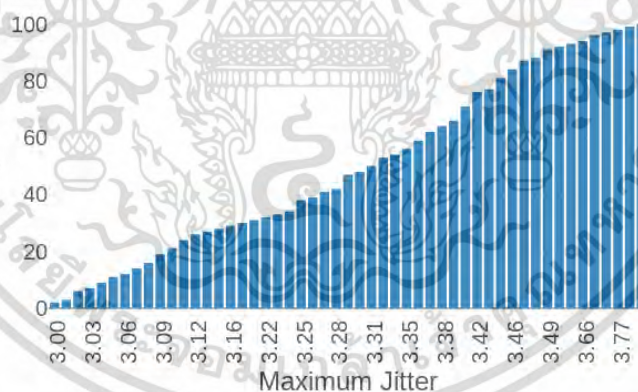
จากรูปที่ 3.4 คือการแสดงผลลัพธ์ความสัมพันธ์ระหว่าง ค่า Round trip time ที่ใช้ในการส่งข้อมูล เมื่อมีสภาพแวดล้อมการสูญหายของข้อมูลต่าง ๆ ของโพรโทคอล MQTT และ CoAP เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานาน นีเมอนุญาตหนาไปเซประเษนคานการคําไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ตัวอย่างการแสดงผลรูปแบบแผนภูมิแบบเส้น [3]

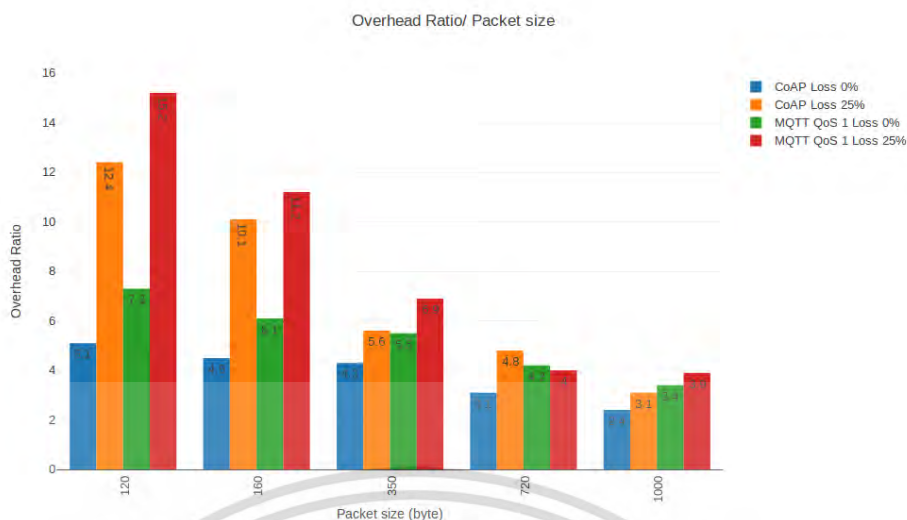
จากรูปที่ 3.5 แสดงผลลัพธ์ความสัมพันธ์ระหว่าง ค่า Round trip time ที่ใช้ในการส่งข้อมูลของโพรโทคอล MQTT เมื่อกำหนดให้ระดับ QoS แตกต่างกันและขนาดของแพ็คเก็ตที่ส่งต่างกันซึ่งการแสดงผลด้วยแผนภูมิแบบเส้นสื่อความหมายว่าเป็น QoS ระดับเดียวกัน

Jitter Cumulative Histogram (MQTT 10ms 512)



รูปที่ 3.6 ตัวอย่างการแสดงผลรูปแบบแผนภูมิแบบฮิสโตแกรมสะสม

จากรูปที่ 3.6 การแสดงผลด้วยแผนภูมิแบบฮิสโตแกรมสะสม ซึ่งแผนภูมิประเภทนี้จะแสดงให้เห็นถึงการเกาะกลุ่มของข้อมูล จากตัวอย่างแสดงถึงค่าเวลาหน่วงไม่คงที่ (Jitter) ของส่งแพ็คเก็ตโดยโพรโทคอล MQTT ในเครือข่ายมีอัตราค่าการสูญหายของแพ็คเก็ต แสดงให้เห็นว่า 80% ค่าเวลาหน่วงไม่คงที่ (Jitter) อยู่ระหว่าง 3.00 – 3.40 วินาที



รูปที่ 3.7 ตัวอย่างการแสดงผลลัพธ์แบบแผนภูมิแบบแท่ง

จากรูปที่ 3.7 การแสดงผลด้วยแผนภูมิแบบแท่งซึ่งจากรูปตัวอย่างแสดงถึงความสัมพันธ์ระหว่างข้อมูลส่วนเกินกับขนาดของแพ็คเกจที่ส่ง

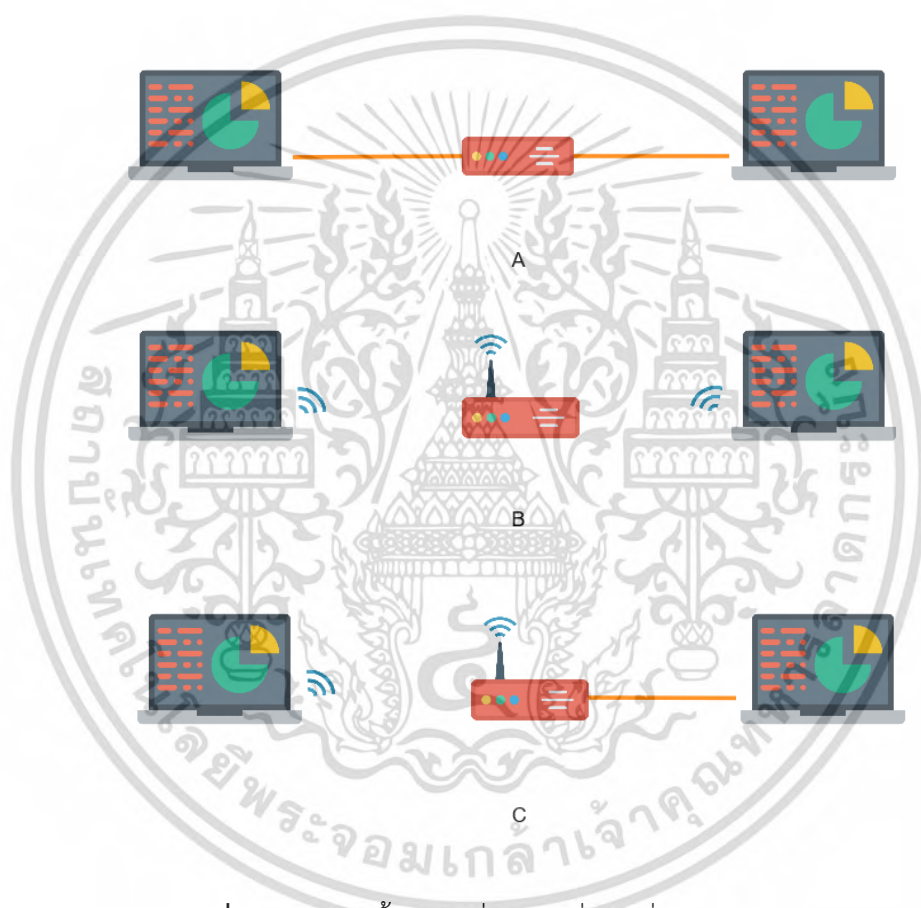
3.4 แนวคิดการแก้ไขปัญหา

จากปัญหาที่ได้อธิบายไปแล้วในบทที่ 1 และได้ขยายความจากการศึกษาวิจัยที่เกี่ยวข้องในบทที่ 3 ซึ่งแสดงให้เห็นว่าในการทดสอบวัดประสิทธิภาพโพรโทคอลไอโอทีโดยการทดลองแบบใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง ไม่มีเครื่องชุดที่สนับสนุนการทดลองโดยตรง อีกทั้งผู้ทดสอบต้องจัดทำตัวควบคุมการทดลองบางส่วนขึ้นมาเอง ดังนั้นจึงขอเสนอแนวคิดว่าแก้ไขปัญหาคือ จัดทำชุดทดลองหรือทดสอบสำหรับวัดประสิทธิภาพโพรโทคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลอง ชุดเครื่องมือจะช่วยให้ผู้ทดสอบลดขั้นตอนการเตรียมการ การจัดทำตัวควบคุมการส่ง ตัวควบคุมสภาพแวดล้อม ตัวจัดการการเก็บผลลัพธ์ และตัวจัดการการแสดงผล พร้อมสามารถทำการทดสอบแบบอัตโนมัติ

3.5 การออกแบบระบบ

จากการศึกษาวิธีการทดสอบโพรโทคอลไอโอทีและการสนับสนุนจากเครื่องมือซึ่งจะเห็นได้ว่าผู้ทดสอบจะทำการพัฒนาส่วนต่าง ๆ ขึ้นมาเองเป็นส่วนใหญ่ ซึ่งในการออกแบบจะได้นำส่วนการควบคุมต่าง ๆ จัดให้อยู่ในชุดโปรแกรมที่ผู้ทดสอบหรือสามารถใช้งานได้ง่ายและลดระยะเวลาในการเตรียมการและดำเนินการโดยการตัดส่วนที่ผู้ทดสอบต้องพัฒนาขึ้นเองออก ซึ่งมีส่วนสำคัญต่าง ๆ แบ่งออกเป็น 7 ส่วนดังนี้ 1.) การจัดเตรียมอุปกรณ์ในการทดลอง 2.) การนำไปทดสอบโพรโทคอลไอโอทีมาทดสอบ 3.) ตัวควบคุมการส่งข้อมูล 4.) แพ็คเกจ 5.) การควบคุมสภาพแวดล้อม 6.) ตัวรับข้อมูลและเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างเครือข่ายแบบ A ประกอบด้วยอุปกรณ์เครือข่าย 1 ตัว อุปกรณ์ทดสอบซึ่งทำหน้าที่รับส่งข้อมูลในตัวเดียวกัน 1 ตัวและอุปกรณ์แม่ข่าย 1 ตัว โครงสร้างเครือข่ายแบบ B แตกต่างจากแบบ A คือมีอุปกรณ์ทดสอบแยกตัวส่งกับตัวรับ อุปกรณ์ที่ใช้สำหรับดำเนินการทดสอบรับส่งข้อมูลจำเป็นต้องเป็นแพลตฟอร์มที่รองรับการทำงานของโปรแกรม Python เนื่องจากชุดทดสอบถูกพัฒนาจากภาษา python เป็นหลัก อุปกรณ์ที่รองรับในการใช้งาน อาทิเช่น เครื่องคอมพิวเตอร์แบบตั้งโต๊ะ แบบพกพา หรือ คอมพิวเตอร์แบบสมองกลฝังตัว หรือเป็นโปรแกรมประยุกต์สร้างอุปกรณ์จำลอง อาทิเช่น VMware [14], VisualBox [15], QEMU [16] เทคโนโลยีเครือข่ายสามารถเลือกใช้ได้ทั้งแบบสาย ไร้สายหรือแบบผสม ดังรูปที่ 3.10



รูปที่ 3.10 โครงสร้างเครือข่ายแบบต่างๆ ที่ชุดทดสอบรองรับ

3.5.2 การนำไปทดลองโอโอทีมาทดสอบ

จากวิธีดั้งเดิมที่พบในงานวิจัยที่เกี่ยวข้อง ผู้ทดสอบจะนำ Library ของโพรโทคอลโอโอทีมาเขียนโปรแกรมสำหรับดำเนินการทดสอบด้วยตัวเอง ซึ่งปัจจุบันโพรโทคอลโอโอทีส่วนมากจะมีการรองรับภาษา Python อาทิเช่น MQTT CoAP AMQP และอื่น ๆ แต่สำหรับชุดทดสอบที่พัฒนานี้จะมีการออกแบบให้ผู้ทดสอบสามารถเลือกทดสอบโพรโทคอลโอโอที MQTT MQTT-SN CoAP และ AMQP ได้ทันทีโดยไม่ต้องเขียนโปรแกรมเพิ่มเติม โดยชุดทดสอบจะมีโปรแกรมประกอบด้วยกัน 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนคือ 1.) โปรแกรมดำเนินการภาคส่ง 2.) โปรแกรมดำเนินการภาครับ 3.) โปรแกรมดำเนินการแม่ข่าย นอกจากนี้ผู้ที่ทดสอบสามารถเลือกโพรโทคอลไอโอทีโดยที่ผู้ทดสอบไม่ต้องเขียนโปรแกรมด้วยตัวเองแล้ว หากผู้ทดสอบต้องการทดสอบโพรโทคอลไอโอทีที่นอกเหนือจากโพรโทคอลดังกล่าวหรือต้องการพัฒนาขึ้นเอง ชุดทดสอบจะมีไฟล์ โปรแกรมดำเนินการภาคส่ง โปรแกรมดำเนินการภาครับ และโปรแกรมดำเนินการแม่ข่ายสำหรับผู้ทดสอบได้พัฒนาเพิ่มเติม โดยในการใช้งานโปรแกรมทั้ง 3 ส่วนนี้จะต้องถูกติดตั้งไว้ดังนี้ โปรแกรมดำเนินการภาคส่งและภาครับจะต้องติดตั้งไว้อุปกรณ์ประมวลผลสามารถใช้อุปกรณ์เดียวกันหรือแยกกันตามแต่สะดวก โปรแกรมดำเนินการแม่ข่ายจะต้องติดตั้งแยกต่างหากอีกหนึ่งอุปกรณ์

3.5.3 การควบคุมการรับส่งข้อมูล

การควบคุมการส่งข้อมูลจะอยู่ในส่วนของโปรแกรมภาคส่ง โดยจะมีการควบคุม ความถี่ในการส่ง ขนาดข้อมูล คุณภาพบริการ (แตกต่างกันไปตาม โพรโทคอล) จำนวนครั้งในการส่ง (Loop) พร้อมดำเนินการใส่เวลาที่แพ็คเก็ตถูกส่งและหมายเลขประจำแพ็คเก็ตเพื่อใช้ในการคำนวณตัวแปรวัดผลซึ่งจะอธิบายในหัวข้อ ตัวรับข้อมูล ประมวลผลและการเก็บผล การกำหนดค่าตัวแปรตั้งต้นเกี่ยวกับการส่งสามารถใส่ได้มากกว่าหนึ่งค่า ทำให้ผู้ทดสอบสามารถตั้งค่า กรณีทั้งหมดที่ต้องการทดสอบ ชุดทดสอบจะทำการดำเนินการทดสอบให้ทุกกรณีที่กำหนดไว้โดยจำนวนกรณีเกี่ยวกับการส่งข้อมูลสามารถคำนวณได้ดังสมการที่ 3.1

$$S_{case} = (pks_{case})(qos_{case})(pr_{case}) \quad (3.1)$$

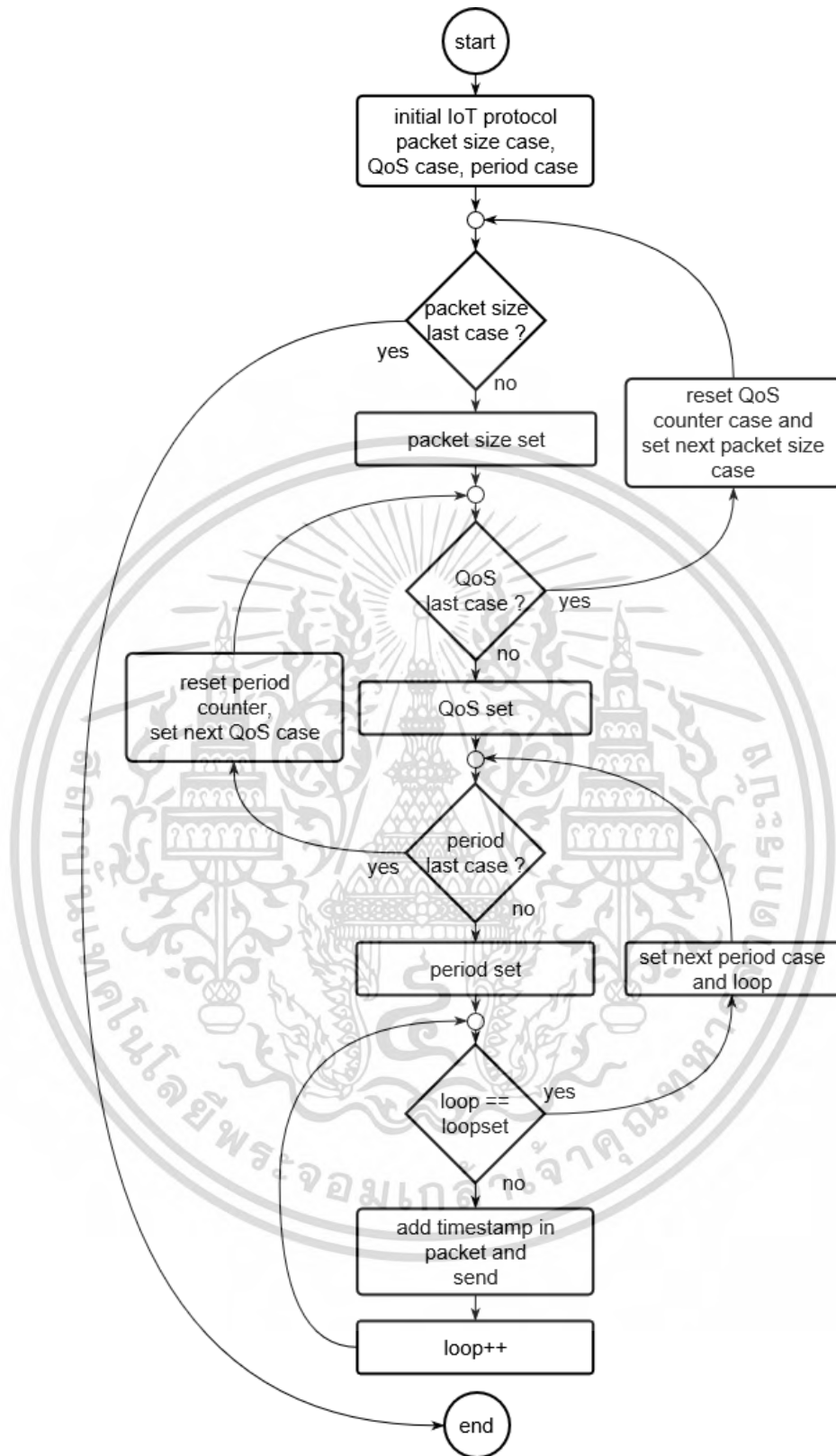
S_{case} = Total case of Sending control

pks_{case} = Total case of packet size

qos_{case} = Total case of QoS

pr_{case} = Total case of period sending

จากสมการแสดงการคำนวณถึงจำนวนกรณีทั้งหมดของที่ควบคุมด้วยการส่งข้อมูลจะเห็นว่า ยังไม่มีตัวแปรจำนวนครั้งของการส่ง ซึ่งตัวแปรนี้ถูกในการกำหนดจำนวนครั้งในการส่งแพ็คเก็ตในแต่ละกรณีนั่นเองโดยได้ออกแบบการทำงานของการควบคุมการส่งข้อมูลดังรูปโฟลว์ชาร์ทที่ 3.11



รูปที่ 3.11 โฟลว์ชาร์ทการทำงานการควบคุมการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 แพ็คเก็ต

ในชุดการทดสอบออกแบบให้ผู้ทดสอบสามารถที่จะเลือกขนาดของข้อมูลที่รับส่งได้ อีกทั้งในแพ็คเก็ตจะต้องประกอบด้วยส่วนระบุหมายเลขแพ็คเก็ต เพื่อให้สามารถตรวจสอบได้ว่าแพ็คเก็ตใดสูญหายเมื่อถูกส่งออกไปแล้วไม่ได้รับกลับมา โดยส่วนระบุแพ็คเก็ตจะมีค่าสูงสุดเท่ากับจำนวนครั้งในการส่งข้อมูลที่ผู้ทดสอบกำหนดไว้ อีกส่วนหนึ่งคือเวลาที่ถูกรับส่งออกไป เพื่อใช้ในการคำนวณค่าเวลาเดินทางไป-กลับของแพ็คเก็ต (RTT) และค่าเวลาหน่วงไม่คงที่ (jitter) เมื่อแพ็คเก็ตไปถึงภาครับข้อมูล

ID	Time stamp				pk size		QoS	IP	Loop		Loss rate		Latency	Bandwidth					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...	N

รูปที่ 3.12 แสดงข้อมูลที่ถูกละไว้ในแพ็คเก็ต

จากรูปที่ 3.12 แสดงถึงวิธีการสร้างแพ็คเก็ต โดยในแพ็คเก็ตจะเริ่มต้นจากการสร้าง ไบต์ array ตามขนาดที่ผู้ทดสอบกำหนด (ไบต์) จากนั้นจะแทรกส่วนระบุแพ็คเก็ตลงในตำแหน่งที่ 0 และ 1 เป็นตัวเลขจำนวนเต็ม 2 ไบต์ เวลาที่ส่ง 4 ไบต์ ขนาดของแพ็คเก็ต 2 ไบต์ ค่าคุณภาพบริการ 1 ไบต์ ค่าความถี่ในการส่ง 2 ไบต์ ค่าจำนวนครั้งการส่งต่อหนึ่งกรณี 2 ไบต์ ควบคุมอัตราการสูญหายของข้อมูลในเครือข่าย (loss rate) 2 ไบต์ ควบคุมค่าความล่าช้าแฝงในเครือข่าย (latency time) 2 ไบต์ และควบคุมความกว้างการรับส่งเครือข่าย (bandwidth) 2 ไบต์ ฉะนั้นขนาดแพ็คเก็ตที่น้อยที่สุดที่กำหนดได้คือ 19 ไบต์

3.5.5 การควบคุมสภาพแวดล้อม

ชุดทดสอบออกแบบให้ผู้ทดสอบสามารถควบคุมสภาพแวดล้อม โดยจะถูกตั้งค่าไว้ยังโปรแกรมภาคส่งประกอบด้วย ควบคุมอัตราการสูญหายของข้อมูลในเครือข่าย (loss rate) ควบคุมค่าความล่าช้าแฝงในเครือข่าย (latency time) และควบคุมความกว้างการรับส่งเครือข่าย (bandwidth) โดยชุดทดสอบได้นำเครื่องควบคุมสภาพแวดล้อมเครือข่าย NetEM มาร่วมใช้งาน ค่าตัวแปรตั้งต้นสภาพแวดล้อมผู้ทดสอบสามารถกำหนดค่าตัวแปรเหล่านี้ได้มากกว่าหนึ่งค่า ระบบจะสามารถดำเนินการทดสอบได้ในทุก ๆ กรณีที่กำหนดโดยอัตโนมัติ โดยจำนวนกรณีสภาพแวดล้อมทั้งหมดแสดงได้ดังสมการที่ 3.2

$$En_{case} = (Lr_{case})(Lc_{case})(Bw_{case}) \quad (3.2)$$

En_{case} = Total case of Environment control

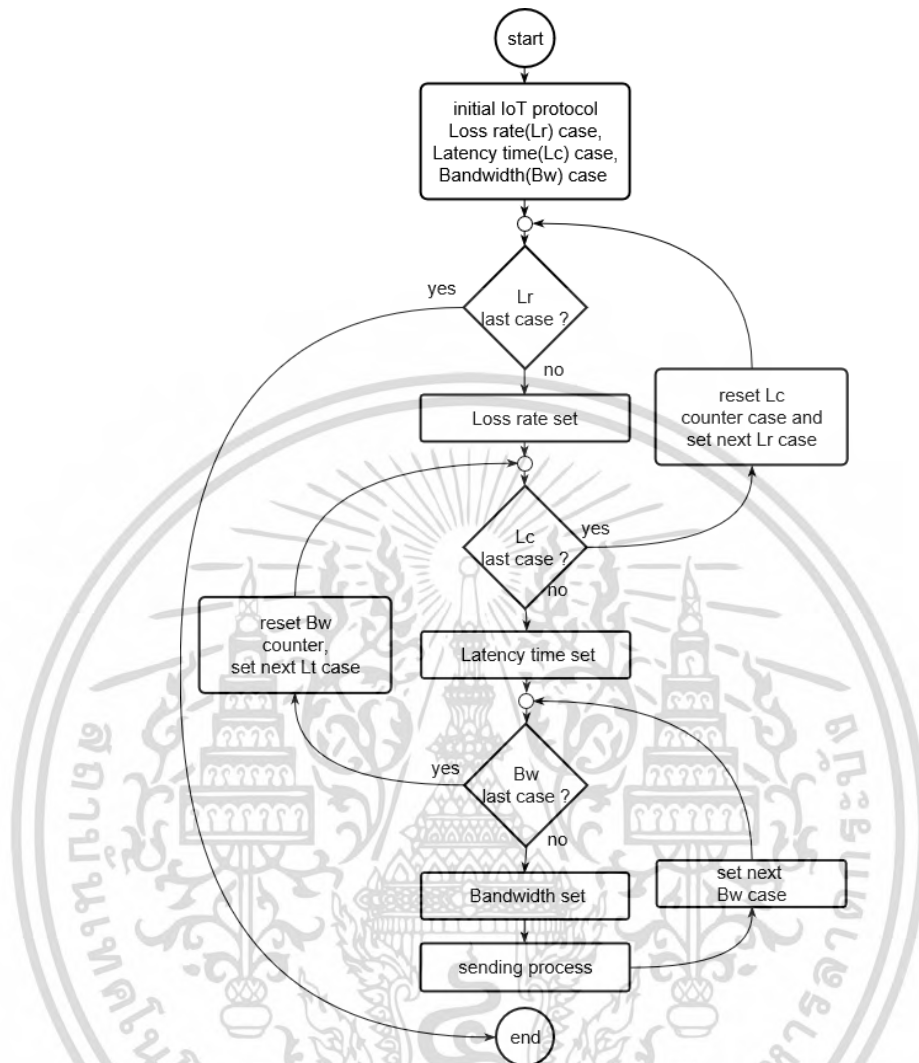
Lr_{case} = Total case of Loss rate

Lc_{case} = Total case of latency time

Bw_{case} = Total case of bandwidth

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบส่วนควบคุมสภาพแวดล้อมแสดงดังรูปโฟลว์ชาร์ทที่ 3.13



รูปที่ 3.13 โฟลว์ชาร์ทแสดงการทำงานส่วนควบคุมสภาพแวดล้อม

3.5.6 ตัวรับข้อมูล ประมวลผลและการบันทึกผลลัพธ์

ตัวรับข้อมูลจะอยู่ในโปรแกรมภาครับ ซึ่งจะมีหน้ารับข้อมูลจากที่ส่งมาจากตัวส่งข้อมูล จากนั้นประมวลผลและเก็บผลลัพธ์ การคำนวณค่าตัวแปรวัดผลต่าง ๆ ซึ่งประกอบด้วย เวลาที่ใช้ในการรับส่งแพ็คเก็ตไป-กลับ (Round trip time : RTT) อัตราการสูญหายของแพ็คเก็ตข้อมูล (Loss rate) ค่าเวลาหน่วงไม่คงที่ (Jitter) และ ข้อมูลส่วนเกิน (Overhead)

การเก็บผลการทดสอบ ในแต่ละครั้งที่รับส่งแพ็คเก็ตจะบันทึกผล ค่าเวลารับส่งไป-กลับ (RTT) ค่าดีเลย์ที่ไม่คงที่ (Jitter) และค่าตัวแปรตั้งต้น ขนาดของแพ็คเก็ต ความถี่ในการส่งข้อมูล ระดับคุณภาพบริการ จำนวนรอบในการทดลอง อัตราการสูญหายของข้อมูล ค่าความล่าช้าแฝงในเครือข่าย ค่าความกว้างการรับส่งเครือข่าย และยังรวมถึงชื่อของโปรโตคอลที่ผู้ทดสอบได้กำหนดไว้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

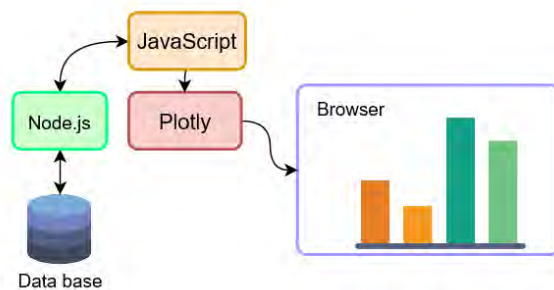
โดยข้อมูลที่เก็บมีลักษณะรูปแบบโครงสร้างคงที่ ดังนั้นการเลือกใช้การเก็บข้อมูลด้วยฐานข้อมูล SQL จึงเหมาะสมในการนำมาใช้เก็บข้อมูล จึงเลือกใช้ฐานข้อมูล MySQL [18] ซึ่งได้รับความนิยมมากอย่างมาก การทํางานส่วนการบันทึกผลดังแสดงให้เห็นในรูปโฟลว์ชาร์ทที่ 3.14



รูปที่ 3.14 โฟลว์ชาร์ทกำกรับข้อมูล

3.5.7 การแสดงผลลัพธ์

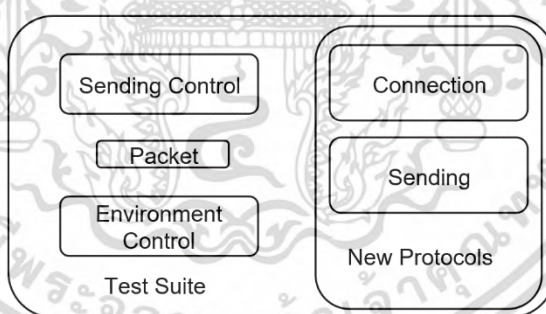
ส่วนการแสดงผลลัพธ์ เลือกใช้การแสดงผลด้วย web application โดยใช้ plotly [19] ตัวควบคุมการแสดงผลกราฟฟิค ระบบจะดึงข้อมูลผลลัพธ์ที่ได้จากการทดลองจากฐานข้อมูลด้วยโปรแกรมที่พัฒนาขึ้นด้วย Node js [20] ซึ่งจะอยู่ในรูปแบบของกราฟ โดยผู้ทดสอบสามารถเลือกแสดงความสัมพันธ์ระหว่างตัวแปรตั้งต้นและตัวแปรวัดผลที่ต้องการ และสามารถแสดงผลกราฟได้ทั้งกราฟเส้นและกราฟแท่ง และกราฟแบบฮิสโทแกรมค่าสะสม รูปที่ 3.15 แสดงถึงส่วนการทํางานของส่วนแสดงผลลัพธ์



รูปที่ 3.15 ไดอะแกรมการทำงานของส่วนแสดงผลลัพธ์

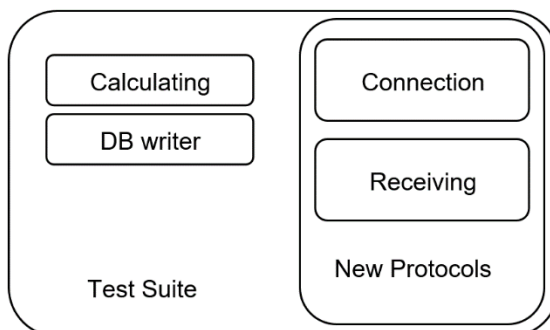
3.5.8 ส่วนการนำเข้าโปรโทคอลไอโอทีด้วยมือ

แม้ว่าชุดทดสอบวัดประสิทธิภาพโปรโทคอลไอโอทีจะมีโปรโทคอลไอโอทีให้ผู้ใช้สามารถเลือกทดสอบได้ทันทีโดยไม่ต้องพัฒนาโปรแกรมใด ๆ เพิ่มเติม แต่หากผู้ทดสอบมีความต้องการที่จะทดสอบโปรโทคอลไอโอทีนอกเหนือจากที่มีอยู่ ผู้ทดสอบสามารถนำเข้าโปรโทคอลไอโอทีที่ต้องการพัฒนารวมเข้ากันกับชุดทดสอบได้ โดยชุดทดสอบจะมีไฟล์โปรแกรม Python ที่เป็นมีเฉพาะส่วนการดำเนินการทดสอบต่าง ๆ แต่ยังไม่ได้นำเข้าโปรโทคอลไอโอทีใด ๆ ซึ่งโครงสร้างจะถูกแบ่งออกเป็น 3 ส่วนหลัก ๆ คือ ตัวดำเนินการส่งข้อมูล ตัวดำเนินการรับข้อมูล และตัวดำเนินการแม่ข่ายไอโอที (ถ้าต้องใช้) โดยแต่ละส่วนมีดังต่อไปนี้



รูปที่ 3.16 การนำเข้าโปรโทคอลกับส่วนตัวดำเนินการส่งข้อมูล

ส่วนที่ 1 การนำเข้าโปรโทคอลใหม่เพื่อใช้กับชุดทดสอบ รูปที่ 3.16 ในส่วนตัวดำเนินการส่งข้อมูล ผู้ทดสอบจะต้องพัฒนาส่วนการเชื่อมต่อและส่วนการส่งข้อมูลของโปรโทคอลใหม่เพื่อให้ชุดทดสอบสามารถควบคุมการเชื่อมต่อและสามารถส่งแพ็คเกจที่ใช้ในการทดสอบผ่านโปรโทคอลใหม่นี้ได้ ส่วนการควบคุมสภาพแวดล้อมเครือข่ายชุดทดสอบจะจัดการให้ตามที่ผู้ตั้งค่าเช่นเดิม



รูปที่ 3.17 การนำเข้าโปรโตคอลกับส่วนตัวดำเนินการรับข้อมูล

ส่วนที่ 2 การนำเข้าโปรโตคอลใหม่เพื่อใช้กับชุดทดสอบ รูปที่ 3.17 ในส่วนตัวดำเนินการรับข้อมูล ผู้ทดสอบจะต้องพัฒนาส่วนการเชื่อมต่อและส่วนการรับข้อมูลของโปรโตคอลใหม่ เพื่อให้ชุดทดสอบสามารถนำแพ็คเกจทดสอบที่ส่งมาจากตัวส่งข้อมูล เพื่อที่จะนำไปคำนวณตัวแปรวัดผล และบันทึกลงฐานข้อมูลของชุดทดสอบต่อไป ซึ่งส่วนการคำนวณตัวแปรวัดผลและการบันทึกลงฐานข้อมูลชุดทดสอบจะจัดการให้เช่นเดิม

ส่วนที่ 3 ตัวดำเนินการแม่ข่ายไอโอทีสำหรับโปรโตคอลไอโอทีที่จำเป็นต้องมีแม่ข่ายผู้ทดสอบสามารถพัฒนาแยกต่างหากได้ โดยจะต้องตั้งค่าในส่วนของการส่งข้อมูลและส่วนการรับข้อมูลให้เชื่อมต่อไปยังแม่ข่ายไอโอทีของผู้ทดสอบโดยตรง

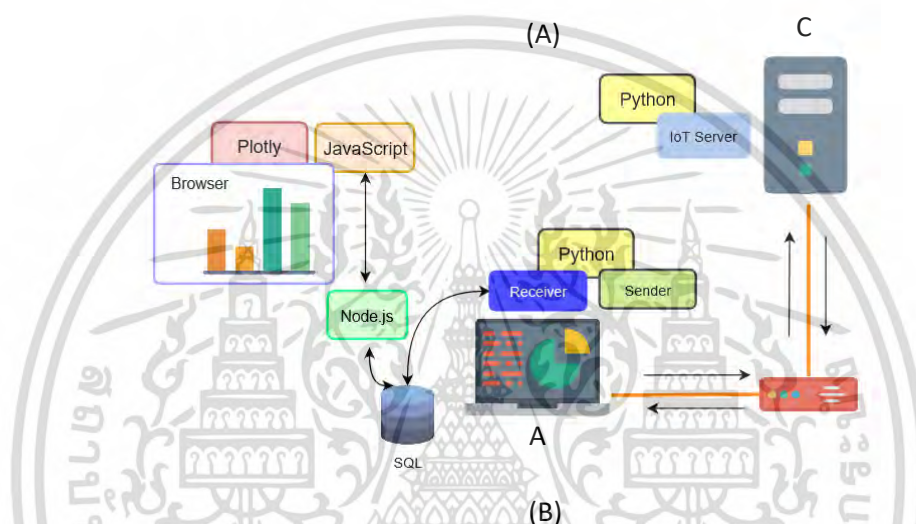
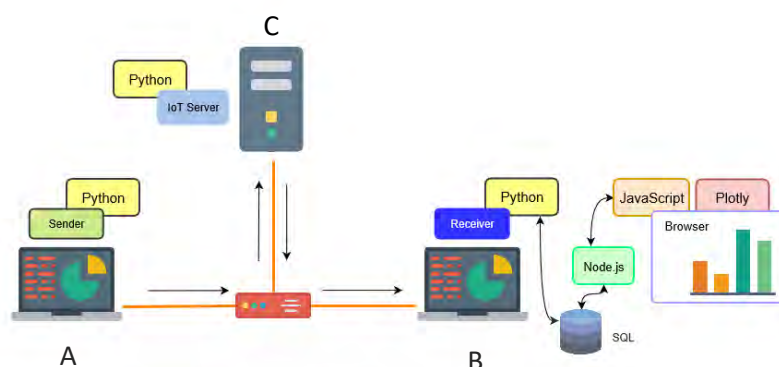
3.6 ชุดทดลองที่พัฒนาเสร็จสิ้น

จากหัวข้อที่ 3.5 ได้อธิบายการออกแบบส่วนต่าง ๆ ของระบบเป็นที่เรียบร้อยแล้ว ในหัวข้อนี้จะนำเสนอชุดทดลองทดสอบโปรโตคอลไอโอทีที่ใช้อุปกรณ์หรืออุปกรณ์จำลองที่ได้พัฒนาเสร็จสิ้นแล้ว โดยอธิบายส่วนต่าง ๆ ในการใช้งานตามหัวข้อดังนี้ 1.) การติดตั้ง 2.) การใช้งานส่วนทดลองทดสอบ 3.) การใช้งานส่วนแสดงผล

3.6.1 การติดตั้ง

ชุดทดลองทดสอบที่พัฒนาขึ้นประกอบด้วยโปรแกรมทั้งหมด 5 ส่วนคือ โปรแกรมส่วนภาคส่ง โปรแกรมส่วนภาครับ โปรแกรมแม่ข่าย โปรแกรมแสดงผลและฐานข้อมูล ซึ่งโปรแกรมส่วนการส่ง โปรแกรมส่วนภาครับและโปรแกรมแม่ข่ายพัฒนาขึ้นด้วยภาษา Python ดังนั้นผู้ทดสอบจะต้องติดตั้งตัวดำเนินการ Python ซึ่งดาวน์โหลดได้ที่ <https://www.python.org/> โปรแกรมส่วนแสดงผลพัฒนาด้วย Node js ผู้ทดสอบสามารถดาวน์โหลดและติดตั้งได้จาก <https://nodejs.org/en/> และฐานข้อมูล MySQL ใช้สำหรับกับผลลัพธ์ในการทดสอบสามารถดาวน์โหลดได้จากชุดโปรแกรม <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-16-04>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 โครงสร้างการติดตั้งชุดทดสอบ

จากรูปที่ 3.18 แสดงถึงโครงสร้างการติดตั้งโปรแกรมชุดทดสอบวัดประสิทธิภาพโพรโทคอลไอโอทีที่ทั้งแบบส่วนรับกับส่วนส่งแยกกัน หรือรวมกันโดยที่ภาพ 3.18(A) โปรแกรมส่วนส่ง (Sender) กับโปรแกรมส่วนรับจะถูกติดตั้งอยู่อุปกรณ์คนละตัว A และ B ตามลำดับ และโปรแกรมส่วนแม่ข่ายติดตั้งอยู่ที่อุปกรณ์แยกอีกหนึ่งตัวคืออุปกรณ์ C ส่วนภาพ 3.18(B) โปรแกรมส่วนส่งและส่วนรับติดตั้งอยู่ที่อุปกรณ์เดียวกันคืออุปกรณ์ A

3.6.2 ตัวอย่างการใช้งานชุดทดสอบวัดประสิทธิภาพโพรโทคอลไอโอที

การใช้งานชุดทดสอบวัดประสิทธิภาพโพรโทคอลไอโอที เริ่มด้วยการเริ่มโปรแกรมแม่ข่าย Server.sh การตั้งค่าจะต้องเลือกโพรโทคอลไอโอทีที่ต้องการเปิดแม่ข่าย ตั้งค่าพอร์ท ที่อุปกรณ์แม่ข่าย แสดงดังรูปที่ 3.19

```

IoT protocols performance testing v.1.8
IoT Server part
select IoT protocols [MQTT, CoAP, AMQP, MQTT-SN]
MQTT
insert port
5005
you slected MQTT on port 5005 confirm (y/n) ?
y
MQTT Borker running on port 5005 : press ctrl+c to coltrol menu

```

รูปที่ 3.19 การเริ่มใช้โปรแกรมแม่ข่าย

ถัดมาเริ่มตั้งค่าโปรแกรมส่วนรับข้อมูลที่อุปกรณ์รับข้อมูลโดยเริ่มโปรแกรม Receiver.sh การตั้งค่าจะต้องเลือกโพรโทคอลไอโอทีที่ต้องการทดสอบ พร้อมกับตั้งค่าการเชื่อมต่อมายังแม่ข่ายซึ่งแต่ละโพรโทคอลไอโอทีอาจมีค่าการตั้งค่าที่แตกต่างกันซึ่งโปรแกรมจะแสดงตัวอย่างให้ผู้ทดสอบเพื่อความสะดวก แสดงดังรูปที่ 3.20

```

IoT protocols performance testing v.1.8
IoT Receiver part
select IoT protocols [MQTT, CoAP, AMQP, MQTT-SN]
MQTT
insert server ip
192.168.1.1
insert port
5005
-----
Insert Data base ip:
192.168.1.1
Insert Data base user:
user
Insert Data base password:
1234
Insert Data base table name:
MQTT02
-----Recicer config-----
Protocol : MQTT
Server : 192.168.1.1 port 5005
Data base ip 192.168.1.1 user:user password:1234 table name:MQTT02
-----
confirm (y/n) ?
y
press enter to start...

```

รูปที่ 3.20 การเริ่มใช้โปรแกรมส่วนรับข้อมูล

ถัดมาการตั้งค่าโปรแกรมส่วนส่งข้อมูลที่อุปกรณ์ส่งข้อมูลโดยเริ่มโปรแกรม Sender.sh ดังรูปที่ 3.21 ซึ่งการตั้งค่าจะต้องเลือกโพรโทคอลไอโอทีที่ต้องการทดสอบเช่นเดียวกับส่วนรับ พร้อมกับตั้งค่าการเชื่อมต่อกับแม่ข่าย และกำหนดตัวแปรตั้งต้นดังนี้ จำนวนครั้งกำส่งข้อมูลทดสอบแต่ละกรณี (loop case) ขนาดของแพ็คเก็ตที่ส่ง (packet size) ระดับคุณภาพบริการ (QoS) อัตราการสูญหายของข้อมูล (loss rate) ค่าเวลาแฝงในเครือข่าย (latency delay) ความกว้างของเครือข่าย (Bandwidth) ตัวแปรแต่ละตัวสามารถใส่ได้มากกว่า 1 ค่ายกเว้นจำนวนครั้งในการส่ง

```

IoT protocols performance testing v.1.0
IoT Sender part
select IoT protocols [MQTT, CoAP, AMQP, MQTT-SN]
MQTT
Insert server ip
192.168.1.1
Insert port
5005
-----
Insert sending control case
QoS of MQTT : 0, 1, 2 : insert in form of array, example 0,1,2
0,1,2
Sending period(frequency) case : (ms : mllisecond) : insert in form of array, ex
ample 50,500,1000
50,100,500
Packet size case : (byte), minimum 20 byte , maximum 100k byte : insert in form
of array, example 20,50,100
20,100,1000
Sending round(Loop) : one value , example 1000
1000
Insert environment control case
Loss rate case : percentage(%), example 10,20,50
10,20,50
Latency time case : (ms : mllisecond) example 100,200,500
100,200,500
Bandwidth case : (kbps) example 100,500,1000
100,500,1000
-----sending config-----
Protocol : MQTT
Server : 192.168.1.1 port 5005
QoS cases : [0,1,2]
Sending period cases : [50,100,500]
Packet size cases : [20,100,1000]
Sending round : 1000
Loss rate cases : [10,20,50]
Latency time cases : [100,200,500]
Bandwidth cases : [100,500,1000]
-----
confirm (y/n) ?
y
press enter to start...

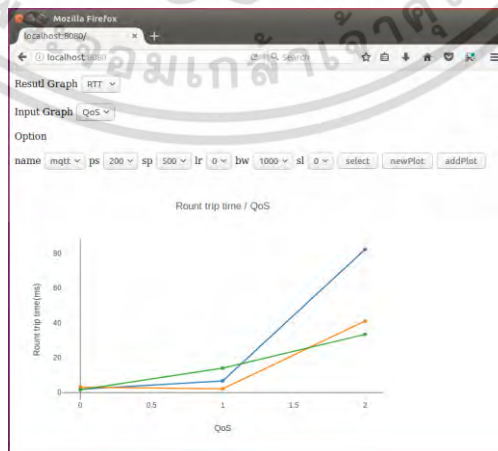
```

รูปที่ 3.21 ตัวอย่างการใช้งานโปรแกรมส่วนส่งข้อมูล

เมื่อตั้งค่าเสร็จสิ้นแล้วสามารถส่งเริ่มต้นดำเนินการทดสอบได้ที่โปรแกรม Sender.sh ด้วยคำสั่ง start ชุดการทดสอบจะทำการทดสอบจนกระทั่งครบวงการทำงานเสร็จสิ้น

3.6.3 ส่วนแสดงผลลัพธ์

เมื่อทดสอบสิ้นสุดทุกกรณีก็ตามที่กำหนดไว้เรียบร้อยแล้วสามารถดูผลการทดลองได้ที่ <http://localhost/iotprotocaltest:8888> ซึ่งจะปรากฏเว็บแอปพลิเคชันสำหรับแสดงผลการทดลองต่างๆ ซึ่งผู้ทดสอบสามารถเลือกความสัมพันธ์ของค่าตัวแปรต่างๆ ได้ตามที่ต้องการ การแสดงผลกราฟสามารถเลือกเป็นแบบเส้นหรือแบบแท่ง และกราฟแบบค่าสะสม แสดงดังรูปที่ 3.22



รูป 3.22 ตัวอย่างโปรแกรมแสดงผลของผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การทดสอบและอุปกรณ์ที่ใช้ในการทดสอบ

ในการทดสอบชุดทดสอบวัดประสิทธิภาพโพรโทคอลไอโอที จะทำการนำผลการทดสอบของโพรโทคอล MQTT CoAP และ AMQP มาเปรียบเทียบกับผลการทดสอบที่ได้จากงานวิจัยที่เกี่ยวข้อง ซึ่งในการทดสอบจะแสดงผลลัพธ์ที่ใช้ตัวแปรตั้งต้นและตัวแปรวัดผล ค่าเดียวกันกับงานวิจัยที่เกี่ยวข้อง จะมีเพียงแต่อุปกรณ์ที่ใช้ในการทดสอบกับชุดทดสอบเท่านั้นซึ่งแตกต่างกัน เนื่องจากข้อจำกัดเกี่ยวกับอุปกรณ์ที่อาจจะคุณสมบัติแตกต่างกัน จึงจะทำให้ค่าผลลัพธ์ในเชิงปริมาณต่างกัน แต่ในเชิงของแนวโน้มหรือพฤติกรรมของตัวแปรวัดผล ควรจะเป็นไปในทิศทางเดียวกัน ชุดในการทดสอบโดยชุดทดสอบจะมีทดสอบโดยใช้อุปกรณ์จริงและอุปกรณ์จำลอง เพื่อเป็นการแสดงให้เห็นว่า ชุดทดสอบสามารถรองรับในการทำงานกับอุปกรณ์จริงและอุปกรณ์จำลองได้จริง รายชื่ออุปกรณ์และคุณสมบัติเป็นไปตามตารางที่ 3.1 และ 3.2

ตารางที่ 3.1 รายชื่ออุปกรณ์จริงที่ใช้ในการทดสอบ

ชื่ออุปกรณ์	หน้าที่	คุณสมบัติ
Raspberry Pi 3 Model B	อุปกรณ์ตัวส่งข้อมูล	CPU : 4x ARM Cortex-A53, 1.2GHz RAM : 1GB LPDDR2 900MHz LAN : 10/100Mbps Wireless : IEEE 802.11n OS : Linux(Raspbian)
Thinkpad x61	อุปกรณ์ตัวรับข้อมูล	CPU : Intel C2D 2.0GHz T7300 RAM : 2GB DDR2 PC5200 LAN : 1Gbp Wireless : IEEE 802.11a/b/g/n OS : Linux(Ubuntu)
Asus ux410uq	อุปกรณ์แม่ข่ายไอโอที	CPU : Intel i7 2.7GHz RAM : 8GB DDR4 LAN : 1Gbp Wireless : IEEE 802.11ac OS : Windows 10
TP-Link TL-WR842N	อุปกรณ์เครือข่าย	Wire Interface: - 4x 10/100Mbps LAN Ports - 1x 10/100Mbps WAN Port Wireless Standards - IEEE 802.11n/g/b

ตารางที่ 3.2 รายชื่ออุปกรณ์จำลองที่ใช้ในการทดสอบ

ชื่ออุปกรณ์	หน้าที่	คุณสมบัติ
Pub (Virtual Box)	อุปกรณ์ตัวส่งข้อมูล	CPU : 1 Core RAM : 1GB LAN : 1Gpbs OS : Linux (Ubuntu)
Sub (Virtual Box)	อุปกรณ์ตัวรับข้อมูล	CPU : 1 Core RAM : 1GB LAN : 1Gpbs OS : Linux (Ubuntu)
Server (Virtual Box)	อุปกรณ์แม่ข่ายไอโอที	CPU : 1 Core RAM : 1GB LAN : 1Gpbs OS : Linux (Ubuntu)
Internal Network (Virtual Box)	อุปกรณ์เครือข่าย	Virtual Box Network 1Gpbs
Asus ux410uq	อุปกรณ์เจ้าบ้าน(Host)	CPU : Intel i7 2.7GHz RAM : 8GB DDR4 LAN : 1Gbp Wireless : IEEE 802.11ac OS : Windows 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ในบทนี้จะกล่าวถึงทั้งหมด 2 หัวข้อนั้นคือ 1.) ผลการทดสอบวัดประสิทธิภาพโพรโทคอลไอโอทีโดยชุดทดสอบที่พัฒนาขึ้น โดยเปรียบเทียบกับผลการทดสอบกับงานวิจัยที่เกี่ยวข้อง เพื่อเป็นการทดสอบว่าผลลัพธ์ที่ได้จากชุดทดสอบที่พัฒนาขึ้นสามารถนำไปใช้อ้างอิงได้ และ 2.) คือผลลัพธ์จากการพัฒนาชุดทดสอบวัดประสิทธิภาพโพรโทคอลไอโอทีที่สามารถช่วยเหลือผู้ทดสอบได้อย่างไรบ้างเมื่อเปรียบเทียบกับระบบดั้งเดิม

4.1 ผลการทดสอบวัดประสิทธิภาพโพรโทคอลไอโอทีโดยชุดทดสอบที่พัฒนาขึ้น

ในการทดสอบจะนำผลลัพธ์ของตัวแปรวัดผล ระยะเวลาส่งข้อมูลไป-กลับ (Round Trip Time: RTT) อัตราการสูญหายของข้อมูล (Loss rate) ค่าเวลาหน่วงที่ไม่คงที่ (Jitter) ข้อมูลส่วนเกิน (Overhead) โดยผลลัพธ์แต่ละอย่างจะนำมาจากงานวิจัยที่เกี่ยวข้องในแต่ละงานที่มีการนำเสนอผลลัพธ์นั้น ๆ ซึ่งผลลัพธ์นั้นอาจจะไม่ใช่ค่าที่ตรงกันเนื่องจากอุปกรณ์การที่ใช้ในการทดสอบเปรียบเทียบกับแต่ละงานวิจัยนั้นแตกต่างกัน แต่ได้นำเสนอในส่วนของแนวโน้มผลลัพธ์หรือความคล้ายคลึง โดยในการทดสอบจะใช้โพรโทคอล MQTT กับ CoAP ในการทดสอบเป็นหลักแต่จะมีในบางการทดสอบจะใช้โพรโทคอล AMQP เข้ามาร่วมด้วย ระบบเครือข่ายในการทดสอบจะปรับเปลี่ยนไปตามการทดสอบในงานวิจัยที่นำมาเปรียบเทียบ ซึ่งจะมีทั้งเครือข่ายแบบสายและไร้สาย

4.1.1 ระยะเวลาส่งข้อมูลไป-กลับ (Round Trip Time : RTT)

การทดสอบที่ 1 การทดสอบระยะเวลาส่งข้อมูลไป-กลับ (RTT) เมื่อมีค่าเวลาแฝง (Latency time) ค่าต่างๆ โดยได้ยกตัวอย่างงานวิจัยของ Yuang Chen และ Thomas Kunz มาท การเปรียบเทียบโดยในงานวิจัยดังกล่าวได้ทดสอบในเครือข่ายแบบไร้สาย แต่ในการทดสอบด้วยชุดทดสอบจะทำการทดสอบในเครือข่ายทั้งแบบสายและไร้สาย โดยมีค่าตัวแปรตั้งต้นอื่น ๆ ดังตารางที่ 4.1

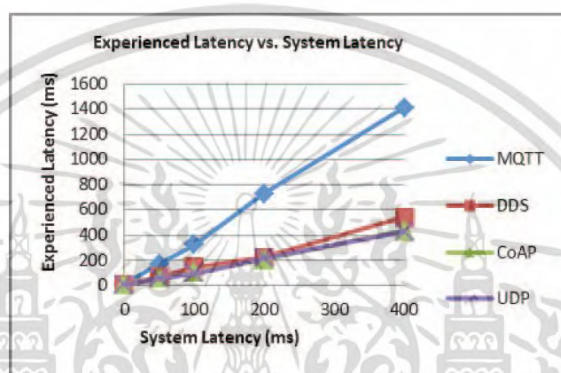
ตารางที่ 4.1 ค่าตัวแปรตั้งต้นในการทดสอบที่ 1

ตัวแปรตั้งต้น	
ขนาดแพ็คเก็ต	409 ไบต์
ความถี่การส่งแพ็คเก็ต	250 มิลลิวินาที

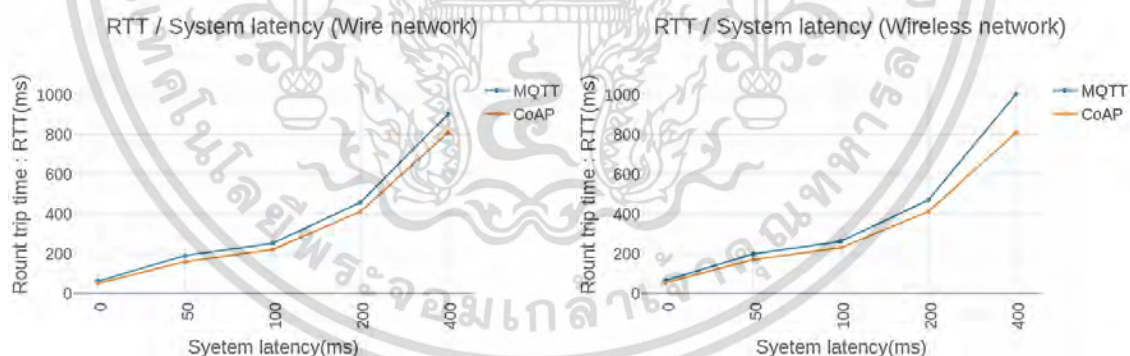
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 (ต่อ)

ตัวแปรตั้งต้น		
คุณภาพบริการ	MQTT ระดับ 0	CoAP ไม่ใช่
อัตราการสูญหายข้อมูลในเครือข่าย	ไม่มี	
ค่าเวลาแฝงในการรับ/ส่งข้อมูล	0 50 100 200 และ 400 มิลลิวินาที	
ปริมาณการรับ/ส่งข้อมูลของการ์ดเครือข่าย	ไม่จำกัด	



รูปที่ 4.1 ระยะเวลาส่งข้อมูลไป-กลับในแต่ละค่าเวลาแฝงจากงานวิจัยเปรียบเทียบ [5]

รูปที่ 4.2 ระยะเวลาส่งข้อมูลไป-กลับในแต่ละค่าเวลาแฝงจากชุดทดสอบ
ในเครือข่ายแบบสาย (ข่าย) และไร้สาย (ขวา)

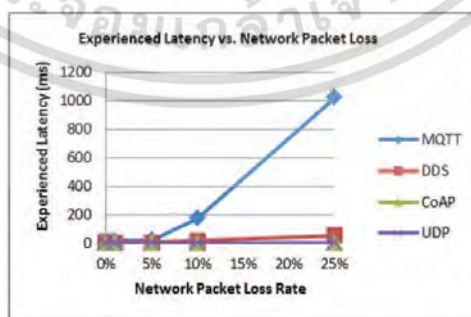
จากรูปที่ 4.1 ซึ่งแสดงถึงผลลัพธ์ของระยะเวลาส่งข้อมูลไปยังตัวรับเมื่อกำหนดค่าเวลาแฝงจากงานของ Yuang Chen และ Thomas Kunz จะเห็นได้ว่าในทุกๆ ค่าเวลาแฝงโพรโทคอล CoAP จะมีค่าระยะเวลาส่งข้อมูลไป-กลับน้อยกว่าโพรโทคอล MQTT ส่วนผลลัพธ์ที่ได้จากชุดทดสอบในรูปที่ 4.2 แบ่ง

ออกเป็นการทดสอบในเครือข่ายแบบสาย (ซ้าย) และไร้สาย (ขวา) มีระยะเวลาส่งข้อมูลไป-กลับของ CoAP น้อยกว่า MQTT เช่นเดียวกับงานวิจัยเปรียบเทียบ โดยผลลัพธ์จากการทดสอบด้วยชุดทดสอบพบว่าในเครือข่ายแบบสายมีระยะเวลาส่งข้อมูลไป-กลับน้อยกว่าเครือข่ายแบบไร้สายเล็กน้อย และสังเกตเห็นว่าระยะเวลาส่งข้อมูลไป-กลับมีค่าเป็นสองเท่าของเวลาแฝงที่ใช้ในการทดสอบ จากการทดสอบจะเห็นได้ว่าเมื่อในระบบหรือเครือข่ายมีค่าเวลาแฝงเพิ่มขึ้น ระยะเวลาในการส่งข้อมูลจะเพิ่มตามไปด้วยไม่ว่าจะเป็นเครือข่ายแบบสายหรือไร้สาย หรือโพรโทคอลไอโอทีที่เป็น TCP-base หรือ UDP-base

การทดสอบที่ 2 การทดสอบระยะเวลาส่งข้อมูลไป-กลับ(RTT) เมื่อมีอัตราการสูญหายข้อมูลในเครือข่าย(Loss rate) เพิ่มขึ้น โดยได้ยกตัวอย่างงานวิจัยของ Yuang Chen และ Thomas Kunz มาทำการเปรียบเทียบ โดยในงานวิจัยที่นำมาเปรียบเทียบได้ทำการทดสอบในเครือข่ายแบบไร้สาย ส่วนการทดสอบโดยใช้ชุดทดสอบได้ทำการทดสอบในเครือข่ายแบบสายและไร้สาย โดยมีค่าตัวแปรตั้งต้นดังตารางที่ 4.2

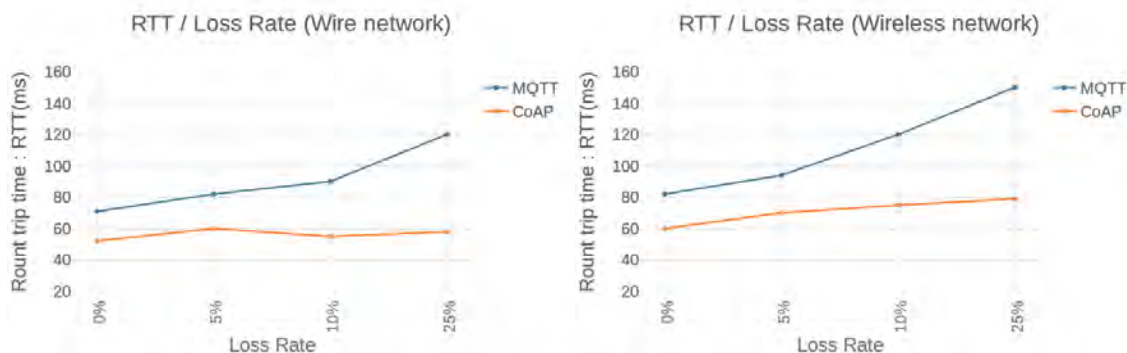
ตารางที่ 4.2 ค่าตัวแปรตั้งต้นในการทดสอบที่ 2

ตัวแปรตั้งต้น	
ขนาดแพ็คเกจ	409 ไบต์
ความถี่การส่งแพ็คเกจ	250 มิลลิวินาที
คุณภาพบริการ	MQTT ระดับ 0 CoAP ไม่ใช่
อัตราการสูญหายข้อมูลในเครือข่าย	ร้อยละ 0 5 10 และ 25
ค่าเวลาแฝงในการรับ/ส่งข้อมูล	ไม่มี
ปริมาณการรับ/ส่งข้อมูลของการทดสอบเครือข่าย	ไม่จำกัด



รูปที่ 4.3 ระยะเวลาส่งข้อมูลไป-กลับเมื่อมีอัตราการสูญหายข้อมูลในเครือข่ายเพิ่มขึ้นจากงานวิจัยเปรียบเทียบ [5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ระยะเวลาส่งข้อมูลไป-กลับเมื่อมีอัตราการสูญหายข้อมูลในเครือข่ายเพิ่มขึ้นจากชุดทดสอบเครือข่ายแบบสาย (ซ้าย) แบบไร้สาย (ขวา)

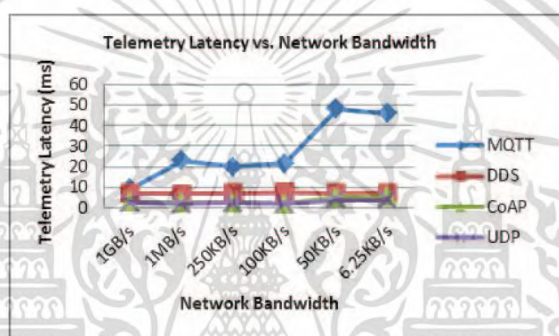
ผลลัพธ์ที่ได้จากงานวิจัยของ Yuang Chen และ Thomas Kunz ในรูปที่ 4.3 และผลลัพธ์จากชุดทดสอบในรูปที่ 4.4 แสดงถึงเวลาที่ข้อมูลถูกส่งจากตัวส่งไป-กลับ ซึ่งจะเห็นว่าโพรโทคอล MQTT มีอัตราการเพิ่มขึ้นของระยะเวลาส่งข้อมูลไป-กลับมากกว่าโพรโทคอล CoAP ซึ่งโพรโทคอล CoAP อัตราการสูญหายมีผลกับเวลาส่งข้อมูลไป-กลับน้อยมาก สาเหตุเป็นเพราะโพรโทคอล MQTT เป็น TCP-base ซึ่งจะมีการแก้ไขเมื่อเกิดการสูญหายของข้อมูลทำให้ TCP จะพยายามส่งข้อมูลให้สำเร็จในทุก ๆ ครั้งที่ทำ การส่งข้อมูลไปยังปลายทาง ส่วน CoAP เป็น UDP-base ซึ่งไม่มีส่วนแก้ไขปัญหาเมื่อเกิดการสูญหายของข้อมูล จึงทำให้ UDP จะไม่เสียเวลาในการพยายามส่งข้อมูลให้ถึงปลายทาง ทำให้ในแง่ของความเร็วโพรโทคอล CoAP จะสามารถส่งข้อมูลได้รวดเร็วกว่าโพรโทคอล MQTT

ผลลัพธ์ในการทดสอบจากชุดทดสอบพบว่า เมื่อมีอัตราการสูญหายของข้อมูลในเครือข่ายเครือข่ายแบบสาย โพรโทคอล CoAP จะมีค่าเวลาในการส่งข้อมูลไป-กลับในทั้งสองเครือข่ายใกล้เคียงกัน ส่วนโพรโทคอล MQTT พบว่าในเครือข่ายแบบสายจะมีค่าเวลาในการส่งข้อมูลไป-กลับน้อยกว่าในเครือข่ายแบบไร้สาย สาเหตุอาจเป็นเพราะในเครือข่ายแบบสายมีความเร็วและเสถียรภาพมากกว่าเครือข่ายแบบไร้สาย

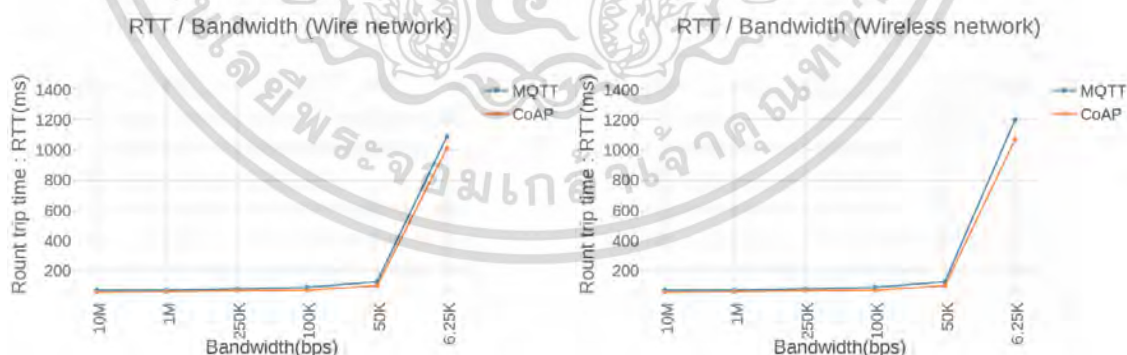
การทดสอบที่ 3 การทดสอบระยะเวลาส่งข้อมูลไป-กลับ(RTT) เมื่อตัวแปรตั้งต้นค่าความกว้างในการรับ-ส่งข้อมูลในเครือข่าย(Bandwidth) ที่แตกต่างกัน โดยได้ยกตัวอย่างงานวิจัยของ Yuang Chen และ Thomas Kunz มาทำการเปรียบเทียบ โดยในงานวิจัยดังกล่าวทำการทดสอบในเครือข่ายแบบไร้สาย ส่วนการทดสอบด้วยชุดทดสอบจะทำการทดสอบทั้งในเครือข่ายแบบสายและแบบไร้สาย โดยมีค่าตัวแปรตั้งต้นดังตารางที่ 4.3

ตารางที่ 4.3 ค่าตัวแปรตั้งต้นในการทดสอบที่ 3

ตัวแปรตั้งต้น		
ขนาดแพ็คเกจ	409 ไบต์	
ความถี่การส่งแพ็คเกจ	250 มิลลิวินาที	
คุณภาพบริการ	MQTT ระดับ 0	CoAP ไม่ใช่
อัตราการสูญหายข้อมูลในเครือข่าย	ไม่มี	
ค่าเวลาแฝงในการรับ/ส่งข้อมูล	ไม่มี	
ปริมาณการรับ/ส่งข้อมูลของการ์ดเครือข่าย	1Gbps, 1M – 6.25Kbps	



รูปที่ 4.5 ระยะเวลาส่งข้อมูลไป-กลับเมื่อความกว้างในการรับ-ส่งข้อมูลของเครือข่ายแตกต่างกันจากงานวิจัยเปรียบเทียบ [5]



รูปที่ 4.6 ระยะเวลาส่งข้อมูลไป-กลับเมื่อความกว้างในการรับ-ส่งข้อมูลของเครือข่ายแตกต่างกันจากชุดทดสอบในเครือข่ายแบบสาย (ข่าย) และไร้สาย (ขวา)

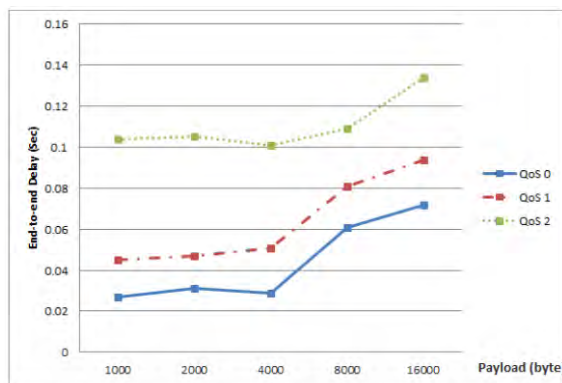
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบจากงานวิจัยของ Yuang Chen และ Thomas Kunz ในรูปที่ 4.5 พบว่าโพรโทคอล CoAP สามารถส่งแพ็คเกจโดยมีเวลาในการส่งข้อมูลไป-กลับใกล้เคียงกันในทุกความกว้างในการรับ-ส่งข้อมูลของเครือข่าย แต่ในโพรโทคอล MQTT มีแนวโน้มที่จะใช้เวลาในการส่งข้อมูลไป-กลับเพิ่มมากขึ้นเมื่อมีค่าความกว้างของเครือข่ายลดลง ส่วนผลลัพธ์จากชุดทดสอบในรูปที่ 4.6 โพรโทคอลทั้ง CoAP และ MQTT เมื่อความกว้างในการรับ-ส่งข้อมูลของเครือข่าย 10Mbps ถึง 50Kbps ทั้งสองโพรโทคอลมีค่าการส่งข้อมูลไป-กลับไม่เกิน 200 มิลลิวินาที แต่พบว่ามีค่าความแตกต่างจากงานวิจัยที่นำมาเปรียบเทียบเมื่อความกว้างในการรับ-ส่งข้อมูลของเครือข่ายเท่ากับ 6.25 Kbps โดยโพรโทคอลทั้ง CoAP และ MQTT ใช้เวลาในการส่งข้อมูลไป-กลับประมาณ 1 วินาที ซึ่งอาจเป็นไปได้ว่า แพ็คเกจที่ใช้ในการทดสอบมีขนาด 409 ไบต์ซึ่งเท่ากับ 3.2K บิต และเมื่อค่าความกว้างในการรับ-ส่งของเครือข่ายอยู่ที่ 6.25Kbps $6.25 \times 1,024 \text{ b/s} = 6,400 \text{ bps}$ ในการส่งข้อมูล 3,272 บิต ไปและกลับอาจจะใช้เวลาอยู่ที่ประมาณ 1 วินาที และจากการทดสอบนี้แสดงให้เห็นว่า ทั้งสองโพรโทคอลมีแนวโน้มที่จะใช้เวลาในการรับ-ส่งข้อมูลเพิ่มมากขึ้นเมื่อมีความกว้างในการรับ-ส่งข้อมูลน้อยลง

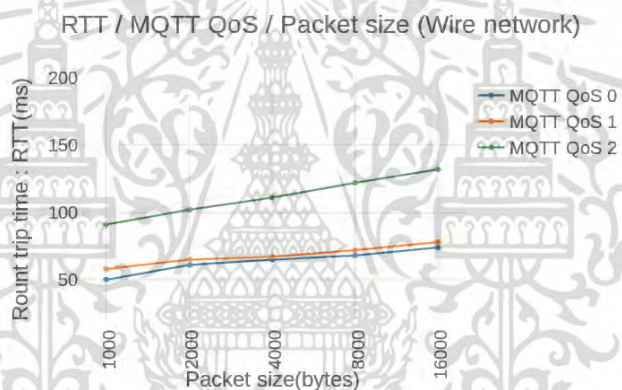
การทดสอบที่ 4 การทดสอบระยะเวลาส่งข้อมูลไป-กลับ(RTT) โดยมีคุณภาพบริการ(QoS) และขนาดของแพ็คเกจในการทดสอบต่างกัน โดยได้ยกตัวอย่างงานวิจัยของ Shinho Lee และคณะ ซึ่งทำการทดสอบคุณภาพบริการของโพรโทคอล MQTT ในเครือข่ายแบบสายและไร้สายที่เป็นมาตรฐาน 3G สำหรับการทดสอบด้วยชุดทดสอบจะใช้เครือข่ายแบบสายและไร้สายแต่เป็นมาตรฐาน IEEE 802.11 มาทำการเปรียบเทียบโดยมีค่าตัวแปรตั้งต้นดังตารางที่ 4.4

ตารางที่ 4.4 ค่าตัวแปรตั้งต้นในการทดสอบที่ 4

ตัวแปรตั้งต้น	
ขนาดแพ็คเกจ	1K 2K 4K 8K และ 16K ไบต์
ความถี่การส่งแพ็คเกจ	250 มิลลิวินาที
คุณภาพบริการ	ระดับ 0 1 และ 2
อัตราการสูญหายข้อมูลในเครือข่าย	ไม่มี
ค่าเวลาแฝงในการรับ/ส่งข้อมูล	ไม่มี
ปริมาณการรับ/ส่งข้อมูลของการทดสอบเครือข่าย	1Gbps, 1M – 6.25Kbps



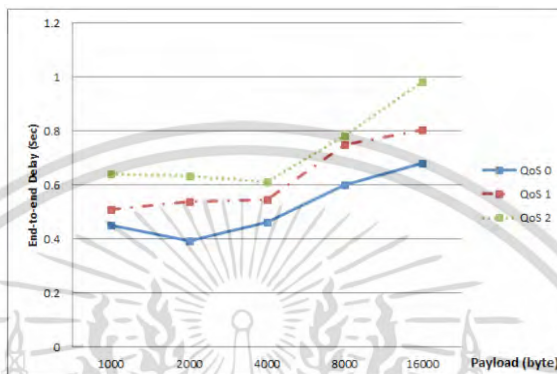
รูปที่ 4.7 ระยะเวลาส่งข้อมูลไป-กลับพรโทคอล MQTT เมื่อคุณภาพบริการและขนาดของแพ็คเก็ตที่ต่างกันในเครือข่ายแบบสายจากงานวิจัยเปรียบเทียบ [3]



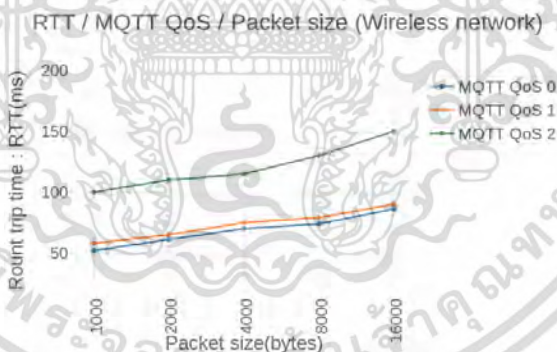
รูปที่ 4.8 ระยะเวลาส่งข้อมูลไป-กลับพรโทคอล MQTT เมื่อคุณภาพบริการและขนาดของแพ็คเก็ตที่ต่างกันเครือข่ายแบบสายจากชุดทดสอบ

ผลลัพธ์จากการทดสอบจากงานวิจัยของ Shinho Lee และคณะ ในรูปที่ 4.7 แสดงให้เห็นว่าในการรับ-ส่งข้อมูลในเครือข่ายแบบสายโดยใช้โพรโทคอล MQTT เมื่อขนาดของแพ็คเก็ตเพิ่มมากขึ้น เวลาที่ใช้ในการรับ-ส่งจะมีแนวโน้มเพิ่มขึ้นตามไปด้วย ส่วนระดับของคุณภาพบริการจะมีผลต่อเวลาในการรับ-ส่งข้อมูลตามไปด้วย โดยจากกราฟระดับคุณภาพบริการที่ 0 และ 1 จะใช้เวลาในการรับ-ส่งข้อมูลที่ใกล้เคียงกันแต่ระดับคุณภาพบริการที่ 2 จะใช้เวลาในการส่งข้อมูลไป-กลับมากกว่าระดับที่ 0 และ 1 อย่างเห็นได้ชัด ส่วนผลลัพธ์ที่ได้จากชุดทดสอบให้ความสัมพันธ์ที่แบบเดียวกับงานวิจัยของ Shinho Lee และคณะ โดยเวลาที่ใช้ในการรับ-ส่งมีแนวโน้มเพิ่มมากขึ้นเมื่อขนาดของแพ็คเก็ตที่ส่งเพิ่มขึ้น และเวลาในระดับคุณภาพบริการที่ 0 และ 1 ใช้เวลาในการรับ-ส่งข้อมูลใกล้เคียงกัน ส่วนระดับที่ 2 จะเพิ่มสูงออกไป

ผลการทดสอบในเงื่อนไขเดียวกันกับเครือข่ายแบบไร้สายโดยใช้ระบบ 3G ผลลัพธ์จากงานวิจัยของ Shinho Lee และคณะ พบว่าเวลาที่ใช้ในการรับ-ส่งข้อมูลในเครือข่ายแบบไร้สายน้อยกว่าเครือข่ายแบบสาย โดยเวลาที่มีแนวโน้มเพิ่มมากขึ้นเมื่อแพ็คเกจมีขนาดใหญ่ขึ้นเช่นเดียวกับการทดสอบในเครือข่ายแบบสาย แสดงได้ดังรูปที่ 4.9



รูปที่ 4.9 ระยะเวลาส่งข้อมูลไป-กลับโพรโทคอล MQTT เมื่อคุณภาพบริการและขนาดของแพ็คเกจแตกต่างกันในเครือข่ายแบบไร้สายจากงานวิจัยเปรียบเทียบ [3]



รูปที่ 4.10 ระยะเวลาส่งข้อมูลไป-กลับโพรโทคอล MQTT เมื่อคุณภาพบริการและขนาดของแพ็คเกจที่แตกต่างกันในเครือข่ายแบบไร้สายจากชุดทดสอบ

ในขณะเดียวกันผลลัพธ์จากการทดสอบด้วยชุดทดสอบพบว่า ระยะเวลาในการส่งข้อมูลไป-กลับมีแนวโน้มเพิ่มมากขึ้นเมื่อแพ็คเกจมีขนาดใหญ่ขึ้นและระดับคุณภาพบริการที่มากขึ้น แต่กับพบว่าเวลาที่ใช้ในการรับ-ส่งข้อมูลในเครือข่ายแบบไร้สายช้ากว่าความเวลาที่ใช้ในเครือข่ายแบบสาย แต่เนื่องจากการทดสอบจากงานวิจัยที่เปรียบเทียบได้ใช้การทดสอบโดยมีเครื่องแม่ข่ายไอโอที (MQTT Broker) ติดตั้งอยู่

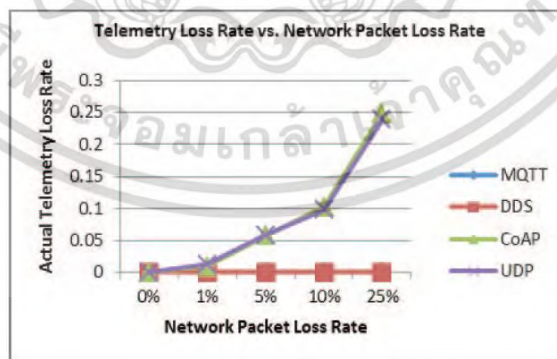
ในอินเทอร์เน็ต และการทดสอบด้วยเครือข่ายแบบไร้สายเป็นเทคโนโลยี 3G ซึ่งอาจเป็นไปได้ว่ามี ความเร็วมากกว่าเครือข่ายแบบสายที่ใช้ในการทดสอบ

4.1.2 อัตราการสูญหายของข้อมูล (Loss rate)

การทดสอบที่ 5 การทดสอบอัตราการสูญหายของข้อมูล(Loss rate) เมื่ออัตราการสูญหายใน เครือข่าย(Loss rate) ต่าง ๆ โดยได้ยกตัวอย่างงานวิจัยของ Yuang Chen Thomas Kunz ซึ่งทำการ ทดสอบในเครือข่ายแบบไร้สาย มาท การเปรียบเทียบกับชุดทดสอบซึ่งจะท การทดสอบในเครือข่ายแบบ สายและแบบไร้สาย โดยจะใช้โพรโทคอล MQTT และโพรโทคอล CoAP และมีค่าตัวแปรตั้งต้นดังตารางที่ 4.5 ในการทดสอบ

ตารางที่ 4.5 ค่าตัวแปรตั้งต้นในการทดสอบที่ 5

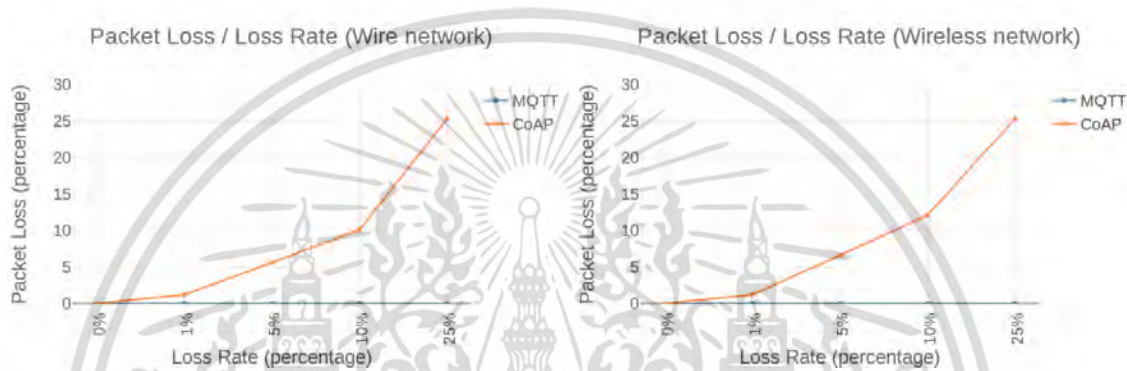
ตัวแปรตั้งต้น	
ขนาดแพ็คเกจ	409 ไบต์
ความถี่การส่งแพ็คเกจ	250 มิลลิวินาที
คุณภาพบริการ	MQTT ระดับ 0 CoAP ไม่ใช้
อัตราการสูญหายข้อมูลในเครือข่าย	ร้อยละ 0 5 10 และ 25
ค่าเวลาแฝงในการรับ/ส่งข้อมูล	ไม่มี
ปริมาณการรับ/ส่งข้อมูลของการ์ดเครือข่าย	ไม่จำกัด



รูปที่ 4.11 อัตราการสูญหายข้อมูลเมื่อมีอัตราการสูญหายข้อมูลของเครือข่ายต่างๆ จากงานวิจัย เปรียบเทียบ [5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบจากงานวิจัยของ Yuang Chen และ Thomas Kunz ในรูปที่ 4.11 พบว่าโพรโทคอล CoAP อัตราการเกิดข้อมูลสูญหายค่าใกล้เคียงกับค่าอัตราการสูญหายในสภาพแวดล้อมเครือข่ายที่กำหนด ส่วนโปรแกรม MQTT ไม่มีแพ็คเกจเกิดสูญหาย ส่วนผลลัพธ์จากชุดทดสอบ ในรูปที่ 4.12 พบว่าในการทดสอบกับเครือข่ายแบบสายและไร้สาย โพรโทคอล CoAP อัตราการเกิดข้อมูลสูญหายมีค่าใกล้เคียงกับค่าอัตราการสูญหายในสภาพแวดล้อมเครือข่ายที่กำหนด ส่วนโปรแกรม MQTT ไม่มีแพ็คเกจเกิดสูญหาย เช่นเดียวกับผลลัพธ์ที่ได้จากงานวิจัยที่นำมาเปรียบเทียบ



รูปที่ 4.12 อัตราการสูญหายข้อมูลเมื่อมีอัตราการสูญหายข้อมูลของเครือข่ายต่างๆ จากชุดทดสอบในเครือข่ายแบบสาย (ซ้าย) และไร้สาย (ขวา)

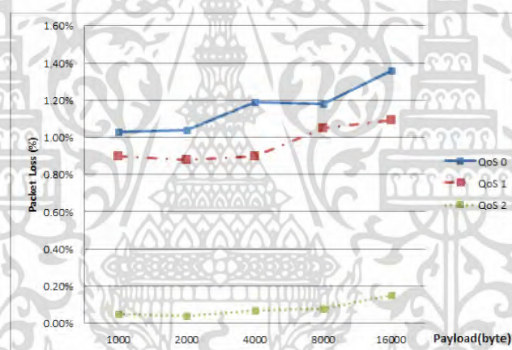
สาเหตุที่ไม่พบอัตราแพ็คเกจเกิดสูญหายในโพรโทคอล MQTT เนื่องจากเป็น TCP-base ซึ่งแม้ว่าจะเลือกใช้คุณภาพบริการของโพรโทคอล MQTT ในระดับ 0 ซึ่งไม่มีการตอบกลับจากปลายทางเมื่อได้รับข้อมูลแล้วก็ตาม แต่ด้วยความที่เป็น TCP-base ซึ่งจะมีการส่งข้อมูลใหม่เมื่อไม่ได้รับการตอบรับจากปลายทางว่าได้รับข้อมูลเรียบร้อยแล้ว จึงทำให้ในการทดสอบจากชุดทดสอบทั้งในเครือข่ายทั้ง 2 แบบ ไม่พบแพ็คเกจเกิดสูญหายในโพรโทคอล MQTT แต่อย่างใด แต่ในโพรโทคอล CoAP ซึ่งเป็น UDP-base ซึ่งไม่มีตัวจัดการในการตรวจเมื่อข้อมูลไปถึงปลายทางเรียบร้อยแล้ว จึงทำให้มีอัตราสูญหายแพ็คเกจมีค่าใกล้เคียงกับอัตราการสูญหายข้อมูลของเครือข่ายที่กำหนด โดยพบว่าการทดสอบจากชุดทดสอบเครือข่ายแบบสายมีอัตราการสูญหายข้อมูลของโพรโทคอล CoAP น้อยกว่าเครือข่ายแบบไร้สายอยู่เล็กน้อย

การทดสอบที่ 6 การทดสอบอัตราการสูญหายของข้อมูล(Loss rate) เมื่อกำหนดคุณภาพบริการและขนาดแพ็คเกจที่แตกต่างกัน โดยได้ยกตัวอย่างงานวิจัยของ Shinho Lee และคณะมาทำการเปรียบเทียบ ซึ่งงานวิจัยดังกล่าวทำการทดสอบโดยใช้โพรโทคอล MQTT ในเครือข่ายแบบไร้สาย ส่วน

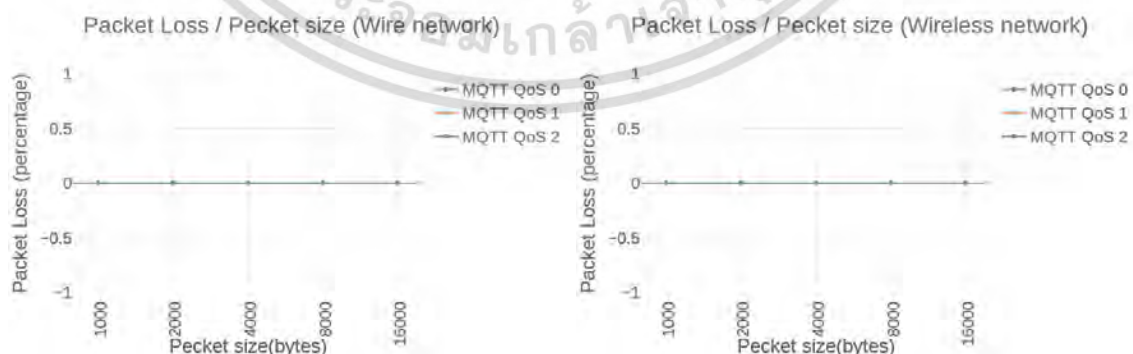
การทดสอบโดยชุดทดสอบทำการทดสอบในเครือข่ายแบบสายและไร้สาย โดยมีค่าตัวแปรตั้งต้นดังตารางที่ 4.6

ตารางที่ 4.6 ค่าตัวแปรตั้งต้นในการทดสอบที่ 6

ตัวแปรตั้งต้น	
ขนาดแพ็คเกจ	1K 2K 4K 8K และ 16K ไบต์
ความถี่การส่งแพ็คเกจ	250 มิลลิวินาที
คุณภาพบริการ	MQTT ระดับ 0 1 และ 2
อัตราการสูญหายข้อมูลในเครือข่าย	ไม่มี
ค่าเวลาแฝงในการรับ/ส่งข้อมูล	ไม่มี
ปริมาณการรับ/ส่งข้อมูลของการทดสอบเครือข่าย	ไม่จำกัด



รูปที่ 4.13 อัตราการสูญหายของข้อมูล เมื่อ กำหนดคุณภาพบริการและขนาดของแพ็คเกจของโพรโทคอล MQTT ในเครือข่ายแบบไร้สายจากงานวิจัยเปรียบเทียบ [3]



รูปที่ 4.14 อัตราการสูญหายของข้อมูล เมื่อ กำหนดคุณภาพบริการและขนาดของแพ็คเกจของโพรโทคอล MQTT ในเครือข่ายแบบสายและไร้สายจากชุดทดสอบ

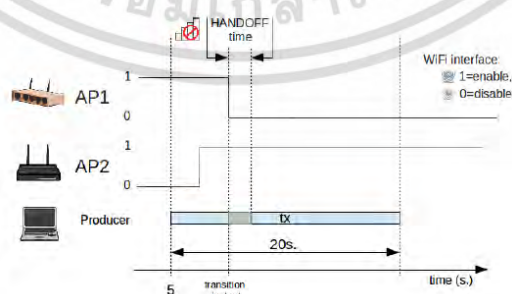
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.13 ในงานวิจัยที่นำมาเปรียบเทียบพบว่า ในการเลือกใช้คุณภาพบริการของโพรโทคอล MQTT ระดับที่ 0 และ 1 ในเครือข่ายแบบไร้สายมีอัตราการสูญหายของข้อมูลประมาณร้อยละ 1 ส่วนคุณภาพบริการระดับที่ 2 พบว่าอัตราการสูญหายของข้อมูลน้อยกว่าร้อยละ 0.2

จากการทดสอบอัตราการเกิดข้อมูลสูญหาย ซึ่งผลลัพธ์จากการทดสอบด้วยชุดทดสอบดังในรูปที่ 4.14 โพรโทคอล MQTT ไม่พบอัตราการเกิดข้อมูลสูญหายใด ๆ ในทุกกรณีทั้งในเครือข่ายแบบสายและไร้สายและการทดสอบด้วยการกำหนดอัตราการเกิดข้อมูลสูญหายให้กับสภาพแวดล้อมเครือข่าย เพราะว่าโพรโทคอล MQTT เป็น TCP-base ซึ่งจะมีการส่งข้อมูลใหม่เมื่อไม่ได้รับการตอบรับจากปลายทางว่าได้รับข้อมูลเรียบร้อยแล้ว จึงทำให้ในการทดสอบจากชุดทดสอบทั้งในเครือข่ายทั้ง 2 แบบ ไม่พบแพ็คเกจสูญหายในโพรโทคอล MQTT แต่อย่างใด ส่วนสาเหตุที่พบอัตราการสูญหายข้อมูลจากผลการทดสอบจากงานวิจัยเปรียบเทียบ อาจเป็นไปได้ว่าเครือข่ายไร้สายแบบเทคโนโลยี 3G กับ IEEE 802.11 มีกลไกในการป้องกันการเกิดข้อมูลสูญหายที่แตกต่างกัน

4.1.3 ค่าเวลาหน่วงไม่คงที่ (Jitter)

การทดสอบที่ 7 การทดสอบค่าเวลาหน่วงไม่คงที่ จะนำการทดสอบจากงานวิจัยของ Jorge E. Luzuriaga และคณะมาเปรียบเทียบ ซึ่งได้ทำการทดสอบกับโพรโทคอล MQTT และ AMQP เพื่อทดสอบการแก้ไขปัญหาเมื่อการเชื่อมต่อถูกตัดขาดและเมื่อการเชื่อมต่อสามารถกลับมาใช้ได้อีกครั้ง ข้อมูลที่ถูกส่งในขณะที่การเชื่อมต่อถูกตัดขาดจะเป็นอย่างไร โดยวิธีการทดสอบของ Jorge E. Luzuriaga และคณะคือการใช้ตัวปล่อยสัญญาณเครือข่ายไร้สาย 2 ชุด โดยช่วงแรกจะเปิดให้เครื่องลูกข่ายรับ-ส่งข้อมูลกับเครื่องแม่ข่ายโดยใช้ตัวปล่อยสัญญาณชุดที่ 1 ก่อนจากนั้นจะปิดตัวปล่อยชุดที่ 1 ลงและเปิดตัวปล่อยสัญญาณชุดที่ 2 ให้เครื่องลูกข่าย เป็นการตัดการเชื่อมต่อ ในการทดสอบได้ใช้เวลา 20 วินาทีโดยใช้ขนาดของแพ็คเกจ 512 ไบต์และส่งข้อมูลทุก ๆ 500 มิลลิวินาที ดังรูปที่ 4.15

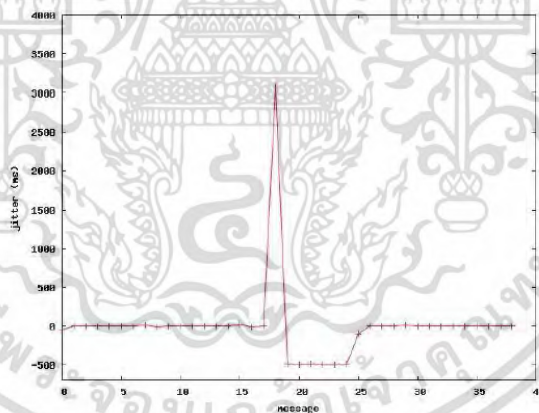


รูปที่ 4.15 วิธีการทดสอบค่าเวลาหน่วงไม่คงที่ของในงานวิจัยของ Jorge E. Luzuriaga และคณะ [6]

เนื่องจากชุดทดสอบที่พัฒนาขึ้นยังไม่สามารถทำการเปิด-ปิดตัวปล่อยสัญญาณได้อย่างที่งานวิจัยที่นำมาเปรียบเทียบ จึงใช้วิธีการกำหนดอัตราการเกิดข้อมูลสูญหายที่ร้อยละ 100 แทนการตัดการเชื่อมต่อและใช้การกำหนดอัตราการเกิดข้อมูลสูญหายที่ร้อยละ 0 ในการทำให้เครือข่ายกลับมาเชื่อมต่อกันได้ปกติ โดยตัวแปรตั้งต้นในการทดสอบสรุปได้ดังตารางที่ 4.7

ตารางที่ 4.7 ค่าตัวแปรตั้งต้นในการทดสอบที่ 7

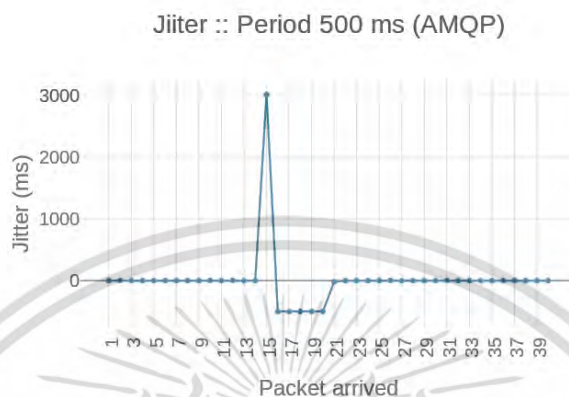
ตัวแปรตั้งต้น		
ขนาดแพ็คเกจ	512 ไบต์	
ความถี่การส่งแพ็คเกจ	500 มิลลิวินาที	
คุณภาพบริการ	MQTT ระดับ 0	AMQP ไม่ใช่
อัตราการสูญหายข้อมูลในเครือข่าย	คุมควบพิเศษ	
ค่าเวลาแฝงในการรับ/ส่งข้อมูล	ไม่มี	
ปริมาณการรับ/ส่งข้อมูลของการทดสอบเครือข่าย	ไม่จำกัด	



รูปที่ 4.16 ค่าเวลาหน่วงไปคงที่ของโพรโทคอล MQTT จากงานวิจัยเปรียบเทียบ [6]

จากการทดสอบโพรโทคอล MQTT ของ Jorge E. Luzuriaga และคณะ ซึ่งได้ทำการสลับจุดปล่อยสัญญาณในช่วงประมาณการส่งข้อมูลครั้งที่ 17 และได้ผลลัพธ์ดังรูปที่ 4.16 ซึ่งเห็นได้ชัดในช่วงการส่งข้อมูลในช่วงเริ่มต้นค่าเวลาหน่วงไม่คงที่จะอยู่ที่ประมาณ 0 มิลลิวินาที ซึ่งแสดงถึงการรับ-ส่งข้อมูลเป็นไปด้วยความราบเรียบสม่ำเสมอ ส่วนในครั้งที่ 18 มีค่าเวลาหน่วงไม่คงที่ที่ 3,000 มิลลิวินาทีซึ่งเป็น

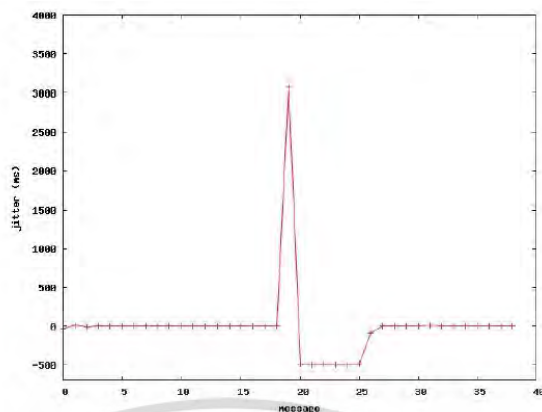
ช่วงเวลาที่การเชื่อมต่อกับมาเป็นปกติได้อีกครั้งและแพ็คเก็ตที่ 19 ถึง 24 มีค่าเวลาหน่วยไม่คงที่ที่ -500 มิลลิวินาที



รูปที่ 4.17 ค่าเวลาหน่วงไปคงที่ของโปรโตคอล MQTT จากงานชุดทดสอบ

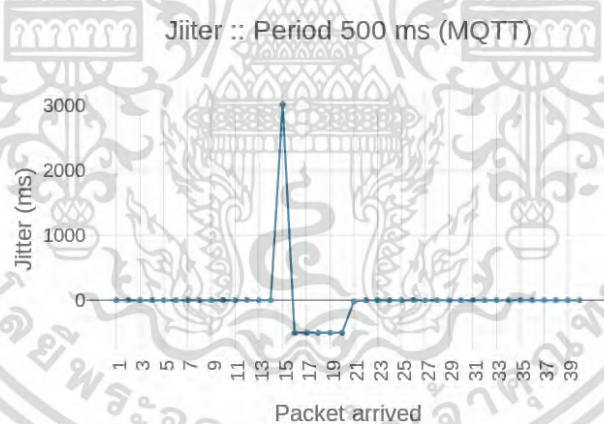
ผลลัพธ์การทดสอบจากชุดทดสอบในรูปที่ 4.17 ซึ่งได้นำโปรโตคอล MQTT มาทดสอบ เช่นเดียวกับงานวิจัยที่นำมาเปรียบเทียบ ซึ่งได้ทำการตัดการเชื่อมต่อที่การส่งแพ็คเก็ตครั้งที่ 15 และตัดการเชื่อมต่อเป็นเวลา 3 วินาทีเพื่อดูค่าเวลาหน่วงไม่คงที่ที่เกิดขึ้น ซึ่งผลลัพธ์ที่ออกมาให้ผลลัพธ์ที่ใกล้เคียงกับของ Jorge E. Luzuriaga และคณะ

การทดสอบโดยใช้โปรโตคอล AMQP จากงานวิจัยของ Jorge E. Luzuriaga และคณะ ซึ่งได้ทดสอบได้ใช้ตัวแปรตั้งต้น ดังตารางที่ 4.7 ซึ่งผลลัพธ์แสดงดังรูปที่ 4.18 แสดงให้เห็นว่าโปรโตคอล AMQP สามารถนำข้อมูลที่ถูกส่งขณะที่การเชื่อมต่อของเครือข่ายถูกตัดมาส่งเมื่อการเชื่อมต่อกับมาสู่สภาวะปกติได้ ครั้งที่ 19 มีค่าเวลาหน่วงไม่คงที่ที่ 3,000 มิลลิวินาที ซึ่งเป็นช่วงเวลาที่การเชื่อมต่อกับมาเป็นปกติได้อีกครั้งและแพ็คเก็ตที่ 20 ถึง 25 มีค่าเวลาหน่วยไม่คงที่ที่ -500 มิลลิวินาที



รูปที่ 4.18 ค่าเวลาหน่วงไม่คงที่ของโพรโทคอล AMQP จากงานวิจัยเปรียบเทียบ [6]

ส่วนผลลัพธ์จากการทดสอบด้วยชุดทดสอบในรูปที่ 4.19 โดยใช้โพรโทคอล AMQP และทำการทดสอบตามการทดสอบในงานวิจัย Jorge E. Luzuriaga และคณะ ซึ่งผลลัพธ์ออกมาคล้ายคลึงกับผลลัพธ์ที่ได้จากการทดสอบโพรโทคอล MQTT



รูปที่ 4.19 ค่าเวลาหน่วงไม่คงที่ของโพรโทคอล AMQP จากชุดทดสอบ

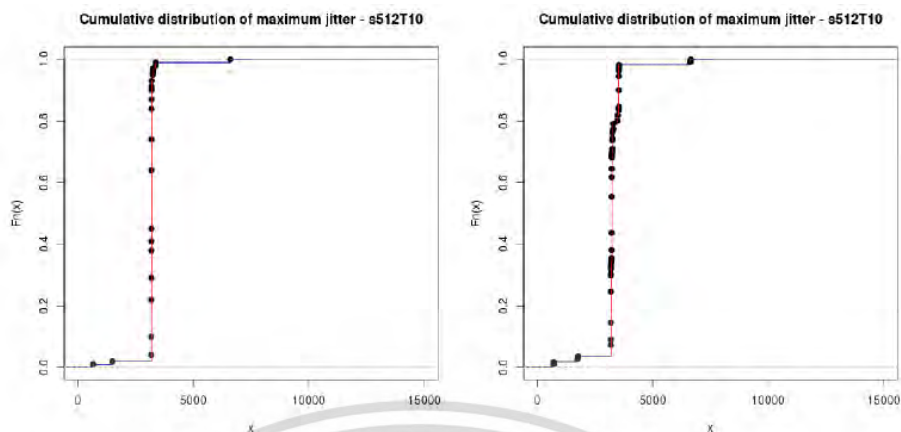
Jorge E. Luzuriaga และคณะ ได้อธิบายไว้ว่าทั้งโพรโทคอล MQTT และ AMQP มีการจัดการแก้ไขปัญหาเมื่อการเชื่อมต่อของเครือข่ายถูกตัดขาด โพรโทคอลทั้ง 2 ได้ทำการเก็บลำดับของข้อมูลที่ถูกรับและส่งและทำการส่งออกเมื่อการเชื่อมต่อของเครือข่ายกลับมาเป็นปกติ และอธิบายการเกิดค่าเวลาหน่วงไม่คงที่ในช่วง 3,000 มิลลิวินาที ไว้ว่าจากการส่งข้อมูลด้วยความถี่ที่ 500 มิลลิวินาที เมื่อมีแพ็คเก็ตถูกส่งมาหลังจากแพ็คเก็ตก่อนหน้า 3,000 มิลลิวินาที จะประมาณได้ว่าการมาของแพ็คเก็ตนั้นจะมีแพ็ค

เกิดที่ทำการส่งในขณะที่การเชื่อมต่อของเครือข่ายถูกตัดหรือมีปัญหาเป็นจำนวนประมาณ 6 แพ็คเก็ต และการเกิดค่าเวลาหน่วงไม่คงที่ -500 ซึ่งเท่ากับกับความถี่ในการส่งข้อมูลเป็นการบอกว่าแพ็คเก็ตนั้นถูกส่งมาพร้อมกันกับแพ็คเก็ตก่อนหน้า และการทดสอบด้วยชุดทดสอบจะเห็นได้ว่าทั้งโปรโตคอล MQTT และ AMQP เมื่อมีการตัดการเชื่อมต่อหลังการส่งข้อมูลครั้งที่ 14 เป็นเวลา 3 วินาที ท าให้แพ็คเก็ตที่ 15 – 20 ถูกส่งมาพร้อมกันเมื่อการเชื่อมต่อกับมาและล าดับของข้อมูลยังคงเป็นไปตามล าดับตามเดิม

การทดสอบที่ 8 การทดสอบวัดค่าเวลาหน่วงไม่คงที่ของโปรโตคอล MQTT และ AMQP กับขนาดของแพ็คเก็ตและความถี่ในการส่งซึ่งในงานวิจัยของ Jorge E. Luzuriaga และคณะ ได้ทำการเปรียบเทียบโดยใช้ขนาดของแพ็คเก็ต 512 ไบต์ และ 6K ไบต์ และใช้ความถี่ในการส่งข้อมูล 10 มิลลิวินาที โดยสรุปค่าตัวแปรตั้งต้นดังตารางที่ 4.8

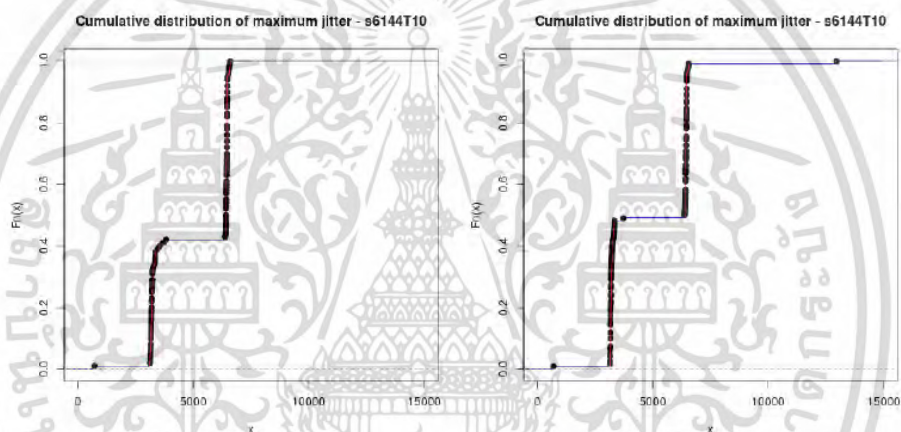
ตารางที่ 4.8 ค่าตัวแปรตั้งต้นในการทดสอบที่ 8

ตัวแปรตั้งต้น	
ขนาดแพ็คเก็ต	512 และ 6K ไบต์
ความถี่การส่งแพ็คเก็ต	10 และ 500 มิลลิวินาที
คุณภาพบริการ	MQTT ระดับ 0 AMQP ไม่ใช่
อัตราการสูญหายข้อมูลในเครือข่าย	คุมควบพิเศษ
ค่าเวลาแฝงในการรับ/ส่งข้อมูล	ไม่มี
ปริมาณการรับ/ส่งข้อมูลของการดเครือข่าย	ไม่จ ากัด



(a) message size = 512 Bytes

(b) message size = 512 Bytes



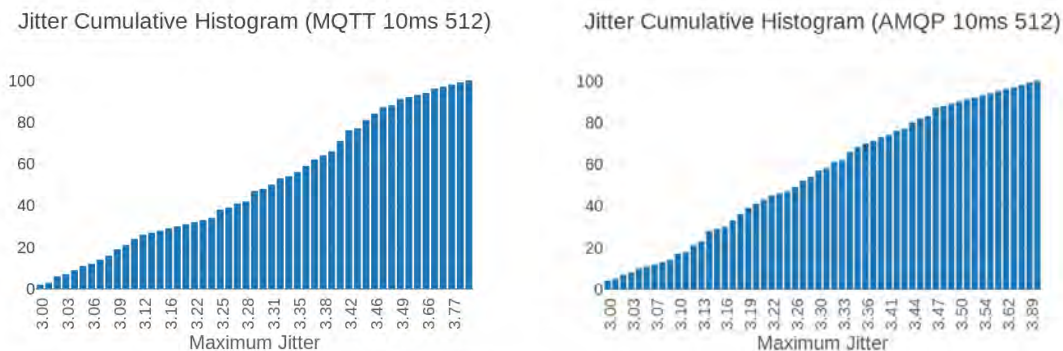
(c) message size = 6 KBytes

(d) message size = 6 KBytes

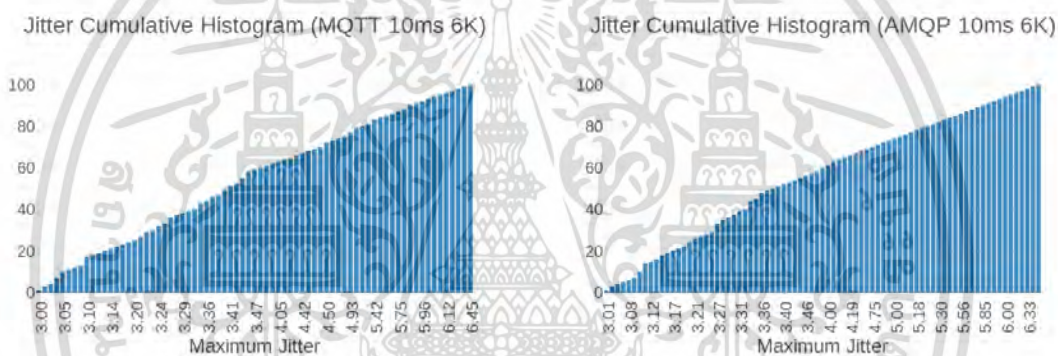
รูปที่ 4.20 ค่าเวลาหน่วงไม่คงที่ที่มากที่สุดของโปรโตคอล MQTT(ขวา) และ AMQP(ซ้าย) ที่ความถี่ในการส่ง 10 มิลลิวินาที กับขนาดแพ็คเก็ตที่ 512 ไบต์และ 6K ไบต์ จากงานวิจัยเปรียบเทียบ [6]

จากรูปที่ 4.20 แสดงการทดสอบโปรโตคอล MQTT และ AMQP Jorge E. Luzuriaga และคณะอธิบายว่าในการส่งข้อมูลที่ความถี่ 10 มิลลิวินาที ในการส่งแพ็คเก็ตขนาด 512 ไบต์ จากการทดสอบทั้งหมดโปรโตคอลทั้งสองมีค่าเวลาหน่วงไม่คงที่มากที่สุดประมาณ 3.3 วินาที แต่เมื่อขนาดแพ็คเก็ตเพิ่มเป็น 6K ไบต์ ค่าเวลาหน่วงไม่คงที่ครั้งหนึ่งของการทดสอบมีค่าประมาณ 3.3 วินาที และอีกครั้งหนึ่งที่ 7 วินาที โดย Jorge E. Luzuriaga และคณะให้เหตุผลว่าเวลาที่ 7 วินาที มาจากการแยกกันของแพ็คเก็ตการส่งข้อมูลขนาด 6K ไบต์ด้วยความถี่ที่ 10 มิลลิวินาที หากเครือข่ายหรือการรับ-ส่งข้อมูลของอุปกรณ์ไม่สามารถรองรับได้อาจทำให้ระบบหยุดการดำเนินงานลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 ผลการทดสอบค่าเวลาหน่วงไม่คงที่ที่มากที่สุดของโพรโทคอล MQTT(ซ้าย) และ AMQP(ขวา) ที่ความถี่ในการส่ง 10 มิลลิวินาที กับขนาดแพ็คเก็ตที่ 512 ไบต์ จากชุดทดสอบ



รูปที่ 4.22 ผลการทดสอบค่าเวลาหน่วงไม่คงที่ที่มากที่สุดของโพรโทคอล MQTT(ซ้าย) และ AMQP(ขวา) ที่ความถี่ในการส่ง 10 มิลลิวินาที กับขนาดแพ็คเก็ตที่ 6K ไบต์ จากชุดทดสอบ

จากรูปที่ 4.21 คือผลการทดสอบด้วยชุดทดสอบจากกราฟแสดงค่าเวลาหน่วงไม่คงที่ในการทดสอบทั้งหมด 100 ครั้งแบบ Cumulative Histogram แสดงให้เห็นว่าการส่งข้อมูลขนาด 512 ไบต์ ทั้งในโพรโทคอล MQTT และ AMQP เกิดค่าเวลาหน่วงไม่คงที่มากที่สุดอยู่ในช่วง 3 – 3.8 วินาที ทั้งโพรโทคอล MQTT และ AMQP ร้อยละ 80 ของค่าเวลาหน่วงไม่คงที่อยู่ในช่วง 3.00 – 3.50 วินาที และจากรูปที่ 4.22 เป็นผลการทดสอบจากชุดทดสอบด้วยการส่งข้อมูล 6K ไบต์ โพรโทคอลทั้งสองมีค่าเกิดค่าเวลาหน่วงไม่คงที่อยู่ในช่วงประมาณ 3 – 6.4 วินาที โดยประมาณร้อยละ 70 ค่าเวลาหน่วงไม่คงที่อยู่ในช่วง 3 – 4 วินาที ร้อยละ 15 อยู่ในช่วง 5 – 6 วินาที และอีกร้อยละ 15 อยู่ในช่วงมากกว่า 6 วินาที

ในการทดสอบค่าเวลาหน่วงไม่คงที่ที่สามารถใช้บอกความสัมพันธ์ในการรับ-ส่งข้อมูลของระบบ เช่นระบบที่ต้องการความเป็นเวลาจริง หากเลือกความถี่ในการส่งข้อมูลต่ำจะทำให้เสียความเป็นเวลาจริง

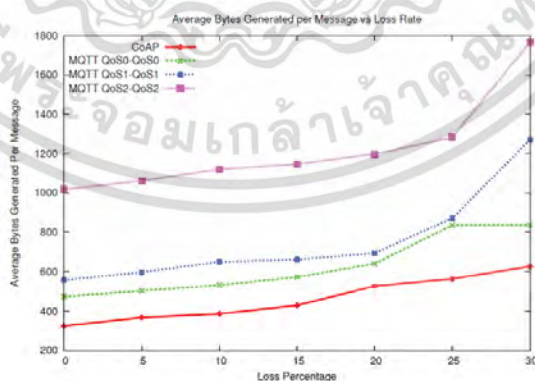
แต่หากเลือกความถี่ที่สูงเกินไป ซึ่งอาจสูงเกินกว่าอุปกรณ์สื่อสารจะรองรับได้ อาจทำให้ระบบทำงานไม่ราบรื่นหรือจนกระทั่งหยุดการทำงาน ดังนั้นจึงต้องหาจุดเหมาะสมของระบบเพื่อให้การรับ-ส่งข้อมูลเป็นไปอย่างราบรื่น

4.1.4 ข้อมูลส่วนเกิน (Overhead)

การทดสอบที่ 9 การทดสอบค่าข้อมูลส่วนเกินเมื่อกำหนดอัตราการสูญหายของข้อมูลในเครือข่าย โดยงานวิจัยของ Dinesh Thangavel และคณะซึ่งได้ทำการทดสอบโปรโตคอล MQTT และ CoAP ในเครือข่ายแบบสาย ส่วนการทดสอบโดยใช้ชุดทดสอบจะทำการทดสอบในเครือข่ายแบบสายและแบบไร้สาย โดยได้กำหนดค่าตัวแปรตั้งต้นดังตารางที่ 4.9

ตารางที่ 4.9 ค่าตัวแปรตั้งต้นในการทดสอบที่ 9

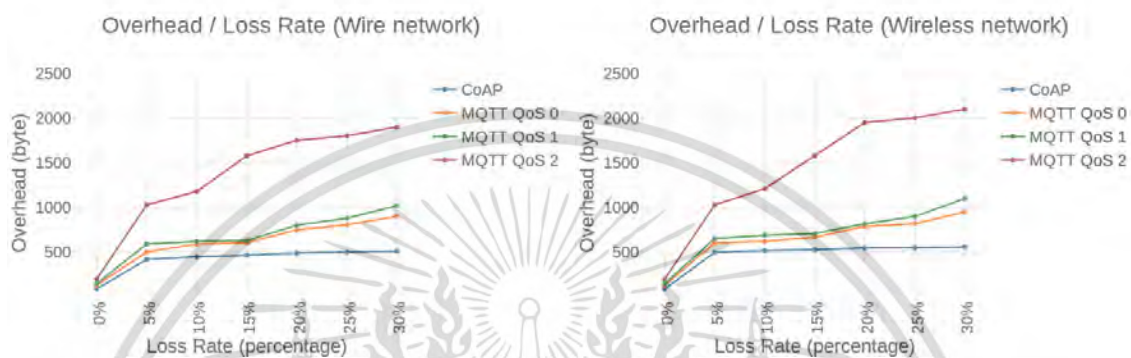
ตัวแปรตั้งต้น	
ขนาดแพ็คเกจ	60 ไบต์
ความถี่การส่งแพ็คเกจ	200 มิลลิวินาที
คุณภาพบริการ	MQTT ระดับ 0, 1 CoAP ใช้
อัตราการสูญหายข้อมูลในเครือข่าย	ร้อยละ 0 - 30
คาเวลาแฝงในการรับ/ส่งข้อมูล	ไม่มี
ปริมาณการรับ/ส่งข้อมูลของการ์ดเครือข่าย	ไม่มีจำกัด



รูปที่ 4.23 ข้อมูลส่วนเกินของโปรโตคอล CoAP และ MQTT เมื่ออัตราการสูญหายข้อมูลของเครือข่ายแตกต่างกันจากงานวิจัยเปรียบเทียบ [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดสอบจากงานวิจัยของ Dinesh Thangavel และคณะจากรูปที่ 4.23 คือการทดสอบส่งข้อมูลขนาด 60 ไบต์ในสภาพแวดล้อมเครือข่ายมีอัตราการสูญหายของข้อมูล ร้อยละ 0 5 10 15 20 25 30 ตามล าดับผลคือโพรโทคอล CoAP มีค่าเฉลี่ยของสร้างข้อมูลส่วนเกินน้อยที่สุดในทุก ๆ อัตราการสูญหายของข้อมูล ตามมาด้วยโพรโทคอล MQTT ที่ใช้ค่าคุณภาพบริการที่ระดับ 0 1 และ 2 ตั่วล าดับ



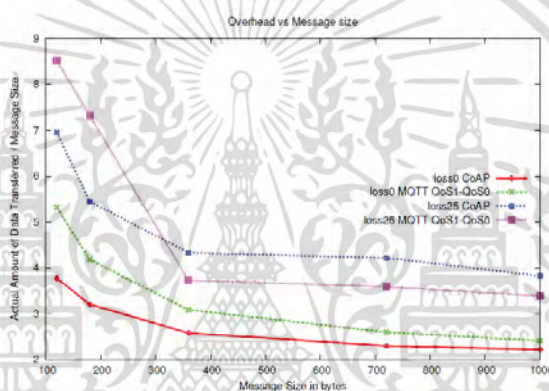
รูปที่ 4.24 ข้อมูลส่วนเกินของโพรโทคอล CoAP และ MQTT เมื่ออัตราการสูญหายข้อมูลของเครือข่ายแตกต่างกันจากชุดทดสอบในเครือข่ายแบบสาย (ข่าย) และไร้สาย (ขวา)

ผลลัพธ์จากการทดสอบด้วยชุดทดสอบดังรูปที่ 4.24 โดยผลที่ออกมามีแนวโน้มไปในทิศทางเดียวกันทั้งในเครือข่ายแบบสายและไร้สาย โดยโพรโทคอล CoAP มีค่าเฉลี่ยการสร้างข้อมูลส่วนเกินน้อยที่สุด และโพรโทคอล MQTT ที่ใช้เลือกใช้ค่าคุณภาพบริการระดับ 2 มีค่าเฉลี่ยการสร้างข้อมูลส่วนเกินมากที่สุด โดยในเครือข่ายแบบไร้สายเมื่อมีอัตราการสูญหายข้อมูลเพิ่มมากขึ้นจะมีข้อมูลส่วนเกินมากกว่าเครือข่ายแบบสาย ส่วนสาเหตุที่โพรโทคอล MQTT มีข้อมูลส่วนเกินมากกว่า CoAP เกิดจากเมื่อเกิดอัตราการสูญหายของข้อมูลในเครือข่าย ทำให้โพรโทคอล MQTT ซึ่งเป็น TCP-base จะทำการส่งข้อมูลใหม่เมื่อปลายทางไม่ตอบกลับถึงการได้รับข้อมูล และด้วยกลไกการรับ-ส่งข้อมูลในคุณภาพบริการระดับที่ 1 และ 2 จะมีกลไกตอบโต้กัน 2 ครั้งและ 4 ครั้งตามล าดับทำให้เกิดเป็นข้อมูลส่วนเกิน

การทดสอบที่ 10 การทดสอบอัตราส่วนของข้อมูลส่วนเกินกับข้อมูล ได้ยกตัวอย่างงานวิจัยของ Dinesh Thangavel และคณะซึ่งได้ทำการทดสอบโพรโทคอล MQTT และ CoAP ซึ่งได้ทำการทดสอบในเครือข่ายแบบสาย ส่วนการทดสอบโดยใช้ชุดทดสอบจะทำการทดสอบในเครือข่ายแบบสายและแบบไร้สาย โดยได้กำหนดค่าตามปรังตั้งต้นดังตารางที่ 4.10

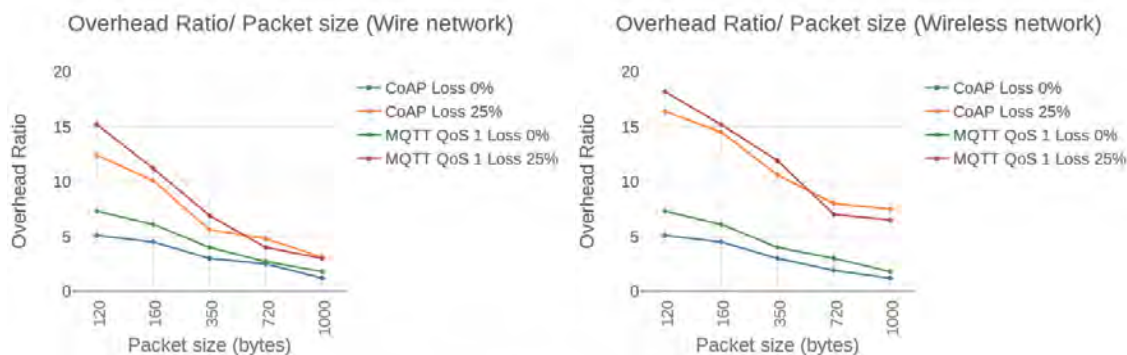
ตารางที่ 4.10 ค่าตัวแปรตั้งต้นในการทดสอบที่ 10

ตัวแปรตั้งต้น	
ขนาดแพ็คเกจ	120 160 350 720 และ 1K ไบต์
ความถี่การส่งแพ็คเกจ	200 มิลลิวินาที
คุณภาพบริการ	MQTT ระดับ 0, 1 CoAP ใช้
อัตราการสูญหายข้อมูลในเครือข่าย	ร้อยละ 0 และ 25
ค่าเวลาแฝงในการรับ/ส่งข้อมูล	ไม่มี
ปริมาณการรับ/ส่งข้อมูลของการ์ดเครือข่าย	ไม่จำกัด



รูปที่ 4.25 อัตราส่วนของข้อมูลกับข้อมูลที่ถูกส่งของของโพรโทคอล CoAP และ MQTT จากงานวิจัยเปรียบเทียบ [4]

ผลลัพธ์ที่ Dinesh Thangavel และคณะนำเสนอค่าอัตราส่วนระหว่างขนาดข้อมูลกับขนาดข้อมูลที่ถูกออกไปจริงๆ ผลลัพธ์ที่แสดงในรูปที่ 4.25 แสดงให้เห็นว่าในสภาพแวดล้อมเครือข่ายปกติ การส่งแพ็คเกจเมื่อเทียบอัตราส่วนระหว่างข้อมูลที่ส่งกับข้อมูลที่รวมส่วนหัวโพรโทคอลและข้อมูลอื่นๆที่ใช้ประกอบการรับ-ส่งแล้ว การส่งแพ็คเกจขนาดเล็กมีอัตราส่วนของข้อมูลส่วนเกินมากกว่าการส่งแพ็คเกจขนาดใหญ่ ทั้งในโพรโทคอล CoAP และ MQTT และพบว่าเมื่อมีอัตราการสูญหายข้อมูลในเครือข่ายจะทำให้มีข้อมูลส่วนเกินเพิ่มขึ้น และสภาพแวดล้อมที่มีอัตราการสูญหายข้อมูล การส่งแพ็คเกจขนาดที่น้อยกว่า 300 ไบต์ โพรโทคอล CoAP มีค่าข้อมูลส่วนเกินน้อยกว่า MQTT แต่ในการส่งแพ็คเกจขนาดที่มากกว่า 300 ไบต์ โพรโทคอล CoAP มีค่าข้อมูลส่วนเกินมากกว่า MQTT

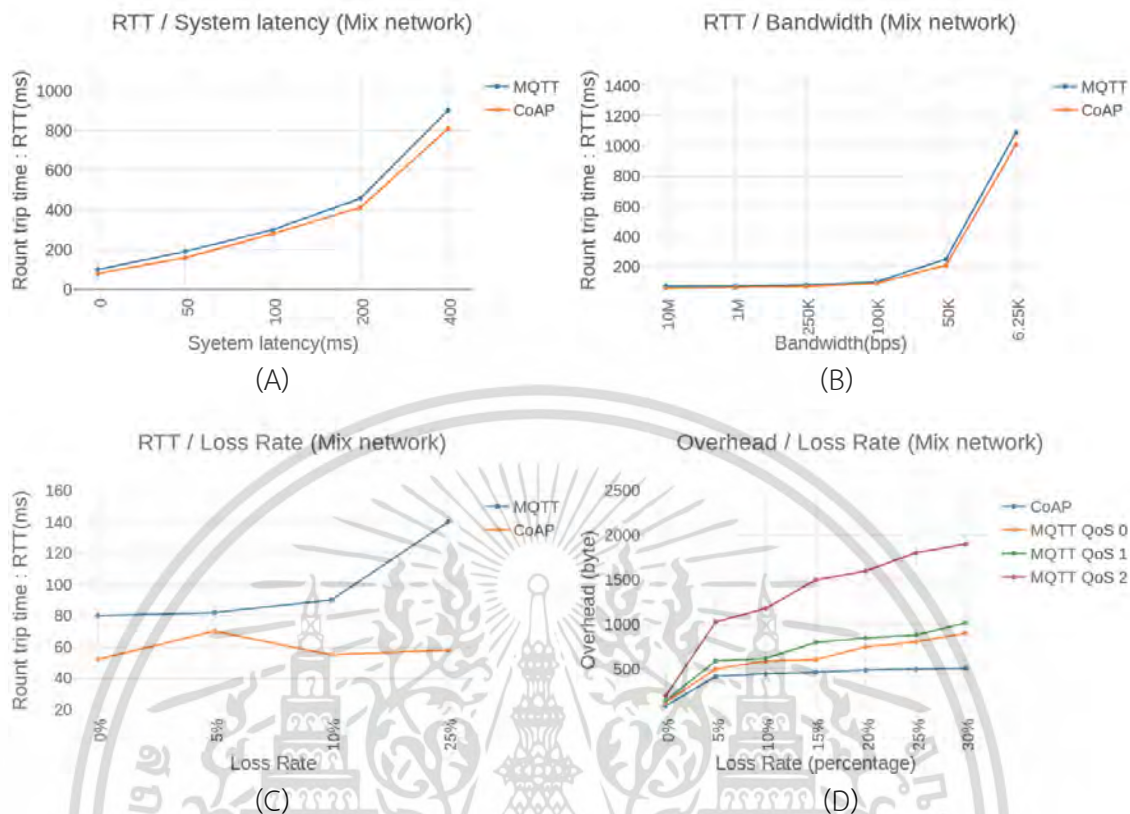


รูปที่ 4.26 อัตราส่วนของข้อมูลกับข้อมูลที่ถูกส่งของของโพรโทคอล CoAP และ MQTT จากจากชุดทดสอบในเครือข่ายแบบสาย(ซ้าย) และไร้สาย(ขวา)

ผลการทดสอบจากชุดทดสอบดังรูปที่ 4.26 ให้แนวโน้มผลลัพธ์ที่ใกล้เคียงกับงานวิจัยของ Dinesh Thangavel และคณะ โดยผลลัพธ์จากชุดทดสอบ พบว่าในสภาพแวดล้อมที่กำหนดอัตราการเกิดข้อมูลสูญหายร้อยละ 25 โพรโทคอล MQTT จะมีอัตราส่วนของข้อมูลส่วนเกินน้อยกว่าโพรโทคอล CoAP ในช่วงขนาดของแพ็คเกจ 720 ไบต์ โดยสาเหตุที่ท ให้โพรโทคอล MQTT มีอัตราส่วนของข้อมูลส่วนเกินน้อยกว่าโพรโทคอล CoAP เมื่อมีอัตราการสูญหายข้อมูลในเครือข่าย จากการศึกษางานวิจัยที่นำมาทดสอบมีความเป็นไปได้ว่า ในสภาพแวดล้อมที่มีอัตราการสูญหายข้อมูลในเครือข่าย โพรโทคอล CoAP เกิดข้อมูลสูญหายมากกว่า MQTT ท ให้ CoAP ซึ่งเปิดให้คุณภาพบริการซึ่งจะท การส่งข้อมูลใหม่ จึงเป็นสาเหตุให้มีอัตราส่วนของข้อมูลส่วนเกินมากขึ้น

4.1.5 การทดสอบชุดทดสอบกับเครือข่ายแบบผสม (Mix network)

จากการออกแบบชุดทดสอบวัดประสิทธิภาพโพรโทคอลไอโอทีจะสามารถรองรับการทดสอบกับเครือข่ายได้ทั้งหมด 3 แบบคือ เครือข่ายแบบสาย เครือข่ายแบบไร้สายและเครือข่ายแบบผสมคือการใช้เครือข่ายแบบสายและแบบไร้สายรวมด้วยกัน ในส่วนของการทดสอบกับเครือข่ายแบบสายและไร้สายได้แสดงตัวอย่างการทดสอบโดยชุดทดสอบซึ่งนำมาเปรียบเทียบแนวโน้มในการทดสอบตัวแปรวัดผลกับงานวิจัยที่ได้หยิบยกมาเปรียบซึ่งจะอยู่ในหัวข้อที่ 4.1.1 – 4.1.4 และในส่วนนี้จะเป็นการน แสดงตัวอย่างผลการทดสอบโดยชุดทดสอบกับเครือข่ายแบบผสม



รูปที่ 4.27 การทดสอบในเครือข่ายแบบผสม

ในการทดสอบโดยใช้ชุดทดสอบในเครือข่ายแบบผสมโดยได้ทำการทดสอบ ตัวแปรวัดผล เวลาในการส่งข้อมูลไป-กลับ (RTT) เมื่อมีค่าเวลาแฝงในระบบ (System latency) เพิ่มขึ้น แสดงดังรูปที่ 4.27 (A) เวลาในการส่งข้อมูลไป-กลับ (RTT) เมื่อมีค่าความกว้างของเครือข่าย (Bandwidth) ลดลงแสดงดังรูปที่ 4.27 (B) เวลาในการส่งข้อมูลไป-กลับ (RTT) เมื่อมีอัตราการสูญหายของข้อมูลในเครือข่าย (Loss rate) เพิ่มขึ้น แสดงดังรูปที่ 4.27 (C) ข้อมูลส่วนเกิน (Overhead) เมื่อมีอัตราการสูญหายของข้อมูลในเครือข่าย (Loss rate) เพิ่มขึ้น แสดงดังรูปที่ 4.27 (D) โดยความสัมพันธ์ของผลลัพธ์จากการทดสอบในเครือข่ายแบบผสมมีแนวโน้มในทิศทางเดียวกันกับการทดสอบในเครือข่ายแบบสายและไร้สาย ซึ่งเป็นการทดสอบการรับรองการท างานในเครือข่ายแบบต่าง ๆ ของชุดทดสอบตามที่ได้ออกแบบ

4.2 เปรียบเทียบการเตรียมการแบบดั้งเดิมกับใช้ชุดทดสอบโพรโทคอลไอโอที

จากหัวข้อที่แล้วได้นำเสนอผลลัพธ์ในการทดสอบค่าตัวแปรวัดผลต่าง ๆ จากชุดทดสอบด้วยการเปรียบเทียบกับทดสอบจากงานวิจัยที่เกี่ยวข้องกับการทดสอบโพรโทคอลไอโอที เป็นที่เรียบร้อยแล้ว ในหัวข้อนี้จะทบทวนสรุปเปรียบเทียบในการเตรียมการสำหรับทดสอบโพรโทคอลไอโอทีโดยใช้อุปกรณ์จริงหรืออุปกรณ์จำลองในรูปแบบดั้งเดิมกับการเตรียมการโดยใช้ชุดทดสอบโพรโทคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลองโดยในการเปรียบเทียบสรุปได้ดังตารางที่ 4.1

ตารางที่ 4.1 เปรียบเทียบการเตรียมการแบบดั้งเดิมกับใช้ชุดทดสอบโพรโทคอลไอโอที

หัวข้อ	วิธีดั้งเดิม	ชุดทดสอบโพรโทคอลไอโอที
การทดสอบโพรโทคอลไอโอทีมาทดสอบ	ผู้ทดสอบจำเป็นต้องพัฒนาโปรแกรมในการเรียกใช้ API ของโพรโทคอลไอโอทีนั้น ๆ เพื่อนำมาทดสอบ ดังนั้นจำเป็นต้องมีความเข้าใจในการเขียนโปรแกรมคอมพิวเตอร์	มีโพรโทคอลไอโอทีให้เลือกใช้ได้แบบสำเร็จรูป MQTT MQTT-SN CoAP AMQP ทั้งสำหรับลูกข่ายและแม่ข่ายนอกจากนี้ผู้ทดสอบยังสามารถนำโพรโทคอลที่สนใจมาพัฒนาเพิ่มเติมเพื่อทดสอบได้
ตัวแปรตั้งต้น	ขึ้นอยู่กับความต้องการของผู้ทดสอบ	จำนวนครั้งในการส่งข้อมูล, ขนาดของข้อมูลที่ส่ง, ความถี่ในการส่งข้อมูล, คุณภาพบริการ, อัตราการสูญหายของข้อมูลในเครือข่าย, ค่าเวลาหน่วงในเครือข่าย, ค่าความกว้างของเครือข่าย
ส่วนควบคุมการส่งข้อมูล	ผู้ทดสอบจำเป็นต้องพัฒนาขึ้นเอง	จัดการแบบอัตโนมัติ
ส่วนควบคุมสภาพแวดล้อม	ผู้ทดสอบจำเป็นต้องพัฒนาขึ้นเอง	จัดการแบบอัตโนมัติ
ตัวแปรวัดผล	ขึ้นอยู่กับความต้องการของผู้ทดสอบ	เวลาในการรับ-ส่งข้อมูล, อัตราการสูญหายของข้อมูล, ค่าเวลาหน่วงไม่คงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.11 (ต่อ)

หัวข้อ	วิธีดั้งเดิม	ชุดทดสอบโพรโทคอลไอโอที
รูปแบบการเก็บผลลัพธ์	ขึ้นอยู่กับผู้ทดสอบ อาจจะเป็น log ไฟล์ หรือ ฐานข้อมูล	ฐานข้อมูลแบบ SQL สามารถนำไปใช้ได้สะดวก
แสดงผลลัพธ์	ผู้ทดสอบจำเป็นต้องนำผลลัพธ์มาแสดงผลเอง	มีระบบแสดงผลลัพธ์ได้ทันทีเมื่อการทดสอบเสร็จสิ้นด้วยเว็บเบราว์เซอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

บทนี้ประกอบด้วย 2 หัวข้อได้แก่ สรุปผลการวิจัยพัฒนาชุดทดสอบโพรโทคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลอง และ ข้อเสนอแนะสำหรับการพัฒนาต่อยอด ซึ่งมีเนื้อหาดังต่อไปนี้

5.1 สรุปผลการวิจัย

จุดเริ่มต้นของการวิจัยพัฒนาชุดทดสอบโพรโทคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลองเริ่มมาจากการศึกษาวิจัยที่เกี่ยวข้องกับการทดสอบโพรโทคอลโดยใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง โดยแต่ละงานวิจัยก็ได้นำเสนอ ปัญหา วิธีการทดสอบ และผลการทดสอบ และจากการศึกษาพบว่า ในกระบวนการทดสอบนั้นยังไม่มีเครื่องมือที่ช่วยลดขั้นตอนการเตรียมการในกระบวนการต่าง ๆ เช่น การควบคุมการรับส่งข้อมูล การควบคุมสภาพแวดล้อม การคำนวณผลและการเก็บผล และการแสดงผลลัพธ์ การเตรียมการเหล่านี้ผู้ทดสอบจะต้องจัดเตรียมขึ้นเองทั้งหมด แม้ในบางกระบวนการอาจจะมีเครื่องมือที่ช่วยในการทำงานอยู่บ้าง แต่ก็ยังไม่สามารถทำการทดสอบได้แบบอัตโนมัติ อีกทั้งในแต่ละงานวิจัยจะให้ความสนใจกับค่าตัวแปรวัดผลต่างกัน ดังนั้นชุดทดสอบโพรโทคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลองนี้จะสามารถทำการทดสอบและคำนวณค่าตัวแปรวัดผลพื้นฐานที่สำคัญ โดยในการใช้งานผู้ทดสอบสามารถเลือกโพรโทคอลไอโอที MQTT MQTT-SN CoAP AMQP ในการทดสอบได้ทันที โดยที่ไม่ต้องทำการพัฒนาโปรแกรมใด ๆ และสามารถกำหนดตัวแปรตั้งต้น การส่งข้อมูลและสภาพแวดล้อมเครือข่าย ในตอนเริ่มทดสอบ ชุดทดสอบจะทำการทดสอบในทุก ๆ เงื่อนไขที่กำหนดไว้ คำนวณตัวแปรวัดผล พร้อมบันทึกผลลัพธ์ลงในฐานข้อมูล จากนั้นแสดงผลลัพธ์ในรูปแบบกราฟเมื่อกระบวนการทดสอบสิ้นสุดลง ด้วยความพร้อมใช้ และความสามารถในการทดสอบแบบอัตโนมัติ ทำให้ชุดทดสอบโพรโทคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลองสามารถช่วยลดภาระงานในการเตรียมการทดสอบ

5.2 ข้อเสนอแนะในการพัฒนาต่อยอด

ในการพัฒนาชุดทดสอบโพรโทคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลอง ยังมีสิ่งที่น่าสนใจให้สามารถพัฒนาต่อไปได้โดยจะขอสรุปเป็นหัวข้อ ดังนี้

1. ชุดทดสอบโพรโทคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลองในขณะนี้จะรองรับอุปกรณ์จริงได้เพียง Desktop PC, Laptop, Server และ Linux Embedded จะยังไม่รองรับอุปกรณ์จำพวก Microcontroller ในส่วนนี้เป็นจุดที่น่าสนใจในการพัฒนาต่อ
2. ชุดทดสอบโพรโทคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลองในขณะนี้ รองรับการเชื่อมต่ออุปกรณ์ทดสอบได้มากที่สุด 3 อุปกรณ์เท่านั้นคือ 1 ตัวส่ง 1 ตัวรับ และ 1 แม่ข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่บนสื่อสาธารณะ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาให้สามารถรองรับการทดสอบแบบมีการส่งรับข้อมูลมากกว่า 2 ตัว ในส่วนนี้เป็นจุดที่น่าสนใจในการพัฒนาต่อ

3. ชุดทดสอบโพรโทคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลองในขณะนี้ รับรองการทดสอบตัวโพรโทคอล MQTT MQTT-SN CoAP AMQP ซึ่งในอนาคตจะต้องมีโพรโทคอลไอโอทีชนิดใหม่ถูกพัฒนาขึ้น การพัฒนาชุดทดสอบในรับรองกับโพรโทคอลใหม่ ๆ เป็นสิ่งที่น่าสนใจในการพัฒนาต่อ
4. พัฒนาเพิ่มตัวแปรสำคัญอื่น ๆ หรือการวิเคราะห์ผล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Saadeh H., Almobaideen W., and Sabri K. E. “Internet of Things: A review to support IoT architecture’s design”. **2nd Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS)**. 2017. pp.1-7.
- [2] AL-Shargabi B., Sabri O. “Internet of Things: an Exploration Study of Opportunities and Challenges”. **International Conference on Engineering & MIS (ICEMIS)**. 2017. pp.1-4.
- [3] Lee S., Kim H., Hong D.-k. “Correlation analysis of MQTT loss and delay according to QoS level” **Information Networking(ICOIN)**. 2013. pp.714-717.
- [4] Thagavel D., Ma X., Valera A. “Performance evaluation of MQTT and CoAP via a common middleware” **Intelligent Sensors Sensor Networks and Information Processing (ISSNIP)**. 2014. pp.1-6.
- [5] Chen Y., Kunz T. “Performance evaluation of IoT protocols under a constrained wireless access network” **Selected Topics in Mobile & Wireless Networking (MoWNet)**. 2016. pp.1-7.
- [6] Luzuriaga J. E., Perez M., Boronat P. “A comparative evaluation of AMQP and MQTT protocols over unstable and mobile network” **Consumer Communications and Networking Conference(CCNC)**. 2015. pp.931-936.
- [7] Govindan K., Azad A. P. “End-to-end service assurance in IoT MQTT-SN” **12th Annual IEEE Consumer Communications and Networking Conference(CCNC)**. 2015. pp.290-296.
- [8] Python 2.7.14, “Python”. [Online]. Available : <https://www.python.org/>. [Accessed 30 March 2018].

- [9] Eclipse Foundation. 2018. “**Eclipse Mosquitto**”. [Online]. Available : <https://projects.eclipse.org/projects/technology.mosquitto>. [Accessed 30 March 2018].
- [10] Giacomo T. 2018. “**CoAPthon**”. [Online]. Available : <https://github.com/Tanganelli/CoAPthon>. [Accessed 30 March 2018].
- [12] The Linux Foundation. 2018. “**NetEM**”. [Online]. Available : <https://wiki.linuxfoundation.org/networking/netem>. [Accessed 30 March 2018].
- [13] TATA Consultancy Services. 2018. “**WANem**”. [Online]. Available : <http://wanem.sourceforge.net/>. [Accessed 30 March 2018].
- [14] VMware Inc. 2018. “**VMware**”. [Online]. Available : <https://www.vmware.com/asean.html>. [Accessed 30 March 2018].
- [15] ORACLE. 2018. “**VirtualBox**”. [Online]. Available : <https://www.virtualbox.org/>. [Accessed 30 March 2018].
- [16] QEMU. 2018. “**QEMU**”. [Online]. Available : <https://www.qemu.org/>. [Accessed 30 March 2018].
- [17] Python Software Foundation. 2018. “**Socket-Low-level networking interface**”. [Online]. Available : <https://docs.python.org/2/library/socket.html>. [Accessed 30 March 2018].
- [18] DigitalOcean. 2018. “**MySQL on Ubuntu 16.04**”. [Online]. Available : <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-16-04>. [Accessed 30 March 2018].
- [19] Plotly. 2018. “**Plotly**”. [Online]. Available : <https://plot.ly/javascript/>. [Accessed 30 March 2018].
- [20] The Linux Foundation. 2018. “**Node Js**”. [Online]. Available : <https://nodejs.org/en/>. [Accessed 30 March 2018].

- [21] www.itilttd-india.com. [Online] .Availabie : <http://www.itilttd-india.com/loT/loT%20Verticals.jpg> . [Accessed 30 March 2018].
- [22] Pakanati C., Padmavathamma M., Reddy N. R. “Performance Comparison of TCP, UDP, and TFRC in Wired Networks” **IEEE International Conference on Computational Intelligence and Communication Technology**. 2016. pp.257-263.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดวัดประสิทธิภาพโพรโทคอลไอโอที

สำหรับการทดสอบโดยใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง

IoT Protocol Performance Testing Suite for Real Equipment or Virtual Equipment

วิศิษฐ์ ฟู่อาลี¹ วัชรระ ด้ตรวิริยะ¹

¹สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

1 ซอยฉลองกรุง 1 เขตลาดกระบัง กรุงเทพมหานคร 10520

บทคัดย่อ

วิธีการทดลองวัดประสิทธิภาพของโพรโทคอลไอโอทีโดยใช้
อุปกรณ์จริงหรืออุปกรณ์จำลอง ซึ่งเป็นวิธีที่นิยมกันในปัจจุบันเนื่องจาก
เป็นการวิธีที่สะดวกและให้ผลลัพธ์ที่น่าเชื่อถือ จากการศึกษางานวิจัยที่
เกี่ยวข้องพบว่าส่วนใหญ่แล้วผู้วิจัยจะต้องพัฒนาโปรแกรมเพื่อดำเนินการ
การทดลอง เช่น ควบคุมการรับส่งข้อมูล ควบคุมสภาพแวดล้อมเครือข่าย
การเก็บผล และการแสดงผล ซึ่งงานวิจัยนี้สังเกตเห็นถึงความซับซ้อนและ
ภาระในการเตรียมการทดลองจึงได้พัฒนาชุดวัดประสิทธิภาพ
โพรโทคอลไอโอทีสำหรับการทดลองโดยใช้อุปกรณ์จริงหรืออุปกรณ์
จำลอง โดยชุดทดลองจะมีส่วนควบคุมการทดลอง ส่วนควบคุมการ
รับส่งข้อมูล ส่วนควบคุมสภาพแวดล้อมเครือข่าย ส่วนการเก็บผล และ
ส่วนแสดงผลลัพธ์ พร้อมดำเนินการทดลองในเงื่อนไขต่าง ๆ แบบ
อัตโนมัติ ชุดทดลองจะให้ผลการทดลองพื้นฐานที่ครบถ้วน อาทิเช่น ค่า
เวลาไปกลับ(RTT) อัตราการสูญหายของข้อมูล (Loss rate) ค่าหน่วงเวลา
ไม่คงที่ (Jitter) ส่วนเกินข้อมูลจากการส่ง (Overhead)

Abstract

The Internet of Thing performance testing based on real or
virtual equipment are popular solutions due to convenience and reliable
result. However, from survey, the researchers had to develop the
program implementing with process to simulate specific environment,
such as packet sending and receiving management, network
environment management, result calculating and result display. It might
take several times on doing them. To reduce these workload and make it
less complicated, the IoT performance testing suite for real or virtual
equipment is, therefore, given in this research. With this testing suite,
process management, packet sending and receiving management,
network environment management, result calculating result display and
automate case process testing are included. And the results could
deliver in form of round trip time(RTT), loss rate, Jitter and Overhead.

บทนำ

อินเทอร์เน็ตทุกสรรพสิ่ง(Internet of thing : IoT) เรียกสั้นๆว่า
ไอโอที[1] คือแนวคิดที่จะเชื่อมต่อสิ่งต่าง ๆ เข้าด้วยกันด้วยเครือข่าย เพื่อ
แลกเปลี่ยนข้อมูลที่จำเป็น เช่น รถยนต์สามารถบอกความเร็วให้กับ
รถยนต์คันหลัง เซ็นเซอร์วัดความชื้นในดินสามารถสั่งการหัวฉีดน้ำให้
รดน้ำ หรือกระทั่งส่งข้อมูลไปยังศูนย์กลางข้อมูลเพื่อใช้ประโยชน์ต่อไป
เนื่องจากอุปกรณ์เครือข่ายในปัจจุบันมีขนาดเล็ก บริโภคพลังงานต่ำ ราคา
ไม่สูง ส่งเสริมแนวคิดไอโอทีเป็นอย่างดี ซึ่งเทคโนโลยีไอโอทีมีการ
ออกแบบ โพรโทคอล(Protocol)สื่อสารใหม่แยกออกจากโพรโทคอล
เครือข่ายเดิม อุปกรณ์ไอโอทีที่ส่วนมากมีความเร็วประมวลผลต่ำ หน่วย
ความต่ำ และอาจใช้พลังงานแบตเตอรี่ ดังนั้นจำเป็นต้องทำงานให้น้อย
ที่สุดเพื่อประหยัดพลังงาน จึงมีการสร้างและออกแบบ โพรโทคอลไอโอที
โดยโพรโทคอลเหล่านี้จะมีขนาดหัวโพรโทคอลเล็ก(header) และการ
ทำงานที่แตกต่างกันออกไป

เนื่องจากปัจจุบันมีโพรโทคอลไอโอทีมากมายให้เลือกใช้ ซึ่ง
แต่ละโพรโทคอลไอโอทีมีความเหมาะสมกับการประยุกต์ใช้งานแตกต่างกัน
จึงมีงานวิจัยทดลองวัดประสิทธิภาพไอโอทีที่โพรโทคอล โดยทดลอง
กับอุปกรณ์จริงหรืออุปกรณ์จำลอง(Virtual box, VMWare) [2]



รูปที่ 1 การทดลองโดยใช้อุปกรณ์จริง[3]

โดยมีเงื่อนไขในการทดลองแตกต่างกัน ซึ่งก่อนทำการทดลองจะพัฒนา
โปรแกรม ควบคุมกระบวนการทดลองเช่น การรับหรือส่งข้อมูล การ
ควบคุมสภาพแวดล้อมของเครือข่าย วิธีคำนวณผลลัพธ์ การเก็บผล

บทความวิจัย

การประชุมวิชาการ งานวิจัยและพัฒนาเชิงประยุกต์ ครั้งที่ 10

10th ECTI-CARD 2018, Phitsanulok Thailand

รวมถึงวิธีการแสดงเพื่อให้ง่ายต่อการวิเคราะห์ผล กระบวนการเหล่านี้ยังไม่ชัดเจนหรือขาดประสิทธิภาพให้ใช้งาน ผู้วิจัยจะต้องพัฒนาขึ้นเอง ดังนั้นจุดประสงค์ของงานวิจัยนี้คือการพัฒนาชุดวัดประสิทธิภาพโปรโตคอลไอโอทีสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลอง ที่จะช่วยลดภาระและระยะเวลาในการเตรียมการและการทดลอง

เครื่องมือดั้งเดิมในงานวิจัยที่ผ่านมา

หัวข้อนี้จะพูดถึงงานวิจัยเกี่ยวกับการทดลองทดสอบวัดประสิทธิภาพโปรโตคอลไอโอทีและเครื่องมือที่ใช้ดำเนินการทดลองโดยเริ่มจากงานวิจัยแรก Yuang Chen และ Thomas Kunz เปรียบเทียบโปรโตคอล MQTT CoAP และ DDS ในเครือข่ายไร้สายที่จำกัด [3] ทดลองโดยใช้อุปกรณ์จริง นำเสนอผลลัพธ์ อัตราการสูญหายข้อมูล (loss rate) เวลาส่งข้อมูลไปกลับ (RTT) การบริโภคแบนด์วิธ ในสภาพแวดล้อมเครือข่ายต่าง ๆ โดยใช้ NetEM เป็นตัวควบคุมสภาพแวดล้อม TBF ควบคุมแบนด์วิธ งานวิจัยที่สอง Thangavel D. และคณะทำการทดลองวัดประสิทธิภาพโปรโตคอล MQTT และ CoAP เมื่อใช้ทางออกร่วมกัน [2] ทดลองโดยใช้อุปกรณ์จริงและอุปกรณ์จำลอง นำเสนออัตราการสูญหายข้อมูล (loss rate) เวลาส่งข้อมูลไปกลับ (RTT) และใช้ WireShark [11] วัดข้อมูลส่วนเกินการส่งข้อมูล (Overhead) โดยใช้ WANem ควบคุมสภาพแวดล้อมเครือข่าย งานวิจัยที่สาม Jorge E. Luzuriaga และคณะทดลองวัดประสิทธิภาพโปรโตคอล AMQP กับ MQTT ในเครือข่ายอุปกรณ์ไร้สายที่ไม่มีเสถียรภาพ [4] โดยทดลองด้วยอุปกรณ์จริง นำเสนอการวัดผลค่าเวลาไม่คงที่ (Jitter) โดยทำการตัดสัญญาณสื่อสารด้วยวิธีการสลับจุดรับสัญญาณ (Access point) จากจุดหนึ่งไปยังอีกจุดหนึ่ง งานวิจัยที่สี่ Shinho Lee และคณะศึกษาความสัมพันธ์ของคุณภาพบริการ (QoS) กับเวลาการรับส่งข้อมูลของโปรโตคอล MQTT [5] โดยการทดลองเลือกใช้อุปกรณ์จริงแต่ใช้เครื่องแม่ข่ายที่อยู่ในอินเทอร์เน็ต

ตัวแปรที่เกี่ยวข้อง

จากการศึกษางานวิจัยพบตัวแปรที่ใช้ในการทดลอง แบ่งออกเป็น 2 ชนิด คือ ตัวแปรตั้งต้น และ ตัวแปรวัดผล ดังนี้

ตัวแปรตั้งต้น ประกอบด้วย 1) ขนาดแพ็คเกจ (Packet size) คือขนาดข้อมูลที่ได้รับ/ส่งส่วนมากมีหน่วยเป็นไบต์ (byte) 2) ความถี่ในการส่งแพ็คเกจ เช่น 10ms/packet 50ms/packet 100ms/packet 3) คุณภาพบริการ (QoS) บางโปรโตคอลไอโอทีที่มีการรันคิวการรับส่งข้อมูลเช่นโปรโตคอล MQTT มีคุณภาพบริการอยู่ 3 ระดับ โปรโตคอล CoAP มี 2 ระดับ 4) อัตราการสูญหายข้อมูล (Loss rate) หรือร้อยละของข้อมูลสูญหายเช่นร้อยละ 5, 10, 20 5) ค่าเวลาแฝงการรับ/ส่งข้อมูล (Latency time) ในการรับส่งข้อมูลจะมีกระบวนการทำงานหลายส่วนกว่าที่ข้อมูลจะถูกส่งออกจากต้นทางไปยังปลายทางเวลาส่วนนี้คือเวลาที่แฝงอยู่ในระบบ เช่น 50ms, 100ms, 200ms 6) ปริมาณการรับ/ส่งข้อมูลของกริดเครือข่าย

หรือแบนด์วิธ (bandwidth) อาทิเช่น 100Kbps 1000kbps หรือ 1 Gbps ตัวแปรตั้งต้นที่ 1 2 และ 3 เป็นตัวแปรตั้งต้นกับการส่งข้อมูล ส่วน 4 5 และ 6 เป็นตัวแปรตั้งต้นสภาพแวดล้อมเครือข่าย

ตัวแปรวัดผล 1) เวลาในการส่งข้อมูลไปกลับ (RTT :: Round Trip Time) บางงานวิจัยเรียกว่าเวลาหน่วง End-to-End [6] ค่ารวมได้จากผลต่างของเวลาที่ได้รับแพ็คเกจกับเวลาที่ส่งแพ็คเกจ 2) อัตราการสูญหายข้อมูล (Loss rate) คือร้อยละของจำนวนแพ็คเกจที่ส่งกับจำนวนที่ได้รับจริง 3) ค่าเวลาหน่วงไม่คงที่ [4] (Jitter) การส่งข้อมูลแต่ละครั้งมักมีค่าเวลาแฝงซึ่งในอุดมคติจะเท่ากันทุกครั้งที่ส่งแต่ในความเป็นจริงจะมีปัจจัยอื่นที่มีผลต่อค่าเวลาแฝงซึ่งทำให้เวลาหน่วงไม่คงที่ โดยสามารถคำนวณได้จากสมการที่ 1

$$J_n = (t'_n - t'_{n-1}) - T \quad (1)$$

J_n = ค่าเวลาหน่วงไม่คงที่ของแพ็คเกจ n (Jitter)

t'_n = เวลาถึงของแพ็คเกจที่ n

t'_{n-1} = เวลาถึงของแพ็คเกจที่ $n-1$

T = ค่าหน่วงเวลาส่ง

4) ข้อมูลส่วนเกินจากการส่ง [2] (Overhead) คือจำนวนข้อมูลจริงที่ได้รับซึ่งคำนวณได้ดังสมการที่ 2

$$oh = dr - d \quad (2)$$

oh = ขนาดของ Overhead

dr = ข้อมูลทั้งหมดที่ได้รับจากการ์เครือข่าย

d = ข้อมูลที่ต้องการส่ง

การออกแบบและพัฒนาชุดทดลอง

จากการศึกษาการทดลองวัดประสิทธิภาพโปรโตคอลไอโอทีและกระบวนการทดลองซึ่งพบวิธีดำเนินการที่คล้ายคลึงกันแบ่งออกเป็น 4 ส่วน ได้แก่ ส่วนควบคุมการส่ง/รับข้อมูล ส่วนควบคุมสภาพแวดล้อมเครือข่าย ส่วนการเก็บผลลัพธ์ และการแสดงผลลัพธ์ แต่ชุดทดลองจะเพิ่มส่วนควบคุมการทดลองให้สามารถทำงานแบบอัตโนมัติ จึงออกแบบชุดทดลองดังรูปที่ 2 ประกอบด้วยตัวส่งข้อมูล (A) ตัวรับข้อมูล (B) ซึ่งใช้อุปกรณ์จริงหรืออุปกรณ์จำลอง อุปกรณ์เครือข่ายแบบสาย ไร้สาย หรือแบบผสมและเป็นแบบท้องถิ่นหรืออินเทอร์เน็ตก็ได้ดังรูปที่ 3 ชุดทดลองรับรองตัวส่งและตัวรับข้อมูลอย่างละ 1 อุปกรณ์ อัตราจะอธิบายในแต่ละส่วนดังนี้ 1) ตัวควบคุมการส่ง/รับข้อมูล แบ่งเป็น 2 ส่วนคือตัวส่งและตัวรับ ตัวส่งจะส่งแพ็คเกจไปยังตัวรับด้วยโปรโตคอลไอโอทีที่ผู้ใช้งานกำหนด และตัวรับจะส่งรับแพ็คเกจกลับมาด้วยตัวส่งเป็นอันจบ

10th ECTI-CARD 2018 “การประชุมครั้งที่ 10 งานเทคโนโลยีเพื่อตอบสนองนโยบายประเทศไทย 4.0”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

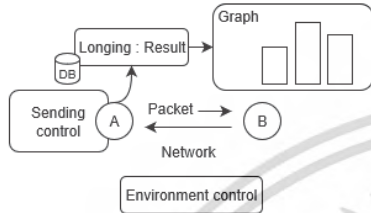
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทความวิจัย

การประชุมวิชาการ งานวิจัยและพัฒนาเชิงประยุกต์ ครั้งที่ 10

10th ECTI-CARD 2018, Phitsanulok Thailand

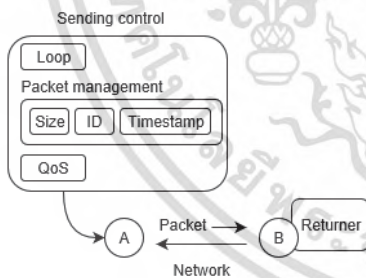
กระบวนการ ส่วนนี้ซึ่งกำหนดจำนวนรอบในการทดลอง(Loop) ตัวจัดการแพ็คเกจ ขนาดของแพ็คเกจ(packet size) ไอดีของแพ็คเกจ และการบันทึกเวลาในการส่งหรือรับ ค่าคุณภาพบริการ(QoS) ดังในรูปที่ 4
 2) ส่วนควบคุมสภาพแวดล้อมเครือข่าย ประกอบด้วย การควบคุม อัตราการสูญหายข้อมูล (loss rate) ค่าเวลาแฝง (latency time) และแบนด์วิดท์ (bandwidth) โดยตัวควบคุมนี้จะทำงานร่วมกับ NetEM



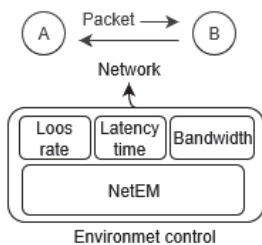
รูปที่ 2 โครงสร้างการดำเนินการของชุดทดลอง



รูปที่ 3 รูปแบบเครือข่ายที่ชุดทดลองรองรับ

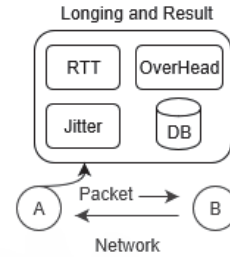


รูปที่ 4 โครงสร้างส่วนควบคุมการส่ง/รับข้อมูล



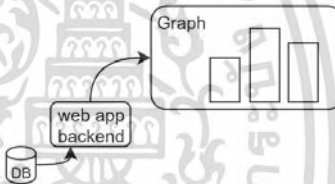
รูปที่ 5 โครงสร้างส่วนควบคุมสภาพแวดล้อมเครือข่าย

3) ส่วนการเก็บผลลัพธ์ รวมถึงจำนวนผลลัพธ์ของตัวแปรวัดผล เวลาไปกลับของแพ็คเกจ(RTT) ค่าช่วงเวลาไม่คงที่ (Jitter) และข้อมูลส่วนเกิน



รูปที่ 6 โครงสร้างส่วนการเก็บผลลัพธ์

จากการส่ง(Overhead) จากนั้นเก็บลงฐานข้อมูลดังรูปที่ 6 และสุดท้าย 4) ส่วนการแสดงผลลัพธ์ สามารถดูผลลัพธ์ได้ทันทีเมื่อการทดลองเสร็จสิ้นเลือกเปรียบเทียบตัวแปรตั้งต้นกับตัวแปรวัดผลได้สะดวกในรูปแบบกราฟเส้นหรือกราฟแท่ง โดยใช้ plotly[12] ซึ่งใช้งานผ่านบราวเซอร์ดังรูปที่ 7



รูปที่ 7 โครงสร้างส่วนการแสดงผลลัพธ์

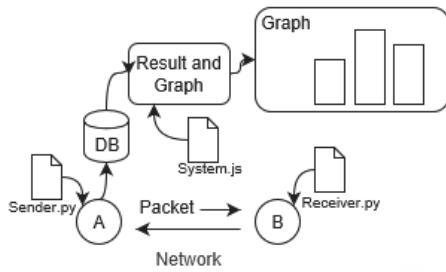
ชุดทดลองที่พัฒนาเสร็จสิ้นและวิธีใช้งาน

ชุดทดลองโปรโตคอลไอโอทีที่พัฒนาขึ้นจำเป็นต้องมีโปรแกรมอื่น ๆ ประกอบการใช้งาน เพื่อให้ระบบสามารถทำงานได้ดังนี้ ติดตั้ง Python[7] NodeJs[8] XAMPP[9] และ NetEM[10] ในอุปกรณ์ตัวส่งเนื่องจากอุปกรณ์ตัวส่งเป็นตัวควบคุมการทดลอง ควบคุมการรับ/ส่งข้อมูล ควบคุมสภาพแวดล้อมเครือข่าย เก็บผลลัพธ์ และแสดงผลเมื่อการทดลองเสร็จสิ้น การใช้งานชุดทดลองโปรโตคอลไอโอทีที่ผู้ใช้งานต้องนำโปรโตคอลที่ต้องการทดสอบรวมเข้ากับไฟล์ Sender.py ซึ่งอยู่ที่ตัวส่ง และ Receiver.py ซึ่งอยู่ที่ตัวรับ โดยสามารถกำหนดค่าตัวแปรต่าง ๆ ที่ใช้ในการทดสอบในไฟล์ Sender.py ซึ่งประกอบด้วย จำนวนรอบในการทดสอบแต่ละกรณี (loop case) ขนาดของแพ็คเกจที่ส่ง (packet size) ระดับคุณภาพบริการ(QoS) อัตราการสูญหายของข้อมูล (loss rate) ค่าเวลาแฝงในเครือข่าย(latency delay) ความกว้างของเครือข่าย(Bandwidth) การกำหนดค่าตัวแปรตั้งต้นสามารถกำหนดได้มากกว่าหนึ่งค่า ชุดทดลองจะตรวจสอบกรณีและดำเนินการจนครบทุกกรณี ถัดมากำหนดชื่อฐานข้อมูลชื่อตารางที่ไฟล์ System.js เมื่อนำเข้าโปรโตคอลและกำหนด

บทความวิจัย

การประชุมวิชาการ งานวิจัยและพัฒนาเชิงประยุกต์ ครั้งที่ 10

10th ECTI-CARD 2018, Phitsanulok Thailand



รูปที่ 8 โครงสร้างไฟล์ชุดทดลอง

```
#sending case
loop = 100
qos = (0, 1, 2)
payload_size = {50, 100, 200} # byte
sending_prepid = {100, 200, 500} #ms

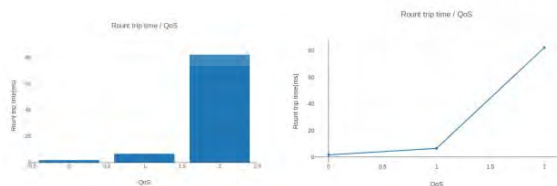
#environment case
loss_rate = {0, 10, 20} # 0%, 10%, 20%
sys_latency = {0, 50, 100} # 100ms
bandwidth = {1000, 500, 200} # kbps
```

รูปที่ 9 การกำหนดค่าตัวแปรต่าง ๆ

ค่าตัวแปรต่าง ๆ เรียบร้อยแล้วสามารถเริ่มกระบวนการทดสอบด้วยโปรแกรม Run.sh ชุดทดลองจะเริ่มทดลองโดยอัตโนมัติจนกระทั่งครบทุกกรณีที่ได้กำหนดไว้ เมื่อการทดสอบสิ้นสุดลงสามารถดูผลการทดลองได้ที่ <http://localhost/iotprotocaltest:8888> จะปรากฏเว็บแอปพลิเคชันสำหรับแสดงผลการทดลองต่าง ๆ ซึ่งสามารถเลือกความสำคัญของค่าตัวแปรต่าง ๆ ได้ตามที่ต้องการ ในรูปแบบกราฟเส้นหรือแบบแท่ง



รูปที่ 10 โปรแกรมเลือกแสดงผลของผลการทดลอง



รูปที่ 11 การแสดงผลลัพธ์จากชุดทดลอง

สรุป

การพัฒนาชุดทดลองวัดประสิทธิภาพไอโอทีโปรโตคอลสำหรับอุปกรณ์จริงหรืออุปกรณ์จำลอง ซึ่งมีส่วนควบคุมการทดลอง ส่วนตัวควบคุมการส่ง/รับข้อมูล ส่วนควบคุมสภาพแวดล้อมเครือข่าย ส่วนการเก็บผลลัพธ์ ส่วนการแสดงผล พร้อมทำงานได้ทันทีทำให้ผู้ใช้งานไม่ต้องดำเนินการพัฒนาโปรแกรมขึ้นเองทำให้ลดภาระงานลงได้ อีกทั้งสามารถทำการทดลองแบบอัตโนมัติทำให้การทดลองรวดเร็วขึ้น ซึ่งในการพัฒนาต่อไป ตั้งใจพัฒนาการทดลองแบบอุปกรณ์รับส่งมากกว่า 1 ตัว

เอกสารอ้างอิง

- [1] H. Saadeh, W. Almobaideen, and K. E. Sabri, "Internet of Things: A review to support IoT architecture's design," Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS), Jordan, Dec 2017.
- [2] D. Thagavel, X. Ma, A. Valera, "Performance evaluation of MQTT and CoAP via a common middleware," Intelligent Sensors Sensor Networks and Information Processing (ISSNIP), Singapore, April 2014.
- [3] Y. Chen, T. Kunz, "Performance evaluation of IoT protocols under a constrained wireless access network," Selected Topics in Mobile & Wireless Networking (MoWNet), Egypt, April 2016.
- [4] J. E. Luzuriage, M. Perez, P. Boronat, "A comparative evaluation of AMQP and MQTT protocols over unstable and mobile network," Consumer Communications and Networking Conference(CCNC), USA, Jan 2015.
- [5] S. Lee, H. Kim, D.-k. Hong, "Correlation analysis of MQTT loss and delay according to QoS level," Information Networking(ICOIN), Thailand, April 2013.
- [6] K. Govindan, A. P. Azad, "End-to-end service assurance in IoT MQTT-SN," Consumer Communications and Networking Conference(CCNC), USA, Jan 2015.
- [7] Python 3.6.4, "Python", <https://www.python.org/>, 2018.
- [8] NodeJs 8.9.4, "Node Js", <https://nodejs.org/en/>, 2018
- [9] XAMPP 7.2.1, "XAMPP", <https://www.apachefriends.org/index.html/>, 2018
- [10] NetEM, "NetEM", <https://wiki.linuxfoundation.org/networking/netem>, 2018
- [11] WireShark, "WireShark", <https://www.wireshark.org/>, 2018
- [12] Plotly.js/"plotly", <https://plot.ly/javascript/> 2018

ประวัติผู้เขียน

ชื่อ-นามสกุล นายวิปศย์ ปุยสำลี
วัน เดือน ปีเกิด 3 มีนาคม 2533
ที่อยู่ 147/1 หมู่ 1 ตำบลสะพานหิน อำเภอหนองมะโมง
จังหวัดชัยนาท 17120
ประวัติการศึกษา 2555 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร วิทยาเขตพระนครเหนือ

ประสบการณ์การทำงานและผลงานวิจัย

พ.ศ. 2556-2559 ผู้ช่วยนักวิจัย ห้องปฏิบัติการสมองกลฝังตัว (Embedded System Laboratory) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2558 วิทยากรบรรยาย “ระบบอัจฉริยะเพื่อการตรวจสอบย้อนกลับผลิตภัณฑ์สินค้าพลาสติก” ณ สหกรณ์พลาสติกและสัตว์น้ำ จังหวัด ฉะเชิงเทรา
พ.ศ. 2558 ผลงานวิจัย “Mobile RFID Reader Applied to Food Processing System” ณ งานประชุมวิชาการ “The 2014 International Conference on Embedded Systems and Intelligent Technology 2014” ณ เมือง กวาง ประเทศเกาหลีใต้
พ.ศ. 2560 หัวหน้าฝ่ายวิจัยพัฒนาผลิตภัณฑ์ บริษัทลิ่งคิฟว์ จำกัด
พ.ศ. 2561 ผลงานวิจัย “IoT Protocol Performance Testing Suite for Real Equipment or Virtual Equipment” ณ งานประชุมวิชาการ ECTI-CARD 2018 จังหวัดพิษณุโลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้