การพัฒนาโพรโทคอลของเกตเวย์สำหรับการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M เพื่อการประยุกต์ใช้ในระบบการจัดการพลังงานภายในอาคาร

นายธีรพันธุ์ กลิ่นประทุม

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2557
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DEVELOPMENT OF GATEWAY PROTOCOL FOR INTERWORKING BETWEEN IEEE1888 STANDARD AND ETSI M2M STANDARD FOR BUILDING ENERGY MANAGEMENT SYSTEM APPLICATION

Mr. Teerapan Klinpratum

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2014

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาโพรโทคอลของเกตเวย์สำหรับการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M เพื่อการประยุกต์ใช้ในระบบการจัดการพลังงานภายในอาคาร
โดย	นายธีรพันธุ์ กลิ่นประทุม
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร. ชัยเชษฐ์ สายวิจิตร

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร. บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. แนบบุญ หุนเจริญ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์ ดร. ชัยเชษฐ์ สายวิจิตร)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. เชาวน์ดิศ อัศวกุล)

.....กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร. กิตติพันธุ์ เตชะกิตติโรจน์)

ธีรพันธุ์ กลิ่นประทุม : การพัฒนาโพรโทคอลของเกตเวย์สำหรับการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M เพื่อการประยุกต์ใช้ในระบบการจัดการพลังงานภายในอาคาร (DEVELOPMENT OF GATEWAY PROTOCOL FOR INTERWORKING BETWEEN IEEE1888 STANDARD AND ETSI M2M STANDARD FOR BUILDING ENERGY MANAGEMENT SYSTEM APPLICATION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร. ชัยเชษฐ์ สายวิจิตร, 60 หน้า.

ในอนาคต อุปกรณ์อิเล็กทรอนิกส์จำนวนมากจะถูกเชื่อมต่อกับอินเตอร์เนตเพื่อให้แต่ละอุปกรณ์สามารถส่งข้อมูลสื่อสารกันได้ด้วยตัวเอง รวมไปถึงถูกเฝ้าสังเกตและถูกควบคุมจากระยะไกลได้ เป็นเหตุให้ขนาดของสถาปัตยกรรมทางอินเตอร์เนตขยายตัวอย่างรวดเร็ว และเพื่อรองรับกับโครงสร้างเครือข่ายที่กำลังเพิ่มขึ้นในอนาคต จึงมีการค้นคว้าและวิจัยเพื่อให้เกิดการทำงานร่วมกันระหว่างหลากหลายเทคโนโยลีการเข้าถึงและอุปกรณ์หลายๆชนิด ทำให้เกิดเป็นแพลตฟอร์มที่สามารถใช้ในการสื่อสารแบบแมชชีน-ทู-แมชชีน (Machine-to-Machine communication) ทั้งผ่านสายส่งและแบบไร้สาย ในปัจจุบันมีมาตรฐานแมชชีน-ทู-แมชชีน 2 มาตรฐานที่มีจุดประสงค์เดียวกัน คือ จัดหาตัวกลางในการทำงานร่วมกันระหว่างเทคโนโลยีที่มีอยู่แล้วนั่นคือ มาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M

ในวิทยานิพนธ์ฉบับนี้ มีจุดมุ่งหมายเพื่อพัฒนาโพรโทคอลของเกตเวย์สำหรับการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M โดยมีการแสดงการเปรียบเทียบโครงสร้างและความสามารถของทั้งสองมาตรฐาน และออกแบบการทำงานร่วมกันของสองแพลตฟอร์มของทั้งสองมาตรฐานนี้ โดยใช้ Node.js ในการออกแบบการดำเนินงานโปรแกรมด้วยภาษา JavaScript ในวิทยานิพนธ์ฉบับนี้เรียกตัวกลางในการทำงานร่วมกันว่า เกตเวย์จำลอง (Proxy gateway) ซึ่งเป็นตัวกลางในการทำงานร่วมกันผ่านโครงข่าย (Network Interworking proxy) โดยมีจุดประสงค์เพื่อให้เกิดการแลกเปลี่ยนข้อมูลจากหน่วยเก็บข้อมูลระหว่างมาตรฐาน IEEE1888 ด้วยโพรโทคอล WRITE และ FETCH ผ่านโพรโทคอล SOAP และ HTTP ด้วยข้อมูลรูปแบบ XML และมาตรฐาน ETSI M2M ด้วยระเบียบวิธี CREATE และ RETRIEVE ผ่านการดำเนินการ RESTful ด้วยข้อมูลรูปแบบ JSON

ภาควิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2557

5570240621 : MAJOR ELECTRICAL ENGINEERING
KEYWORDS: FUTURE INTERNET ARCHITECTURE / M2M COMMUNICATION / INTERWORKING PROXY / SENSOR NETWORKS

TEERAPAN KLINPRATUM: DEVELOPMENT OF GATEWAY PROTOCOL FOR INTERWORKING BETWEEN IEEE1888 STANDARD AND ETSI M2M STANDARD FOR BUILDING ENERGY MANAGEMENT SYSTEM APPLICATION. ADVISOR: ASST. PROF. CHAIYACHET SAIVICHIT, Ph.D., 60 pp.

In the near future, many electronic devices will be equipped with internet connectivity in which they can communicate with one another. They will also be capable of being monitored and automatic controlled from remote location. As a result, IoT (Internet of Thing) architecture shall thrive in the soon future. Interoperability among various access technologies and many kind of devices become one of crucial concern for the upcoming extensive network infrastructure. So providing a common platform that can be used for Machine-2-Machine (M2M) communication though both wireless and wired networks supporting various kinds of devices is most required nowadays. Two M2M - related standards recently released are IEEE1888 standard and ETSI M2M standard.

In this thesis, it aims to the development of protocol for interworking between IEEE1888 standard and ETSI M2M standard. This shows comparison of two standards based on supported features and capabilities and proposes a model for interworking between two prototype platforms from both standards by using Node.js to design code platform with JavaScript language. The network interworking proxy called "Proxy gateway" is used for data exchange between IEEE1888 standard with WRITE and FETCH protocol by using XML format data communicated though SOAP and HTTP protocol and ETSI M2M standard with CREATE and RETRIEVE operation by using JSON format data communicated though RESTful service.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Department: Electrical Engineering Student's Signature .....
Field of Study: Electrical Engineering Advisor's Signature .....
Academic Year: 2014

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ ด้วยความช่วยเหลือของผู้ช่วยศาสตราจารย์ ดร. ชัยเชษฐ์ สายวิจิตร อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งกรุณาให้ความรู้ที่เป็นประโยชน์ในการวิจัย และให้คำแนะนำต่าง ๆ รวมไปถึงคำวิจารณ์ต่อผู้ทำวิจัยมาโดยตลอด ผู้วิจัยจึงขอกราบขอบพระคุณมา ณ ที่นี้

ขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร. แนบบุญ หุนเจริญ ประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร. เชาวน์ดิศ อัศวกุล กรรมการสอบวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร. กิตติพันธุ์ เตชะกิตติโรจน์ กรรมการสอบวิทยานิพนธ์จากภายนอกมหาวิทยาลัย ที่ได้สละเวลาตรวจสอบและให้คำแนะนำเพื่อให้วิทยานิพนธ์ฉบับนี้สมบูรณ์ยิ่งขึ้น

ขอขอบคุณความคิดเห็น และข้อเสนอแนะดีๆ รวมทั้งกำลังใจจากพี่ๆ เพื่อนๆ และน้องๆทุกคน ตลอดจนการสละเวลาช่วยขัดเกลาและตรวจแก้ผลงานวิจัยจาก อาจารย์ภัทรชาติ โกมลกิติ นางสาวศศิรมย์ เทียนน้อย นายธีรยุทธ บุญเกิด นางสาวพรณิภา รัตนาวดี นายธนากร อินทสุทธิ์ และเพื่อนๆ พีๆ น้องๆในแลปที่ไม่ได้กล่าวถึง นอกจากนี้ยังได้รับโอกาสในการฝึกฝนวิธีการนำเสนอผลงานร่วมกับอาจารย์และเพื่อนๆในกลุ่มสัมมนากลุ่มเนตเวิร์ก (Network group) นำโดย ผศ.ดร.เชาวน์ดิศ อัศวกุล และ ผศ.ดร ชัยเชษฐ์ สายวิจิตร ซึ่งมีส่วนสำคัญในการเสริมสร้างความมั่นใจและความกล้าแสดงความคิดเห็นในเชิงวิชาการมากขึ้น

ขอกราบขอบพระคุณบิดา มารดา และพี่ชาย ที่เป็นกำลังใจและกำลังทรัพย์ตลอดมา รวมทั้งให้โอกาสผู้วิจัยได้ศึกษาต่อในระดับปริญญามหาบัณฑิต

ขอขอบคุณห้องปฏิบัติการวิจัยระบบโทรคมนาคม ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย สำหรับทรัพยากรต่างๆในการศึกษา ค้นคว้าและวิจัย

ขอบคุณโครงการ CUBEMS ที่มอบความรู้และข้อมูลให้ในการทำวิจัยชิ้นนี้ และโครงการ UNIFI ที่มอบโปรแกรม OpenMTC และขอบคุณทีมงานที่คอยให้ความรู้และคำแนะนำในการออกแบบงานชิ้นนี้

งานวิจัยชิ้นนี้ได้รับทุนสนับสนุนจากโครงการขับเคลื่อนการวิจัย กองทุนรัชดาภิเษกสมโภช (Special Task Force for Activating Research (STAR) ภายใต้กลุ่มวิจัยโครงข่ายไร้สายและอินเทอร์เน็ตอนาคต (Wireless Network and Future Internet Research Group) จุฬาลงกรณ์มหาวิทยาลัย

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ง

บทคัดย่อภาษาอังกฤษ.....จจ

กิตติกรรมประกาศ.....ฉฉ

สารบัญ.....ชช

สารบัญตาราง.....ญญ

สารบัญรูป.....ฎฎ

บทที่ 1 บทนำ.....11

1.1 ความเป็นมาและความสำคัญ.....1

1.2 แนวทางของวิทยานิพนธ์.....22

1.3 วัตถุประสงค์.....22

1.4 ขอบเขตของวิทยานิพนธ์.....22

1.5 ขั้นตอนการดำเนินงาน.....33

1.6 ประโยชน์ที่คาดว่าจะได้รับ.....33

1.7 ประมวลวิทยานิพนธ์.....44

บทที่ 2 ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง.....5

2.1 มาตรฐาน IEEE1888.....5

2.1.1 สถาปัตยกรรมของ IEEE1888.....55

2.1.2 ลำดับการสื่อสารของโพรโทคอล.....66

2.1.3 โพรโทคอลการสื่อสารระหว่างส่วนประกอบ.....77

2.1.4 โครงสร้างของข้อมูลที่ใช้ในการสื่อสาร.....1010

2.1.5 การจัดการตำแหน่งข้อมูลด้วยโครงสร้างเซตของตำแหน่งข้อมูล.....11

2.1.6 การจัดการโครงสร้างของตำแหน่งข้อมูลตามความหมายด้วยสารบบทั่วไป.....1212

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

2.2 มาตรฐาน ETSI M2M 13
2.2.1 สถาปัตยกรรมของมาตรฐาน ETSI M2M 13
2.2.2 โปรแกรมประยุกต์ต่อประสานของเอ็มทูเอ็ม 17
2.3 การเปรียบเทียบเชิงวิเคราะห์ระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M 20
2.3.1 การเปรียบเทียบสถาปัตยกรรม 20
2.3.2 การเปรียบเทียบรูปแบบข้อมูลที่ใช้ในการสื่อสาร 22
2.3.3 การเปรียบเทียบโพรโทคอลการสื่อสาร 23
2.3.4 การเปรียบเทียบการดำเนินการ (Operations) 23
2.3.5 การเปรียบเทียบโครงสร้างการจัดการตำแหน่งข้อมูล 24
บทที่ 3 โพรโทคอลในการแลกเปลี่ยนข้อมูลที่นำเสนอ 26
3.1 การพัฒนาโพรโทคอลของเกตเวย์ 26
3.2 กระบวนการทำงาน 27
3.2.1 กระบวนการ 1: ร้องขอข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 27
3.2.2 กระบวนการ 2: ร้องขอข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M 28
3.3 สภาพแวดล้อมของระบบ 28
3.3 สรุป 31
บทที่ 4 การทดสอบและวิเคราะห์ผลการทดสอบการดำเนินการของโพรโทคอล 32
4.1 ส่วนประกอบของโพรโทคอลที่ใช้ทำการทดสอบ 32
4.1.1 เกตเวย์จำลอง 32
4.1.2 หน่วยเก็บข้อมูลของ CUBEMS 32
4.1.3 หน่วยเก็บข้อมูลของ OpenMTC 33
4.1.4 ข้อมูลจากเซนเซอร์ 10 จุด 33
4.1.5 ตัวชี้วัดในการทดสอบ 34

หน้า

4.2 การทดสอบและผลการทดสอบของโพรโทคอลที่นำเสนอ 37
4.2.1 การทดสอบเบื้องต้น 37
4.2.2 การทดสอบโครงสร้างการจำลอง 42
4.2.3 การเปรียบเทียบสมรรถนะของโครงสร้างการจำลองในรูปแบบต่างๆ 47
4.3 สรุป 53
บทที่ 5 บทสรุปและข้อเสนอแนะ 55
5.1 บทสรุป 55
5.2 ข้อเสนอแนะ 56
รายการอ้างอิง 57
ประวัติผู้เขียนวิทยานิพนธ์ 60

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

ตารางที่ 1 การทำงานของกรณีต่างๆในระบบสื่อสารตามมาตรฐาน IEEE1888 [3]..... 7

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญรูป

รูปที่ 1 สถาปัตยกรรมตามมาตรฐาน IEEE1888[3] 5
รูปที่ 2 ลำดับการสื่อสารของโพรโทคอล[3] 6
รูปที่ 3 โพรโทคอล FETCH [3] 7
รูปที่ 4 โพรโทคอล WRITE [3] 8
รูปที่ 5 โพรโทคอล TRAP (รูปแบบการนำไปใช้งานแบบทั่วไป) [3] 9
รูปที่ 6 โพรโทคอล TRAP (รูปแบบการนำไปใช้งานในทางปฏิบัติ) [3] 9
รูปที่ 7 โครงสร้างข้อมูลที่ใช้ในการสื่อสาร [3] 11
รูปที่ 8 รูปแบบโครงสร้างเซตของตำแหน่งข้อมูล [3] 11
รูปที่ 9 โครงสร้างเซตของตำแหน่งข้อมูลตามสารบบทั่วไป [6] 12
รูปที่ 10 ตัวอย่างการสร้างตัวแบบเซตโปรแกรมประยุกต์ [6] 13
รูปที่ 11 สถาปัตยกรรมของเอ็มทูเอ็ม[5] 13
รูปที่ 12 สถาปัตยกรรมของ โอเพ็นเอ็มทีซี [7] 16
รูปที่ 13 กระบวนการตามโครงสร้างแบบ RESTful 18
รูปที่ 14 ตัวอย่างการไหลของข้อความของแบบแผนพื้นฐาน 19
รูปที่ 15 ตัวอย่างโครงสร้างของทรัพยากร M2M sclBase (M2M sclBase resource tree) [9] 20
รูปที่ 16 สถาปัตยกรรมของมาตรฐาน ETSI M2M 21
รูปที่ 17 สถาปัตยกรรมของมาตรฐาน IEEE1888 21
รูปที่ 18 ตัวอย่างรูปแบบข้อมูลที่ใช้ในการสื่อสารตามมาตรฐาน ETSI M2M 22
รูปที่ 19 ตัวอย่างรูปแบบข้อมูลที่ใช้ในการสื่อสารตามาตรฐาน IEEE1888 22
รูปที่ 20 ตัวอย่างโครงสร้างการจัดการส่วนประกอบตามมาตรฐาน ETSI M2M 24
รูปที่ 21 ตัวอย่างโครงสร้างการจัดการส่วนประกอบตามมาตรฐาน IEEE1888 25
รูปที่ 22 รูปแบบการทำงานร่วมกันระหว่างมาตรฐานIEEE1888 กับมาตรฐาน ETSI M2M 27

รูปที่ 23 แผนภาพกระบวนการทำงานของเกตเวย์จำลองในกระบวนการ 1 [11]..... 27
รูปที่ 24 แผนภาพกระบวนการทำงานของเกตเวย์จำลองในกระบวนการ 2 [11]..... 28
รูปที่ 25 ภาพรวมส่วนประกอบของโครงการ CU-BEMS..... 29
รูปที่ 26 สถาปัตยกรรมของโครงร่างเอ็มทูเอ็มที่ถูกปรับให้เกิดการเชื่อมต่อ..... 30
รูปที่ 27 คอมพิวเตอร์โน้ตบุ๊กที่ใช้เป็นเกตเวย์จำลอง..... 32
รูปที่ 28 เครื่อง server ที่ใช้เป็นหน่วยเก็บข้อมูลของ CUBEMS..... 33
รูปที่ 29 คอมพิวเตอร์ตั้งโต๊ะที่ใช้เป็นหน่วยเก็บข้อมูลของ OpenMTC..... 33
รูปที่ 30 แสดงตำแหน่งเซนเซอร์ที่นำข้อมูลมาใช้ทดสอบ..... 34
รูปที่ 31 เวลาที่เกตเวย์จำลองส่งการร้องขอข้อมูลไปและได้รับข้อมูลตอบกลับจากหน่วยเก็บข้อมูล..... 35
รูปที่ 32 ปริมาณของข้อมูลที่ส่ง และปริมาณข้อมูลที่ใช้ในการสื่อสารจริง..... 36
รูปที่ 33 ข้อมูลปริมาณงานที่ใช้ในการสื่อสาร..... 37
รูปที่ 34 การดำเนินการส่งข้อมูลของเกตเวย์จำลองไปยังหน่วยเก็บข้อมูล..... 38
รูปที่ 35 เปรียบเทียบเวลาที่ใช้ในการทำงานของโพรโทคอล WRITE และการดำเนินการ CREATE..... 38
รูปที่ 36 เปรียบเทียบปริมาณงานของโพรโทคอล WRITE และการดำเนินการ CREATE..... 39
รูปที่ 37 เปรียบเทียบโอเวอร์เฮดของข้อมูลของโพรโทคอล WRITE และการดำเนินการ CREATE..... 39
รูปที่ 38 การดำเนินการดึงข้อมูลของเกตเวย์จำลองจากหน่วยเก็บข้อมูล..... 40
รูปที่ 39 เปรียบเทียบเวลาที่ใช้ในการทำงานของโพรโทคอล FETCH และการดำเนินการ RETRIEVE..... 40
รูปที่ 40 เปรียบเทียบปริมาณงานของโพรโทคอล FETCH และการดำเนินการ RETRIEVE..... 41
รูปที่ 41 เปรียบเทียบโอเวอร์เฮดของข้อมูลของโพรโทคอล FETCH และการดำเนินการ RETRIEVE..... 41
รูปที่ 42 การดำเนินการของเกตเวย์จำลองในกระบวนการ 1..... 42
รูปที่ 43 แผนภาพกระบวนการทำงานของเกตเวย์จำลองในการทดสอบตามกระบวนการ 1..... 43

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

รูปที่ 44 การดำเนินการของเกตเวย์จำลองในกระบวนการ 2 43
รูปที่ 45 แผนภาพการทำงานของเกตเวย์จำลองในการทดสอบตามกระบวนการ 2 43
รูปที่ 46 เวลาที่ใช้ในการทำงานของกระบวนการ 1 44
รูปที่ 47 เวลาที่ใช้ในการทำงานของกระบวนการ 2 44
รูปที่ 48 โอเวอร์เฮดของข้อมูลของกระบวนการ 1 45
รูปที่ 49 โอเวอร์เฮดของข้อมูลของกระบวนการ 2 45
รูปที่ 50 ปริมาณงานของกระบวนการ 1 46
รูปที่ 51 ปริมาณงานของกระบวนการ 2 46
รูปที่ 52 รูปแบบการจำลองโครงสร้างแบบ ContainerID 48
รูปที่ 53 รูปแบบการจำลองโครงสร้างแบบ SearchString 49
รูปที่ 54 รูปแบบการจำลองโครงสร้างแบบ LocationContainer 49
รูปที่ 55 เวลาที่ใช้ในการทำงานของโครงสร้างการจำลองรูปแบบต่างๆในกระบวนการ 1 50
รูปที่ 56 โอเวอร์เฮดของข้อมูลของโครงสร้างการจำลองรูปแบบต่างๆในกระบวนการ 1 50
รูปที่ 57 ปริมาณงานของโครงสร้างการจำลองรูปแบบต่างๆในกระบวนการ 1 51
รูปที่ 58 เวลาที่ใช้ในการทำงานของโครงสร้างการจำลองแบบต่างๆในกระบวนการ 1 52
รูปที่ 59 โอเวอร์เฮดของข้อมูลของโครงสร้างการจำลองแบบต่างๆในกระบวนการ 2 52
รูปที่ 60 ปริมาณงานของโครงสร้างการจำลองแบบต่างๆในกระบวนการ 2 52

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

เทคโนโลยีการติดต่อสื่อสารและการแลกเปลี่ยนข้อมูลในสังคมปัจจุบันได้เข้ามามีบทบาทอย่างมาก จนนับเป็นสิ่งที่จำเป็นสำหรับการติดต่อสื่อสารระหว่างมนุษย์ องค์กรหรือประเทศ ซึ่งเทคโนโลยีดังกล่าวมีหลากหลายรูปแบบ เช่น ZigBee Wifi Bluetooth WiMAX ADSL Cable 3G 4G เป็นต้น และเนื่องจากการเติบโตของตลาดเทคโนโลยีการสื่อสารรวมไปถึงการค้นคว้าวิจัยที่มุ่งไปสู่การสื่อสารที่มนุษย์เข้ามามีบทบาทน้อยที่สุด ที่เรียกว่า การสื่อสารแบบแมชชีน-ทู-แมชชีน (machine-to-machine communication) [1] เพื่อรองรับกับการใช้งานอุปกรณ์สื่อสารผ่านเทคโนโลยีที่แตกต่างกัน จึงทำให้เกิดมาตรฐานที่จัดหาตัวกลางในการเชื่อมโยงการสื่อสารและรองรับกับปริมาณการใช้บริการที่เพิ่มมากขึ้น ซึ่งทำให้โครงข่ายการสื่อสารขยายขอบเขตมากขึ้นไปด้วย [2]

สถาบันวิชาชีพวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ (Institute of Electrical and Electronics Engineers, IEEE) ได้ออกมาตรฐานเปิด (Open Standard) เกี่ยวข้องกับการจัดการพลังงานโดยใช้ชื่อว่า "โพรโทคอลโครงข่ายการควบคุมชุมชนสีเขียวอย่างแพร่หลาย (Ubiquitous Green Community Control Network Protocol) หรือ IEEE1888" [3] โดยมีวัตถุประสงค์หลักคือ การเชื่อมโยงระบบการจัดการพลังงานทุกรูปแบบเพื่อที่จะสื่อสารระหว่างกันผ่านโครงข่ายทีซีพี/ไอพี (TCP/IP) ไม่ว่าจะเป็นโพรโทคอลแบบใดก็ตาม เช่น BACnet LonWorks Modbus ZigBee เป็นต้น เพื่อลดการขาดระบบการสื่อสาร และลดการทำลายสิ่งแวดล้อม ทั้งนี้ ทุก ๆ โครงข่ายที่มีอยู่เดิมจะสามารถติดต่อสื่อสารกับระบบการจัดการพลังงานตามมาตรฐาน IEEE1888 นี้ได้ต้องสื่อสารด้วยภาษาเดียวกันนั่นคือ XML และมีลักษณะของตัวระบุรูปแบบทรัพยากร (URI: Uniform resource identifier) เป็นรูปแบบเดียวกัน ซึ่งโครงข่ายต่าง ๆ ที่กล่าวข้างต้นจะสามารถทำงานร่วมกันได้ต้องมีตัวแปลงภาษาจากภาษาเดิมในเทคโนโลยีต่าง ๆ ให้เป็นภาษาตามมาตรฐาน IEEE1888 โดยใช้เกตเวย์ (Gateway) เพื่อเชื่อมโยงให้เข้ากับโครงข่ายทีซีพี/ไอพี และเพื่อนำมาประยุกต์ใช้ในระบบการจัดการพลังงานตามมาตรฐาน IEEE1888 นี้ได้ [4]

สถาบันมาตรฐานโทรคมนาคมแห่งยุโรป (European Telecommunications Standard Institute: ETSI) ได้ออกมาตรฐานเกี่ยวกับการสื่อสารแบบแมชชีน-ทู-แมชชีน หรือเรียกสั้น ๆ ว่า เอ็ม ทูเอ็ม (M2M) ซึ่งมีวัตถุประสงค์เพื่อรองรับการขยายตัวของโครงข่ายสื่อสาร ปริมาณของข้อมูลที่สูงขึ้น และสถาปัตยกรรมการสื่อสารที่กำลังเติบโตอย่างรวดเร็วในอนาคต พร้อมกับใช้ประโยชน์จากการสื่อสารผ่านโครงข่ายทีซีพี/ไอพี ด้วยระบบตามมาตรฐาน 3GPP, TISPAN และ 3GPP2 ในการจัดหา

ตัวกลางที่สามารถรองรับการติดต่อสื่อสารผ่านเทคโนโลยีการสื่อสารหลากหลายรูปแบบเช่น 2G 3G 4G LTE เป็นต้น ซึ่งช่วยให้เกิดการแลกเปลี่ยนข้อมูลข้ามเทคโนโลยี [5]

ดังนั้น หากโพรโทคอลตามมาตรฐานทั้งสองมาตรฐานนี้สามารถแลกเปลี่ยนข้อมูลกันได้ จะเป็นอีกหนทางหนึ่ง ที่จะพัฒนาเทคโนโลยีการติดต่อสื่อสาร โดยการเชื่อมต่อมาตรฐานที่ถูกพัฒนาและใช้งานบนพื้นที่ที่แตกต่างกันบนโลก นั่นคือสามารถสื่อสารจากมาตรฐานหนึ่งซึ่งเกิดขึ้นในประเทศญี่ปุ่นไปยังอีกมาตรฐานหนึ่งซึ่งเกิดขึ้นในแถบยุโรป และจะเป็นประโยชน์อย่างยิ่งในอนาคตอันใกล้เนื่องจากโครงข่ายการสื่อสารกำลังเป็นที่นิยมและแพร่หลายไปทั่วโลก มีจำนวนผู้ใช้บริการผ่านเทคโนโลยีต่าง ๆ เพิ่มขึ้นอย่างมากมาย ซึ่งการทำให้มาตรฐานสองมาตรฐานสามารถสื่อสารแลกเปลี่ยนข้อมูลกันได้ จะก่อให้เกิดประโยชน์ทั้งในประเทศ และพัฒนาไปถึงระดับโลกเลยทีเดียว

1.2 แนวทางของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้เสนอการพัฒนาโพรโทคอลของเกตเวย์ในการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M ในการดึงข้อมูลมาจากหน่วยเก็บข้อมูลของมาตรฐานหนึ่ง และส่งไปเก็บยังหน่วยเก็บข้อมูลของอีกมาตรฐานหนึ่ง ซึ่งจะต้องมีการแปลงรูปแบบของการส่งต่อข้อมูลและลักษณะของข้อมูลให้สอดคล้องกันกับทั้งสองมาตรฐาน ทั้งนี้จะกล่าวถึงวิธีการตามกระบวนการต่างๆ ในบทที่ 3

ในขั้นตอนของการพัฒนาโพรโทคอลจะใช้ภาษา JavaScript ตามการดำเนินการโปรแกรมแบบ Node.js ในการออกแบบโปรแกรมเพื่อให้สอดคล้องกับรูปแบบการร้องขอ และรูปแบบการส่งต่อข้อมูล รวมไปถึงลักษณะของข้อมูลของทั้งสองมาตรฐานข้างต้นด้วย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.3 วัตถุประสงค์

เพื่อพัฒนาโพรโทคอลของเกตเวย์ที่ใช้สำหรับการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M ในการดึงข้อมูลตัวรับรู้จากหน่วยเก็บข้อมูลของมาตรฐานหนึ่งและแปลงข้อมูลเพื่อส่งไปเก็บในหน่วยเก็บข้อมูลของอีกมาตรฐานหนึ่งได้

1.4 ขอบเขตของวิทยานิพนธ์

1. วิเคราะห์เปรียบเทียบประสิทธิภาพระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M ตามหัวข้อดังนี้ รูปแบบโพรโทคอลสื่อสาร ส่วนประกอบของโพรโทคอล รูปแบบการดำเนินการในโพรโทคอล และรูปแบบของข้อมูลที่รองรับในการสื่อสาร

2. พัฒนาโพรโทคอลของเกตเวย์เพื่อใช้สำหรับการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M ในการดึงข้อมูลที่ถูกบันทึกจากตัวรับรู้ไร้สาย ZigBee จากหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 ด้วยโพรโทคอล FETCH และแปลงรูปแบบของข้อมูลให้สอดคล้องกับมาตรฐาน ETSI M2M จากนั้นส่งข้อมูลไปเก็บในหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M พร้อมกับใช้ในการดึงข้อมูลที่ถูกบันทึกในหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M ด้วยระเบียบวิธี RETRIEVE และแปลงรูปแบบของข้อมูลให้สอดคล้องกับมาตรฐาน IEEE1888 จากนั้นส่งข้อมูลไปเก็บในหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 ด้วยโพรโทคอล WRITE

1.5 ขั้นตอนการดำเนินงาน

1. ศึกษาระเบียบวิธีการสื่อสารตามมาตรฐาน IEEE1888
2. ศึกษาระเบียบวิธีการสื่อสารตามมาตรฐาน ETSI M2M
3. เปรียบเทียบโครงสร้าง วิธีการสื่อสาร ความสามารถและสมรรถนะของมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M
4. วิเคราะห์ผลการเปรียบเทียบสองมาตรฐาน
5. พัฒนาโพรโทคอลของเกตเวย์เพื่อใช้สำหรับการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M
6. วิเคราะห์และประเมินผลการดำเนินงานของโพรโทคอลของเกตเวย์ทำงานร่วมกันระหว่างสองมาตรฐาน
7. สรุปผลและรวบรวมข้อมูลเพื่อจัดทำเล่มวิทยานิพนธ์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.6 ประโยชน์ที่คาดว่าจะได้รับ

องค์ความรู้ในการพัฒนาเกตเวย์ที่ใช้สำหรับการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M ในการดึงข้อมูลตัวรับรู้จากหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 แล้วแปลงข้อมูลเพื่อส่งไปเก็บในหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M และดึงข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M แล้วแปลงข้อมูลเพื่อส่งไปเก็บในหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 และองค์ความรู้ในการทดสอบสมรรถนะการทำงานร่วมกันของทั้งสองแพลตฟอร์ม

1.7 ประมวลวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งรายละเอียดออกเป็น 5 บท ดังนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์และแนวทางของวิทยานิพนธ์ ขอบเขตของวิทยานิพนธ์ ขั้นตอนการดำเนินงาน ประโยชน์ที่คาดว่าจะได้รับ และประมวลวิทยานิพนธ์

บทที่ 2 ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง โดยกล่าวถึงความรู้พื้นฐานของงานวิจัย ประกอบด้วย มาตรฐาน IEEE1888 มาตรฐาน ETSI M2M และการเปรียบเทียบระหว่างทั้งสองมาตรฐาน

บทที่ 3 อธิบายเกี่ยวกับโครงสร้างของระบบและกระบวนการทำงานของเกตเวย์จำลอง รวมไปถึงสภาพแวดล้อมของระบบที่ใช้ในการทดสอบ

บทที่ 4 บทนี้จะกล่าวถึงผลการทดสอบ และวิเคราะห์ผลการทดสอบโพรโทคอลที่นำเสนอซึ่งประกอบไปด้วย การทดสอบเบื้องต้น การทดสอบโครงสร้างการจำลอง และการเปรียบเทียบสมรรถนะของโครงสร้างการจำลองแบบต่างๆ

บทที่ 5 บทสรุป กล่าวถึงบทสรุป และข้อเสนอแนะของงานวิจัยในอนาคต

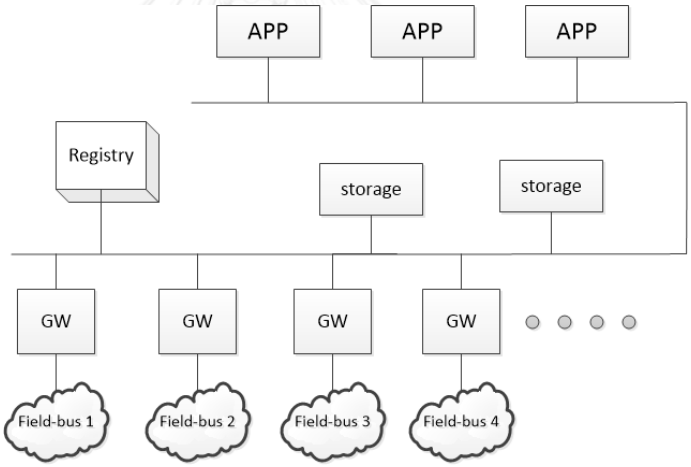
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 2
ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง

2.1 มาตรฐาน IEEE1888

2.1.1 สถาปัตยกรรมของ IEEE1888

สถาปัตยกรรมของมาตรฐาน IEEE1888 (IEEE1888 architecture) แบ่งออกเป็น 2 ระนาบ ได้แก่ ระนาบควบคุม (Control plane) และ ระนาบข้อมูล (Data plane) โดยระนาบควบคุม คือ ส่วนของรีจิสทรี (Registry) ในส่วนระนาบข้อมูลแบ่งออกเป็น 3 ส่วนประกอบ ได้แก่ เกตเวย์ (Gateway; GW) หน่วยเก็บข้อมูล (Data Storage) และ โปรแกรมประยุกต์ (Application; APP) ซึ่งทั้งสองระนาบจะสื่อสารระหว่างกันบนพื้นฐานของโครงข่ายทีซีพี/ไอพี โครงสร้างของสถาปัตยกรรมตามมาตรฐาน IEEE1888 แสดงดังรูปที่ 1



รูปที่ 1 สถาปัตยกรรมตามมาตรฐาน IEEE1888[3]

จากรูปที่ 1 ส่วนประกอบของสถาปัตยกรรมตามมาตรฐาน IEEE1888 มีดังนี้

2.1.1.1 เกตเวย์ (Gateway)

ทำหน้าที่เป็นตัวกลางในการเชื่อมต่อระหว่างโครงข่ายตัวรับรู้และตัวกระตุ้น กับโครงข่ายทีซีพี/ไอพีเพื่อให้การสื่อสารเป็นไปตามรูปแบบของมาตรฐาน IEEE1888

2.1.1.2 หน่วยเก็บข้อมูล (Storage)

ทำหน้าที่บันทึกข้อมูลที่ได้จากการวัดของตัวรับรู้ที่ติดตั้งตามสถานที่ต่าง ๆ ตามลำดับเวลาที่บันทึกข้อมูล

2.1.1.3 โปรแกรมประยุกต์ (Application)

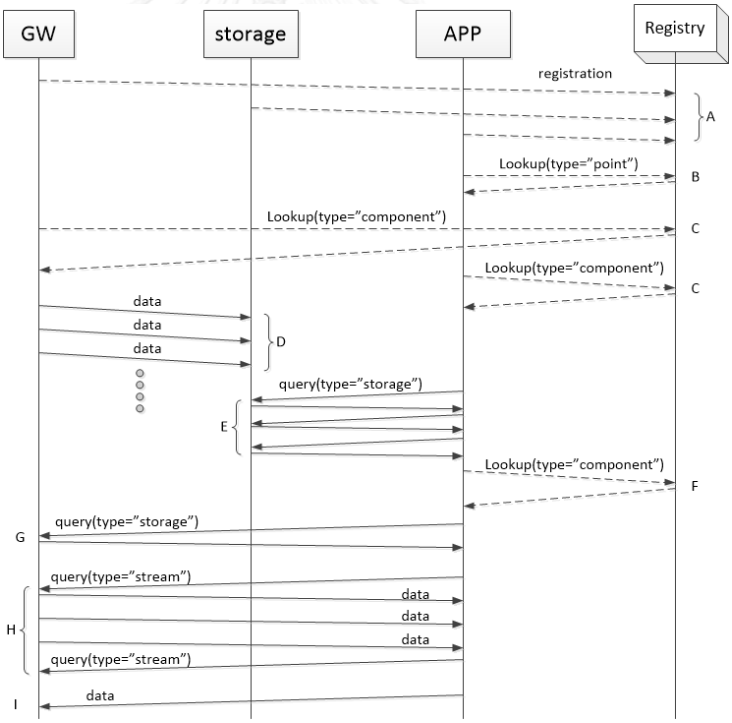
ทำหน้าที่เฝ้าสังเกตสถานะของตัวรับรู้ที่ถูกบันทึกในหน่วยเก็บข้อมูลเกี่ยวกับระบบการจัดการพลังงานไฟฟ้าที่สามารถตรวจสอบข้อมูลย้อนหลังได้โดยแสดงเป็นรูปกราฟ เพื่อวิเคราะห์และตรวจสอบแนวโน้มของข้อมูลในรายชั่วโมง รายวัน รายสัปดาห์ รายเดือน หรือรายปี เพื่อวางแผนการใช้งานของเครื่องใช้ไฟฟ้าในอนาคตได้

2.1.1.4 รีจิสทรี (Registry)

ทำหน้าที่เป็นระบบศูนย์กลางระหว่างส่วนประกอบทั้ง 3 ส่วน ซึ่งได้แก่ เกตเวย์ หน่วยเก็บข้อมูล และโปรแกรมประยุกต์ เพื่อควบคุมการเชื่อมโยงในการสื่อสาร

2.1.2 ลำดับการสื่อสารของโพรโทคอล

เมื่อส่วนประกอบหนึ่งต้องการที่จะเข้าถึงข้อมูล ขั้นตอนแรกส่วนประกอบนั้นต้องทำการเชื่อมต่อกับรีจิสทรี หลังจากนั้นรีจิสทรีจะทำการตอบกลับด้วย access URIs ของส่วนประกอบที่ทำหน้าที่จัดการข้อมูลและเชื่อมต่อกับ Point ID ดังรูปที่ 2



รูปที่ 2 ลำดับการสื่อสารของโพรโทคอล[3]

ลำดับการสื่อสารของโพรโทคอล (Typical sequence) สามารถอธิบายด้วยการทำงานของกรณีต่าง ๆ ในระบบสื่อสารตามมาตรฐาน IEEE1888 แสดงดังตารางที่ 1

ตารางที่ 1 การทำงานของกรณีต่างๆในระบบสื่อสารตามมาตรฐาน IEEE1888 [3]

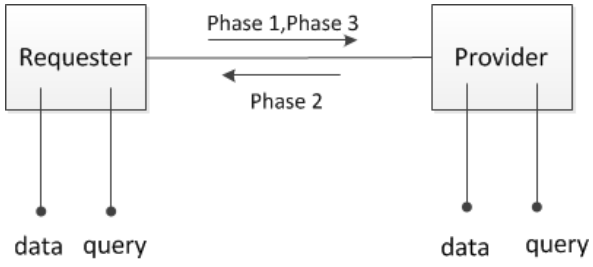
กรณี	การทำงาน
A	การลงทะเบียน Point ID ของส่วนประกอบต่าง ๆ ได้แก่ เกตเวย์ หน่วยเก็บข้อมูล และโปรแกรมประยุกต์
B	การค้นหา Point โดยใช้โพรโทคอลเรียกถาม (Query protocol)
C	ค้นหา URI
D	เกตเวย์ส่งข้อมูลจากโครงข่ายตัวรับรู้ไปยังหน่วยเก็บข้อมูล
E	โปรแกรมประยุกต์ร้องขอข้อมูลจากหน่วยเก็บข้อมูลโดยเริ่มดึงข้อมูลจาก เวลาเริ่มต้นของการร้องขอ จนกระทั่งเวลาสุดท้ายของการร้องขอ
F	การค้นหา URI ของเกตเวย์โดยระบุ Point
G	โปรแกรมประยุกต์กำหนดค่าต่าง ๆ ที่เกตเวย์
H	โปรแกรมประยุกต์ร้องขอให้ส่งข้อมูล เพื่ออัพเดตข้อมูลไปยังเกตเวย์เพื่อตั้งค่าการทำงาน และเกตเวย์จะตอบกลับมาพร้อมกับข้อมูลที่เกิดเหตุการณ์ตรงกับข้อมูลที่โปรแกรมประยุกต์ได้อัพเดตไป
I	โปรแกรมประยุกต์ส่งข้อมูลเพื่อสั่งการทำงานไปยังเกตเวย์

2.1.3 โพรโทคอลการสื่อสารระหว่างส่วนประกอบ

โพรโทคอลหลักที่ใช้ในการสื่อสารกันระหว่างส่วนประกอบต่าง ๆ นั้น มีอยู่ 3 โพรโทคอล คือ FETCH, WRITE และ TRAP ซึ่งมีรายละเอียดการทำงานของแต่ละโพรโทคอล ดังนี้

2.1.3.1 โพรโทคอล FETCH

โพรโทคอล FETCH ใช้สำหรับร้องขอข้อมูลจากส่วนประกอบที่ต้องการทำการเรียกดูข้อมูลระยะไกล โดยส่วนประกอบของการดำเนินการประกอบไปด้วย ผู้ร้องขอข้อมูล (Requester) และผู้ให้บริการข้อมูล (Provider) ซึ่งมีการรับส่งข้อมูลดังรูปที่ 3



รูปที่ 3 โพรโทคอล FETCH [3]

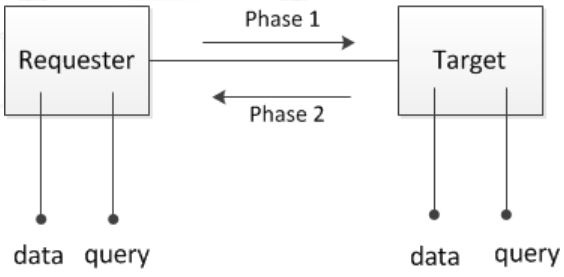
ช่วงที่ 1 ผู้ร้องขอข้อมูล ส่งร้องขอข้อมูลไปยังผู้ให้บริการข้อมูล โดยการส่งช่วงเวลาของข้อมูลที่ต้องการไปยังผู้ให้ข้อมูล และระบุขนาดสูงสุดของข้อมูล (Acceptable size) ที่สามารถรับได้ด้วย

ช่วงที่ 2 ผู้ให้บริการข้อมูลส่งข้อมูลกลับมายังผู้ร้องขอข้อมูล โดยที่จะมีการแบ่งข้อมูลออกเป็นชุดย่อย ๆ ถ้าหากข้อมูลที่ร้องขอนั้นมีขนาดใหญ่กว่าปริมาณข้อมูลที่ร้องขอได้ระบุไป และชุดข้อมูลนั้นจะกลับไปพร้อมกับตัวชี้ตำแหน่งของข้อมูล (Cursor) ถ้าหากผู้ร้องขอข้อมูลไม่ได้รับตัวชี้ตำแหน่งข้อมูลมาพร้อมกับชุดข้อมูล แสดงว่าข้อมูลที่ได้รับมาเป็นข้อมูลชุดสุดท้าย จึงเป็นอันสิ้นสุดการทำงานของโพรโทคอล FETCH ไม่มีการกลับมาทำงานในช่วงที่ 3 อีกครั้ง

ช่วงที่ 3 ถ้าหากผู้ร้องขอข้อมูลยังได้รับตำแหน่งของข้อมูลมาพร้อมกับชุดข้อมูลด้วย แสดงว่าข้อมูลยังไม่สิ้นสุด ผู้ร้องขอข้อมูลจะต้องทำการร้องขอข้อมูลพร้อมกับตัวชี้ตำแหน่งไปยังผู้ให้ข้อมูลอีกครั้ง และเข้าสู่การทำงานช่วงที่ 2 ต่อไป

2.1.3.2 โพรโทคอล WRITE

โพรโทคอล WRITE ทำหน้าที่ถ่ายโอนข้อมูลไปยังส่วนประกอบหนึ่งที่ต้องการส่งข้อมูลไปถึง โดยการดำเนินการคือ ผู้ร้องขอส่งข้อมูล (Requester) ไปยังส่วนประกอบเป้าหมาย (target) ซึ่งมีการรับส่งข้อมูลดังรูปที่ 4



รูปที่ 4 โพรโทคอล WRITE [3]

ช่วงที่ 1 ผู้ส่งข้อมูลทำหน้าที่ส่งข้อมูลที่ต้องการส่งไปยังส่วนประกอบเป้าหมาย

ช่วงที่ 2 ส่วนประกอบเป้าหมายส่งตอบกลับไปยังผู้ขอส่งข้อมูลว่า การส่งข้อมูลสำเร็จ หรือเกิดข้อผิดพลาดในการส่ง

2.1.3.3 โพรโทคอล TRAP

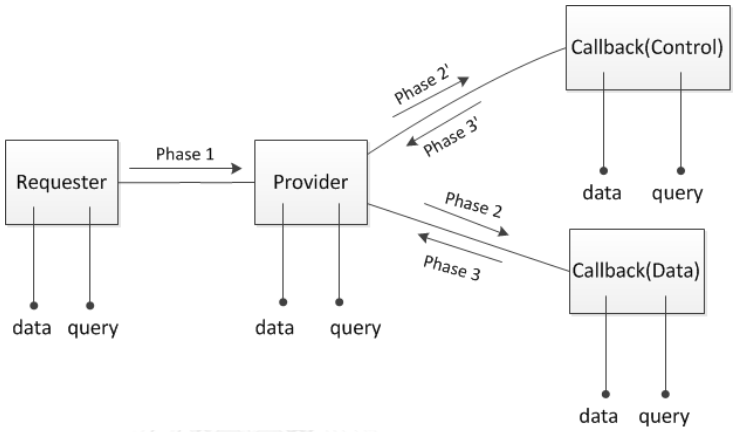
โพรโทคอล TRAP ใช้สำหรับการชี้แจงเหตุการณ์ล่วงหน้าที่ต้องการแจ้งเตือน (Notification) และถ่ายโอนข้อมูลเมื่อเกิดเหตุการณ์ตามที่ชี้แจงไว้ โดยมีการระบุชื่อของส่วนประกอบต่าง ๆ ดังนี้

ผู้ร้องขอข้อมูล (Requester) คือ ส่วนประกอบที่ทำการตั้งค่าเหตุการณ์ที่ต้องการให้มีการแจ้งเตือนไปยังผู้ให้บริการข้อมูล

ผู้ให้บริการข้อมูล (Provider) คือ ส่วนประกอบที่ส่งข้อมูลเมื่อได้รับการแจ้งเตือนจากการที่เกิดเหตุการณ์ตรงกับที่ผู้ร้องขอข้อมูลได้ชี้แจงไว้

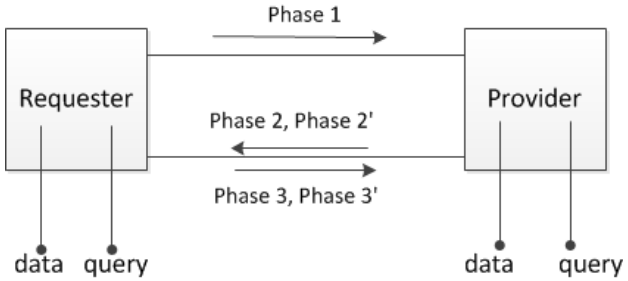
ผู้เรียกกลับข้อมูล (Callback (Data)) คือ ส่วนประกอบที่รับข้อมูลจากผู้ให้บริการข้อมูล

ผู้เรียกกลับการควบคุม (Callback(Control)) คือ ส่วนประกอบที่รับสัญญาณควบคุมจากผู้ให้บริการข้อมูล



รูปที่ 5 โพรโทคอล TRAP (รูปแบบการนำไปใช้งานแบบทั่วไป) [3]

จากรูปที่ 5 แสดงให้เห็นถึงการทำงานร่วมกันของส่วนประกอบต่าง ๆ แต่ในทางปฏิบัติเพื่อที่จะให้เห็นถึงการแบ่งหมวดหมู่อย่างชัดเจนของการสื่อสารข้อมูลระหว่างส่วนประกอบต่าง ๆ โดยที่ผู้ให้บริการข้อมูล ผู้เรียกกลับข้อมูลและผู้เรียกกลับการควบคุมเป็นส่วนประกอบเดียวกัน แสดงดังรูปที่ 6



รูปที่ 6 โพรโทคอล TRAP (รูปแบบการนำไปใช้งานในทางปฏิบัติ) [3]

ช่วงที่ 1 ผู้ร้องขอข้อมูลชี้แจงเหตุการณ์ล่วงหน้าที่ต้องการให้มีการแจ้งเตือนไปยังผู้ให้บริการข้อมูลโดยต้องระบุเวลาหมดอายุของเหตุการณ์ในหน่วยวินาที เช่น TTL(Time To Leave) =

500 ms และระบุ URI สำหรับการตอบกลับของผู้เรียกกลับข้อมูล และผู้เรียกกลับการควบคุม

ช่วงที่ 2 เมื่อเกิดเหตุการณ์ที่ตรงกับที่ผู้ร้องขอได้ชี้แจงไว้ ผู้ให้บริการข้อมูลจะตอบกลับไปยัง URI ของผู้เรียกกลับข้อมูล

ช่วงที่ 3 ผู้เรียกกลับข้อมูลตอบกลับไปยังผู้ให้บริการข้อมูลว่า ได้รับข้อมูลเรียบร้อยแล้วหรือเกิดความผิดพลาดขึ้น

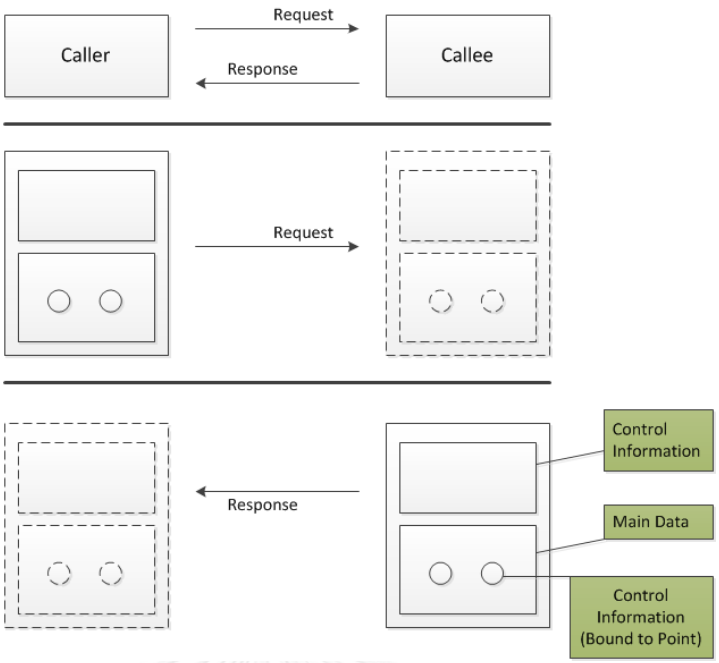
ช่วงที่ 2' ถ้าหากในช่วงที่ 3 เกิดความผิดพลาดในการส่งข้อมูลขึ้น ผู้ให้บริการข้อมูลจะตอบกลับไปยัง URI ของผู้เรียกกลับการควบคุมว่าเกิดความผิดพลาดขึ้นในช่วงที่ 3

ช่วงที่ 3' ผู้เรียกกลับการควบคุมตอบกลับไปยังผู้ให้บริการข้อมูลว่า ได้รับข้อมูลเรียบร้อยแล้วหรือเกิดความผิดพลาดขึ้น

2.1.4 โครงสร้างของข้อมูลที่ใช้ในการสื่อสาร

การสื่อสารระหว่างส่วนประกอบหนึ่งกับอีกส่วนประกอบหนึ่ง (Component-to-component) ได้นำโพรโทคอล RPC (Remote Procedure Call) มาใช้งานดังแสดงในรูปที่ 7 โดยผู้ร้องขอ (Caller) ทำหน้าที่ส่งข้อมูลที่เรียกว่า ข้อมูลร้องขอ (Request-data) ไปยัง ผู้รับ (Receiver) และผู้รับทำหน้าที่ตอบกลับข้อมูลที่เรียกว่า ข้อมูลตอบกลับ (Response-data) ไปยังผู้ร้องขอ โดยมีโครงสร้างของข้อมูลของการร้องขอและการตอบกลับที่เหมือนกัน

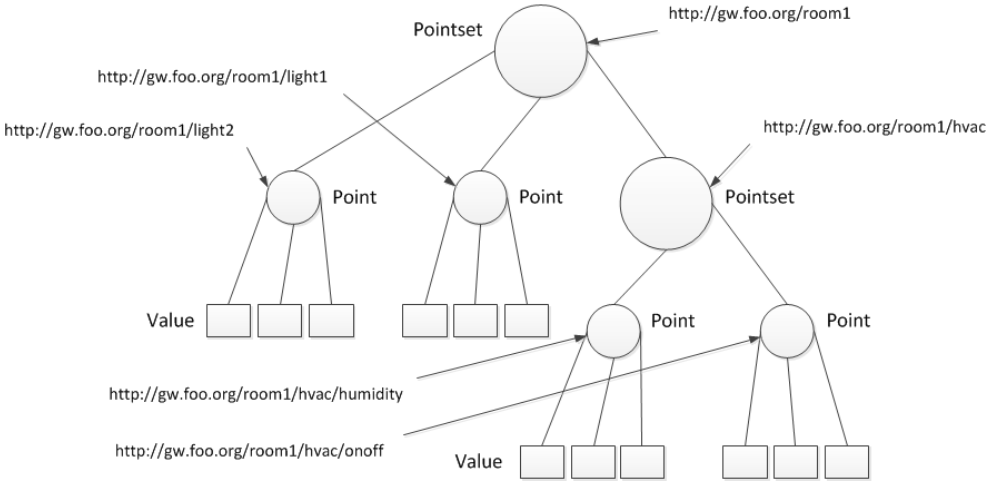
ข้อมูลของ RPC นั้นจะถูกแบ่งออกเป็น 2 ส่วน คือ ส่วนหัว (Header part) และส่วนใจความ (Body part) โดยส่วนหัวประกอบไปด้วยข้อมูลที่ทำการควบคุม ตัวอย่างเช่น ข้อมูลแสดงการรับทราบ (Acknowledgement) และข้อมูลแสดงความผิดพลาด (false information) เป็นต้น ส่วนใจความประกอบไปด้วยวัตถุ (Object) ของตำแหน่งข้อมูล (Point) หรือเซตของตำแหน่งข้อมูล (Pointset) ตัวอย่างเช่น ข้อมูลที่อ่านมาจากตัวรับรู้ เป็นต้น



รูปที่ 7 โครงสร้างข้อมูลที่ใช้ในการสื่อสาร [3]

2.1.5 การจัดการตำแหน่งข้อมูลด้วยโครงสร้างเซตของตำแหน่งข้อมูล

การจัดการตำแหน่งข้อมูลด้วยโครงสร้างเซตของตำแหน่งข้อมูลซึ่งโครงสร้างนี้มีลักษณะเป็นการจัดการแบบลำดับขั้น (Hierarchy) ของตำแหน่งข้อมูล เซตของตำแหน่งข้อมูล (Pointset tree) ประกอบด้วยตำแหน่งข้อมูลและเซตของตำแหน่งข้อมูล โดยที่ตำแหน่งข้อมูล 1 ตำแหน่งประกอบด้วยส่วนย่อยเพียง 1 ลำดับคือ ค่าของข้อมูล (value) ในค่าของข้อมูลจะบรรจุค่าจากเซนเซอร์หรือคำสั่งของตัวกระตุ้น แสดงโครงสร้างทั่วไปของตำแหน่งข้อมูล แสดงดังรูปที่ 8



รูปที่ 8 รูปแบบโครงสร้างเซตของตำแหน่งข้อมูล [3]

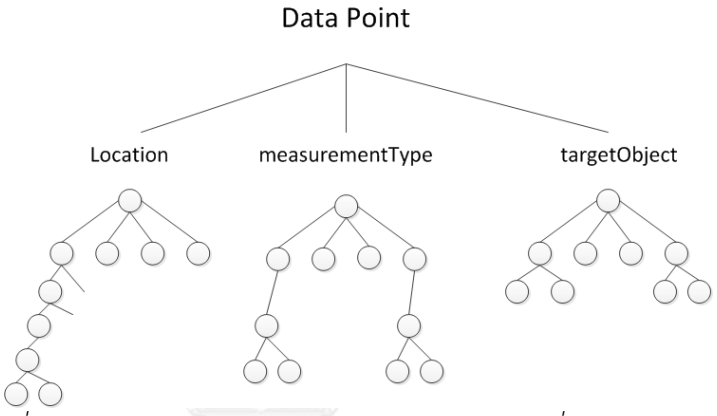
2.1.6 การจัดการโครงสร้างของตำแหน่งข้อมูลตามความหมายด้วยสารบบทั่วไป

วิธีหนึ่งของสารบบทั่วไป (ubiquitous directory) แบ่งการจัดการตำแหน่งของข้อมูลออกเป็น 3 ส่วน ดังนี้

ส่วนที่ 1 ตำแหน่งที่ตั้ง (Location model) เป็นส่วนที่บ่งบอกถึงตำแหน่งที่ตั้งจริงของข้อมูล

ส่วนที่ 2 ชนิดของการวัด (measurementType model) เป็นส่วนที่บ่งบอกถึงชนิดของข้อมูลซึ่งมีความจำเป็นในขั้นการดำเนินการข้อมูล

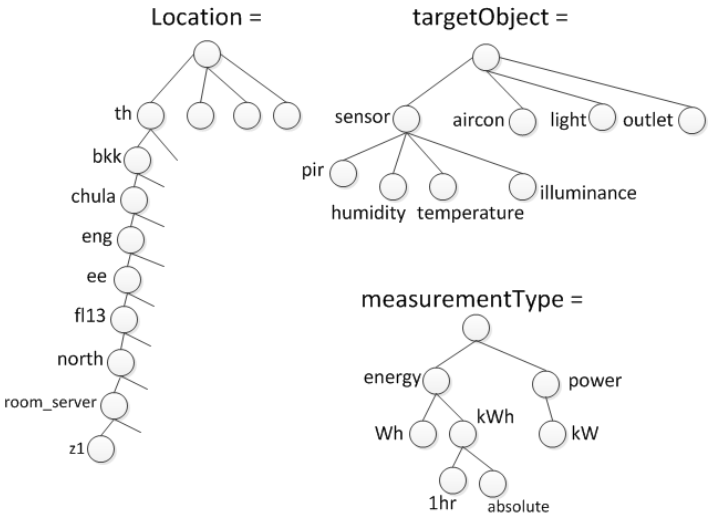
ส่วนที่ 3 วัตถุเป้าหมาย (TargetObject model) เป็นส่วนที่บ่งบอกว่า วัตถุหรืออุปกรณ์ที่ถูกตรวจสอบอยู่



รูปที่ 9 โครงสร้างเซตของตำแหน่งข้อมูลตามสารบบทั่วไป [6]

จากวิธีการจัดการโครงสร้างตามสารบบทั่วไป นำมาเขียนยกตัวอย่างการออกแบบโครงสร้างของจุดข้อมูลได้ดังรูปที่ 10 ซึ่งอิงข้อมูลจากตำแหน่งของการติดตั้งอุปกรณ์ที่ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มีรายละเอียดดังนี้

1) **สถานที่** คือตามสถานที่ตั้งของอุปกรณ์
   - ประเทศไทย กรุงเทพมหานคร จุฬาลงกรณ์มหาวิทยาลัย, คณะวิศวกรรมศาสตร์, ภาควิชาวิศวกรรมไฟฟ้า, ตึก 4, ชั้น 13, พื้นที่ทางทิศเหนือ, ห้องserver, โซน 1
     Location = /th/bkk/chula/eng/eng4/fl13/north/room_server/z1
2) **อุปกรณ์**
   - ตัวรับรู้, วัดค่าอุณหภูมิ
     targetObject = /sensor/temperature
3) **ชนิดของข้อมูล**
   - พลังงาน, มีหน่วยเป็น kWh, วัดทุก ๆ 1 ชั่วโมง
     measurementType = /energy/kWh/1hr

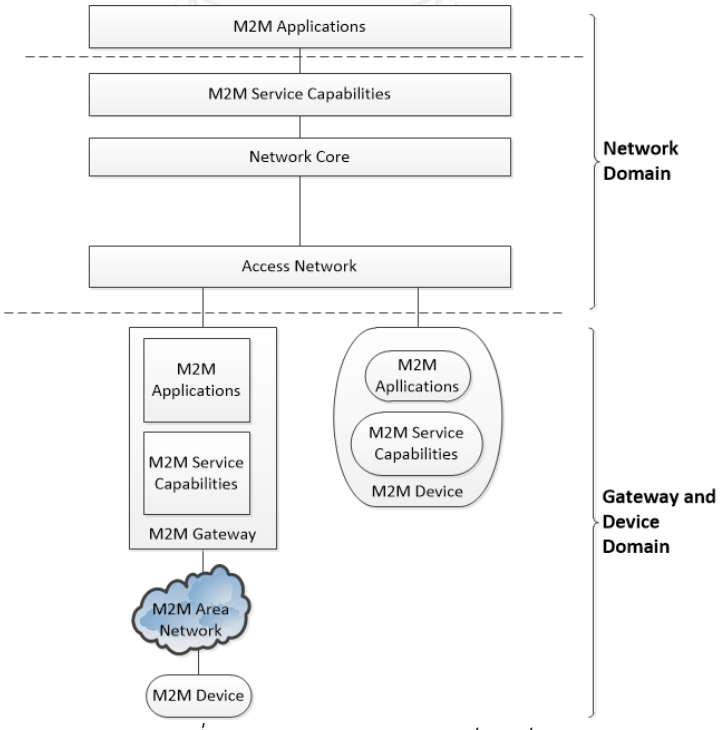


รูปที่ 10 ตัวอย่างการสร้างตัวแบบเซตโปรแกรมประยุกต์ [6]

2.2 มาตรฐาน ETSI M2M

2.2.1 สถาปัตยกรรมของมาตรฐาน ETSI M2M

สถาบันมาตรฐานโทรคมนาคมแห่งยุโรป ได้กำหนดมาตรฐานเกี่ยวกับการสื่อสารแบบแมชชีน-ทู-แมชชีน เพื่อรองรับกับปริมาณการใช้งานที่กำลังเพิ่มขึ้นอย่างรวดเร็ว โดยมาตรฐานที่ระบุถึงโครงสร้างของเอ็มทูเอ็ม (M2M architecture) คือมาตรฐาน ETSI TS 102 690 [5] ดังแสดงในรูปที่ 11



รูปที่ 11 สถาปัตยกรรมของเอ็มทูเอ็ม[5]

จากรูปที่ 11 แสดงให้เห็นว่าสถาปัตยกรรมของมาตรฐาน ETSI M2M แบ่งโครงสร้างของเอ็มทูเอ็มออกเป็น 2 ส่วน ดังนี้

2.2.1.1 โดเมนอุปกรณ์และเกตเวย์ (Device and gateway domain)

1) **อุปกรณ์เอ็มทูเอ็ม (M2M devices)** ทำหน้าที่ใช้โปรแกรมประยุกต์เอ็มทูเอ็ม (M2M application) โดยอาศัยความสามารถในการบริการของเอ็มทูเอ็ม (M2M service capability)
2) **โครงข่ายพื้นที่เอ็มทูเอ็ม (M2M Area Network)** ทำหน้าที่เชื่อมโยงระหว่างอุปกรณ์เอ็มทูเอ็ม กับเกตเวย์เอ็มทูเอ็ม (M2M gateway) โดยแบ่งการเชื่อมโยงออกเป็น 2 ลักษณะดังนี้
    2.1) โครงข่ายส่วนบุคคล (Personal area network) เช่น IEEE802.15.1 Zigbee Bluetooth IETF Roll ISA 100.11a เป็นต้น
    2.2) โครงข่ายเฉพาะที่ (Local network) เช่น PLC M-BUS Wireless M-BUS KNX เป็นต้น
3) **เกตเวย์เอ็มทูเอ็ม** ทำหน้าที่ในการใช้งานโปรแกรมประยุกต์เอ็มทูเอ็มซึ่งเป็นโปรแกรมที่จัดการและจัดเก็บข้อมูลหลากหลายรูปแบบ โดยอาศัยความสามารถในการบริการเอ็มทูเอ็มและเป็นตัวกลางระหว่างอุปกรณ์เอ็มทูเอ็มกับโดเมนโครงข่าย (Network domain) เกตเวย์เอ็มทูเอ็มสามารถทำงานร่วมกับอุปกรณ์ที่มีรูปแบบต่างกัน เช่น Android Arduino Raspberry Pi เป็นต้น และสามารถสนับสนุนโพรโทคอลหลากหลายเทคโนโลยี เช่น Zigbee FS20 Bluetooth เป็นต้น

2.2.1.2 โดเมนโครงข่าย ประกอบไปด้วย

1) **โครงข่ายการเข้าถึง (Access network)** ทำหน้าที่อนุญาตให้โดเมนอุปกรณ์และเกตเวย์เอ็มทูเอ็มติดต่อสื่อสารกับศูนย์กลางของโครงข่าย (Core network) ผ่านเทคโนโลยีต่างๆ ดังนี้ xDSL, HFC, satellite, GERAN, UTRAN, eUTRAN, W-LAN and WiMAX
2) **ศูนย์กลางของโครงข่าย** จัดหาการเชื่อมโยงไอพี ฟังก์ชันการควบคุมการให้บริการและการสื่อสาร การติดต่อซึ่งกันและกัน และการใช้งานข้ามเขต (roaming)

3) ความสามารถในการบริการของเอ็มทูเอ็ม มีความสามารถดังนี้

- สามารถจัดหาฟังก์ชันเอ็มทูเอ็ม (M2M functions) และแสดงฟังก์ชันผ่านส่วนต่อประสานแบบเปิด (Open Interface)
- สามารถใช้ฟังก์ชันของศูนย์กลางของโครงข่าย
- สามารถทำให้การพัฒนาโปรแกรมประยุกต์เป็นไปอย่างง่ายและเหมาะสม

ซึ่งความสามารถในการบริการของเอ็มทูเอ็ม แบ่งออกเป็นความสามารถต่างๆ ดังนี้

3.1) การเปิดใช้งานโปรแกรมประยุกต์(Application Enablement (xAE))

3.2) การสื่อสารทั่วไป (Generic Communication (xGC))

3.3) ความสามารถในการเข้าถึง, การกำหนดเลขที่อยู่ และแหล่งเก็บข้อมูล (Reachability, Addressing and Repository (xRAR))

3.4) การเลือกใช้การสื่อสาร (Communication Selection (xCS))

3.5) การจัดการข้อมูลในระยะไกล (Remote Entity Management (xREM))

3.6) ความปลอดภัย (SECurity (xSEC))

3.7) ประวัติและการรักษาข้อมูล (History and Data Retention (xHDR))

3.8) การจัดการรายการเปลี่ยนแปลง (Transaction Management (xTM))

3.9) นายหน้าค่าทดแทน (Compensation Broker (xCB))

3.10) การแสดงออกของผู้ควบคุมหน่วยงานโทรคมนาคม (Telco Operator Exposure (xTOE))

3.11) ตัวแทนในการทำงานร่วมกัน (Interworking Proxy (xIP))

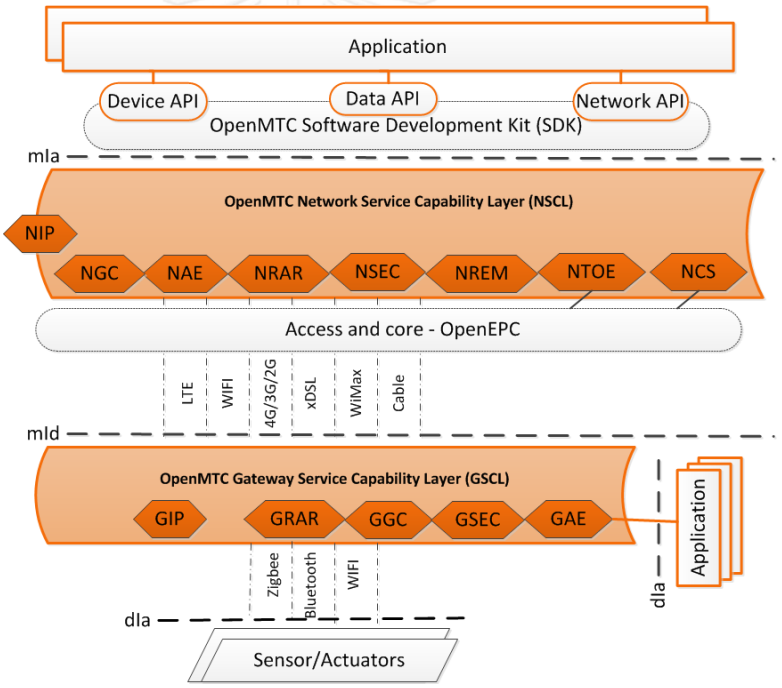
โดยที่ความสามารถในการบริการถูกนำมาใช้ในส่วนต่างๆ ดังนี้

- ชั้นความสามารถในการบริการโครงข่าย (Network Service Capabilities Layer (NSCL)) คือ ความสามารถในการบริการของเอ็มทูเอ็มในโดนเมนโครงข่าย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

- ชั้นความสามารถในการบริการเกตเวย์ (Gateway Service Capabilities Layer (GSCL)) คือ ความสามารถในการบริการของเอ็มทูเอ็มในเกตเวย์เอ็มทูเอ็ม
- ชั้นความสามารถในการบริการอุปกรณ์ (Device Service Capabilities Layer (DSCL)) คือ ความสามารถในการบริการของเอ็มทูเอ็มในอุปกรณ์เอ็มทูเอ็ม

จากแพลตฟอร์มของโอเพ็นเอ็มทีซี (OpenMTC platform) [4][7] ซึ่งถูกออกแบบอิงตามมาตรฐาน ETSI M2M แสดงให้เห็นถึงชั้นความสามารถในการบริการเอ็มทูเอ็ม และตัวอย่างการนำความสามารถในการบริการของเอ็มทูเอ็มมาใช้ในโอเพ็นเอ็มทีซี ซึ่งแยกตามส่วนประกอบในโครงสร้างของเอ็มทูเอ็มดังรูปที่ 12



รูปที่ 12 สถาปัตยกรรมของ โอเพ็นเอ็มทีซี [7]

สำหรับส่วนต่อประสาน (Interface) แต่ละส่วน ซึ่งตามข้อกำหนดของมาตรฐาน ETSI M2M แบ่งออกเป็น 3 ส่วนต่อประสานดังแสดงในรูปที่ 12 ดังนี้

- mIa (M2M application interface) เป็นส่วนเชื่อมต่อระหว่างโปรแกรมประยุกต์โครงข่ายกับชั้นความสามารถในการบริการโครงข่าย

- dIa (Device application interface) เป็นส่วนเชื่อมต่อระหว่างโปรแกรมประยุกต์เกตเวย์กับชั้นความสามารถในการบริการเกตเวย์ และเป็นส่วนเชื่อมต่อระหว่างโปรแกรมประยุกต์อุปกรณ์กับชั้นความสามารถในการบริการอุปกรณ์หรือชั้นความสามารถในการบริการเกตเวย์
- mId (M2M to device interface) เป็นส่วนเชื่อมต่อระหว่างชั้นความสามารถในการบริการเกตเวย์หรือชั้นความสามารถในการบริการอุปกรณ์กับชั้นความสามารถในการบริการโครงข่าย

4) โปรแกรมประยุกต์เอ็มทูเอ็ม (M2M application)) ทำหน้าที่ดำเนินการตามตรรกะการให้บริการ และใช้ความสามารถในการบริการของเอ็มทูเอ็มผ่านส่วนต่อประสานแบบเปิด โดยที่โปรแกรมประยุกต์เอ็มทูเอ็มถูกนำมาใช้ในส่วนต่างๆ ดังนี้

4.1) โปรแกรมประยุกต์โครงข่าย (Network application(NA)) คือ โปรแกรมประยุกต์เอ็มทูเอ็มที่เชื่อมต่อกับชั้นความสามารถในการบริการโครงข่าย

4.2) โปรแกรมประยุกต์เกตเวย์ (Gateway application(GA)) คือ โปรแกรมประยุกต์เอ็มทูเอ็มที่เชื่อมต่อกับชั้นความสามารถในการบริการเกตเวย์

4.3) โปรแกรมประยุกต์อุปกรณ์ (Device application(DA)) คือ โปรแกรมประยุกต์เอ็มทูเอ็มที่เชื่อมต่อกับชั้นความสามารถในการบริการอุปกรณ์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.2.2 โปรแกรมประยุกต์ต่อประสานของเอ็มทูเอ็ม

ในมาตรฐาน ETSI M2M วิธีการที่โปรแกรมประยุกต์เอ็มทูเอ็ม และชั้นความสามารถในการบริการเอ็มทูเอ็ม ใช้ในการแลกเปลี่ยนข้อมูลระหว่างกันคือ วิธีการตามโครงสร้าง RESTful ซึ่งเป็นการส่งต่อการแสดงตำแหน่งของทรัพยากรนั้นๆ โดยที่มาตรฐาน ETSI M2M ได้กำหนดโครงสร้างของทรัพยากรให้อยู่ในชั้นความสามารถในการบริการที่เกี่ยวข้อง โครงสร้างแบบ RESTful ใช้ในการจัดการทรัพยากรแบ่งออกเป็น 4 กระบวนการ แสดงดังรูปที่ 13

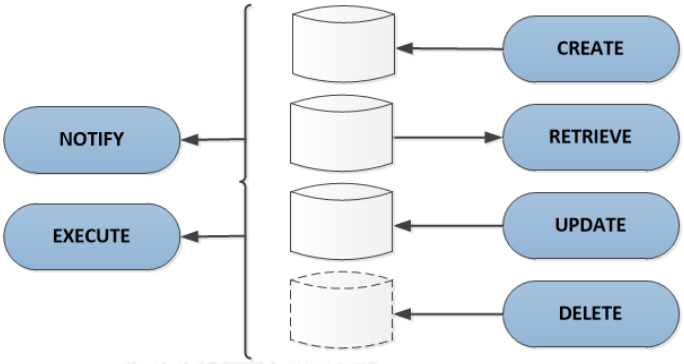
RESTful มี 4 กระบวนการดังนี้ [5]

- CRATE สร้างทรัพยากร
- RETRIEVE อ่านเนื้อหาในทรัพยากร

- UPDATE ปรับแก้เนื้อหาในทรัพยากร
- DELETE ลบทรัพยากร

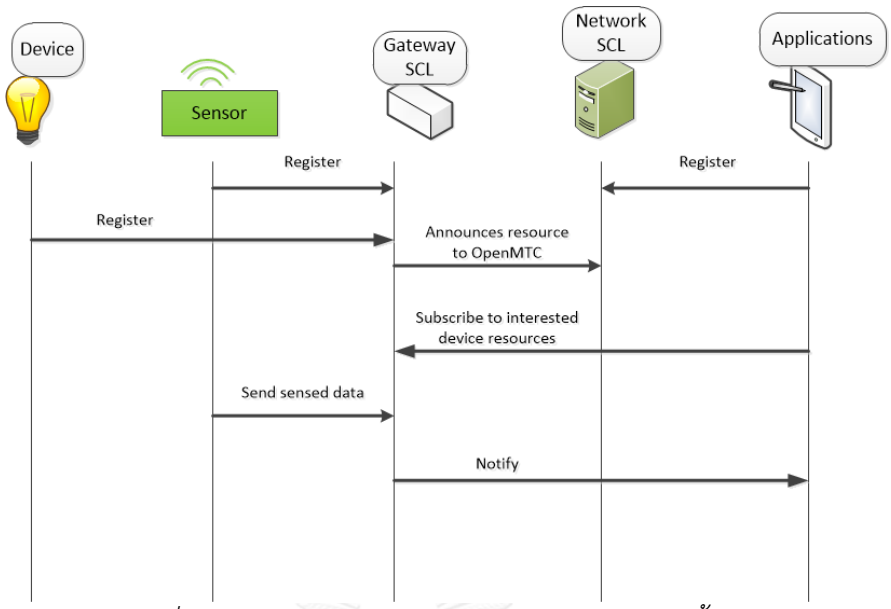
และยังมีกระบวนการเพิ่มเติมดังนี้

- NOTIFY แจ้งเกี่ยวกับเนื้อหาใหม่หรือมีการเปลี่ยนแปลงเกิดขึ้นให้ทรัพยากรทราบตามลำดับการลงทะเบียน
- EXECUTE กระทำการจัดการตามภารกิจหรือคำสั่งที่ถูกกำหนดโดยทรัพยากร



รูปที่ 13 กระบวนการตามโครงสร้างแบบ RESTful

แสดงแบบแผนพื้นฐานของการติดต่อสื่อสารในเอ็มทูเอ็มผ่านโอเพ็นเอ็มทีซี ซึ่งโปรแกรมประยุกต์สามารถค้นหาและลงทะเบียนไว้ที่ชั้นความสามารถในการบริการโครงข่าย เพื่อเป็นทรัพยากรที่ถูกบันทึกเรียงตามลำดับในระบบตามเกณฑ์ที่เอ็มทูเอ็มกำหนดไว้ เช่น การนำไปใช้ประโยชน์ได้ (functionality) ช่วงเวลาในการลงทะเบียน เป็นต้น เมื่อมีเหตุการณ์หรือ ข้อมูลใหม่เกิดขึ้น ข้อมูลจะถูกส่งมาที่เกตเวย์และนำไปสู่การประกาศไปที่โปรแกรมประยุกต์ที่มาลงทะเบียนไว้เรียงตามลำดับแสดงดังรูปที่ 14



รูปที่ 14 ตัวอย่างการไหลของข้อความของแบบแผนพื้นฐาน

(Message flow of a simple scenario) [8]

จากรูปที่ 14 แสดงขั้นตอนตามลำดับการไหลของข้อความ สามารถอธิบายได้ดังนี้

ขั้นตอนที่ 1 โปรแกรมประยุกต์และอุปกรณ์ต่างๆ ทำการลงทะเบียนที่ชั้นความสามารถในการบริการที่แต่ละอุปกรณ์หรือโปรแกรมประยุกต์เชื่อมต่ออยู่

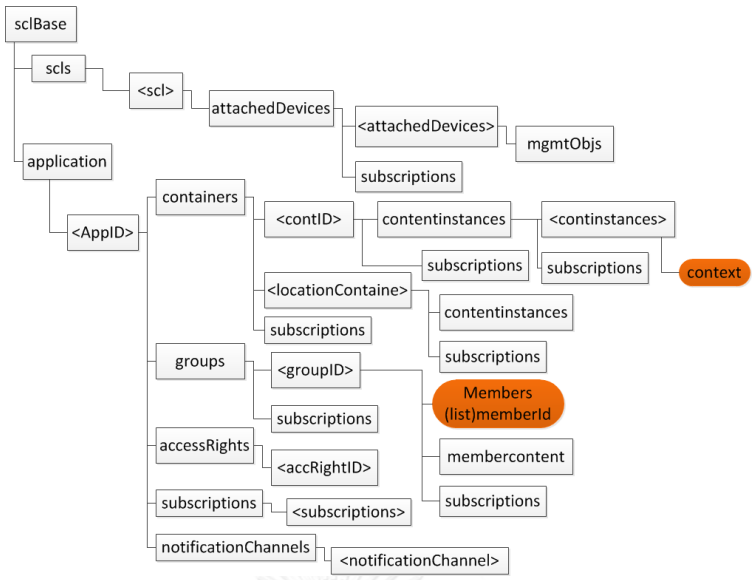
ขั้นตอนที่ 2 ชั้นความสามารถในการบริการเกตเวย์ทำการประกาศโปรแกรมประยุกต์ที่เชื่อมต่ออยู่ เช่น โปรแกรมประยุกต์เกตเวย์ โปรแกรมประยุกต์อุปกรณ์ เป็นต้น ไปที่ชั้นความสามารถในการบริการโครงข่าย

ขั้นตอนที่ 3 โปรแกรมประยุกต์โครงข่าย ต้องการรับข้อมูลจากตัวรับรู้ โดยทำการส่ง RETRIEVE พร้อมกับระเบียบวิธี subscribe มาที่ชั้นความสามารถในการบริการโครงข่าย และชั้นความสามารถในการบริการโครงข่ายทำการส่งซ้ำไปที่ชั้นความสามารถในการบริการเกตเวย์

ขั้นตอนที่ 4 ตัวรับรู้ตรวจจับสภาพแวดล้อม และส่งข้อมูลด้วยกระบวนการ CREATE มาที่ชั้นความสามารถในการบริการเกตเวย์

ขั้นตอนที่ 5 ชั้นความสามารถในการบริการเกตเวย์ทำการประกาศข้อมูลไปยังชั้นความสามารถในการบริการโครงข่าย และชั้นความสามารถในการบริการโครงข่ายใช้กระบวนการ NOTIFY ส่งข้อมูลไปยังโปรแกรมประยุกต์โครงข่าย

จากการดำเนินการตามการไหลของข้อมูล จะมีการสร้างส่วนประกอบต่างๆขึ้นในโครงสร้างของทรัพยากรเพื่อรองรับข้อมูลที่ไหลไปตามลำดับ จึงทำให้เกิดเป็นโครงสร้างที่มีลำดับขั้น แสดงดังรูปที่ 15



รูปที่ 15 ตัวอย่างโครงสร้างของทรัพยากร M2M sclBase (M2M sclBase resource tree) [9]

2.3 การเปรียบเทียบเชิงวิเคราะห์ระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M

ในการที่มาตรฐาน IEEE 1888 และมาตรฐาน ETSI M2M จะสามารถทำงานร่วมกันได้นั้น เราจำเป็นต้องทราบรายละเอียดต่างๆของทั้งสองมาตรฐาน เช่น ส่วนประกอบ วิธีการดำเนินการ โพรโทคอลที่ใช้ในการสื่อสาร ภาษาที่ใช้ในการสื่อสาร เป็นต้น โดยเฉพาะอย่างยิ่งการศึกษาหาแนวทางในการเชื่อมต่อการทำงานของทั้งสองมาตรฐานเข้าด้วยกันนั้น จึงมีการเปรียบเทียบส่วนประกอบและการทำงานต่างๆ เพื่อศึกษาส่วนเชื่อมต่อและวิธีการเชื่อมต่อระหว่างสองมาตรฐานแล้วจึงดำเนินการออกแบบตัวกลางในการเชื่อมต่อระหว่างสองมาตรฐาน จากข้อมูลของมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2N สามารถเปรียบเทียบได้ตามหัวข้อดังต่อไปนี้

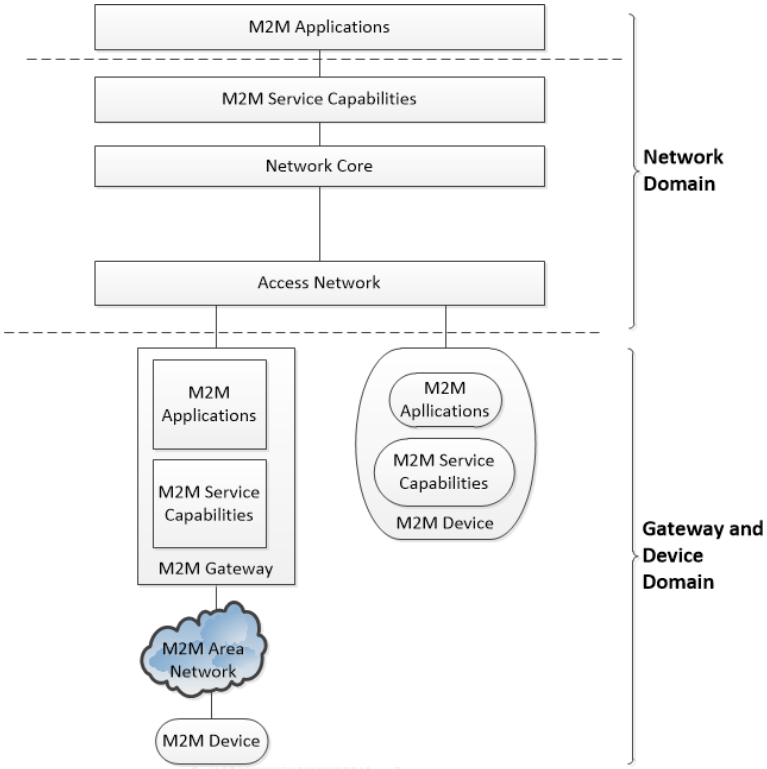
2.3.1 การเปรียบเทียบสถาปัตยกรรม

การเปรียบเทียบสถาปัตยกรรมของทั้งสองมาตรฐานดังแสดงในรูปที่ 16 และรูปที่ 17 ให้เห็นถึงรายละเอียดดังนี้

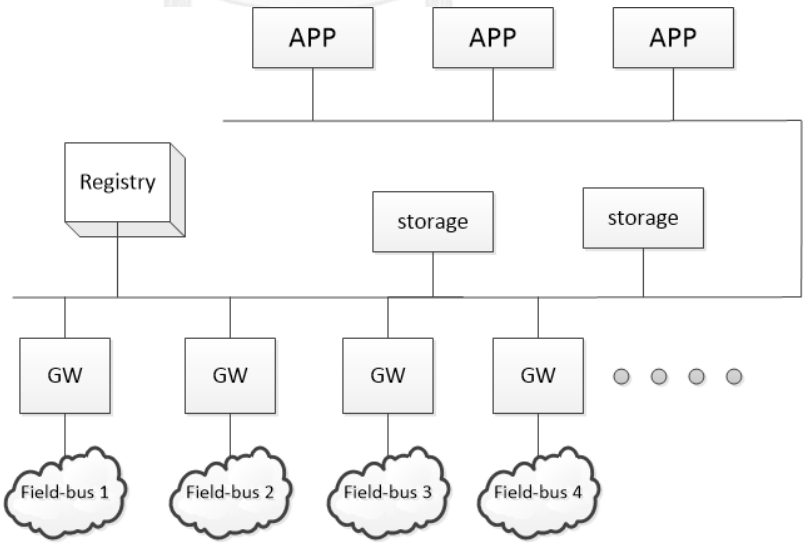
- มาตรฐาน ETSI M2M ประกอบด้วย โดเมนเกตเวย์หรือโดเมนอุปกรณ์ และโดเมนโครงข่าย
- มาตรฐาน IEEE1888 ประกอบด้วย เกตเวย์ หน่วยเก็บข้อมูล รีจิสทรี และโปรแกรมประยุกต์

จากส่วนประกอบดังกล่าว แสดงให้เห็นว่าสถาปัตยกรรมและส่วนประกอบของทั้งสองมาตรฐานมีความคล้ายกันมาก นั่นคือทั้งสองมาตรฐานมีส่วนประกอบที่เป็นส่วนรับข้อมูลจากอุปกรณ์ต่างๆ นั่นคือ เกตเวย์ มีส่วนประกอบที่ทำหน้าที่เป็นส่วนเก็บข้อมูลนั่นคือ หน่วยเก็บข้อมูลและ

RAR (อยู่ในโดเมน) และมีส่วนประกอบที่ทำหน้าที่เป็นส่วนแสดงผลนั่นคือ โปรแกรมประยุกต์ ดังนั้นจึงง่ายต่อการจัดการเพื่อทำให้ทั้งสองมาตรฐานสามารถทำงานร่วมกัน



รูปที่ 16 สถาปัตยกรรมของมาตรฐาน ETSI M2M



รูปที่ 17 สถาปัตยกรรมของมาตรฐาน IEEE1888

2.3.2 การเปรียบเทียบรูปแบบข้อมูลที่ใช้ในการสื่อสาร

การเปรียบเทียบรูปแบบข้อมูลที่ใช้ในการสื่อสารของทั้งสองโพรโทคอลแสดงดังรูปที่ 18 และรูปที่ 19 โดยที่ในรายละเอียดของทั้งสองมาตรฐาน ได้แสดงให้เห็นดังนี้

- โพรโทคอลตามมาตรฐาน IEEE1888 มีรูปแบบของข้อมูลที่สามารถรองรับได้ มีรูปแบบเดียวคือ XML เนื่องจากโพรโทคอล SOAP ใช้เพียงรูปแบบ XML
- โพรโทคอลตามมาตรฐาน ETSI M2M มีรูปแบบของข้อมูลที่สามารถรองรับได้คือ XML, JSON, EXI, Fast Infoset (ตามมาตรฐานกำหนดไว้ว่าแพลตฟอร์มที่ดำเนินการตามมาตรฐานนี้ต้องรองรับได้อย่างน้อย 1 รูปแบบจาก 4 รูปแบบนี้)

จากรูปแบบดังกล่าว แสดงให้เห็นว่าทั้งสองโพรโทคอลมีรูปแบบของข้อมูลที่เหมือนกันคือ XML ดังนั้น การทำให้ทั้งสองมาตรฐานสามารถทำงานร่วมกันได้ควรจะใช้ข้อมูลในรูปแบบ XML หรือรูปแบบข้อมูลที่สามารถแปลงมาเป็นรูปแบบ XML ได้

```
{
  "contentInstance": {
    "content": {
        "$t":
"eyJkYXRhIjp7InRpbWVzdGFtcCI6MjA0NDUsImNvbnN1bWVkIjowLjg0Nn19",
        "contentType": "application/json"
    },
    "id": "contentInstance2478482",
    "creationTime": "2014-06-03T07:40:19.772Z",
    "lastModifiedTime": "2014-06-03T07:40:19.772Z",
    "contentTypes": {
      "contentType": [
        "application/json"
      ]
    },
    "contentSize": 45,
    "href":
"/m2m/applications/GameMeter/containers/1005/contentInstances/c
ontentInstance2478482"
  }
}
```

รูปที่ 18 ตัวอย่างรูปแบบข้อมูลที่ใช้ในการสื่อสารตามมาตรฐาน ETSI M2M

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <ns2:queryRQ xmlns:ns2="http://soap.fiap.org/">
   <transport xmlns="http://gutp.jp/fiap/2009/11/">
    <header>
     <query id="9eed9de4-1c48-4b08-a41d-dac067fc1c0d"
type="storage">
      <key
id="http://bems.ee.eng.chula.ac.th/ee/fl2/east/lectureroom_211/
z1/aircon1/control/max_temp"
       attrName="time" select="maximum" />
     </query>
    </header>
   </transport>?
  </ns2:queryRQ>
 </soapenv:Body>
</soapenv:Envelope>
```

รูปที่ 19 ตัวอย่างรูปแบบข้อมูลที่ใช้ในการสื่อสารตามาตรฐาน IEEE1888

2.3.3 การเปรียบเทียบโพรโทคอลการสื่อสาร

- มาตรฐาน IEEE1888 ใช้ โพรโทคอล SOAP ในการจัดรูปแบบชุดข้อมูล และใช้โพรโทคอล HTTP ในการขนส่งชุดข้อมูลไปยังส่วนประกอบต่างๆ
- มาตรฐาน ETSI M2M ใช้ RESTful APIs ที่จับคู่กับ HTTP นั่นคือ GET, POST, PUT, DELETE, NOTIFY, EXECUTE ในการจัดการข้อมูลและส่วนประกอบต่างๆในหน่วยเก็บข้อมูล และใช้ COAP [10] ซึ่งเป็นโพรโทคอลที่ดำเนินการเกี่ยวกับ request และ response มาเป็นระเบียบวิธีการในการสื่อสารด้วย

จากรายละเอียดของโพรโทคอลดังกล่าว แสดงให้เห็นว่าทั้งสองมาตรฐานใช้โพรโทคอลในการสื่อสารที่แตกต่างกัน ดังนั้นจึงต้องมีตัวกลางในการแปลรูปแบบข้อมูลและคำสั่งให้เป็นไปตามที่โพรโทคอลนั้นต้องการ

2.3.4 การเปรียบเทียบการดำเนินการ (Operations)

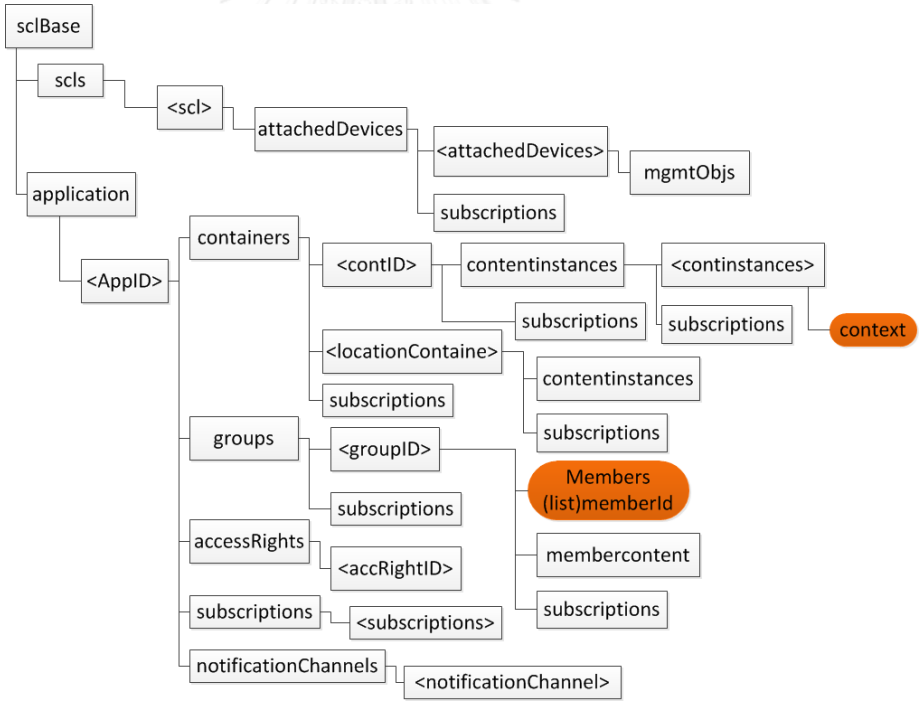
- โพรโทคอลตามมาตรฐาน IEEE1888 ได้ถูกออกแบบรูปแบบการดำเนินการในการส่งและร้องขอข้อมูลระหว่างส่วนประกอบดังนี้
    - FETCH เป็นโพรโทคอลในการดึงข้อมูลจากหน่วยเก็บข้อมูล
    - WRITE เป็นโพรโทคอลในการส่งข้อมูลไปยังหน่วยเก็บข้อมูล
    - TRAP เป็นโพรโทคอลในการร้องขอข้อมูลตามลักษณธข้อมูลที่กำหนดไว้
    - REGISTRATION เป็นโพรโทคอลที่ส่วนประกอบต่างๆ ใช้ในการลงทะเบียน
    - LOOKUP เป็นโพรโทคอลที่ส่วนประกอบใช้ในการค้นหาตำแหน่งข้อมูล
- โพรโทคอลตามมาตรฐาน ETSI M2M มีการดำเนินการตามทฤษฎีของ RESTful (CRUD) ซึ่งทำงานร่วมกับตัวดำเนินการตามโพรโทคอล HTTP ดังนี้
    - CREATE เป็นโพรโทคอลที่ทำงานร่วมกับตัวดำเนินการ POST ของ HTTP ในการส่งข้อมูลไปยังหน่วยเก็บข้อมูล
    - RETRIEVE เป็นโพรโทคอลที่ทำงานร่วมกับตัวดำเนินการ GET ของ HTTP ในการดึงข้อมูลจากหน่วยเก็บข้อมูล
    - UPDATE เป็นโพรโทคอลที่ทำงานร่วมกับตัวดำเนินการ PUT ของ HTTP ในการส่งข้อมูลใหม่ไปบันทึกแทนที่ข้อมูลเก่าในหน่วยเก็บข้อมูล

- DELETE เป็นโพรโทคอลที่ทำงานร่วมกับตัวดำเนินการ DELETE ของ HTTP ในการลบข้อมูลจากหน่วยเก็บข้อมูล
- พร้อมกับมีตัวดำเนินการเสริมคือ NOTIFY และ EXECUTE ในการคงการเชื่อมต่อระหว่างส่วนประกอบ

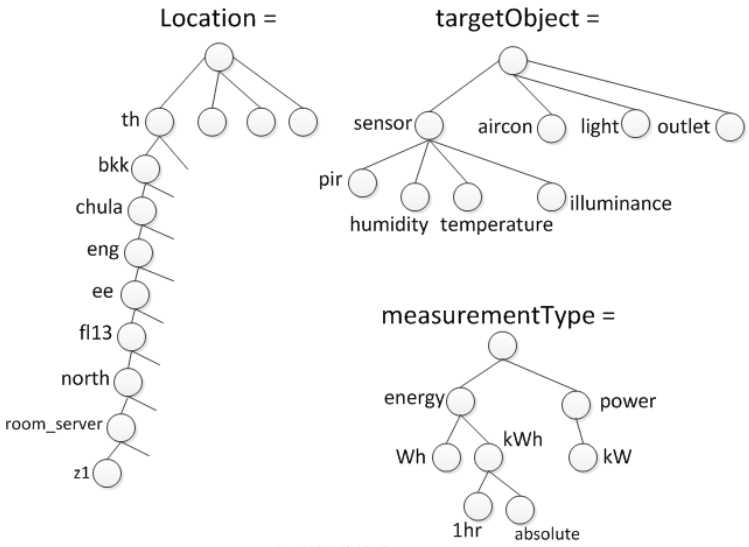
จากการดำเนินการของทั้งสองโพรโทคอล แสดงให้เห็นว่าทั้งสองมาตรฐานมีการดำเนินการที่แตกต่างกัน ดังนั้นหากมีตัวกลางที่สามารถเข้าใจคำสั่งของการดำเนินการของทั้งสองโพรโทคอลนี้จะทำให้ทั้งสองมาตรฐานนี้สามารถทำงานร่วมกัน และแลกเปลี่ยนข้อมูลกันได้

2.3.5 การเปรียบเทียบโครงสร้างการจัดการตำแหน่งข้อมูล

การเปรียบเทียบโครงสร้างการจัดการตำแหน่งข้อมูล (Point management structure) ของทั้งสองมาตรฐานดังแสดงในรูปที่ 20 และรูปที่ 21 ซึ่งจะเห็นว่าทั้งสองโครงสร้างมีความแตกต่างกันมาก จึงจำเป็นต้องมีการออกแบบวิธีการแปลงโครงสร้างการจัดการตำแหน่งข้อมูลให้สอดคล้องกับโครงสร้างของทั้งสองมาตรฐาน



รูปที่ 20 ตัวอย่างโครงสร้างการจัดการส่วนประกอบตามมาตรฐาน ETSI M2M



รูปที่ 21 ตัวอย่างโครงสร้างการจัดการส่วนประกอบตามมาตรฐาน IEEE1888

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 3
โพรโทคอลในการแลกเปลี่ยนข้อมูลที่นำเสนอ

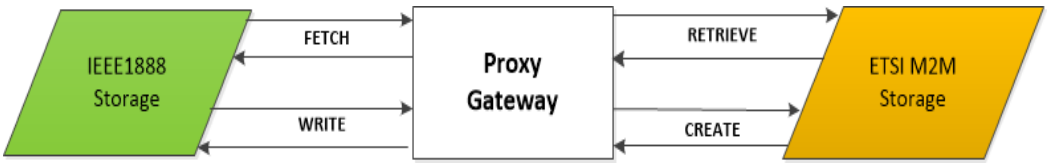
3.1 การพัฒนาโพรโทคอลของเกตเวย์

จากวัตถุประสงค์ของวิทยานิพนธ์ฉบับนี้สำหรับการพัฒนาโพรโทคอลของเกตเวย์เพื่อใช้สำหรับการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M ในการแลกเปลี่ยนข้อมูลที่ถูกบันทึกอยู่ในหน่วยเก็บข้อมูลของทั้งสองมาตรฐาน จึงทำการเปรียบเทียบเพื่อหาวิธีการเชื่อมต่อกันระหว่างสองมาตรฐานนี้ จากผลการเปรียบเทียบแสดงให้เห็นว่าทั้งสองมาตรฐานมีรูปแบบการดำเนินการ และโครงสร้างของส่วนประกอบที่แตกต่างกัน แต่มีส่วนประกอบของสถาปัตยกรรมที่คล้ายคลึงกัน และรูปแบบข้อมูลที่สามารถรองรับได้เหมือนกันคือ XML ดังนั้นการพัฒนาโพรโทคอลของเกตเวย์เพื่อการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M ในการส่งต่อข้อมูล และดึงข้อมูลระหว่างหน่วยเก็บข้อมูลของทั้งสองมาตรฐานได้นั้น จึงต้องมีตัวกลางในการดำเนินการซึ่งในวิทยานิพนธ์ฉบับนี้เรียกว่า เกตเวย์จำลอง (Proxy gateway) และจากวัตถุประสงค์ของวิทยานิพนธ์ฉบับนี้ เพื่อให้เกิดการแลกเปลี่ยนข้อมูลระหว่างหน่วยเก็บข้อมูลของทั้งสองมาตรฐานด้วยการดำเนินการในการดึงข้อมูลจากหน่วยเก็บข้อมูลและส่งข้อมูลไปยังหน่วยเก็บข้อมูล เกตเวย์จำลองนี้จึงควรสามารถดำเนินการตามทั้งสองมาตรฐาน คือ FETCH, WRITE, RETRIEVE, CREATE ได้

คุณสมบัติของเกตเวย์จำลองคือ

- มีรูปแบบข้อมูลในการสื่อสารใช้ XML หรือรูปแบบข้อมูลที่สามารถแปลงให้เป็นรูปแบบ XML ได้
- สามารถแปลงโครงสร้างการจัดการส่วนประกอบจากรูปแบบที่ใช้ในโพรโทคอลหนึ่ง ให้เป็นไปตามโครงสร้างการจัดการส่วนประกอบตามรูปแบบที่ใช้ในอีกโพรโทคอลหนึ่งได้
- สามารถใช้การดำเนินการ FETCH, WRITE, RETRIEVE, CREATE ในการรับข้อมูลจากหน่วยเก็บข้อมูลของทั้งสองมาตรฐาน และส่งต่อข้อมูลไปยังหน่วยเก็บข้อมูลทั้งสองมาตรฐานได้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 22 รูปแบบการทำงานร่วมกันระหว่างมาตรฐานIEEE1888 กับมาตรฐาน ETSI M2M

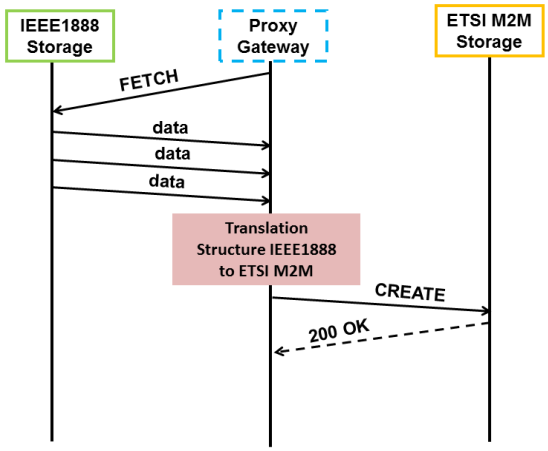
จากรูปที่ 22 แสดงรูปแบบการทำงานของเกตเวย์จำลองที่นำเสนอ ซึ่งมีความสามารถในการดึงข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 ด้วยโพรโทคอล FETCH และสามารถส่งข้อมูลไปเก็บในหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 ด้วยโพรโทคอล WRITE พร้อมกับมีความสามารถในการดึงข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M โดยใช้ระเบียบวิธี RETRIEVE และสามารถส่งข้อมูลไปเก็บในหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M โดยใช้ระเบียบวิธี CREATE ได้

3.2 กระบวนการทำงาน

จากรูปแบบการทำงานของเกตเวย์จำลองข้างต้น จึงทำการออกแบบกระบวนการทำงานในการแลกเปลี่ยนข้อมูลระหว่างหน่วยเก็บข้อมูลของทั้งสองมาตรฐาน โดยแบ่งออกเป็น 2 กระบวนการทำงาน [11] ดังนี้

3.2.1 กระบวนการ 1: ร้องขอข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888

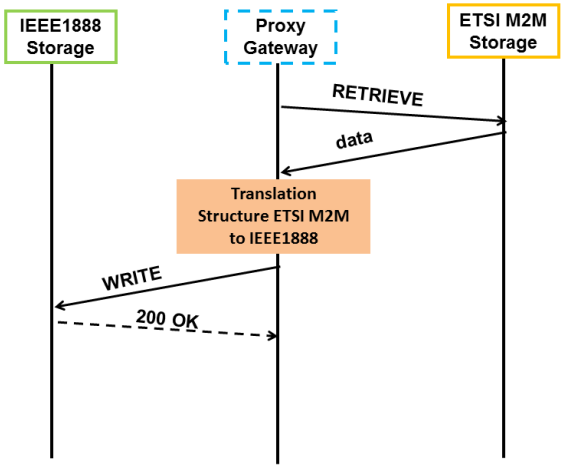
ในกระบวนการนี้ เกตเวย์จำลอง ใช้โพรโทคอล FETCH ในการดึงข้อมูลจาก หน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 และทำการแปลงรูปแบบข้อมูลในการสื่อสารและการระบุตำแหน่งข้อมูลที่ได้รับจากมาตรฐาน IEEE1888 ให้สอดคล้องกับรูปแบบข้อมูลและการระบุตำแหน่งข้อมูลตามมาตรฐาน ETSI M2M จากนั้น เกตเวย์จำลองใช้การดำเนินการ CREATE ในการส่งต่อข้อมูลไปยังหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M หากข้อมูลถูกจัดเก็บในหน่วยเก็บข้อมูลสำเร็จ หน่วยเก็บข้อมูลจะส่งข้อมูลตอบกลับ 200 OK มายังเกตเวย์จำลอง ดังแสดงในรูปที่ 23



รูปที่ 23 แผนภาพกระบวนการทำงานของเกตเวย์จำลองในกระบวนการ 1 [11]

3.2.2 กระบวนการ 2: ร้องขอข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M

ในกระบวนการนี้ เกตเวย์จำลอง ใช้การดำเนินการ RETRIEVE ในการดึงข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M และทำการแปลงรูปแบบข้อมูลในการสื่อสารและการระบุตำแหน่งข้อมูลที่ได้รับจากมาตรฐาน ETSI M2M ให้สอดคล้องกับรูปแบบข้อมูลและการระบุตำแหน่งข้อมูลตามมาตรฐาน IEEE1888 จากนั้น เกตเวย์จำลองใช้โพรโทคอล WRITE ในการส่งต่อข้อมูลไปยังหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 หากข้อมูลถูกจัดเก็บในหน่วยเก็บข้อมูลสำเร็จ หน่วยเก็บข้อมูลจะส่งข้อมูลตอบกลับ 200 OK มายังเกตเวย์จำลองดังแสดงในรูปที่ 24

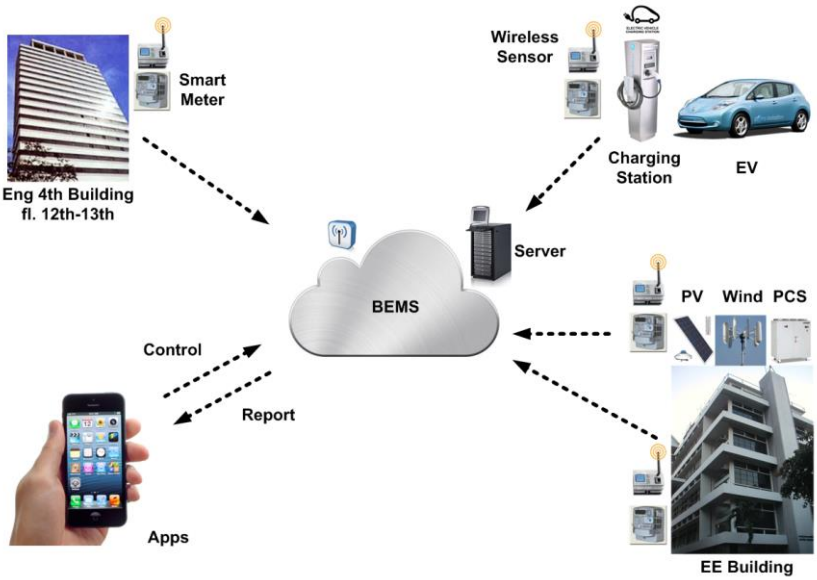


รูปที่ 24 แผนภาพกระบวนการทำงานของเกตเวย์จำลองในกระบวนการ 2 [11]

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.3 สภาพแวดล้อมของระบบ

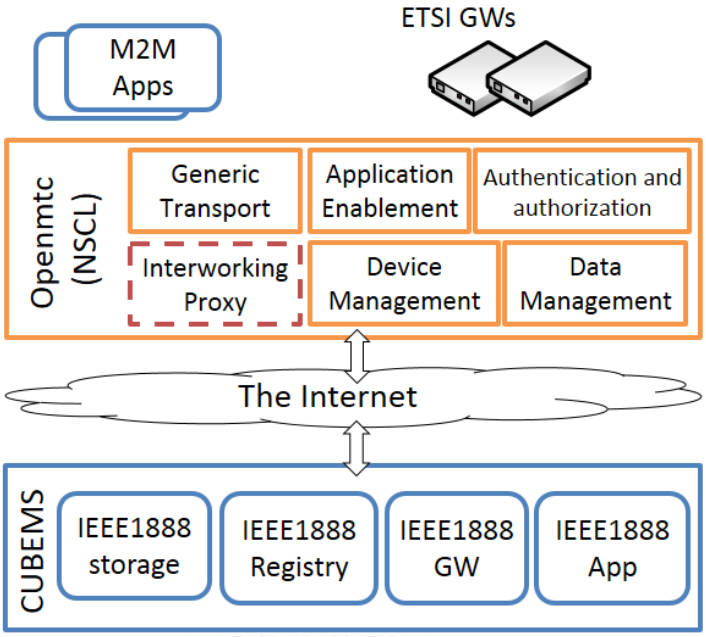
จากกระบวนการทำงานของเกตเวย์จำลองในการแลกเปลี่ยนข้อมูลระหว่างหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 กับหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M ที่กล่าวไว้ในบทที่แล้ว ในการทดสอบการทำงานของกระบวนการทำงานของเกตเวย์จำลองนี้ เราควรนำโพรโทคอลตามมาตรฐานทั้งสองมาตรฐานที่มีการใช้งานจริงในปัจจุบันมาใช้ในการออกแบบการทดสอบ ซึ่งในส่วนของมาตรฐาน IEEE1888 ทางภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาฯ ได้มีการดำเนินโครงการ CUBEMS โดยมีจุดประสงค์เพื่อปรับพฤติกรรมการใช้พลังงานของนิสิตและบุคลากรในภาควิชา นำไปสู่การใช้พลังงานไฟฟ้าที่ลดลง โดยมีการติดตั้งเซนเซอร์ จำนวน 200 จุดโดยประมาณ เพื่อตรวจวัดสภาพแวดล้อมและพฤติกรรมการใช้งาน ติดตั้งมิเตอร์จำนวน 20 จุด เพื่อวัดการใช้พลังงานของทั้งภาควิชา รวมไปถึงติดตั้งจอแสดงผลจำนวน 4 จุด เพื่อแสดงปริมาณการใช้พลังงาน ณ ปัจจุบัน และเพื่อให้นิสิตและบุคลากรตระหนักถึงการช่วยกันประหยัดพลังงานภายในภาควิชา ซึ่งแต่ละส่วนประกอบของโครงการนี้สื่อสารกันโดยใช้โพรโทคอลตามมาตรฐาน IEEE1888 [12]



รูปที่ 25 ภาพรวมส่วนประกอบของโครงการ CU-BEMS

ในส่วนของมาตรฐาน ETSI M2M ทางภาควิชาวิศวกรรมไฟฟ้า ได้รับความร่วมมือและการสนับสนุนจากโครงการ UNIFI (Universities for future internet) ซึ่งเป็นโครงการโดยความร่วมมือกันระหว่าง 5 มหาวิทยาลัยใน 5 ประเทศ ได้แก่ มหาวิทยาลัยเทคโนโลยีเบอร์ลิน ประเทศเยอรมัน มหาวิทยาลัย เคปทาวน์ ประเทศแอฟริกาใต้ มหาวิทยาลัยชิลี ประเทศชิลี มหาวิทยาลัยวิทยาศาสตร์และเทคโนโลยีฮานอย ประเทศเวียดนาม และจุฬาลงกรณ์มหาวิทยาลัย ประเทศไทย เพื่อร่วมกันพัฒนาเทคโนโลยีอินเตอร์เนตในอนาคต (Future internet) และโครงข่ายโทรคมนาคมยุคหน้า (Next generation network) โดยโครงการนี้ได้ให้ความรู้และให้ความช่วยเหลือในการติดตั้งแพลตฟอร์ม OpenMTC ซึ่งดำเนินการด้วยโพรโทคอลตามมาตรฐาน ETSI M2M พร้อมกับมอบโปรแกรมและแนะนำวิธีการปรับปรุงโปรแกรมเพื่อให้สามารถใช้งานกับส่วนอื่นๆได้

โดยการทำให้ทั้งสองโพรโทคอลที่มีการใช้งานอยู่จริงนี้ให้สามารถแลกเปลี่ยนข้อมูลได้ตามความสามารถของเกตเวย์จำลองนั้นจำเป็นต้องออกแบบรูปแบบการเชื่อมต่อระหว่างสองโพรโทคอล และจากการศึกษาการดำเนินการของทั้งสองโพรโทคอลทำให้ทราบว่า สองโพรโทคอลนี้สามารถเชื่อมต่อและแลกเปลี่ยนข้อมูลกันผ่านอินเทอร์เนตได้ จึงทำการออกแบบการทำงานร่วมกันระหว่างสองโพรโทคอลในการแลกเปลี่ยนข้อมูลดังแสดงในรูปที่ 26



รูปที่ 26 สถาปัตยกรรมของโครงร่างเอ็มทูเอ็มที่ถูกปรับให้เกิดการเชื่อมต่อ (Architecture of the Adaptable M2M Framework (AM2MF)) [11]

จากการศึกษาความสามารถในการทำงานร่วมกับโพรโทคอลอื่น ๆ ของ OpenMTC และ CUBEMS ทำให้ทราบว่า OpenMTC ดำเนินการต่างๆตามแบบ Node.js [13] ด้วยภาษา Javascript ผ่าน RESTful API ซึ่งเป็นส่วนต่อประสานแบบเปิด (open interface) และมี interworking proxy เป็นส่วนประกอบที่ทำหน้าที่เชื่อมต่อกับโพรโทคอลอื่นๆผ่านโพรโทคอลทีซีพี/ไอพี และในส่วนของ CUBEMS ซึ่งถูกออกแบบตามมาตรฐาน IEEE1888 สามารถเชื่อมต่อกับโพรโทคอลอื่นได้ โดยโพรโทคอลที่จะมาเชื่อมต่อนั้นต้องสามารถแปลงรูปแบบข้อมูลให้สอดคล้องตามมาตรฐาน IEEE1888 นั่นคือ XML ได้ ทั้งนี้ยังพบว่าทั้ง OpenMTC และ CUBEMS ใช้ทีซีพี/ไอพีเป็นโพรโทคอลในการสื่อสารและส่งข้อมูลกันระหว่างแต่ละส่วนประกอบ

ดังนั้น รูปแบบโพรโทคอลของเกตเวย์จำลองนี้ จึงถูกออกแบบการดำเนินการโดยใช้โปรแกรมของ OpenMTC ซึ่งดำเนินการตามรูปแบบ Node.js ที่ใช้ในการดึงข้อมูลจากหน่วยเก็บข้อมูลและส่งข้อมูลไปยังหน่วยเก็บข้อมูลของ OpenMTC อยู่แล้ว มาปรับปรุงและดัดแปลงให้สามารถแปลงรูปแบบของข้อมูลให้เป็นรูปแบบ XML และตัวระบุตำแหน่งข้อมูลที่สอดคล้องตามรูปแบบของ CUBEMS ได้ จึงทำให้เกตเวย์จำลองสามารถดึงข้อมูลจากหน่วยเก็บข้อมูล และส่งข้อมูลไปยังหน่วยเก็บข้อมูลของ CUBEMS ได้

3.3 สรุป

ในบทนี้ได้นำเสนอโพรโทคอลของเกตเวย์ที่ทำให้เกิดการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M ในการแลกเปลี่ยนข้อมูลระหว่างหน่วยเก็บข้อมูลของทั้งสองมาตรฐาน โดยมีการนำข้อมูลการเปรียบเทียบที่แสดงให้เห็นถึงความคล้ายและความแตกต่างของทั้งสองมาตรฐานมาใช้การพิจารณาออกแบบการดำเนินการของตัวกลางในการแลกเปลี่ยนข้อมูลระหว่างสองมาตรฐานซึ่งในวิทยานิพนธ์ฉบับนี้คือ เกตเวย์จำลอง โดยมีการนำโพรโทคอลในการสื่อสารของทั้งสองมาตรฐานมาทำงานเชื่อมต่อกันในการส่งต่อข้อมูลจากหน่วยเก็บข้อมูลของมาตรฐานหนึ่งไปยังอีกมาตรฐานหนึ่ง นั่นคือการดำเนินการ CREATE, RETRIEVE, FETCH และ WRITE พร้อมกับมีการแปลงรูปแบบข้อมูลให้สอดคล้องกับรูปแบบ XML และ JSON รวมไปถึงสามารถแปลงโครงสร้างการจัดการส่วนประกอบและการระบุตำแหน่งข้อมูลให้สอดคล้องกับทั้งสองมาตรฐานด้วย และในบทนี้ได้กล่าวถึงสภาพแวดล้อมของระบบที่ใช้ในการทดสอบจริงซึ่งเป็นแพลตฟอร์มที่ถูกพัฒนาขึ้นตามมาตรฐานทั้งสองคือ โครงการ CUBEMS ดำเนินการตามมาตรฐาน IEEE 1888 และแพลตฟอร์ม OpenMTC ดำเนินการตามมาตรฐาน ETSI M2M

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 4
การทดสอบและวิเคราะห์ผลการทดสอบการดำเนินการของโพรโทคอล

4.1 ส่วนประกอบของโพรโทคอลที่ใช้ทำการทดสอบ

ส่วนประกอบของโพรโทคอลที่ใช้ในการทดสอบเพื่อเปรียบเทียบความแตกต่างและทดสอบสมรรถนะของโพรโทคอลในรูปแบบต่างๆ มีดังนี้

4.1.1 เกตเวย์จำลอง

เกตเวย์จำลองถูกติดตั้งบนคอมพิวเตอร์โน้ตบุ๊ก ยี่ห้อ DELL รุ่น Latitude E6400 Processor intel Core 2 Duo 2.80 GHz x 2 บนระบบปฏิบัติการ Ubuntu 12.04 LTS 64 bits ตามรูปที่ 27 โดยทำการติดตั้ง Node.js เพื่อเป็นตัวปฏิบัติการในการดำเนินงานโปรแกรมที่เขียนด้วยภาษา JavaScript ให้สามารถดำเนินการแลกเปลี่ยนข้อมูลระหว่างหน่วยเก็บข้อมูลของทั้งสองมาตรฐานได้ กำหนดเลขที่อยู่ไอพี (IP address) เป็น 161.200.90.66

รูปที่ 27 คอมพิวเตอร์โน้ตบุ๊กที่ใช้เป็นเกตเวย์จำลอง

4.1.2 หน่วยเก็บข้อมูลของ CUBEMS

หน่วยเก็บข้อมูลของ CUBEMS ถูกติดตั้งอยู่บนเครื่อง server IBM intel inside Xenon 2.4 GHz RAM 4 GB 1 core เป็นส่วนประกอบของโครงการ CUEMS ในการเก็บข้อมูลตัวรับรู้ทั้งหมดของโครงการ ดังรูปที่ 28 กำหนดเลขที่อยู่ไอพีที่ใช้สำหรับเข้าสู่หน่วยเก็บข้อมูลเป็น 161.200.90.122

รูปที่ 28 เครื่อง server ที่ใช้เป็นหน่วยเก็บข้อมูลของ CUBEMS

4.1.3 หน่วยเก็บข้อมูลของ OpenMTC

หน่วยเก็บข้อมูลของ OpenMTC ถูกติดตั้งอยู่บนเครื่องคอมพิวเตอร์ตั้งโต๊ะ ยี่ห้อ ASUS Processor Intel Core 2 Duo 2.66 GHz x 2 บนระบบปฏิบัติการ Ubuntu 12.04 LTS 64 bits ตามรูปที่ 29 ที่ทำการติดตั้งแพลตฟอร์มของ OpenMTC ไว้ และกำหนดเลขที่อยู่ไอพีเป็น 161.200.90.78

รูปที่ 29 คอมพิวเตอร์ตั้งโต๊ะที่ใช้เป็นหน่วยเก็บข้อมูลของ OpenMTC

4.1.4 ข้อมูลจากเซนเซอร์ 10 จุด

ข้อมูลที่นำมาใช้ในการทดสอบเป็นข้อมูลจากเซนเซอร์ของโครงการ CUBEMS ที่ติดตั้งที่ห้องแลปไฟฟ้าสื่อสาร ชั้น 13 ตึก 4 คณะวิศวกรรมศาสตร์ จุฬาฯ ดังรูปที่ 30 และมีรายชื่อระบุตำแหน่งของข้อมูลดังนี้

1.1 http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z1/sensor1/monitor/temperature

1.2 http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z2/sensor1/monitor/temperature

1.3 http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl /z3/sensor1/monitor/temperature

1.4 http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl /z4/sensor1/monitor/temperature

1.5 http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl /z5/sensor1/monitor/temperature

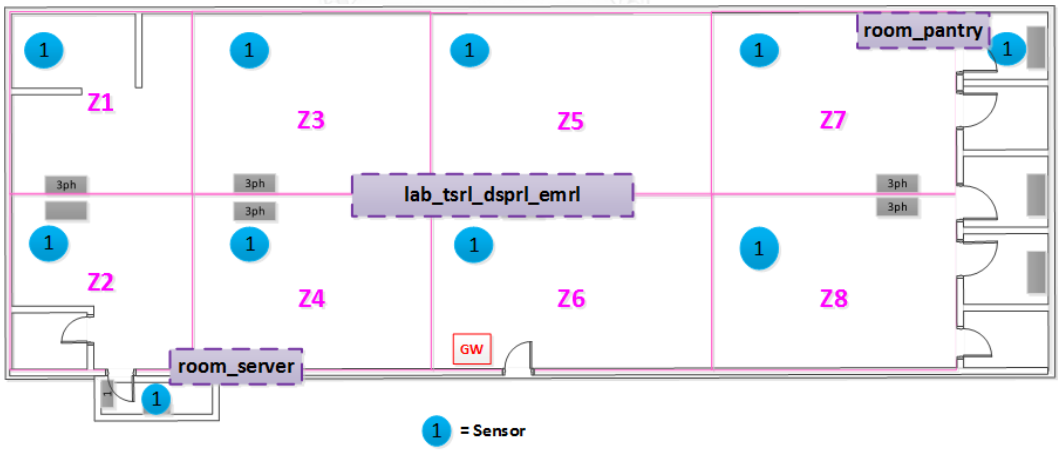
1.6 http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl /z6/sensor1/monitor/temperature

1.7 http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl /z7/sensor1/monitor/temperature

1.8 http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl /z8/sensor1/monitor/temperature

1.9 http://bems.ee.eng.chula.ac.th/eng4/fl13/north/room_pantry/z1 /sensor1/monitor/temperature

1.10 http://bems.ee.eng.chula.ac.th/eng4/fl13/north/room_server /z1/sensor1/monitor/temperature



รูปที่ 30 แสดงตำแหน่งเซนเซอร์ที่นำข้อมูลมาใช้ทดสอบ

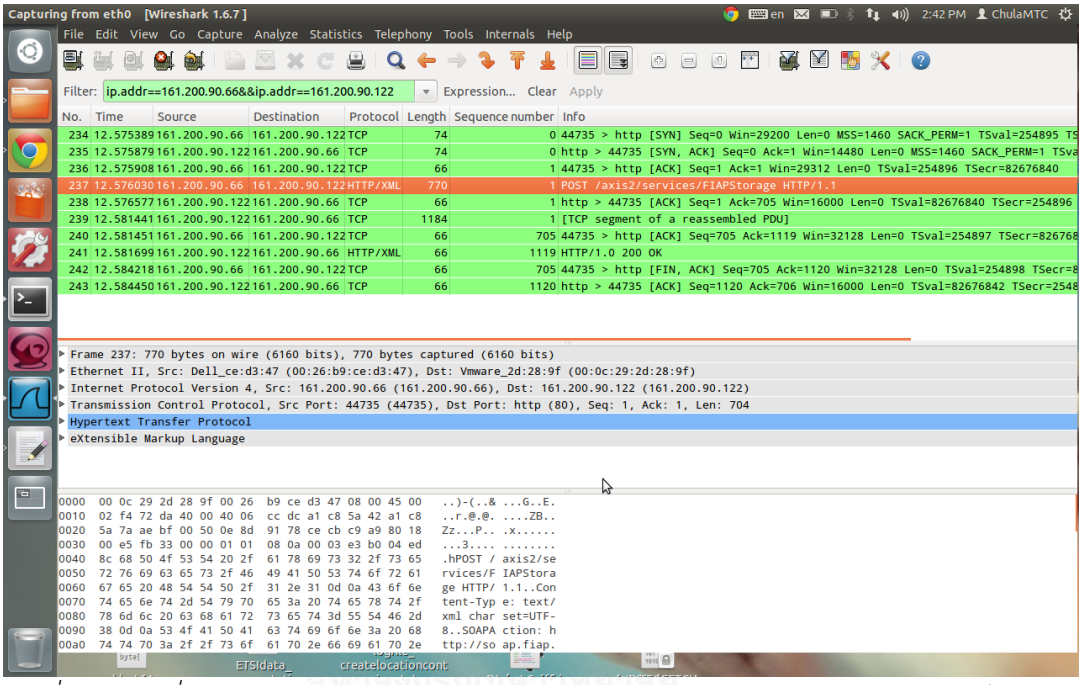
4.1.5 ตัวชี้วัดในการทดสอบ

ตัวชี้วัดในการทดสอบนี้ทำการตรวจวัดด้วยโปรแกรม Wireshark ซึ่งเป็นโปรแกรมที่สามารถตรวจวัดสมรรถนะในการส่งข้อมูลผ่านสายส่งสัญญาณ โดยในวิทยานิพนธ์ฉบับนี้ใช้ตัวชี้วัดในการทดสอบ 3 ชนิด ดังนี้

1) เวลาที่ใช้ในการทำงาน (Round trip time)

เวลาที่ใช้ในการทำงานหาได้โดยการคำนวณช่วงเวลาตั้งแต่เกตเวย์จำลองทำการส่งการร้องขอข้อมูลครั้งแรก ไปจนถึงเวลาที่เกตเวย์จำลองได้รับการตอบกลับข้อมูลจากหน่วยเก็บข้อมูลครั้งสุดท้าย แสดงตัวอย่างเวลาที่เกตเวย์จำลองส่งร้องขอข้อมูล และเวลาที่เกตเวย์จำลองได้รับข้อมูลตอบกลับดังรูปที่ 31

เวลาที่ใช้การทำงาน = เวลาที่ได้รับข้อมูลตอบกลับครั้งสุดท้าย – เวลาที่ส่งร้องขอข้อมูลครั้งแรก



รูปที่ 31 เวลาที่เกตเวย์จำลองส่งการร้องขอข้อมูลไปและได้รับข้อมูลตอบกลับจากหน่วยเก็บข้อมูล

จากรูปที่ 31 แถบสีส้มคือ ข้อมูลที่เกตเวย์ส่งร้องขอข้อมูลครั้งแรกไปยังหน่วยเก็บข้อมูล ส่วนบรรทัดที่โพรโทคอลเป็น HTTP/XML คือ ข้อมูลที่เกตเวย์ได้รับข้อมูลตอบกลับครั้งสุดท้ายจากหน่วยเก็บข้อมูล ทั้งนี้สามารถนำข้อมูลเวลามาคำนวณหาเวลาที่ใช้ในการทำงานได้ดังนี้

เวลาที่เกตเวย์จำลองส่งร้องขอข้อมูลครั้งแรกไปยังหน่วยเก็บข้อมูล คือ 12.576030 วินาที

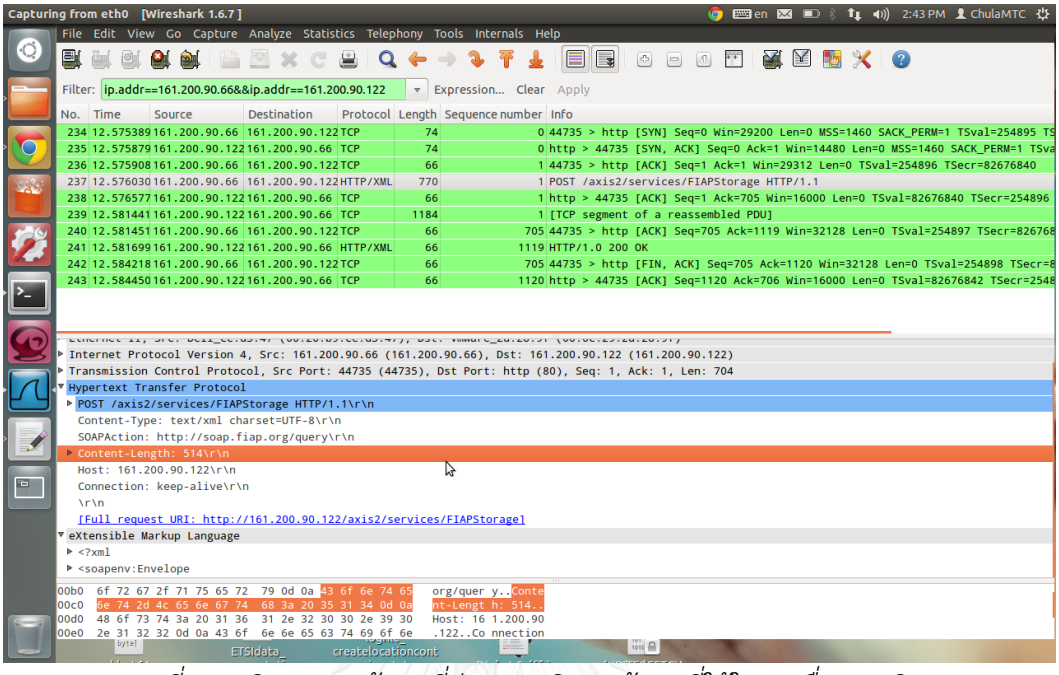
เวลาที่เกตเวย์จำลองได้รับข้อมูลตอบกลับครั้งสุดท้ายจากหน่วยเก็บข้อมูล คือ 12.581699 วินาที

ดังนั้น เวลาที่ใช้ในการทำงาน คือ 12.581699 – 12.576030 = 0.005669 วินาที

2) โอเวอร์เฮดของข้อมูล

ค่าโอเวอร์เฮดของข้อมูลหาได้โดยการคำนวณส่วนต่างของปริมาณข้อมูลที่ต้องการส่ง กับปริมาณข้อมูลที่ใช้จริงในการสื่อสาร โดยทำการรวมค่าส่วนต่างที่เกิดขึ้นทั้งในส่วนของการส่งร้องขอข้อมูลและส่วนที่เป็นการตอบกลับข้อมูลทั้งหมดในการดำเนินการนั้นๆ แสดงดังรูปที่ 32

ค่าโอเวอร์เฮดของข้อมูล = ปริมาณข้อมูลที่ใช้จริงในการสื่อสาร – ปริมาณข้อมูลที่ต้องการส่ง



รูปที่ 32 ปริมาณของข้อมูลที่ส่ง และปริมาณข้อมูลที่ใช้ในการสื่อสารจริง

จากรูปที่ 32 แถบสีเทาคือ ปริมาณข้อมูลที่ใช้จริงในการสื่อสาร ส่วนแถบสีส้มคือ ปริมาณข้อมูลที่ต้องการส่ง ทั้งนี้สามารถนำข้อมูลมาคำนวณหาโอเวอร์เฮดของข้อมูลได้ดังนี้

ปริมาณข้อมูลที่ใช้จริงในการสื่อสารคือ 770 ไบต์

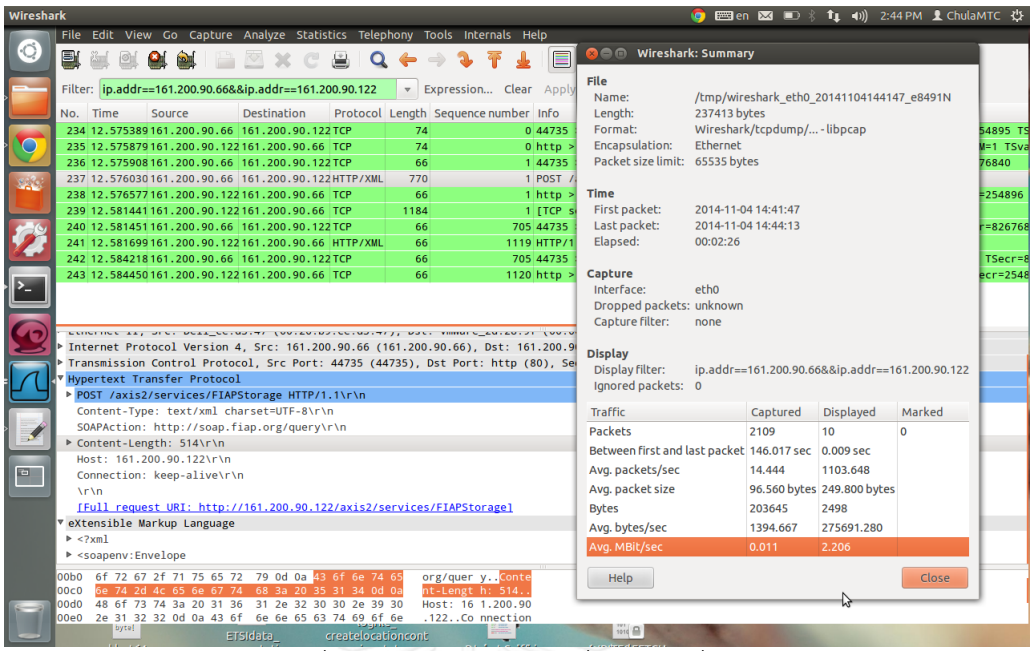
ปริมาณข้อมูลที่ต้องการส่งคือ 514 ไบต์

ค่าโอเวอร์เฮดของข้อมูลคือ 770 - 514 = 256 ไบต์

3) ปริมาณงาน (Throughput)

ค่าปริมาณงานหาได้โดยอาศัยความสามารถของโปรแกรม Wireshark ในการตรวจสอบ โดยเข้าไปที่หัวข้อ Statics และเลือกที่ Summary จากนั้นสังเกตที่แถบ Avg. Mbit/sec ในคอลัมน์ Capture นั่นคือ ค่าปริมาณงานเฉลี่ยในการสื่อสารนั้นๆ ดังแสดงในรูปที่ 33

ค่าปริมาณงานเฉลี่ยในการสื่อสาร คือ 0.011 Mbit/sec



รูปที่ 33 ข้อมูลปริมาณงานที่ใช้ในการสื่อสาร

4.2 การทดสอบและผลการทดสอบของโพรโทคอลที่นำเสนอ

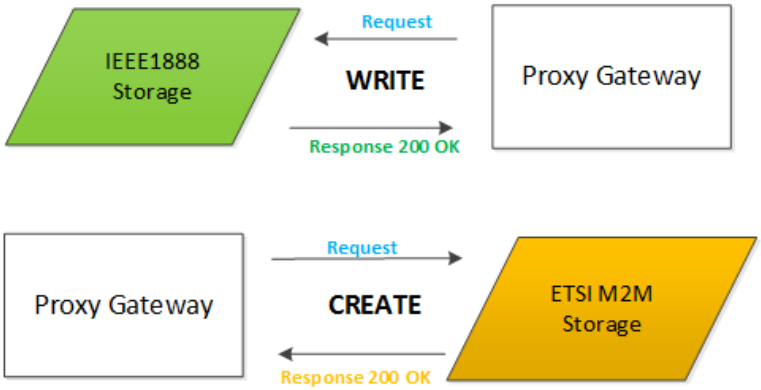
การทดสอบเพื่อเปรียบเทียบและตรวจสอบสมรรถนะการดำเนินการของโพรโทคอลที่นำเสนอตามรูปแบบต่าง ๆ ในวิทยานิพนธ์ฉบับนี้ มี 3 การทดสอบดังนี้

4.2.1 การทดสอบเบื้องต้น

ในขั้นต้น เราทำการทดสอบสมรรถนะของโพรโทคอลที่ใช้ในการสื่อสารของทั้งสองมาตรฐาน โดยทำการทดสอบตามตัวชี้วัด 3 ชนิดคือ เวลาที่ใช้ในการทำงาน, โอเวอร์เฮดของข้อมูล และปริมาณงาน ซึ่งทำการตรวจวัดด้วยโปรแกรม Wireshark สังเกตการส่งการร้องขอข้อมูลและการตอบกลับข้อมูลในระหว่างการดำเนินการแลกเปลี่ยนข้อมูลระหว่างหน่วยเก็บข้อมูลของสองมาตรฐานซึ่งถูกติดตั้งไว้ในคอมพิวเตอร์คนละเครื่อง และกำหนดเลขที่อยู่ไอพีไว้ต่างกันคือ หน่วยเก็บข้อมูลของ CUBEMS กำหนดเป็น 161.200.90.122 และหน่วยเก็บข้อมูลของ OpenMTC กำหนดเป็น 161.200.90.78 โดยทำการทดสอบการส่งข้อมูลและการดึงข้อมูลในการสื่อสารจำนวน 1 จุดข้อมูล (Point ID) จนถึง 10 จุดข้อมูล ในแต่ละการทดสอบจะทำการดำเนินงานโปรแกรมเป็นจำนวน 10 ครั้งแล้วหาค่าเฉลี่ยของผลการดำเนินการนั้น แล้วนำค่าของตัวชี้วัดของการดำเนินการที่ได้มาแสดงการเปรียบเทียบระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M โดยแบ่งการทดสอบออกเป็น 2 การทดสอบคือ การทดสอบดำเนินการส่งข้อมูลไปเก็บยังหน่วยเก็บข้อมูล และการทดสอบการดำเนินการดึงข้อมูลจากหน่วยเก็บข้อมูล

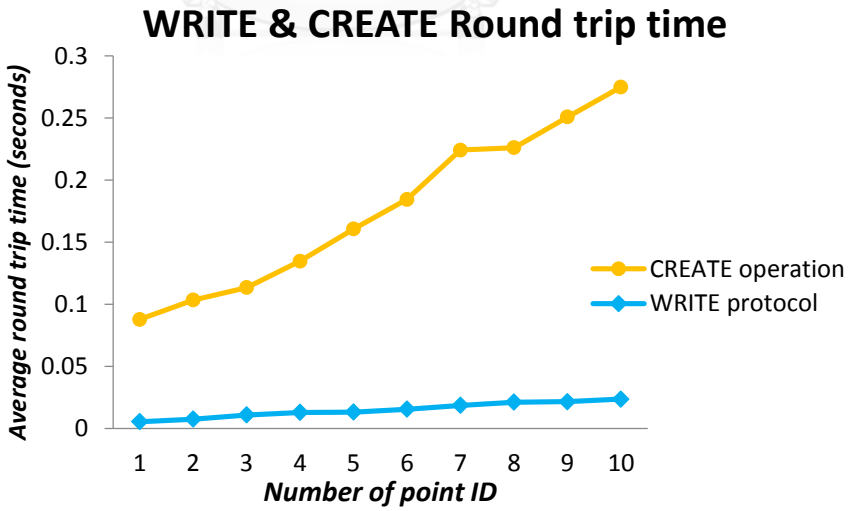
4.2.1.1 เปรียบเทียบสมรรถนะจากการทดสอบในการดำเนินการส่งข้อมูลไปยังหน่วยเก็บข้อมูล

ในการทดสอบนี้ เพื่อทำการเปรียบสมรรถนะของการดำเนินการระหว่างเกตเวย์จำลองใช้โพรโทคอล WRITE ในการส่งข้อมูลไปยังหน่วยเก็บข้อมูลของ CUBEMS กับใช้การดำเนินการ CREATE ในการส่งข้อมูลไปยังหน่วยเก็บข้อมูลของ OpenMTC แสดงการดำเนินการดังรูปที่ 34

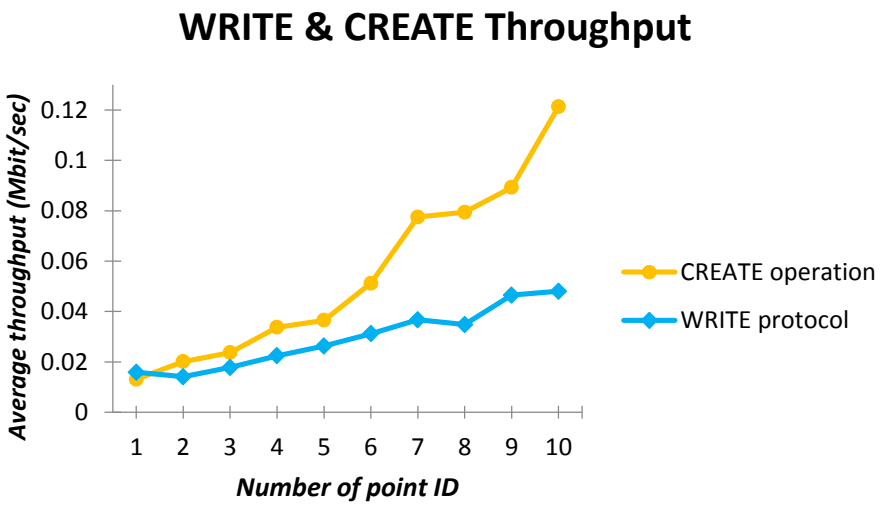


รูปที่ 34 การดำเนินการส่งข้อมูลของเกตเวย์จำลองไปยังหน่วยเก็บข้อมูล

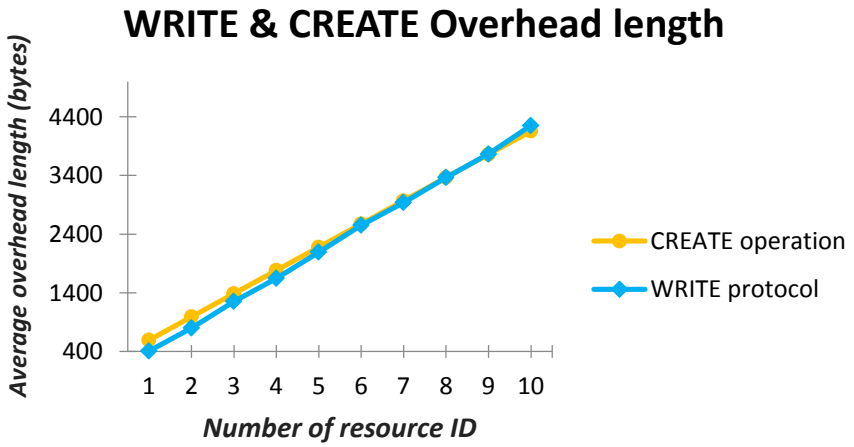
ผลการทดสอบสมรรถนะของการดำเนินการส่งข้อมูลไปยังหน่วยเก็บข้อมูลตามตัวชี้วัดซึ่งคือเวลาที่ใช้ในการทำงาน ปริมาณงาน และโอเวอร์เฮดที่เกิดขึ้น แล้วนำมาเปรียบเทียบกัน แสดงดังรูปที่ 35,รูปที่ 36 และรูปที่ 37



รูปที่ 35 เปรียบเทียบเวลาที่ใช้ในการทำงานของโพรโทคอล WRITE และการดำเนินการ CREATE



รูปที่ 36 เปรียบเทียบปริมาณงานของโพรโทคอล WRITE และการดำเนินการ CREATE



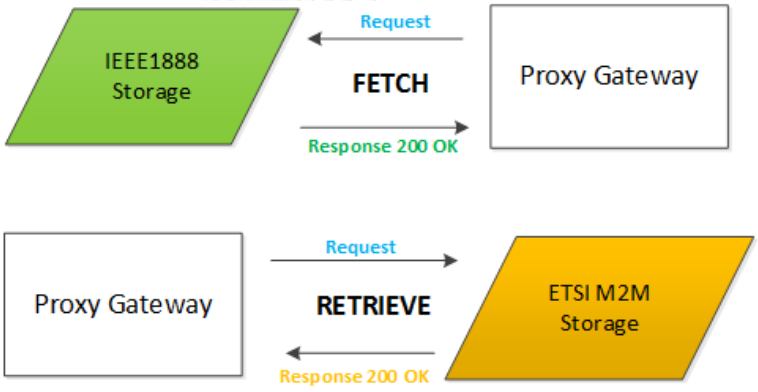
รูปที่ 37 เปรียบเทียบโอเวอร์เฮดของข้อมูลของโพรโทคอล WRITE และการดำเนินการ CREATE

จากรูปที่ 35 สังเกตเห็นว่า การดำเนินการ CREATE ตามมาตรฐาน ETSI M2M ใช้เวลาในการทำงานมากกว่า โพรโทคอล WRITE ตามมาตรฐาน IEEE1888 อันเนื่องจากในขั้นตอนการดำเนินการส่งข้อมูลเข้าไปในหน่วยเก็บข้อมูลของ OpenMTC จะต้องมีการสร้างส่วนประกอบในการเก็บข้อมูลซึ่งประกอบไปด้วย APP, Containers, ContentInstances โดยจะมีการส่งการร้องขอในการสร้างส่วนประกอบ 3 ส่วนประกอบตามลำดับจาก APP ไปจนถึง ContentInstance ซึ่งเป็นส่วนประกอบที่เก็บข้อมูล แต่สำหรับการส่งข้อมูลไปยังหน่วยเก็บข้อมูลของ CUBEMS จะทำการส่งการร้องขอเพียงครั้งเดียว ข้อมูลจะถูกบันทึกไปยังหน่วยเก็บข้อมูลพร้อมกับตำแหน่งของข้อมูลที่ระบุไปพร้อมกับการส่งการร้องขอ ซึ่งส่งผลไปถึงค่าปริมาณงานจากรูปที่ 36 จะเห็นว่าค่าปริมาณงานของการดำเนินการ CREATE มีค่ามากกว่าค่าปริมาณงานของโพรโทคอล WRITE ซึ่งหมายถึงว่าการ

ดำเนินการ CREATE มีความต้องการในการส่งข้อมูลมากกว่าโพรโทคอล WRITE แต่จากรูปที่ 37 จะเห็นว่าค่าโอเวอร์เฮดที่เกิดขึ้นของทั้งสองการดำเนินการมีค่าไม่แตกต่างกันมาก เนื่องจากในการดำเนินการตามโพรโทคอล WRITE จะมีค่าโอเวอร์เฮดที่เกิดขึ้นต่อ 1 จุดข้อมูลคือ 387 ไบท์ และค่าโอเวอร์เฮดที่เกิดขึ้นต่อ 1 จุดข้อมูลของการดำเนินการ CREATE คือ 396 ไบท์ ซึ่งมีค่าแตกต่างกันเพียง 9 ไบท์เท่านั้น จึงทำให้ค่าโอเวอร์เฮดที่เกิดขึ้นในระหว่างการดำเนินงานของทั้งสองการดำเนินการมีค่าใกล้เคียงกัน

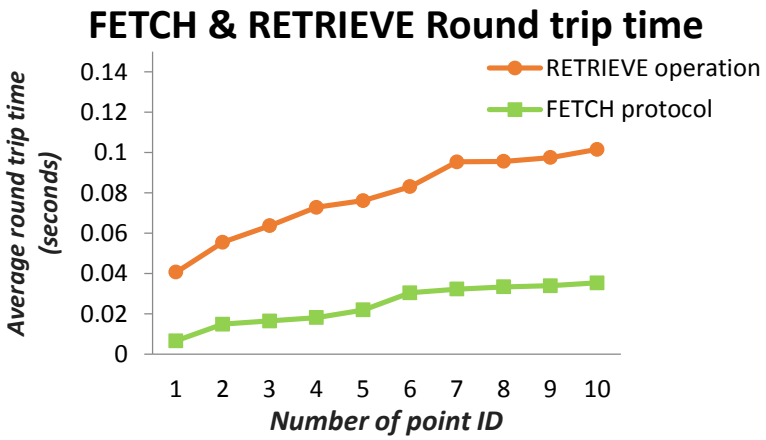
4.2.1.2 เปรียบเทียบสมรรถนะจากการทดสอบในการดำเนินการดึงข้อมูลจากหน่วยเก็บข้อมูล

ในการทดสอบนี้ เพื่อทำการเปรียบสมรรถนะของการดำเนินการระหว่างเกตเวย์จำลองใช้โพรโทคอล FETCH ในการดึงข้อมูลจากหน่วยเก็บข้อมูลของ CUBEMS กับใช้การดำเนินการ RETRIEVE ในการดึงข้อมูลจากหน่วยเก็บข้อมูลของ OpenMTC แสดงดังรูปที่ 38

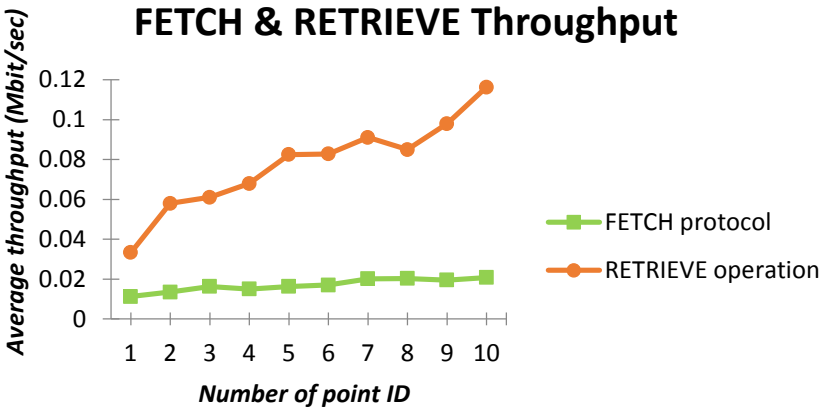


รูปที่ 38 การดำเนินการดึงข้อมูลของเกตเวย์จำลองจากหน่วยเก็บข้อมูล

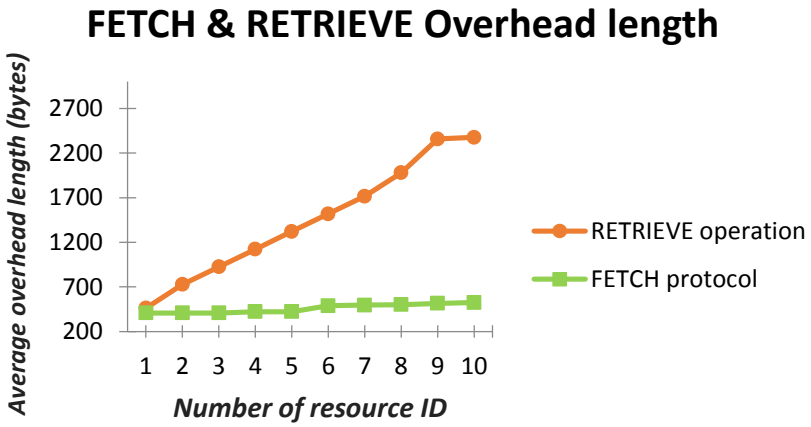
โดยการทดสอบนี้ จะทำการตรวจวัดตามตัวชี้วัด 3 ชนิดคือ เวลาที่ใช้ในการทำงาน โอเวอร์เฮดของข้อมูล และปริมาณงานในการดึงข้อมูลจำนวน 1 จุดข้อมูล ไปจนถึงจำนวน 10 จุดข้อมูล แล้วนำมาเปรียบเทียบกัน แสดงผลการทดสอบดังรูปที่ 39,รูปที่ 40 และรูปที่ 41



รูปที่ 39 เปรียบเทียบเวลาที่ใช้ในการทำงานของโพรโทคอล FETCH และการดำเนินการ RETRIEVE



รูปที่ 40 เปรียบเทียบปริมาณงานของโพรโทคอล FETCH และการดำเนินการ RETRIEVE



รูปที่ 41 เปรียบเทียบโอเวอร์เฮดของข้อมูลของโพรโทคอล FETCH และการดำเนินการ RETRIEVE

จากรูปที่ 39 สังเกตเห็นว่า การดำเนินการ RETRIEVE ตามมาตรฐาน ETSI M2M ใช้เวลาในการทำงานมากกว่า โพรโทคอล FETCH ตามมาตรฐาน IEEE1888 เนื่องจากในการดึงข้อมูลจากหน่วยเก็บข้อมูลของ OpenMTC ด้วยการดำเนินการ RETRIEVE จะทำการตรวจเช็คข้อมูลในทุกส่วนประกอบที่ใช้ในการเก็บข้อมูลตามลำดับเช่นเดียวกับการดำเนินการ CREATE ซึ่งทำให้มีการส่งการร้องขอข้อมูลหลายครั้ง แต่สำหรับการดึงข้อมูลจากหน่วยเก็บข้อมูลของ CUBEMS ด้วยโพรโทคอล FETCH นั้น มีการส่งการร้องขอข้อมูลเพียง 1 ครั้งไม่ว่าจะทำการดึงข้อมูลกี่จุดข้อมูลก็ตาม ซึ่งมีการระบุตัวระบุตำแหน่งข้อมูลที่ต้องการได้ไว้ในข้อความร้องขอข้อมูลนั้นด้วย พร้อมกับได้รับการตอบกลับข้อมูลเพียง 1 ครั้งเช่นกัน ส่งผลต่อค่าโอเวอร์เฮดของข้อมูลที่เกิดขึ้นทำให้ค่าโอเวอร์เฮดของข้อมูลของการดำเนินการ RETRIEVE มีค่ามากกว่าโพรโทคอล FETCH ตามรูปที่ 41 รวมไปถึงค่าปริมาณงานในการดำเนินงานดังรูปที่ 40 ค่าปริมาณงานของการดำเนินการ RETRIEVE มีค่ามากกว่าโพรโทคอล FETCH ซึ่งหมายความว่าการดำเนินงานของการดำเนินการ RETRIEVE มีความต้องการที่

จะส่งข้อมูลในปริมาณที่มากกว่าการดำเนินงานของโพรโทคอล FETCH เนื่องด้วยมีค่าโฮเวอร์เฮดที่เกิดขึ้นในระหว่างการดำเนินงานมากกว่าในจำนวนมาก

4.2.2 การทดสอบโครงสร้างการจำลอง

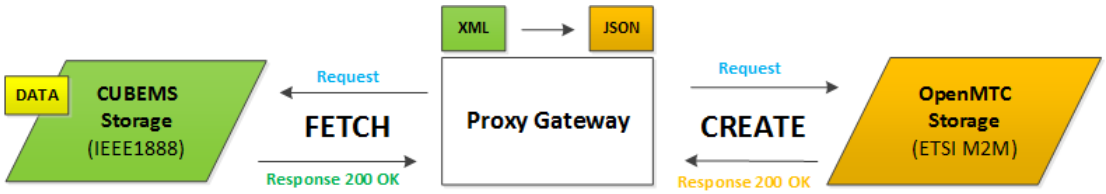
จากการดำเนินการของโพรโทคอลของเกตเวย์จำลองที่ถูกออกแบบให้ทำหน้าที่ดึงข้อมูลและส่งข้อมูลระหว่างหน่วยเก็บข้อมูลด้วยการดำเนินการของทั้งสองมาตรฐานนั้น แบ่งการทำงานออกเป็น 2 กระบวนการดังนี้

1) **กระบวนการ 1 : ร้องขอข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888**
   เป็นกระบวนการที่เกตเวย์จำลองดำเนินการดึงข้อมูลจากหน่วยเก็บข้อมูลของ CUBEMS ด้วยโพรโทคอล FETCH แล้วแปลงรูปแบบข้อมูล ส่งต่อไปยังหน่วยเก็บข้อมูลของ OpenMTC ด้วยการดำเนินการ CREATE
2) **กระบวนการ 2 : ร้องขอข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M**
   เป็นกระบวนการที่เกตเวย์จำลองดำเนินการดึงข้อมูลจากหน่วยเก็บข้อมูลของ OpenMTC ด้วยการดำเนินการ RETRIEVE แล้วแปลงรูปแบบข้อมูล ส่งต่อไปยังหน่วยเก็บข้อมูลของ CUBEMS ด้วยโพรโทคอล WRITE

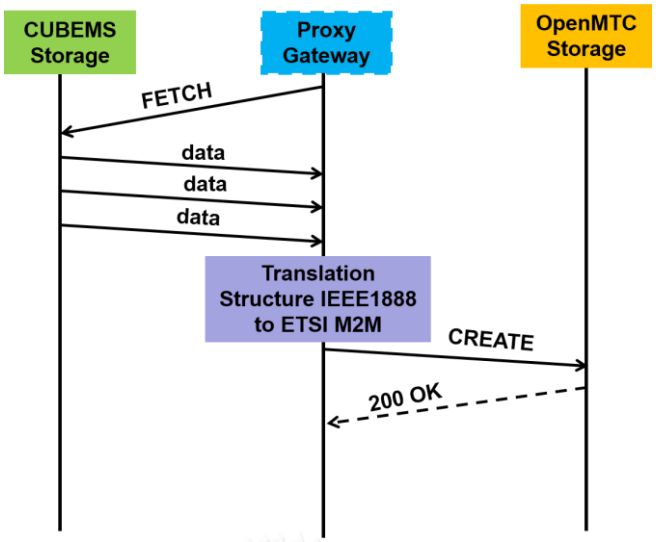
4.2.2.1 การทดสอบการดำเนินการของกระบวนการ 1 และกระบวนการ 2

อธิบายการดำเนินงานของทั้งสองกระบวนการด้วยแผนภาพการทำงานดังนี้

การดำเนินงานของกระบวนการ 1 คือ การดำเนินงานโปรแกรมบนเกตเวย์จำลองเพื่อให้เกิดการดึงข้อมูลด้วยโพรโทคอล FETCH จากหน่วยเก็บข้อมูลของ CUBEMS เกตเวย์จำลองได้รับข้อมูลตอบกลับมาในรูปแบบ XML จากนั้นแปลงรูปแบบข้อมูลจาก XML เป็น JSON แล้วใช้การดำเนินการ CREATE บันทึกข้อมูลลงในหน่วยเก็บข้อมูลของ OpenMTC ดังรูปที่ 42 และรูปที่ 43

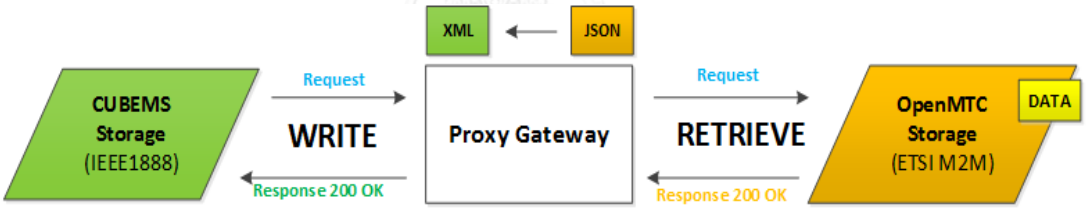


รูปที่ 42 การดำเนินการของเกตเวย์จำลองในกระบวนการ 1

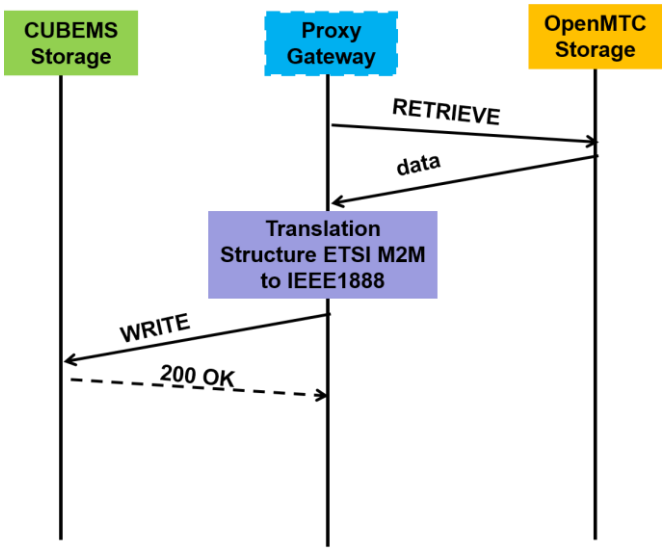


รูปที่ 43 แผนภาพกระบวนการทำงานของเกตเวย์จำลองในการทดสอบตามกระบวนการ 1

การดำเนินงานของกระบวนการ 2 คือ การดำเนินงานโปรแกรมบนเกตเวย์จำลองเพื่อให้เกิดการดึงข้อมูลด้วยการดำเนินการ RETRIEVE จากหน่วยเก็บข้อมูลของ OpenMTC เกตเวย์จำลองได้รับข้อมูลตอบกลับมาในรูปแบบ JSON จากนั้นแปลงรูปแบบข้อมูลจาก JSON เป็น XML แล้วใช้โพรโทคอล WRITE บันทึกข้อมูลลงในหน่วยเก็บข้อมูลของ CUBEMS ดังรูปที่ 44 และรูปที่ 45



รูปที่ 44 การดำเนินการของเกตเวย์จำลองในกระบวนการ 2



รูปที่ 45 แผนภาพการทำงานของเกตเวย์จำลองในการทดสอบตามกระบวนการ 2

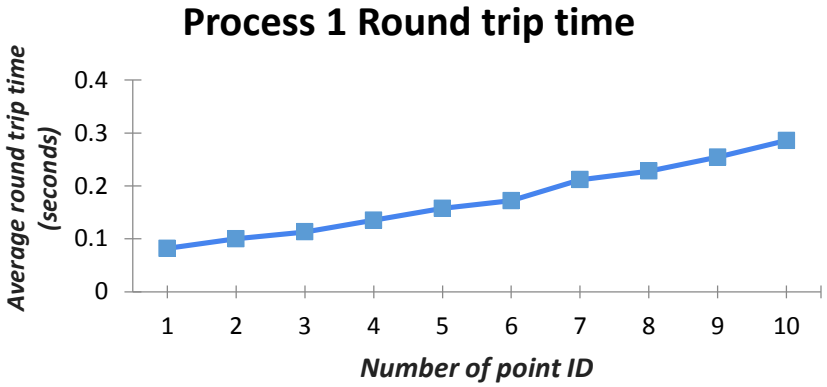
โดยการทดสอบ จะทำการตรวจวัดตามตัวชี้วัด 3 ชนิดคือ เวลาที่ใช้ในการทำงาน โอเวอร์เฮดของข้อมูล และปริมาณในการส่งต่อข้อมูลของทั้งสองกระบวนการ ด้วยข้อมูลจำนวน 1 จุดข้อมูล ไปจนถึงจำนวน 10 จุดข้อมูล โดยทำการทดสอบ 10 ครั้งแล้วดำเนินการหาค่าเฉลี่ย เพื่อให้ทราบถึงขีดจำกัดของการดำเนินการในกระบวนการ 1 และกระบวนการ 2 แล้วนำมาเปรียบเทียบกัน แสดงเป็นผลการทดสอบ

4.2.2.2 ผลการทดสอบสมรรถนะของกระบวนการ 1 และกระบวนการ 2

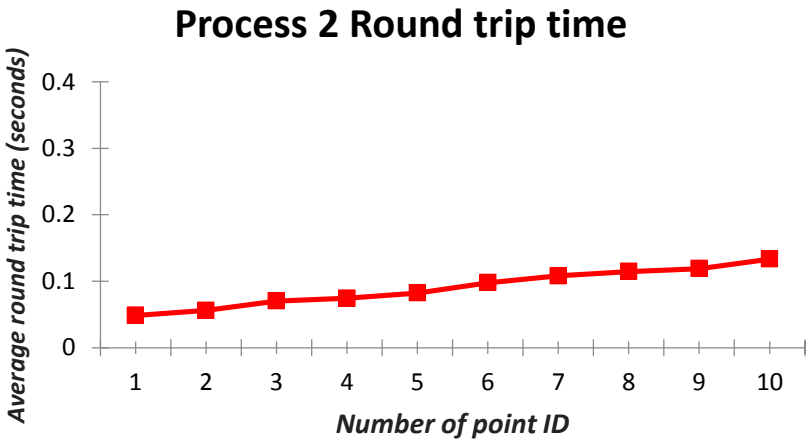
ผลการทดสอบการดำเนินการของกระบวนการ 1 และกระบวนการ 2 แยกตามตัวชี้วัดดังนี้

1) เวลาที่ใช้ในการทำงาน

ผลการทดสอบเวลาที่ใช้ในการทำงานของทั้งกระบวนการ 1 และกระบวนการ 2 แสดงดังรูปที่ 46 และรูปที่ 47



รูปที่ 46 เวลาที่ใช้ในการทำงานของกระบวนการ 1

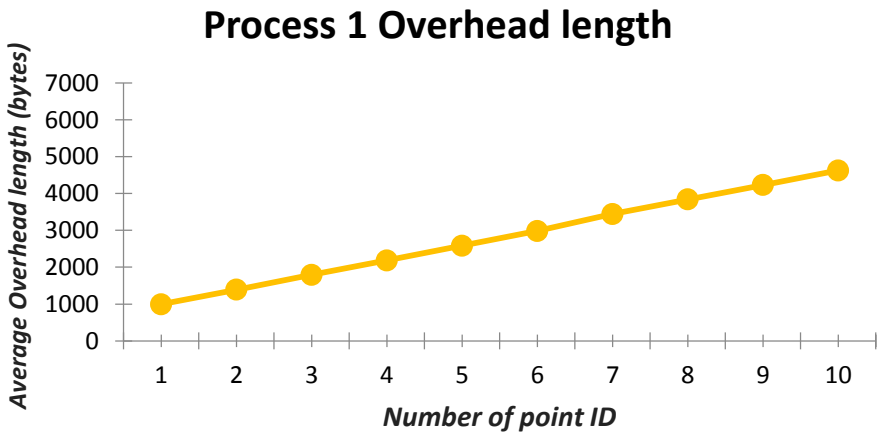


รูปที่ 47 เวลาที่ใช้ในการทำงานของกระบวนการ 2

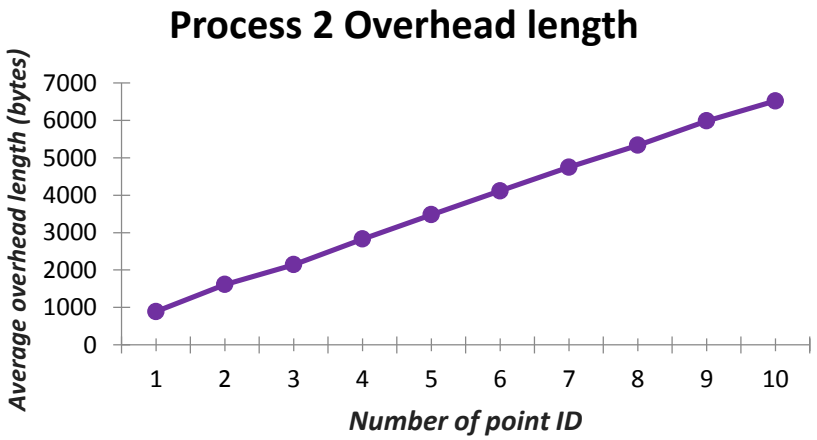
จากรูปที่ 46 และ 47 จะเห็นว่าเวลาที่ใช้ในการทำงานของกระบวนการ 1 มากกว่ากระบวนการ 2 อันเนื่องมาจากในกระบวน 1 ซึ่งประกอบด้วยโพรโทคอล FETCH และการดำเนินการ CREATE มีการส่งการร้องขอในการดำเนินงานมากกว่า โดยเฉพาะการดำเนินการ CREATE ซึ่งมีการส่งการร้องขอเพื่อส่งข้อมูลไปยังหน่วยเก็บข้อมูลของ OpenMTC เพิ่มขึ้น 2 การร้องขอในทุกๆ 1 จุดข้อมูลที่เพิ่มขึ้น ทำให้การดำเนินงานของกระบวนการ 1 ซึ่งมีทั้งการร้องขอ และการตอบกลับ ใช้เวลาในการทำงานมากกว่ากระบวนการ 2

2) โอเวอร์เฮดของข้อมูล

ผลการทดสอบโอเวอร์เฮดของข้อมูลของทั้งกระบวนการ 1 และกระบวนการ 2 แสดงดังรูปที่ 48 และรูปที่ 49



รูปที่ 48 โอเวอร์เฮดของข้อมูลของกระบวนการ 1

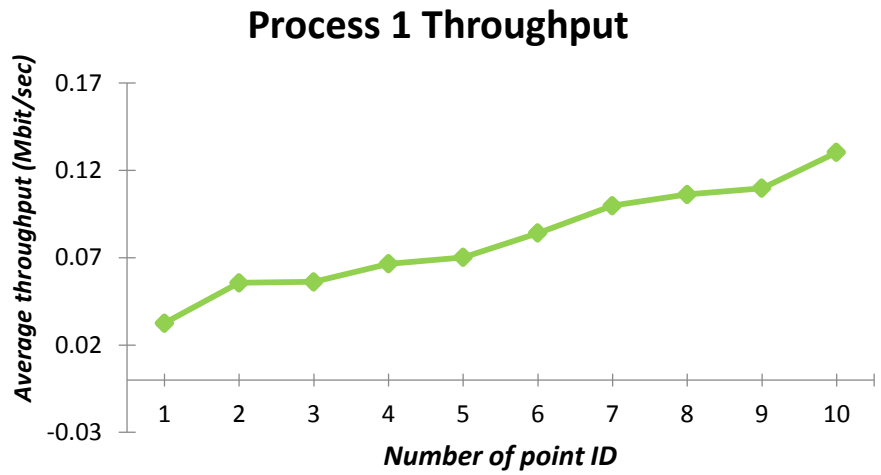


รูปที่ 49 โอเวอร์เฮดของข้อมูลของกระบวนการ 2

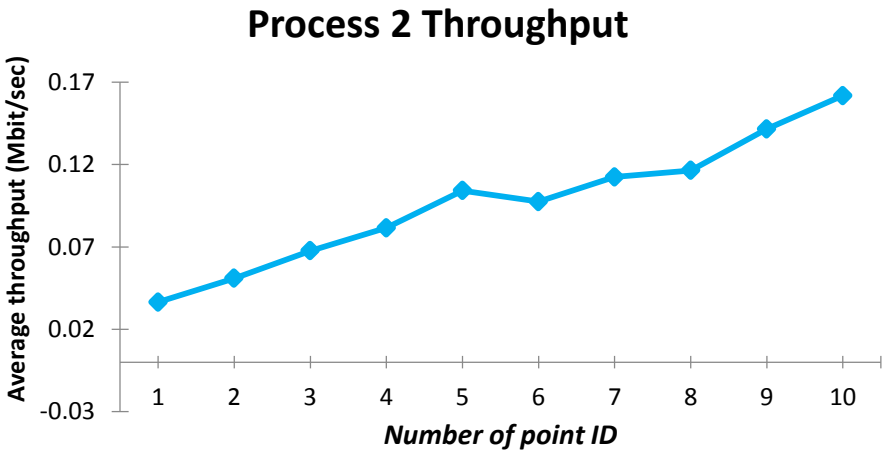
จากรูปที่ 48 และ 49 จะเห็นว่า ค่าโอเวอร์เฮดที่เกิดขึ้นจากการดำเนินการตามกระบวนการ 2 มีค่ามากกว่ากระบวนการ 1 อันเนื่องมาจากในกระบวน 2 ประกอบไปด้วยการดำเนินการ RETRiEVE และโพรโทคอล FETCH มีค่าโอเวอร์เฮดที่เกิดขึ้นเพิ่มขึ้นจำนวน 641 ไบท์ทุกๆ 1 จุดข้อมูล ซึ่งมากกว่ากระบวนการ 1 ที่ประกอบไปด้วยโพรโทคอล FETCH และการดำเนินการ CREATE โดยมีค่าโอเวอร์เฮดที่เกิดขึ้นเพิ่มขึ้น 396 ไบท์ในทุกๆ 1 จุดข้อมูล จึงทำให้ค่าโอเวอร์เฮดรวมที่เกิดขึ้นในกระบวนการ 2 มีค่ามากกว่ากระบวนการ 1

3) ปริมาณงาน

ผลการทดสอบปริมาณงานของทั้งกระบวนการ 1 และกระบวนการ 2 แสดงดังรูปที่ 50 และรูปที่ 51



รูปที่ 50 ปริมาณงานของกระบวนการ 1



รูปที่ 51 ปริมาณงานของกระบวนการ 2

จากรูปที่ 50 และ 51 จะเห็นว่าค่าปริมาณงานจากการดำเนินการตามกระบวนการ 2 มีค่ามากกว่ากระบวนการ 1 อันเป็นผลมากจากค่าโดยรวมของโอเวอร์เฮดที่เกิดขึ้นของกระบวนการ 2 มีค่ามากกว่ากระบวนการ 1 ซึ่งแสดงว่าการดำเนินงานตามกระบวนการ 2 มีความต้องการในการส่งข้อมูลมากกว่ากระบวนการ 1

จากผลการทดสอบสมรรถนะของการดำเนินการตามกระบวนการ 1 และกระบวนการ 2 ของเกตเวย์จำลอง สามารถสรุปได้ว่า การทำงานของกระบวนการ 1 และกระบวนการ 2 ของเกตเวย์จำลองตามตัวชี้วัดทั้งสามชนิดนี้ จะเห็นว่าเกตเวย์จำลองมีสมรรถนะของการดำเนินการตามกระบวน 2 ในการดึงข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M ส่งต่อไปยังหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 ใช้เวลาในการทำงาน มีค่าโอเวอร์เฮดที่เกิดขึ้น และค่าปริมาณที่ส่งในกระบวนการมากกว่ากระบวนการ 1 เนื่องมาจากผลจากการดำเนินงานของโพรโทคอล FETCH ทำงานร่วมกับการดำเนินการ CREATE ทำให้มีจำนวนการส่งการร้องขอและรับการตอบกลับข้อมูลมากกว่าผลจากการทำงานร่วมกันระหว่างการดินการ RETRIEVE กับโพรโทคอล WRITE และยังทำให้ทราบว่า เมื่อจำนวนชุดข้อมูลที่ใช้ในการดำเนินการเพิ่มขึ้น กระบวนการทำงานทั้งสองนี้มีแนวโน้มที่จะใช้เวลาในการดำเนินการ ค่าปริมาณงาน และโอเวอร์เฮดเพิ่มมากขึ้นด้วย ซึ่งอาจส่งผลกระทบโครงข่ายการทำงานในปริมาณจุดข้อมูลที่มาก แต่อาจเป็นผลกระทบที่ยอมรับได้ เนื่องจากเวลาที่ใช้ในการดำเนินการเพิ่มขึ้นเพียงเล็กน้อยเท่านั้น

4.2.3 การเปรียบเทียบสมรรถนะของโครงสร้างการจำลองในรูปแบบต่างๆ

เนื่องจากมาตรฐาน IEEE1888 และมาตรฐาน ETSI M2M มีโครงสร้างในการจัดการตำแหน่งข้อมูลที่แตกต่างกัน นั่นคือ มีวิธีการระบุตำแหน่งที่มาของข้อมูลที่แตกต่างกัน ในมาตรฐาน IEEE1888 การจัดการตำแหน่งข้อมูลจะมีการระบุตำแหน่งข้อมูลลงไปใน Point ID เช่น http://bems.ee.eng.chula.ac.th/eng4/fl13/north/lab_tsrl_dsprl_emrl/z1/sensor1/monitor/temperature เป็นต้น

แต่ในมาตรฐาน ETSI M2M จะมีวิธีการระบุตำแหน่งข้อมูลได้ 3 แบบคือ

1) ContainerID คือการระบุตำแหน่งของข้อมูลไปในชื่อของ Container ยกตัวอย่างเช่น localhost:15000/m2m/applications/ProxyMeter/containers/eng4_fl13_north_lab_tsrl_dsprl_emrl_z1_sensor1_monitor_temperature/contentInstances
2) SearchString คือการระบุข้อมูลลงในหัวข้อมูล SearchString ซึ่งเป็นหัวข้อที่ระบุข้อมูลของ Container นั้นๆเพื่อรองรับการดำเนินการค้นหาข้อมูล โดยไม่ต้องผ่านชื่อ Container ในวิธีนี้จะระบุชื่อ Container เป็นแบบไหนก็ได้

3) LocationContainer คือการระบุข้อมูลลงไปในหัวข้อ LocationContainer ซึ่งเป็นหัวข้อที่ใช้ระบุตำแหน่งของแหล่งข้อมูลที่บันทึกใน Container นั้นๆ

จากความแตกต่างของการระบุที่มาของข้อมูลของทั้งสองมาตรฐาน เป็นผลให้สามารถออกแบบการจำลองโครงสร้างได้เป็น 3 รูปแบบ โดยใช้ชื่อตามวิธีการระบุตำแหน่งข้อมูลตามมาตรฐาน ETSI M2M นั่นคือ

1. ContainerID

เป็นรูปแบบการจำลองโครงสร้างที่มีการดำเนินการคือ เมื่อเกตเวย์จำลองได้รับข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 แล้ว ในส่วนการส่งต่อข้อมูลไปบันทึกในหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M จะใช้วิธีการระบุตำแหน่งข้อมูลไปในชื่อของ container ซึ่งเป็นส่วนเก็บค่าข้อมูล แสดงตามรูปที่ 52


```
{
  "containers": {
    "containerAnncCollection": {
      "namedReference": []
    },
    "containerCollection": {
      "namedReference": [
        {
          "id": "eng4_fl13_north_lab_tsrl_dsprl_emrl_z1_sensor1_monitor_temperature",
          "$t": "/m2m/applications/ProxyMeter/containers
/eng4_fl13_north_lab_tsrl_dsprl_emrl_z1_sensor1_monitor_temperature"
        }
      ]
    },
    "creationTime": "2014-11-23T10:26:51.901Z",
    "lastModifiedTime": "2014-11-23T10:26:51.930Z",
    "locationContainerAnncCollection": {
      "namedReference": []
    },
    "locationContainerCollection": {
      "namedReference": []
    },
    "subscriptionsReference": "/m2m/applications/ProxyMeter/containers/subscriptions"
  }
}
```


รูปที่ 52 รูปแบบการจำลองโครงสร้างแบบ ContainerID

2. SearchString

เป็นรูปแบบการจำลองโครงสร้างที่มีการดำเนินการคือ เมื่อเกตเวย์จำลองได้รับข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 แล้ว ในส่วนของการส่งต่อข้อมูลไปบันทึกในหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M จะใช้วิธีการระบุตำแหน่งข้อมูลไปในหัวข้อ SearchString ซึ่งเป็นส่วนที่ใช้ในการค้นหาค่าข้อมูลด้วยการระบุตำแหน่งของข้อมูล แสดงตามรูปที่ 53

```
{
  "container": {
    "announceTo": {
      "sclList": {
        "reference": []
      }
    },
    "contentInstancesReference": "/m2m/applications/ProxyMeter_ss/containers/1001/contentInstances",
    "creationTime": "2014-11-27T09:35:05.866Z",
    "expirationTime": "2014-11-28T09:35:05.866Z",
    "id": 1001,
    "lastModifiedTime": "2014-11-27T09:35:05.866Z",
    "maxByteSize": -1,
    "maxInstanceAge": "-PT1S",
    "maxNrOfInstances": -1,
    "searchStrings": {
      "searchString": [
        "eng4",
        "fl13",
        "north",
        "lab_tsrl_dsprl_emrl",
        "z1",
        "sensor1",
        "monitor",
        "temperature"
      ]
    },
    "subscriptionsReference": "/m2m/applications/ProxyMeter_ss/containers/1001/subscriptions"
  }
}
```


รูปที่ 53 รูปแบบการจำลองโครงสร้างแบบ SearchString

3. LocationContainer

เป็นรูปแบบการจำลองโครงสร้างที่มีการดำเนินการคือ เมื่อเกตเวย์จำลองได้รับข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 แล้ว ในส่วนของการส่งต่อข้อมูลไปบันทึกในหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M จะใช้วิธีการระบุตำแหน่งข้อมูลไปในหัวข้อ LocationContainer ซึ่งเป็นส่วนใช้ระบุตำแหน่งที่มาของ Container ที่ใช้เก็บค่าข้อมูลนั้นๆ แสดงตามรูปที่ 54


```
{
  "containers": {
    "containerAnncCollection": {
      "namedReference": []
    },
    "containerCollection": {
      "namedReference": [
        {
          "id": 1002,
          "$t": "/m2m/applications/ProxyMeter_location/containers/1002"
        }
      ]
    },
    "creationTime": "2014-11-27T09:37:53.022Z",
    "lastModifiedTime": "2014-11-27T09:37:53.070Z",
    "locationContainerAnncCollection": {
      "namedReference": []
    },
    "locationContainerCollection": {
      "namedReference": [
        {
          "id": "eng4_fl13_north_lab_tsrl_dsprl_emrl_z1_sensor1_monitor_temperature",
          "$t": "/m2m/applications/ProxyMeter_location/containers
/eng4_fl13_north_lab_tsrl_dsprl_emrl_z1_sensor1_monitor_temperature"
        }
      ]
    },
    "subscriptionsReference": "/m2m/applications/ProxyMeter_location/containers/subscriptions"
  }
}
```

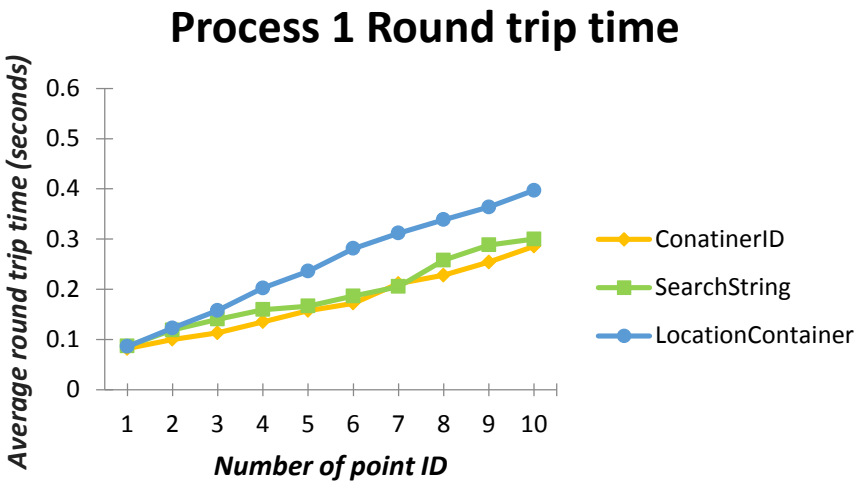

รูปที่ 54 รูปแบบการจำลองโครงสร้างแบบ LocationContainer

จากรูปแบบโครงสร้างการจำลองทั้งสามรูปแบบข้างต้น นำไปสู่การทดสอบสมรรถนะของแต่ละรูปแบบเพื่อให้ทราบว่ารูปแบบใดมีความเหมาะสมกับการดำเนินการของเกตเวย์จำลองในวิทยานิพนธ์ฉบับนี้ และเพื่อให้เป็นประโยชน์ต่อการพัฒนางานต่อไปในอนาคต โดยการทดสอบจะทำ

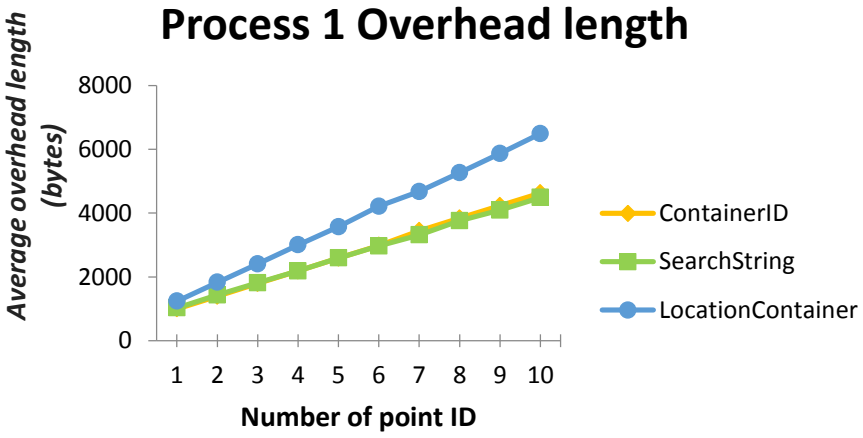
การตรวจวัดตามตัวชี้วัด 3 ชนิดคือ เวลาที่ใช้ในการทำงาน โอเวอร์เฮดของข้อมูล และปริมาณงานตามกระบวนการของเกตเวย์จำลองทั้ง 2 กระบวนการ ด้วยข้อมูลจำนวน 1 จุดข้อมูล ไปจนถึงจำนวน 10 จุดข้อมูล โดยทำการทดสอบ 10 ครั้ง แล้วนำมาหาค่าเฉลี่ย

4.2.3.1 ผลการเปรียบเทียบสมรรถนะของโครงสร้างการจำลองในกระบวนการ 1

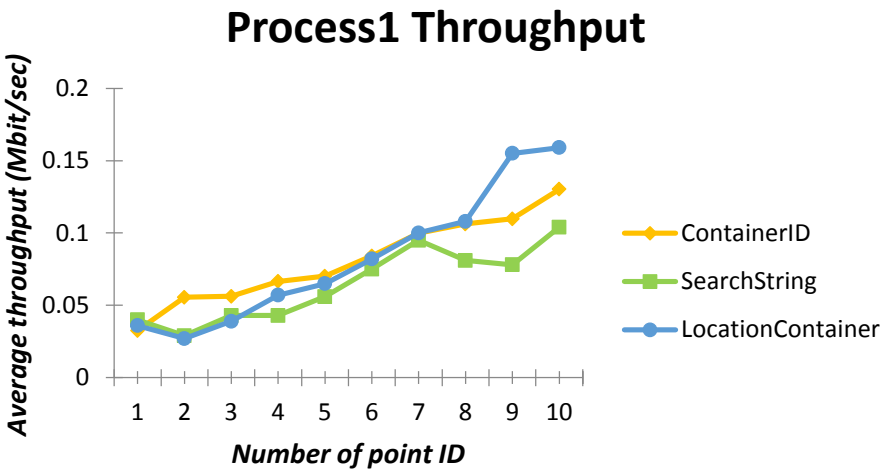
จากการดำเนินการตามกระบวนการ 1 นั่นคือการดำเนินงานโปรแกรมบนเกตเวย์จำลอง เพื่อส่งต่อข้อมูลจากหน่วยเก็บข้อมูลของ CUBEMS ไปเก็บยังหน่วยเก็บข้อมูลของ OpenMTC ซึ่งในหัวข้อนี้ มีการดำเนินงาน 3 โปรแกรมตามรูปแบบของการจำลองโครงสร้าง นั่นคือ ContainerID, Searchstring และ LocaionContainer โดยแต่ละโปรแกรมได้ถูกแก้ไขและเพิ่มเติมจากโปรแกรมโครงสร้างการจำลองเบื้องต้นให้สามารถดำเนินงานเป็นไปตามรูปแบบการจำลองโครงสร้างทั้ง 3 รูปแบบ แสดงผลการทดสอบดังรูปที่ 55,รูปที่ 56 และรูปที่ 57



รูปที่ 55 เวลาที่ใช้ในการทำงานของโครงสร้างการจำลองรูปแบบต่างๆในกระบวนการ 1



รูปที่ 56 โอเวอร์เฮดของข้อมูลของโครงสร้างการจำลองรูปแบบต่างๆในกระบวนการ 1

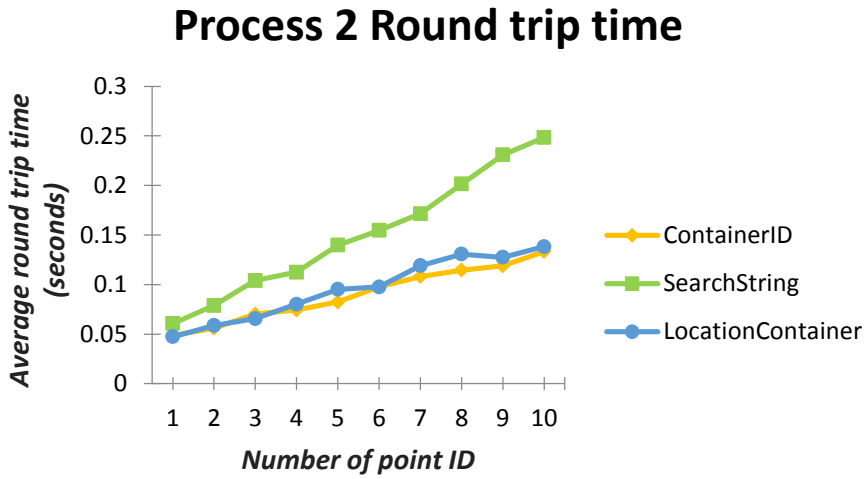


รูปที่ 57 ปริมาณงานของโครงสร้างการจำลองรูปแบบต่างๆในกระบวนการ 1

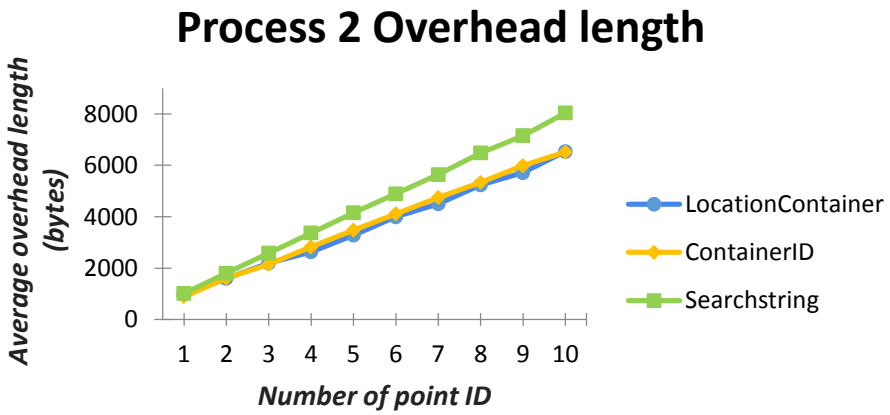
จากรูปผลการทดสอบ จะเห็นว่าในลักษณะการแปลงรูปแบบข้อมูลจากมาตรฐาน IEEE1888 เป็นรูปแบบข้อมูลตามมาตรฐาน ETSI M2M เพื่อนำข้อมูลจากหน่วยเก็บข้อมูลของ CUBEMS ส่งไปเก็บยังหน่วยเก็บข้อมูลของ OpenMTC นั้น รูปแบบการจำลองโครงสร้างแบบ LocationContainer มีค่าเวลาในการทำงาน โอเวอร์เฮดของข้อมูล และค่าปริมาณงานมากที่สุด อันเนื่องมาจากในโปรแกรมตามรูปแบบการจำลองโครงสร้างแบบ LocationContainer ไม่มีฟังก์ชั่นการสร้าง LocationContainer เข้าไปในนหน่วยเก็บข้อมูล จึงต้องมีการเพิ่มฟังก์ชั่นเข้าไปจากโปรแกรมเดิม ในการส่งการร้องขอสร้าง LocationConatiner เข้าไปในหน่วยเก็บข้อมูล ทำให้ต้องใช้เวลาในการทำงานเพิ่มขึ้นและมีค่าโอเวอร์เฮดที่เกิดขึ้นจากการส่งการร้องขอและการได้รับการตอบกลับ โดยที่รูปแบบ ContainerID และ Searchstring สามารถใช้โปรแกรมเดิมในการดำเนินการได้ โดยไม่จำเป็นต้องเพิ่มฟังก์ชั่นใดๆเข้าไป

4.2.3.2 ผลการเปรียบเทียบสมรรถนะของโครงสร้างการจำลองในกระบวนการ 2

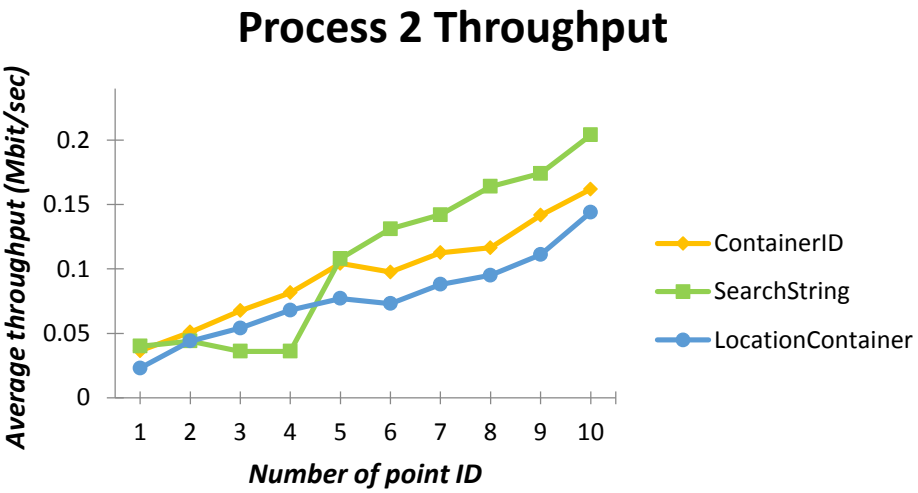
จากการดำเนินการตามกระบวนการ 2 นั่นคือการดำเนินงานโปรแกรมบนเกตเวย์จำลอง เพื่อส่งต่อข้อมูลจากหน่วยเก็บข้อมูลของ OpenMTC ไปเก็บยังหน่วยเก็บข้อมูลของ CUBEMS ซึ่งในหัวข้อนี้ มีการดำเนินงาน 3 โปรแกรมตามรูปแบบของการจำลองโครงสร้าง นั่นคือ ContainerID, Searchstring และ LocaionContainer โดยแต่ละโปรแกรมมีการแก้ไขและเพิ่มเติมจากโปรแกรมโครงสร้างการจำลองเบื้องต้นให้สามารถดำเนินงานเป็นไปตามรูปแบบการจำลองโครงสร้างทั้ง 3 รูปแบบ แสดงผลการทดสอบดังรูปที่ 58,รูปที่ 59 และรูปที่ 60



รูปที่ 58 เวลาที่ใช้ในการทำงานของโครงสร้างการจำลองแบบต่างๆในกระบวนการ 1



รูปที่ 59 โอเวอร์เฮดของข้อมูลของโครงสร้างการจำลองแบบต่างๆในกระบวนการ 2



รูปที่ 60 ปริมาณงานของโครงสร้างการจำลองแบบต่างๆในกระบวนการ 2

จากรูปผลการทดสอบ จะเห็นว่าในลักษณะการแปลงรูปแบบข้อมูลจากมาตรฐาน ETSI M2M เป็นรูปแบบข้อมูลตามมาตรฐาน IEEE1888 เพื่อนำข้อมูลจากหน่วยเก็บข้อมูลของ OpenMTC ส่งไปเก็บยังหน่วยเก็บข้อมูลของ CUBEMS นั้น รูปแบบการจำลองโครงสร้างแบบ Searchstring มีค่าเวลาในการทำงาน และโอเวอร์เฮดที่เกิดขึ้น และปริมาณมากที่สุด อันเนื่องมาจากในโปรแกรมตามรูปแบบการจำลองโครงสร้างแบบ Searchstring มีการเพิ่มฟังก์ชั่นเข้าไปจากโปรแกรมเดิม ในการส่งการร้องขอในการดึงข้อมูลในหัวข้อ Searchstring จากหน่วยเก็บข้อมูล ทำให้ต้องใช้เวลาในการทำงานเพิ่มขึ้นและมีค่าโอเวอร์เฮดที่เกิดขึ้นจากการส่งการร้องขอและการได้รับการตอบกลับ โดยที่รูปแบบ ContainerID และ LocationContainer สามารถใช้โปรแกรมเดิมในการดำเนินการเข้าถึงข้อมูลในหัวข้อดังกล่าวได้ โดยไม่จำเป็นต้องเพิ่มฟังก์ชั่นใดๆเข้าไป

4.3 สรุป

จากผลการทดสอบข้างต้น สามารถสรุปตามหัวข้อการทดสอบได้ดังนี้

1. การทดสอบเบื้องต้นเป็นการเปรียบเทียบสมรรถนะของการดำเนินการระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M พบว่า ในการดำเนินการตามมาตรฐาน IEEE1888 ซึ่งคือ โพรโทคอล WRITE และ FETCH ใช้เวลาในการทำงาน ค่าโอเวอร์เฮดที่เกิดขึ้น และค่าปริมาณงานน้อยกว่าการดำเนินการตามมาตรฐาน ETSI M2M ซึ่งคือการดำเนินการ CREATE และ RETRIEVE โดยสามารถวิเคราะห์ได้ว่า เนื่องจากการดำเนินงานตามการดำเนินการตามมาตรฐาน ETSI M2M มีขั้นตอนในการทำงานที่ซับซ้อนมากกว่า เนื่องจากมีลำดับการดำเนินการตามแบบ RESTful Architecture ในการจัดการส่วนประกอบภายในหน่วยเก็บข้อมูลหลายขั้นตอน ทำให้มีการส่งข้อมูลหลายขั้นตอน นั่นคือ มีการส่งการร้องขอข้อมูลและรับการตอบกลับข้อมูลในจำนวนที่มากกว่า ทำให้ในระหว่างการดำเนินงานต้องใช้เวลาในการทำงานมากกว่า มีค่าโอเวอร์เฮดที่เกิดขึ้นในระหว่างการทำงานมากกว่า และมีค่าปริมาณนั่นคือมีความต้องการในการส่งข้อมูลมากกว่า ดังนั้นอาจสรุปได้ว่าการดำเนินการตามมาฐาน IEEE1888 มีสมรรถนะที่ดีกว่าการดำเนินการตามมาตรฐาน ETSI M2M
2. การทดสอบการทำงานของเกตเวย์จำลอง ซึ่งแบ่งการทำงานออกเป็น 2 กระบวนการ พบว่าในการดำเนินงานของกระบวนการ 1 ซึ่งคือ ข้อมูลจะถูกส่งต่อจากหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 ไปยังหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M พร้อมมีการแปลงรูปแบบข้อมูลจาก XML เป็น JSON จะมีค่าเวลาที่ใช้ในการทำงาน ค่าโอเวอร์เฮดของข้อมูลและปริมาณงานมีค่าน้อยกว่าในการดำเนินงานของกระบวนการ 2 ซึ่งคือ ข้อมูลจะถูกส่งต่อจากหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M ไปยังหน่วยเก็บข้อมูลตามมาตรฐาน

IEEE1888 ซึ่งมีการแปลงรูปแบบข้อมูลจาก JSON เป็น XML อาจวิเคราะห์ได้ว่า การแปลงรูปแบบข้อมูลจากมาตรฐาน IEEE1888 ซึ่งคือ XML เป็นรูปแบบข้อมูลตามมาตรฐาน ETSI M2M ซึ่งคือ JSON มีความรวดเร็วในการทำงาน มีโอเวอร์เฮดที่เกิดขึ้นน้อยกว่า และมีปริมาณงานซึ่งส่งผลกระทบต่อระบบโดยรวมน้อยกว่าการแปลงรูปแบบข้อมูลจากมาตรฐาน ETSI M2M ซึ่งคือ JSON เป็นรูปแบบข้อมูลตามมาตรฐาน IEEE1888 ซึ่งคือ XML ดังนั้นอาจสรุปได้ว่า กระบวนการในการดึงข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 แล้วส่งต่อไปบันทึกในหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M โดยมีการแปลงรูปแบบข้อมูลจาก XML เป็น JSON ร่วมอยู่ด้วย มีสมรรถนะในการทำงานดีว่ากระบวนการในการดึงข้อมูลจากหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M แล้วส่งต่อไปบันทึกในหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 โดยมีการแปลงรูปแบบข้อมูลจาก JSON เป็น XML ร่วมอยู่ด้วย

3. การทดสอบสุดท้ายเพื่อเปรียบเทียบโครงสร้างการจำลอง 3 รูปแบบ ซึ่งคือ ContainerID, SearchString และ LocationContainer ในกระบวนการที่แปลงรูปแบบข้อมูลและส่งต่อข้อมูลจากหน่วยเก็บข้อมูลตามตามมาตรฐาน IEEE1888 ไปยังหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M นั้น รูปแบบการจำลองโครงสร้างแบบ LocationContainer ใช้เวลาในการทำงาน มีค่าโอเวอร์เฮดของข้อมูล และค่าปริมาณงานมากที่สุด อันเป็นผลมาจากการเพิ่มฟังก์ชั่นการสร้างหัวข้อ LocationContainer ในหน่วยเก็บข้อมูล ซึ่งในโปรแกรมเบื้องต้นไม่มี และในกระบวนการที่แปลงรูปแบบข้อมูลและส่งต่อข้อมูลจากหน่วยเก็บข้อมูลตามาตรฐาน ETSI M2M ไปยังหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 รูปแบบการจำลองโครงสร้างแบบ SearchString ใช้เวลาในการทำงาน มีค่าโอเวอร์เฮด และค่าปริมาณมากที่สุด เนื่องมากจากมีการเพิ่มฟังก์ชั่นในการเข้าถึงหัวข้อ SearchString ในหน่วยเก็บข้อมูล ซึ่งในโปรแกรมเบื้องต้นไม่มี จะเห็นได้ว่ารูปแบบการจำลองโครงสร้างแบบ ContainerID ใช้เวลาในการทำงาน มีค่าโอเวอร์เฮดที่เกิดขึ้น และปริมาณงานในระหว่างการดำเนินงานน้อยกว่ารูปแบบอื่นๆ จึงอาจสรุปผลการทดสอบสมรรถนะของรูปแบบการจำลองโครงสร้างทั้ง 3 รูปแบบในการดำเนินงานตามกระบวนการของเกตเวย์จำลอง ได้ว่า รูปแบบการจำลองโครงสร้างแบบ ContainerID ในปริมาณจุดข้อมูลที่มากขึ้น มีแนวโน้มที่จะส่งผลกระทบต่อการทำงานน้อยที่สุด จึงมีความเหมาะสมมากที่สุดที่จะใช้ในการดำเนินการตามกระบวนการทั้ง 2 ของเกตเวย์จำลองที่ถูกออกแบบให้สามารถแลกเปลี่ยนข้อมูลระหว่างหน่วยเก็บข้อมูลตามมาตรฐาน IEEE1888 และหน่วยเก็บข้อมูลตามมาตรฐาน ETSI M2M ซึ่งมีรูปแบบการดำเนินงาน โพรโทคอลที่ใช้ในการสื่อสาร และรูปแบบข้อมูลที่แตกต่างกัน ดังที่ได้กล่าวไว้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

ในวิทยานิพนธ์ฉบับนี้ได้พัฒนาโพรโทคอลของเกตเวย์ในการทำงานร่วมกันระหว่างมาตรฐาน IEEE1888 กับมาตรฐาน ETSI M2M เพื่อใช้ในการแลกเปลี่ยนข้อมูลระหว่างหน่วยเก็บข้อมูลของทั้งสองมาตรฐาน โดยมีการแปลงรูปแบบข้อมูลและโพรโทคอลในการสื่อสารให้สอดคล้องกับทั้งสองมาตรฐานนี้ ในงานวิทยานิพนธ์ได้อธิบายการออกแบบการดำเนินการพัฒนาโพรโทคอลของเกตเวย์ที่ทำให้เกิดการทำงานร่วมกันระหว่างสองมาตรฐาน โดยในวิทยานิพนธ์นี้ใช้ชื่อเรียกเกตเวย์ดังกล่าวว่า เกตเวย์จำลอง ซึ่งในการออกแบบนั้น ได้ออกแบบเกตเวย์จำลองให้สามารถทำงานร่วมกับส่วนประกอบของทั้งสองมาตรฐานที่ทำหน้าที่เชื่อมต่อกับมาตรฐานอื่นได้ โดยใช้แพลตฟอร์มการเขียนโปรแกรมแบบ Node.js ด้วยภาษา JavaScript ซึ่งในระหว่างการส่งผ่านข้อมูลจะมีการแปลงรูปแบบข้อมูลระหว่างรูปแบบ XML กับรูปแบบ JSON ด้วย พร้อมกับมีการแปลงรูปแบบการดำเนินการระหว่างรูปแบบ SOAP กับรูปแบบ RESTful architecture ด้วย

จากผลการทดสอบสมรรถนะของเกตเวย์จำลองด้วยการตรวจวัดเวลาที่ใช้ในการทำงาน ค่าโอเวอร์เฮดของข้อมูล และค่าปริมาณงาน ทำให้ทราบว่า ในกระบวนการส่งผ่านข้อมูลจากมาตรฐาน IEEE1888 ไปยังมาตรฐาน ETSI M2M ใช้เวลาในการทำงาน มีโอเวอร์เฮดของข้อมูลที่เกิดขึ้น และปริมาณงาน น้อยกว่ากระบวนการส่งผ่านข้อมูลจากมาตรฐาน ETSI M2M ไปยังมาตรฐาน IEEE1888 รวมไปถึงการระบุตำแหน่งข้อมูลของทั้งสองมาตรฐานมีวิธีการระบุที่แตกต่างกัน จึงต้องทำการออกแบบโครงสร้างการจำลองให้สอดคล้องกับการระบุตำแหน่งข้อมูลของทั้งสองมาตรฐาน และจากการศึกษาการระบุตำแหน่งของข้อมูลของทั้งสองมาตรฐานทำให้สามารถออกแบบรูปแบบการส่งต่อการระบุตำแหน่งของข้อมูลได้ 3 รูปแบบ คือ ContainerID, SearchString และ LocationContainer โดยจากผลการทดสอบสมรรถนะของทั้ง 3 รูปแบบ แต่ละรูปแบบมีความเหมาะสมที่จะนำไปใช้งานที่แตกต่างกัน ซึ่งรูปแบบการจำลองโครงสร้างที่เหมาะสมกับการดำเนินการของเกตเวย์จำลองตามที่กล่าวในวิทยานิพนธ์ฉบับนี้คือ รูปแบบ ContainerID

5.2 ข้อเสนอแนะ

1. ในวิทยานิพนธ์ฉบับนี้ ได้นำเสนอวิธีการเมื่อทำการดำเนินงานโปรแกรมเพื่อแลกเปลี่ยนข้อมูลระหว่างหน่วยเก็บข้อมูลเบื้องต้น ยังไม่สามารถทำงานแบบเวลาจริง (real time) ได้ ซึ่งสามารถออกแบบการส่งผ่านข้อมูลระหว่างมาตรฐานให้ทำงานแบบเวลาจริงได้โดยการใช้โพรโทคอล TRAP ตามมาตรฐาน IEEE1888 แทนที่โพรโทคอล FETCH ซึ่งสามารถกำหนดให้มีการส่งต่อข้อมูลเมื่อมีการอัพเดตข้อมูลในหน่วยเก็บข้อมูลได้
2. ในวิทยานิพนธ์ฉบับนี้ เป็นการแลกเปลี่ยนข้อมูลระหว่างสองมาตรฐานในระหว่างรูปแบบข้อมูล XML และ JSON ซึ่งในงานวิจัยต่อไป สามารถออกแบให้มีการแลกเปลี่ยนระหว่างรูปแบบข้อมูลอื่นๆ เช่น Fast Infoset EXI เป็นต้น
3. ในวิทยานิพนธ์ฉบับนี้ เป็นการแลกเปลี่ยนข้อมูลตัวรับรู้ระหว่างหน่วยเก็บข้อมูลของทั้งสองมาตรฐาน ซึ่งในงานวิจัยต่อไป สามารถออกแบบการส่งต่อข้อมูลระหว่างข้อมูลตัวรับรู้ของมาตรฐานหนึ่งให้สามารถส่งคำสั่งการไปยังอีกมาตรฐานหนึ่งได้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รายการอ้างอิง

[1] N. Buonaccorsi, C. Cicconetti, R. Mambrini, N. Podias, and P. Russell, "ETSI M2M release 1 demonstration," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, 2012, pp. 1-3.

[2] M. Sneps-Sneppe and D. Namiot, "About M2M standards and their possible extensions," in *Future Internet Communications (BCFIC), 2012 2nd Baltic Congress on*, 2012, pp. 187-193.

[3] "IEEE Standard for Ubiquitous Green Community Control Network Protocol," *IEEE Std 1888-2011,* pp. 1-65, 2011.

[4] C. Ninagawa, H. Yoshida, S. Kondo, and H. Otake, "Data transmission of IEEE1888 communication for wide-area real-time smart grid applications," in *Renewable and Sustainable Energy Conference (IRSEC), 2013 International*, 2013, pp. 509-514.

[5] "ETSI TS Machine-to-Machine communications (M2M); Functional architecture," *ETSI TS 102.690V1.1.1,* 2011.

[6] Hideya Ochiai and H. Esaki, "Toward Open Factility Networking: Semantics Management for Higher-Level Interoperability," 2009.

[7] M. Corici, H. Coskun, A. Elmangoush, A. Kurniawan, M. Tong, T. Magedanz*, et al.*, "OpenMTC: Prototyping Machine Type communication in carrier grade operator networks," in *Globecom Workshops (GC Wkshps), 2012 IEEE*, 2012, pp. 1735-1740.

[8] A. Elmangoush, H. Coskun, S. Wahle, and T. Magedanz, "Design aspects for a reference M2M communication platform for Smart Cities," in *Innovations in Information Technology (IIT), 2013 9th International Conference on*, 2013, pp. 204-209.

[9] A. Elmangoush, T. Magedanz, A. Blotny, and N. Blum, "Design of RESTful APIs for M2M services," in *Intelligence in Next Generation Networks (ICIN), 2012 16th International Conference on*, 2012, pp. 50-56.

[10] C. Bormann, A. P. Castellani, and Z. Shelby, "CoAP: An Application Protocol for Billions of Tiny Internet Nodes," *Internet Computing, IEEE,* vol. 16, pp. 62-67, 2012.

[11] Teerapan Klinpratum, Chaiyachet Saivichit, Asma Elmangoush, and T. Magedanz, "Toward Interconnecting M2M/IoT Standards: Interworking Proxy for IEEE1888 Standard at ETSI M2M Platforms," 2014.

[12] Tankorn Inthasut and C. Aswakul, "Development and Reliability Testing of IEEE1888 Gateway for ZigBee Wireless Sensor Network in Chulalongkorn University's Building Energy Management System," 2014.

[13] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs," *Internet Computing, IEEE,* vol. 14, pp. 80-83, 2010.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ประวัติผู้เขียนวิทยานิพนธ์

นายธีรพันธุ์ กลิ่นประทุม เกิดเมื่อวันที่ 25 มิถุนายน พ.ศ. 2531 จังหวัดสิงห์บุรี เป็นบุตรของนายประเทือง กลิ่นประทุม และนางลักขณา กลิ่นประทุม สำเร็จการศึกษาระดับปริญญาตรี หลักสูตรวิทยาศาสตรบัณฑิต สาขาฟิสิกส์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2554 ด้วยโครงการการออกแบบลักษณะระนาบของนาฬิกาแดดตามสถานที่ต่าง ๆ ด้วยค่าละติจูดและค่าลองจิจูด โดยการเขียนโปรแกรมด้วยภาษา C++ และมีความสนใจในเรื่องระบบจัดการพลังงานแบบใหม่ซึ่งคือ โครงข่ายไฟฟ้าอัจฉริยะ (Smart grid) จึงเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า สาขาไฟฟ้าสื่อสาร ในหน่วยปฏิบัติการวิจัยโครงข่ายไฟฟ้าอัจฉริยะและพลังงานทดแทน (Smart grid and Renewable energy) ในปีการศึกษาถัดมา ณ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย สังกัดห้องปฏิบัติการวิจัยโทรคมนาคม

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University