



รายงานวิจัยฉบับสมบูรณ์

โครงการ

การออกแบบเกณฑ์วิธีเพื่อจัดเส้นทางด้วยอินเทอร์เน็ตโพรโทคอลรุ่นที่ 6
สำหรับการสื่อสารแบบยูนิคาสต์ มัลติคาสต์และแอนนิคาสต์ของเครือข่ายเซ็นเซอร์ไร้สาย

A Design Routing Algorithm for Unicast, Multicast and Anycast Communication in
Internet Protocol Version 6 (IPv6) of Wireless Sensor Networks (WSN)

โดย ผู้ช่วยศาสตราจารย์ ดร. อรรถพ หมั่นสกุล

TNI

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีไทย-ญี่ปุ่น

มีนาคม 2560

รายงานวิจัยฉบับสมบูรณ์

โครงการ

การออกแบบเกณฑ์วิธีเพื่อจัดเส้นทางด้วยอินเทอร์เน็ตโพรโทคอลรุ่นที่ 6
สำหรับการสื่อสารแบบยูนิคาสต์ มัลติคาสต์และแอนนิคาสต์ของเครือข่ายเซ็นเซอร์ไร้สาย

A Design Routing Algorithm for Unicast, Multicast and Anycast Communication in
Internet Protocol Version 6 (IPv6) of Wireless Sensor Networks (WSN)

โดย ผู้ช่วยศาสตราจารย์ ดร. อรรถพล หมั่นสกุล

TNI

สนับสนุนโดยทุนอุดหนุนการวิจัยของสถาบันเทคโนโลยีไทย-ญี่ปุ่น

บทคัดย่อ

สัญญาเลขที่: 1603/A022

ชื่อโครงการ: การออกแบบเกณฑ์วิธีเพื่อจัดเส้นทางด้วยอินเทอร์เน็ตโพรโตคอลรุ่นที่ 6 สำหรับการสื่อสารแบบ
ยูนิคาสต์ มัลติคาสต์และแอนนิกาสต์ของเครือข่ายเซ็นเซอร์ไร้สาย

ชื่อนักวิจัย: ผู้ช่วยศาสตราจารย์ ดร.อรรณพ หมั่นสกุล

อีเมล: annop@tni.ac.th

ระยะเวลาโครงการ: 12 เดือน

ปัจจุบันยังไม่มีอัลกอริทึมเพื่อจัดเส้นทางที่สามารถรองรับการสื่อสารทั้งแบบยูนิคาสต์ มัลติคาสต์และแอนนิกาสต์ของเครือข่ายเซ็นเซอร์ไร้สาย ทั้งสามารถระบุหมายเลขด้วยอินเทอร์เน็ตโพรโตคอลรุ่นที่ 6 เพื่อส่งข้อมูลเข้าสู่เครือข่ายอินเทอร์เน็ต ดังนั้น ผู้วิจัยจึงได้เสนอ M-RPL อัลกอริทึม เพื่อให้การสื่อสารระหว่างเซ็นเซอร์ไร้สายสามารถส่งข้อมูลที่มีความแตกต่างกันภายในอัลกอริทึมเดียวกัน โดยทำการออกแบบพร้อมการทดสอบทั้งบนโปรแกรมจำลองและบนอุปกรณ์จริง

และงานวิจัยนี้ได้นำเทคโนโลยี IoT6 (Internet of Things with IPv6) เพื่อออกแบบและประยุกต์การใช้งานภายในที่อยู่อาศัยที่สามารถรักษาความปลอดภัยและแจ้งเตือนอัคคีภัย โดยได้นำอุปกรณ์ที่ทันสมัยในปัจจุบัน ได้แก่ อุปกรณ์เซ็นเซอร์ไร้สายรุ่น Zolertia Z1 อุปกรณ์ Raspberry Pi 3 อุปกรณ์กล้อง Pi NoIR อุปกรณ์เซ็นเซอร์เปลวไฟ อุปกรณ์เซ็นเซอร์แก๊ส และอุปกรณ์เซ็นเซอร์ตรวจจับความเคลื่อนไหว โดยอุปกรณ์เซ็นเซอร์ไร้สายรุ่น Zolertia Z1 จะมีการอ้างอิงตำแหน่งด้วย IPv6 ซึ่งถือว่าเป็นโพรโตคอลรุ่นใหม่ที่มีข้อดีหลายประการและสามารถรับส่งข้อมูลของอุปกรณ์เซ็นเซอร์ต่างๆ ผ่านอุปกรณ์ Raspberry Pi 3 เพื่อส่งต่อไปยังคลาวด์แพลตฟอร์ม (ที่มีชื่อว่า Ubidots) โดยเมื่อเกิดเหตุฉุกเฉิน ไฟไหม้ แก๊สรั่ว หรือตรวจพบผู้บุกรุก ระบบจะทำการถ่ายภาพและอัปโหลดภาพขึ้นไปยัง Amazon Simple Storage Services เพื่อทำการแจ้งเตือนไปยังผู้ใช้งานผ่านอีเมลหรือข้อความสั้นได้ทันที

กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณทุนอุดหนุนการวิจัยของสถาบันเทคโนโลยีไทย-ญี่ปุ่น ให้การสนับสนุนเงินทุนการวิจัยตลอดโครงการวิจัย ขอขอบพระคุณ รองศาสตราจารย์ ดร.รัตติกกร วรากุลศิริพันธุ์ ที่ให้คำปรึกษาในการทำวิจัยมาโดยตลอด และขอขอบคุณ ผู้ช่วยศาสตราจารย์ตรีรัตน์ เมตต์การุณจิต ที่ให้คำปรึกษาด้านเทคนิค รวมถึงขอขอบคุณนักศึกษา นายสุภโชค เกียรติกิติกุลและนายชรัตน์ ขำแสง ที่ช่วยปฏิบัติการทดลองการวิจัยครั้งนี้จนสำเร็จลุล่วงด้วยดี

สุดท้ายนี้ขอขอบคุณฝ่ายวิจัยและบริการวิชาการ สถาบันเทคโนโลยีไทย-ญี่ปุ่น ทุกท่านที่ให้คำแนะนำที่ดีเสมอมา

ผู้ช่วยศาสตราจารย์ ดร.อรรณพ หมั่นสกุล

หัวหน้าโครงการวิจัย

มีนาคม 2560

TNI

ผลงานจากโครงการวิจัยที่ได้รับทุนอุดหนุนการวิจัยของสถาบันเทคโนโลยีไทย-ญี่ปุ่น

1. ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติ

Annop Monsakul. "M-RPL: A Design Algorithm for WSNs with Mixed Traffic." *Journal of Advances in Computer Networks*, Vol. 4, No. 2, pp. 139-142, June 2016

2. การนำผลงานวิจัยไปใช้ประโยชน์

เชิงวิชาการ (มีการพัฒนาการเรียนการสอน/สร้างนักวิจัยรุ่นใหม่)

3. อื่นๆ (ผลงานตีพิมพ์ในที่ประชุมวิชาการ TNIAC2017)

ศุภโชค เกียรติกิตติกุล ชรัตน์ จำแสง และอรรณพ หมั่นสกุล. "ระบบความปลอดภัยบ้านอัจฉริยะด้วยอินเทอร์เน็ตสำหรับทุกสิ่งก้าวสู่อินเทอร์เน็ตยุคหน้าที่รองรับอินเทอร์เน็ตโพรโตคอลรุ่นที่หก."
ประชุมวิชาการระดับชาติ The TNI Academic Conference 2017 ครั้งที่ 4, พฤษภาคม 2560

สารบัญ

บทคัดย่อ	ข
กิตติกรรมประกาศ	ค
ผลงานจากโครงการวิจัยที่ได้รับทุนอุดหนุนการวิจัยของสถาบันเทคโนโลยีไทย-ญี่ปุ่น	ง
สารบัญ.....	จ
สารบัญรูป.....	ฉ
สารบัญตาราง.....	ฉ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์การศึกษา.....	3
1.3 ขอบเขตของการศึกษา	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	4
บทที่ 2 ทฤษฎีและเทคโนโลยีที่ใช้ในการวิจัย.....	5
2.1 อินเทอร์เน็ตสำหรับทุกสิ่งด้วยไอพีรุ่นที่หก.....	5
2.2 เครือข่ายเซ็นเซอร์ไร้สาย	6
2.3 ไอพีรุ่นที่หก.....	6
2.3.1 แนวคิดของไอพีรุ่นที่หก.....	6
2.3.2 แพ็กเก็ตไอพีรุ่นที่หก	7
2.3.3 ที่อยู่ไอพี	8
2.3.4 กฎของไอพีรุ่นที่หก.....	9
2.3.5 ใช้ไอพีรุ่นที่หกเพื่ออะไร	11
2.3.6 ข้อดีของ ไอพีรุ่นที่หก เมื่อทำงานร่วมกับ IoT	13
2.4 เครือข่ายส่วนบุคคลไร้สายที่ใช้พลังงานต่ำสำหรับไอพีรุ่นที่หก.....	15
2.4.1 ภาพรวมของ LoWPAN	17
2.4.2 การใช้ไอพีบน 6LoWPAN	18
2.4.3 ขั้นตอนการทำงานของ 6LoWPAN.....	20
2.5 ระบบปฏิบัติการ Contiki	20
2.5.1 การติดตั้ง Contiki	21

2.5.2 เครื่องมือจำลอง Instant contiki.....	21
2.6 Zolertia Z1.....	23
2.7 เซ็นเซอร์	24
2.7.1 อนาล็อกเซ็นเซอร์	24
2.7.2 ดิจิทัลเซ็นเซอร์	25
2.8 โพรโทคอลหาเส้นทางสำหรับอุปกรณ์พลังงานต่ำและลดทอนเครือข่าย	26
2.9 IPv6 Tunnel Broker.....	30
2.10 Ubidots	31
2.11 อุปกรณ์ Zolertia Z1	32
2.12 อุปกรณ์ Raspberry Pi	33
2.13 เว็บไซต์ TunnelBroker.net	34
2.14 ไอพีรุ่นที่ 4.....	34
2.15 ไอพีรุ่นที่ 6.....	35
2.16 Ubodot Cloud Platform	38
2.17 กล้อง Raspberry Pi NoIR Camera V2	39
2.18 เซ็นเซอร์ตรวจจับความเคลื่อนไหว	40
2.19 ระบบปฏิบัติการ Contiki OS	42
2.20 คลาวด์แพลตฟอร์ม Amazon Simple Storage Service.....	44
2.21 ชุดพัฒนาซอฟต์แวร์ Boto3.....	44
2.22 ชุดอุปกรณ์เซ็นเซอร์ไร้สายที่ใช้ในการวิจัย.....	45
บทที่ 3 การออกแบบและดำเนินงานวิจัย.....	49
3.1 ออกแบบอัลกอริทึมเพื่อจัดเส้นทาง	49
3.1.1 IPv6 Routing Protocol for LLNs (RPL) 1.....	49
3.1.2 IPv6 Routing Protocol for LLNs (RPL) 2.....	50
3.1.3 Low power and Lossy Networks (LLNs).....	50
3.1.4 Routing over Low power and Lossy networks (ROLL)	51
3.2 ขั้นตอนการดำเนินงาน.....	52
3.2.1 ติดตั้ง VMware และ Contiki.....	52

3.2.2	ทำ Tunnel จากไอพีรุ่นที่สี่เพื่อใช้ไอพีรุ่นที่หก.....	53
3.2.3	สมัครใช้งานและตั้งค่า Ubidots เพื่อรับค่าและแจ้งเตือน	53
3.2.4	ติดตั้งโปรแกรมบนอุปกรณ์ด้วยโปรแกรม Border Router และ Ubidots client บน z1 ที่ต่อกับ เซ็นเซอร์แก๊สและเซ็นเซอร์ไฟ.....	54
3.2.5	จุดไฟแช็คให้ติดไฟหรือแก๊สเพื่อทดลองกับเซ็นเซอร์	54
3.2.6	Ubidots แจ้งเตือนมายังโทรศัพท์ของผู้ใช้งาน	55
3.3	ขั้นตอนการทำงาน	55
3.3.1	การตั้งค่าจาก Ubidot Clients (Z1) เพื่อส่งข้อมูลไปยัง Border Router (Z1).....	56
3.3.2	การส่งข้อมูลจาก Border Router (Z1) ไปยัง Raspberry pi.....	56
3.3.3	การตั้งค่าบน Raspberry pi เพื่อส่งต่อข้อมูลไปยังเราเตอร์	56
3.3.4	การตั้งค่าบนเราเตอร์ เพื่อส่งข้อมูลไปยัง Tunnel Broker	59
3.3.5	จาก Tunnel broker ไปยัง Ubidots	59
3.3.6	จาก Ubidots แจ้งเตือนไปยังผู้ใช้งาน	59
3.3.7	การตั้งค่าบนอุปกรณ์ Zolertia Z1	59
3.3.8	การตั้งค่าบนอุปกรณ์ Raspberry Pi.....	62
3.3.9	การตั้งค่าบนเว็บไซต์ Ubidots.....	69
3.3.10	การตั้งค่าบนเว็บไซต์ Amazon Simple Storage Services	72
บทที่ 4	วิเคราะห์ผลการวิจัย.....	75
4.1	ประสิทธิภาพของ M-RPL อัลกอริทึม	75
4.1.1	ตัวอย่างการออกแบบ	75
4.1.2	M-RPL Algorithm “Pseudo Code”	75
4.1.3	กราฟที่ได้จากการทดสอบ	76
4.1.4	เปรียบเทียบค่าน้ำหนักในการจัดเส้นทาง.....	76
4.2	ผลการวิจัยผ่านอุปกรณ์เซ็นเซอร์ไร้สาย กรณีที่ 1	78
4.2.1	ห้องครัวสำหรับอาหารมังสวิรัตเกิดแก๊สรั่ว	79
4.2.2	ห้องครัวสำหรับอาหารทั่วไปตรวจพบไฟ	80
4.2.3	ตรวจสอบค่าความชื้นและอุณหภูมิในร้านอาหารข้างบ้าน	81
4.3	ผลการวิจัยผ่านอุปกรณ์เซ็นเซอร์ไร้สาย กรณีที่ 2	82
4.3.1	การติดตั้งในสถานที่จริง	82

4.3.2 สามารถแจ้งเตือนผ่านอีเมลล์และข้อความมือถือ	83
4.3.3 สามารถอัปโหลดรูปถ่ายขึ้น Amazon Simple Storage Services	85
4.3.4 แสดงการถ่ายรูปของกล้อง Raspberry Pi NoIR Camera	86
บทที่ 5 บทสรุปและข้อเสนอแนะ	87
5.1 สรุปผลการดำเนินงานวิจัย	87
5.2 ปัญหา	87
5.2.1 ไอพีรุ่นที่สี่แบบสาธารณะ	87
5.2.2 Z1 ต้องเชื่อมต่อ Micro USB	88
5.2.3 ปัญหาที่ Ubidots	88
5.3 ข้อเสนอแนะ	88
5.4 แผนการทำงานในอนาคต	88
5.4.1 พัฒนากลาวด์แพลตฟอร์มของตัวเอง	88
5.4.2 เพิ่มเซ็นเซอร์	88
เอกสารอ้างอิง	89
ภาคผนวก	91

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

สารบัญรูป

รูป

หน้า

รูปที่ 1.1 เครือข่ายของ Wireless Sensor Networks (WSNs) บน เทคโนโลยี IoT6.....	2
รูปที่ 1.2 เครือข่ายเซ็นเซอร์ไร้สายที่ออกแบบสำหรับงานวิจัยโครงการนี้	3
รูปที่ 2.1 วิวัฒนาการของการเชื่อมต่ออินเทอร์เน็ตในอนาคต (ที่มา: Cisco 2011).....	5
รูปที่ 2.2 ชั้นของอินเทอร์เน็ตโพรโทคอล	6
รูปที่ 2.3 ส่วนหัวของไอพีรุ่นที่หก	8
รูปที่ 2.4 ที่อยู่ไอพีรุ่นที่หก	9
รูปที่ 2.5 เน็ตเวิร์คไอดีและอินเทอร์เน็ตเฟสไอดี	11
รูปที่ 2.6 การแลกเปลี่ยนแพ็กเก็ตในไอพีรุ่นที่หก	12
รูปที่ 2.7 ชั้นการทำงานของ 6LoWPAN	20
รูปที่ 2.8 IoT workshop VM.....	22
รูปที่ 2.9 โครงสร้าง Contiki.....	22
รูปที่ 2.10 Z1 motes.....	23
รูปที่ 2.11 เซ็นเซอร์ไฟ	25
รูปที่ 2.12 เซ็นเซอร์แก๊ส.....	25
รูปที่ 2.13 DAG	26
รูปที่ 2.14 DODAG	26
รูปที่ 2.15 Two RPL Instances	27
รูปที่ 2.16 Floating และ Grounded.....	27
รูปที่ 2.17 Control Message	29
รูปที่ 2.18 รูปแบบ DODAG.....	29
รูปที่ 2.19 รูปแบบ DODAG (2)	30
รูปที่ 2.20 ตัวอย่างการทำงาน	31
รูปที่ 2.21 อุปกรณ์ Zolertia Z1.....	32
รูปที่ 2.22 อุปกรณ์ Raspberry Pi.....	33
รูปที่ 2.23 โลโก้เว็บไซต์ tunnelbroker.net	34

รูปที่ 2.24 สแต็กของอินเทอร์เน็ตโปรโตคอล.....	36
รูปที่ 2.25 Network Diagram 6in4 Tunneling	37
รูปที่ 2.26 โลโก้ Ubidots	38
รูปที่ 2.27 กล่อง Raspberry Pi NoIR Camera v2.....	39
รูปที่ 2.28 PIR Motion Sensor	40
รูปที่ 2.29 การทำงานของ PIR Motion Sensor	41
รูปที่ 2.30 การทำงานของ PIR motion sensor เมื่อมีรังสีอินฟราเรดเปลี่ยนแปลง	41
รูปที่ 2.31 ระบบปฏิบัติการ Contiki OS	42
รูปที่ 2.32 โลโก้ Amazon Simple Storage Services	44
รูปที่ 2.33 ขั้นตอนการทำงานของ Amazon S3	44
รูปที่ 2.34 ภาพรวมของอุปกรณ์ที่อยู่ในโครงการวิจัย	45
รูปที่ 2.35 อุปกรณ์ Border Router และ Raspberry pi	46
รูปที่ 2.36 อุปกรณ์เซ็นเซอร์ไร้สาย Z1 และอุปกรณ์เซ็นเซอร์ต่อพ่วงต่าง ๆ.....	46
รูปที่ 3.1 PRL Node Rank เพื่อกำหนดค่า Rank สัมพันธ์กับ DODAG root	49
รูปที่ 3.2 DODAG Example กลุ่มของ parent node และโหนดไม่รับรู้ child node	49
รูปที่ 3.3 DODAG Repair.....	50
รูปที่ 3.4 IoT Workshop และ VMware	52
รูปที่ 3.5 Tunnel Broker	53
รูปที่ 3.6 Ubidots	53
รูปที่ 3.7 ทดสอบเซ็นเซอร์ไฟ	54
รูปที่ 3.8 ทดสอบเซ็นเซอร์แก๊ส.....	55
รูปที่ 3.9 ขั้นตอนการทำงาน.....	55
รูปที่ 3.10 หน้าต่างที่ปรากฏหลังใช้คำสั่ง make connect-router.....	60
รูปที่ 3.11 หน้าต่างหลังแก้ไขไฟล์ project-conf.h	61
รูปที่ 3.12 การทำงานตั้งแต่อุปกรณ์ Raspberry Pi จนถึงเซิร์ฟเวอร์ของ tunnelbroker.net.....	62
รูปที่ 3.13 หน้าต่างการแก้ไขไฟล์ sysctl.conf	63
รูปที่ 3.14 หน้าต่างหลังสร้างอุโมงค์ที่เว็บไซต์ tunnelbroker.net สำเร็จ.....	66
รูปที่ 3.15 หน้าต่างการตั้งค่าของอุปกรณ์ Raspberry Pi	67
รูปที่ 3.16 หน้าต่างการตั้งค่าของ AWS CLI.....	68

รูปที่ 3.17 หน้าต่างการสร้างตัวแปรเพื่อรับค่าจากเซ็นเซอร์	70
รูปที่ 3.18 หน้าต่างการตั้งค่า Events ในเว็บไซต์ Ubidots	71
รูปที่ 3.19 หน้าต่างเมื่อสร้าง Events สำเร็จ	72
รูปที่ 3.20 ขั้นตอนการสร้าง Bucket	73
รูปที่ 3.21 หน้าต่างการขอ Access Key และ Secret Access Key	74
รูปที่ 4.1 กราฟเส้นทางของการสื่อสารภายในเครือข่ายเซ็นเซอร์ไร้สาย	76
รูปที่ 4.2 กรณีสึกษาสถานการณ์จำลองระบบแจ้งเตือนอัคคีภัยในครัวเรือน	78
รูปที่ 4.3 ตารางการหาเส้นทางของ Border Router	79
รูปที่ 4.4 เมลแจ้งเตือนจาก Ubidots	80
รูปที่ 4.5 ข้อความแจ้งเตือนจาก Ubidots	80
รูปที่ 4.6 การแสดงข้อมูลบน Ubidots	81
รูปที่ 4.7 ผังการติดตั้งอุปกรณ์ภายในบ้าน	82
รูปที่ 4.8 หน้าต่างแสดงตารางการเชื่อมต่ออุปกรณ์ผ่าน RPL โพรโทคอล	83
รูปที่ 4.9 ข้อความมือถือที่ถูกส่งมาเมื่อมีการตรวจจับความเคลื่อนไหว	84
รูปที่ 4.10 อีเมลที่ถูกส่งมาเมื่อมีการตรวจจับความเคลื่อนไหว	84
รูปที่ 4.11 รูปถ่ายที่ถูกอัปโหลดขึ้นไป Amazon Simple Storage Services	85
รูปที่ 4.12 ภาพที่ถูกถ่ายหลังจาก Motion2 ตรวจจับความเคลื่อนไหวได้	86
รูปที่ 4.13 ภาพที่ถูกถ่ายหลังจาก Motion1 ตรวจจับความเคลื่อนไหวได้	86

TNI

THAI-NICHI INSTITUTE OF TECHNOLOGY

สารบัญตาราง

ตาราง

หน้า

ตารางที่ 2.1 ไอพีแอดเดรสและโฮสในแต่ละคลาส	35
ตารางที่ 2.2 ตารางโครงสร้างของระบบปฏิบัติการ Contiki OS.....	43
ตารางที่ 3.1 จากการประเมินโปรโตคอลที่มีอยู่แล้ว.....	51
ตารางที่ 4.1 คำนวณน้ำหนักในการจัดเส้นทางมีการสื่อสารแบบยูนีคาสต์ที่เข้าออกภายในโหนดที่ 64 KBPS.....	77
ตารางที่ 4.2 คำนวณน้ำหนักในการจัดเส้นทางมีการสื่อสารแบบยูนีคาสต์ที่เข้าออกภายในโหนดที่ 128 KBPS...	77
ตารางที่ 4.3 คำนวณน้ำหนักในการจัดเส้นทางมีการสื่อสารแบบยูนีคาสต์ที่เข้าออกภายในโหนดที่ 256 KBPS...	77
ตารางที่ 4.4 คำนวณน้ำหนักในการจัดเส้นทางมีการสื่อสารแบบยูนีคาสต์ที่เข้าออกภายในโหนดที่ 512 KBPS...	77
ตารางที่ 4.5 คำนวณน้ำหนักในการจัดเส้นทางมีการสื่อสารแบบยูนีคาสต์ที่เข้าออกภายในโหนดที่ 1024 KBPS.	77

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

บทที่ 1

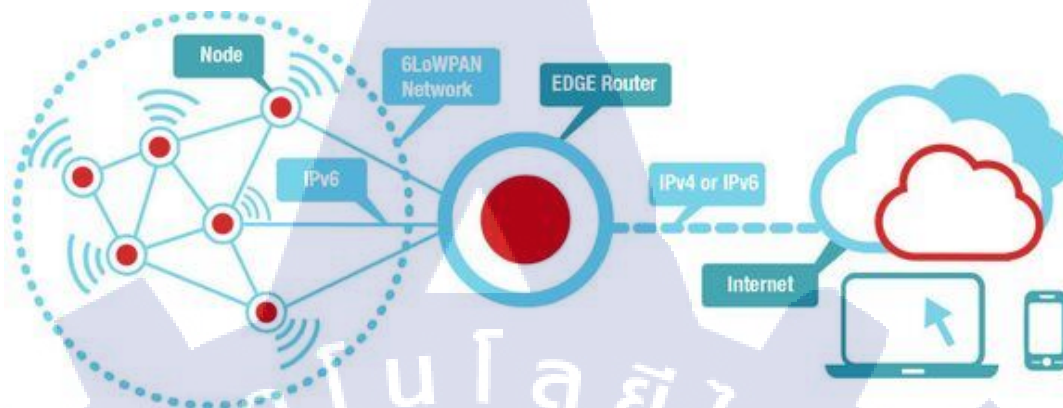
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

Internet of Things (IoT) หรือการเชื่อมต่อของสรรพสิ่ง คือการที่สิ่งของทั้งหลายสามารถเชื่อมต่อกันเป็นเครือข่ายและสามารถสื่อสารกันได้ บางครั้งรู้จักกันในชื่อของ Machine-to-Machine Communication หรือ Internet of Everything หรือ Embedded Internet หรือ Ubiquitous Computing เป็นต้น โดยมีแนวคิดที่ว่าสภาพแวดล้อมต่างๆ ประกอบด้วยทุกสรรพสิ่งที่สามารถสื่อสารและเชื่อมต่อกันได้ผ่านโพรโทคอลการสื่อสารทั้งแบบใช้สายและไร้สาย สรรพสิ่งต่างๆ สามารถระบุตัวตนได้ สามารถมีปฏิสัมพันธ์โต้ตอบและทำงานร่วมกันได้ เพื่อสร้างให้เกิดแอปพลิเคชันหรือบริการรูปแบบใหม่ และในปัจจุบันการเชื่อมต่อของอุปกรณ์ต่างๆ จะนิยมออกแบบมาเป็นแบบเครือข่ายเซ็นเซอร์ไร้สาย (Wireless Sensor Networks) เนื่องจากง่ายต่อการใช้งานและมีความคล่องตัวสูงนั่นเอง เช่น เซ็นเซอร์ไร้สายวัดความชื้น อุณหภูมิ ความเข้มแสง ความเร็ว ความเร่ง สั่นสะเทือน ความเคลื่อนไหว ความลึก ความเป็นกรดหรือด่าง เป็นต้น โดยในปัจจุบันการระบุหมายเลขด้วยอินเทอร์เน็ตโพรโทคอลรุ่นที่ 4 (IPv4) ไม่เพียงพอและกำลังจะหมดลงในอนาคตอันใกล้นี้ ทำให้การใช้งานอินเทอร์เน็ตยุคหน้า (Next Generation Internet หรือ IPv6) เป็นเรื่องที่ต้องดำเนินการอย่างเร่งด่วน ความสำคัญของการมี IPv6 ที่สามารถแจกจ่าย IP address ไม่ซ้ำกันและสามารถเห็นกันได้ทั่วโลก จะช่วยผลักดันการพัฒนา แอปพลิเคชันแบบ peer-to-peer ที่ต้องการ IP address จริงเป็นจำนวนมาก เช่น การทำ File Sharing, Instant Messaging, และ Online Gaming เป็นต้น แอปพลิเคชันเหล่านี้มีข้อจำกัดภายใต้ IPv4 ที่ใช้อยู่ในปัจจุบัน เนื่องจากผู้ใช้งานบางส่วนที่ได้รับจัดสรร IP address ผ่าน NAT (Network Address Translation) ไม่มี IP address จริง จึงไม่สามารถใช้แอปพลิเคชันเหล่านี้ได้ โดยทุกอุปกรณ์ที่เชื่อมต่อกับอินเทอร์เน็ตต้องมีหมายเลขระบุ (IP address) ในการติดต่อสื่อสาร จึงจำเป็นต้องใช้การระบุหมายเลขด้วยอินเทอร์เน็ตโพรโทคอลรุ่นที่ 6 (IPv6) มากำกับเพื่อให้ได้หมายเลขที่ไม่ซ้ำกันและรองรับจำนวนอุปกรณ์ได้อย่างมหาศาล ดังนั้นทุกหน่วยงานจึงมีความจำเป็นอย่างยิ่งที่ต้องเตรียมความพร้อมกับการเปลี่ยนแปลงครั้งนี้

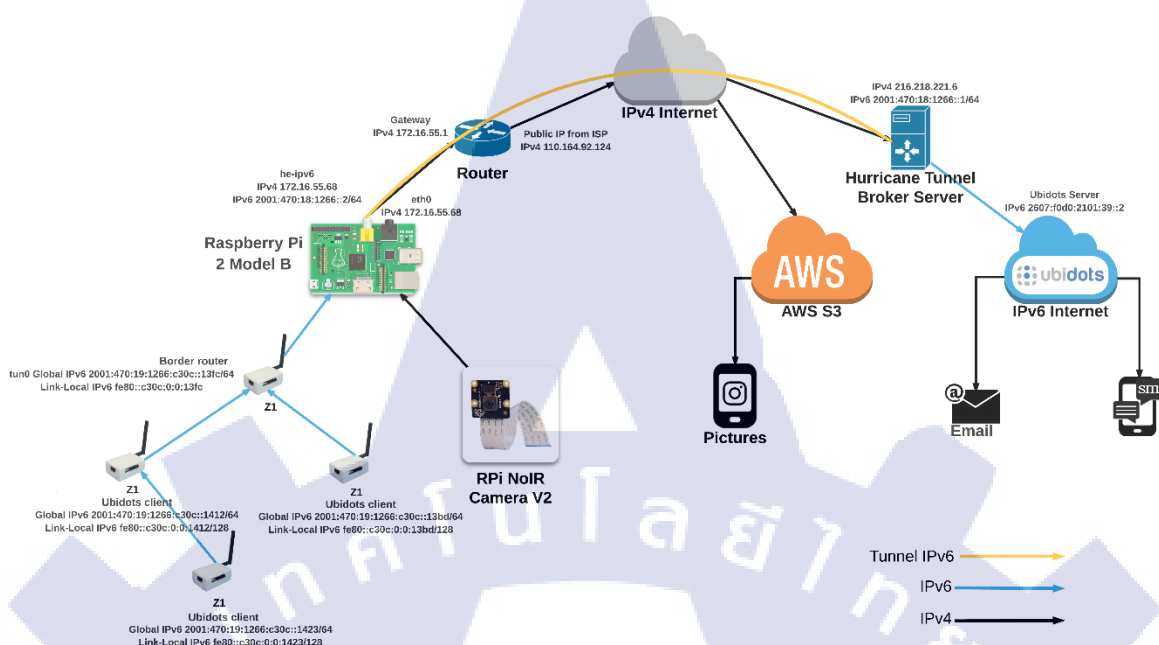
เนื่องจากปัจจุบันยังไม่มี **Routing Algorithm** ที่สามารถรองรับการสื่อสารทั้งแบบ Unicast, Multicast และ Anycast Communication ในเครือข่ายของ Wireless Sensor Networks (WSNs) ตามรูปที่ 1.1 รวมทั้งสามารถระบุหมายเลขด้วยอินเทอร์เน็ตโพรโทคอลรุ่นที่ 6 (IPv6) เพื่อส่งข้อมูลเข้าสู่โครงข่ายอินเทอร์เน็ตต่อไป ดังนั้นผู้วิจัยจึงขอเสนอ Routing Algorithm ดังกล่าว เพื่อให้การสื่อสารระหว่างเซ็นเซอร์ไร้สายสามารถส่งข้อมูลที่มี

ความแตกต่างกันภายใน Algorithm เดียวกัน โดยทำการออกแบบพร้อมการทดสอบทั้งบนโปรแกรมจำลองและบนอุปกรณ์จริง



รูปที่ 1.1 เครือข่ายของ Wireless Sensor Networks (WSNs) บน เทคโนโลยี IoT6

ในปัจจุบันเทคโนโลยี IoT6 เป็นอีกเทคโนโลยีที่มีความทันสมัยและน่าสนใจ เพราะในอุปกรณ์ที่มีขนาดเล็กแต่สามารถเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตบนไอพีรุ่นที่หกและส่งข้อมูลที่ได้รับการตรวจวัดจากเซ็นเซอร์ต่างๆ มากมายหลายชนิด ที่ทำการอัพโหลดขึ้นไปบนอินเทอร์เน็ตได้หลากหลายแพลตฟอร์ม โดยสามารถเข้าถึงข้อมูลที่ได้ตรวจวัดได้ทุกที่ทุกเวลา โดยงานวิจัยนี้ได้นำอุปกรณ์ IoT6 มาติดตั้งภายในบ้านและนำมาช่วยแจ้งเตือนผู้ใช้งานเมื่อเกิด 2 เหตุการณ์หลักๆ ดังนี้ เหตุการณ์แรกคือกรณีที่เกิดเหตุการณ์แก๊สรั่ว ซึ่งเมื่อ Z1 ตรวจจับเจอแก๊สรั่วออกมา จะทำการแจ้งเตือนไปยังผู้ใช้งานผ่านอีเมลและข้อความสั้น เหตุการณ์ที่สองคือเมื่อ Z1 ติดตั้งเซ็นเซอร์การเคลื่อนไหว สามารถแจ้งเตือนหากเกิดผู้บุกรุกภายในบ้านในเวลาที่ไม่มีอยู่บ้าน ซึ่งจะทำการถ่ายรูปและอัพโหลดภาพขึ้นไปยังคลาวด์เพื่อให้ผู้ใช้งานสามารถตรวจเช็คได้ในทันที ตามรูปที่ 1.2



รูปที่ 1.2 เครือข่ายเซ็นเซอร์ไร้สายที่ออกแบบสำหรับงานวิจัยโครงการนี้

1.2 วัตถุประสงค์การศึกษา

- (1) เพื่อออกแบบเกณฑ์วิธีเพื่อจัดเส้นทางด้วยอินเทอร์เน็ตโพรโทคอลรุ่นที่ 6 สำหรับการสื่อสารแบบยูนิคาสต์, มัลติคาสต์ และแอนนิกาสต์ ของเครือข่ายเซ็นเซอร์ไร้สาย
- (2) เพื่อเตรียมความพร้อมของสถาบันให้สามารถใช้งาน IPv6 ในเครือข่ายอินเทอร์เน็ตได้ในอนาคต
- (3) เพื่อศึกษารูปแบบการเชื่อมต่ออุปกรณ์ต่างๆ (Things) และรับส่งข้อมูลภายในเครือข่ายอินเทอร์เน็ต (Internet)
- (4) เพื่อศึกษามาตรฐานการสื่อสาร เช่น 6LoWPAN, CORE, COAP, 6TiSCH เป็นต้น
- (5) เพื่อสร้างเครือข่าย IPv6 ที่รองรับอุปกรณ์ IoT (an IPv6-based IoT)
- (6) เพื่อประกอบการเรียนการสอนในรายวิชาเครือข่ายคอมพิวเตอร์ทั้งภาคทฤษฎีและปฏิบัติ
- (7) เพื่อใช้เป็นเครือข่ายจำลอง (Testbed) สำหรับหัวข้อปัญญานิพนธ์และวิทยานิพนธ์ ของนักศึกษา

1.3 ขอบเขตของการศึกษา

- (1) ออกแบบอัลกอริทึมเพื่อจัดเส้นทางที่สามารถรองรับจากส่งข้อมูลด้วยการสื่อสารแบบยูนิคาสต์, มัลติคาสต์และแอนนิกาสต์ ของเครือข่ายเซ็นเซอร์ไร้สายบนเครือข่าย IPv6

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- (1) สามารถเป็นเครือข่ายจำลอง (Testbed) ที่รองรับการทำวิจัยในเรื่อง Internet of Things (IoT) ด้วย IPv6 ซึ่งผลงานวิจัยทางด้านนี้ยังมีน้อยอยู่มากในปัจจุบัน และสามารถสื่อสารข้อมูลแบบยูนิคาสต์, มัลติคาสต์และแอนนีกาสต์ ของเครือข่ายเซ็นเซอร์ไร้สายบนเครือข่ายอินเทอร์เน็ตด้วย IPv6 สามารถใช้ Routing Algorithm ที่นำเสนอนี้ได้

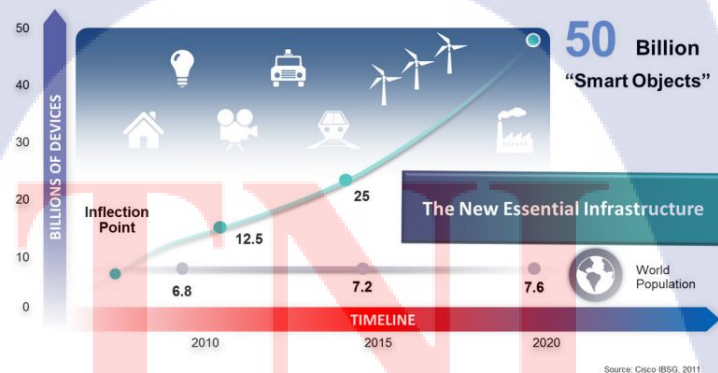
บทที่ 2

ทฤษฎีและเทคโนโลยีที่ใช้ในการวิจัย

2.1 อินเทอร์เน็ตสำหรับทุกสิ่งด้วยไอพีรุ่นที่หก

อินเทอร์เน็ตสำหรับทุกสิ่งด้วยไอพีรุ่นที่หก (Internet of Things with IPv6, IoT6) สร้างขึ้นบนเครือข่ายที่ซับซ้อน เชื่อมต่ออุปกรณ์กว่าพันล้านชิ้นและมนุษย์ เพื่อใช้งานหลากหลายเทคโนโลยี หลายโพรโทคอล หลายแพลตฟอร์ม โดยจุดประสงค์หลักของ IoT คือ การสร้างโลกอัจฉริยะที่ทั้ง ภายนอก ดิจิทัลและโลกเสมือน สร้างสภาพแวดล้อมอัจฉริยะที่จะทำให้เกิดการกระตุ้น ด้านสุขภาพ การขนส่ง เมือง อุตสาหกรรม ตึก อาคาร และสิ่งต่างๆนอกเหนือจากชีวิตประจำวัน

ความคาดหวังต่อการเชื่อมต่อของเครือข่ายอัจฉริยะ เพื่อให้เข้าถึงสารสนเทศได้ไม่ใช่เพียงแค่ทุกเวลาและทุกสถานที่ แต่ยังครอบคลุมถึงทุกสิ่งและทุกคนผ่าน เส้นทาง เครือข่าย บริการต่างๆ ซึ่งจะประสบความสำเร็จโดยมีเป้าหมายที่จะจัดการชีวิตประจำวัน เครื่องมือตรวจวัด ระบุตัวตน อุปกรณ์ระบุตำแหน่ง โดยสร้างขึ้นด้วยที่อยู่ไอพีเพื่อให้กลายเป็นอุปกรณ์อัจฉริยะ ที่สามารถทำการสื่อสารได้ไม่ใช่เพียงอุปกรณ์อื่นๆแต่ยังเชื่อมต่อกับมนุษย์ได้ด้วย โดยคาดหวังที่จะเข้าไปถึงพื้นที่ที่ไม่สามารถเข้าถึงถ้าไม่มีการพัฒนาสร้างการตรวจวัดและการใช้เทคโนโลยีระบุตำแหน่ง



รูปที่ 2.1 วิวัฒนาการของการเชื่อมต่ออินเทอร์เน็ตในอนาคต (ที่มา: Cisco 2011)

2.2 เครือข่ายเซ็นเซอร์ไร้สาย

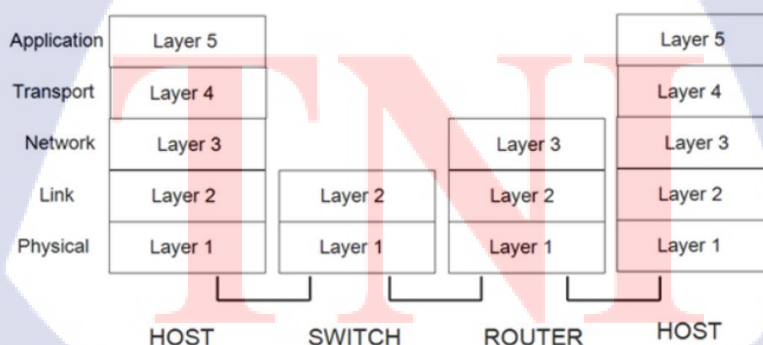
เครือข่ายเซ็นเซอร์ไร้สาย (Wireless Sensors Network, WSN) คือ โหนดเซ็นเซอร์ขนาดเล็กสามารถตั้งค่าเครือข่ายด้วยตัวเองได้ (เรียกว่า Motes) สื่อสารกันด้วยสัญญาณวิทยุ และถูกนำไปใช้ตรวจจับได้จริง เซ็นเซอร์โหนดเป็นพื้นฐานของคอมพิวเตอร์ขนาดเล็ก ทำงานได้หลากหลาย ประกอบด้วยหน่วยประมวลผลที่มีการประมวลผลที่ถูกจำกัด และหน่วยความจำที่ถูกจำกัด สื่อสารผ่านอุปกรณ์วิทยุ ใช้พลังงาน และเซ็นเซอร์หนึ่งอย่างหรือมากกว่านั้น

โหนด (Motes) จะมีรูปทรงและรูปร่างที่ต่างกันขึ้นอยู่กับการใช้งานสามารถให้มีขนาดเล็กมาก ซึ่งหากถูกปล่อยให้ใช้งานมากจะต้องมีผลกระทบน้อย สามารถที่จะชาร์จพลังงานเพื่อใช้งานในห้องทดลอง หลากหลายอุปกรณ์ขนาดเล็กที่แตกต่างกัน ในสถานการณ์ที่มีความหลากหลายมากที่สุด เพื่อให้แน่ใจ หลากหลายของการทำงาน บางแอปพลิเคชันสามารถตรวจสอบ สิ่งแวดล้อม สถาปัตยกรรม สุขภาพและความปลอดภัย

2.3 ไอพีรุ่นที่หก

2.3.1 แนวคิดของไอพีรุ่นที่หก

ไอพีรุ่นที่หก คือ อินเทอร์เน็ตโพรโทคอลรุ่นล่าสุด เพื่อให้เข้าใจว่าทำไมถึงควรที่จะนำมาใช้ กับ IoT หรือเกี่ยวข้องกับโพรโทคอล อื่นอย่าง 6LoWPAN โดยจะอธิบายในภายหลัง โดยไอพีรุ่นที่หกนั้นเป็น คนละโพรโทคอลทำงานร่วมกับไอพีรุ่นที่สี่ไม่ได้



รูปที่ 2.2 ชั้นของอินเทอร์เน็ตโพรโทคอล

ไอพีรุ่นที่หก จะทำงานในชั้นที่สาม คือชั้นเน็ตเวิร์ค ส่วนของข้อมูลจะถูกจัดการในชั้นที่สามจะถูกเรียกว่าแพ็กเก็ต อุปกรณ์ที่เชื่อมต่ออินเทอร์เน็ตเป็นได้ทั้งโฮสหรือเราเตอร์ โฮสจะเป็น พีซี โน้ตบุ๊ก

หรือเซ็นเซอร์บอร์ดก็ได้ รับและส่งข้อมูลแพ็กเก็ต โดยโฮสจะเป็นต้นทางหรือปลายทางของแพ็กเก็ต เราเตอร์นั้นดูแลการส่งต่อข้อมูล และทำหน้าที่เลือกเราเตอร์ถัดไป เพื่อส่งข้อมูลไปให้ถึงปลายทาง เครือข่ายจะประกอบด้วยเครื่องเชื่อมต่อของเราเตอร์มากมาย ที่รับข้อมูลจากช่องทางหนึ่งและส่งไปให้เราเตอร์ถัดไปให้เร็วที่สุดเท่าที่จะเป็นไปได้

2.3.2 แพ็กเก็ตไอพีรุ่นที่หก

ในรูปแบบระดับชั้นก่อนหน้านี้ ทุกชั้นจะมีการแนะนำข้อมูลเลเยอร์ของตัวเองภายในแพ็กเก็ต และใช้ข้อมูลเหล่านั้นประมวลผลชั้นเดียวกันบนอุปกรณ์ไอพีอุปกรณ์อื่น การสนทนาระหว่างระดับชั้นเดียวกันต่างอุปกรณ์จะต้องปฏิบัติตามโปรโตคอล

อินเทอร์เน็ตเลเยอร์มี 5 ชั้นตามลำดับ

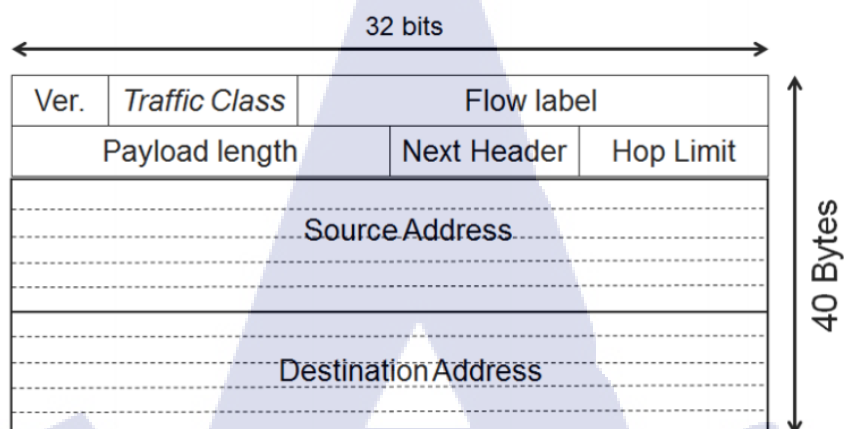
(1) Application เป็นฝั่งของซอฟต์แวร์ถูกพัฒนาโดยโปรแกรมเมอร์ ที่ใช้กองซ้อนเน็ตเวิร์คเพื่อให้บริการเครือข่าย ตัวอย่างเช่นเว็บเบราว์เซอร์ที่เปิดการเชื่อมต่อเครือข่ายเพื่อเชื่อมต่อไปที่เว็บเซิร์ฟเวอร์หรือเว็บเซิร์ฟเวอร์ซอฟต์แวร์ที่ทำงานในเซิร์ฟเวอร์อยู่ภายในเครือข่ายเพื่อตอบสนองคำร้องขอจากเว็บไคลเอนต์ ตัวอย่างแอปพลิเคชันโปรโตคอลเช่น HTTP และ DNS

(2) Transport เป็นชั้นที่ทำงานบนชั้นเน็ตเวิร์คที่คอยบริการเพิ่มเติม เช่น การส่งข้อมูลใหม่จากแพ็กเก็ตที่สูญหายหรือการรับประกันว่าข้อมูลที่ส่งกับข้อมูลที่ได้รับจะต้องเหมือนกัน โดยชั้นนี้แสดงให้เห็นถึงการบริการเครือข่ายไปยังชั้นแอปพลิเคชัน เป็นบริการที่ส่งและรับข้อมูล TCP และ UDP เป็นโปรโตคอลที่ใช้กันทั่วไปภายในอินเทอร์เน็ต

(3) Network ระดับชั้นนี้ดูแลการจัดส่งที่ข้อมูลที่ได้รับจากชั้นการขนส่งข้อมูลไปยังปลายทางให้ถูกต้อง ตลอดจนรับข้อมูลจากชั้นของการเชื่อมต่อที่รับมาจากข้อมูลปลายทางอินเทอร์เน็ตใช้เพียงโปรโตคอลเดียวในชั้นนี้ คือไอพีโดยทั้งต้นทางและปลายทางจะถูกระบุที่อยู่โดยที่อยู่ไอพี

(4) Link ในชั้นนี้ดูแลส่วนของการส่งและรับเฟรม เก็บไบต์ที่ส่งมาจากเน็ตเวิร์กละเยอร์ภายในขอบเขตของ LAN (Local Area Network) ที่ระบุถึงวิธีการที่จะใช้แบ่งปันการใช้งานของโหนดที่ต่างกัน ในชั้นนี้จะมีวิธีการระบุที่อยู่ของมันเป็นเอง ซึ่งจะขึ้นอยู่กับเทคโนโลยีที่ถูกใช้งาน

(5) Physical เป็นชั้นที่ดูแลเกี่ยวกับรายละเอียดและสัญญาณไฟฟ้า การเข้ารหัสและอื่นๆ ซึ่งจำเป็นสำหรับข้อมูลที่จะเดินทางจากโหนดหนึ่งไปอีกโหนดสื่อทางกายภาพก็รวมด้วยทั้งมีสายและไร้สาย



รูปที่ 2.3 ส่วนหัวของไอพีรุ่นที่หก

ชั้นแรกส่วนหัวของไอพีรุ่นที่หก ถูกจำกัดขนาดไว้ที่ 40 ไบต์ ตามมาด้วยข้อมูลในชั้นด้านบนและถูกปรับบางส่วนเพิ่มเข้ามาในส่วนหัวซึ่งจะอธิบายในภายหลัง อย่างที่เห็นว่าจะมีหลายพื้นที่ที่ถูกเก็บไว้ในส่วนหัว ซึ่งถูกจัดเตรียมไว้เพิ่มขึ้นเมื่อเปรียบเทียบกับส่วนหัวของไอพีรุ่นที่สี่

- (1) จำนวนของตัวเลขถูกลดจาก 12 เป็น 8
- (2) ส่วนหัวของไอพีรุ่นที่หก ถูกปรับเป็น 40 ไบต์ และถูกจัดให้เป็นแนวเดียวกัน 64 บิตซึ่งทำให้เราเตอร์ส่งข้อมูลไปได้เร็วขึ้น
- (3) พื้นที่ของที่อยู่เพิ่มจาก 32 เป็น 128 บิต

ส่วนที่สำคัญที่สุดคือต้นทางและปลายทางทุกอุปกรณ์จะต้องมีที่อยู่ไอพี ที่ไม่ซ้ำกันเพื่อระบุที่อยู่บนเครือข่าย โดยใช้ไอพีเพื่อให้เราเตอร์ตัดสินใจในการส่งข้อมูล

ส่วนหัวของไอพีรุ่นที่หกมี 128 บิตสำหรับทุกๆ ไอพีรุ่นที่หก ซึ่งใช้ได้ 2^{128} หรือประมาณ 3.14 คูณด้วย 1,038 หรือ 3.4 ตามด้วยศูนย์สามสิบแปดตัว ในทางตรงข้ามไอพีรุ่นที่สี่ใช้เพียง 32 บิตในการเข้ารหัสแต่ละที่อยู่ซึ่งมีให้ใช้ 2^{32} (4,294,967,296)

2.3.3 ที่อยู่ไอพี

การใช้ 128 บิตสำหรับที่อยู่ไอนำมาซึ่งข้อดีหลายอย่าง

- (1) มีที่อยู่ที่มากตอบสนองความต้องการของอนาคตซึ่งเพียงพอสำหรับนวัตกรรมใหม่ๆ
- (2) ทำให้ง่ายขึ้นเมื่ออุปกรณ์สามารถตั้งค่าเครือข่ายได้ด้วยตนเอง
- (3) ง่ายต่อการมอบหมาย จัดการที่อยู่

(4) มีพื้นที่สำหรับการจัดลำดับชั้นมากขึ้นและสำหรับการรวมเส้นทาง

(5) ความสามารถ IP Security การทำงานแบบปลายทางไปยังปลายทาง

ที่อยู่ไอพีรุ่นที่หกที่มีอยู่ได้ถูกจัดกลุ่มเป็นหลายประเภทดังนี้ (บางอย่างมีในไอพีรุ่นที่สี่)

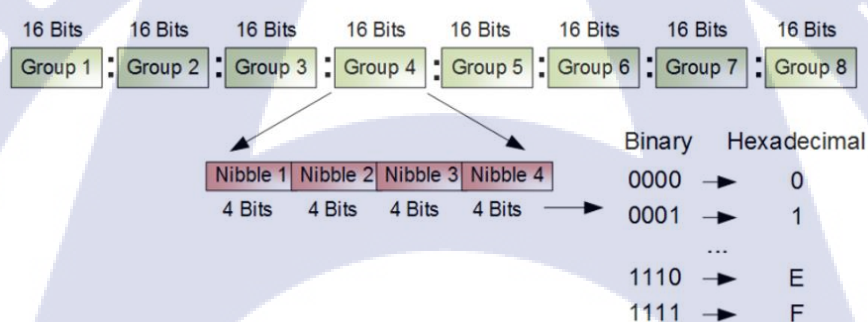
Unicast (one-to-one) ใช้ส่งข้อมูลแพ็กเก็ต จากต้นทางไปยังเป้าหมายเดียว โดยเกิดขึ้นบ่อยมากที่สุด โดยจะกล่าวถึงมันและคลาสย่อยมากขึ้น

Multicast (one-to-many) ใช้ส่งข้อมูลแพ็กเก็ต จากต้นทางไปหลายเป้าหมายที่เป็นไปได้ โดย Multicast จะส่งข้อมูลเดียวกันไปหลายที่

Anycast (one-to-nearest) ใช้ส่งข้อมูลจากต้นทางไปยังกลุ่มปลายทางที่ใกล้ที่สุด

Reserved ที่อยู่หรือกลุ่มพิเศษ ตัวอย่างเช่น ที่อยู่ใช้ในเอกสารหรือตัวอย่าง

2.3.4 กฎของไอพีรุ่นที่หก



รูปที่ 2.4 ที่อยู่ไอพีรุ่นที่หก

ลักษณะของไอพีรุ่นที่หก

(1) 16 บิตจาก 8 กลุ่ม จะถูกแบ่งโดย “:” เลขฐาน 16 จากทุกๆ 4 บิต

(2) ไม่มีการแยกตัวพิมพ์เล็กและตัวพิมพ์ใหญ่

(3) กลุ่มของที่อยู่ (Network Prefix) จะถูกเขียนขนาด Prefix/Prefix Length ซึ่งขนาดของ Prefix จะถูกวัดจากจำนวนบิตของที่อยู่ที่ใช้ทั่วไปในกลุ่ม

(4) เลขศูนย์ซ้ายสุดของแต่ละกลุ่มจะถูกเอาออก

(5) กลุ่มที่มีแต่ 0 หนึ่งกลุ่มหรือมากกว่าจะถูกแทนที่ด้วย “::” ซึ่งทำได้ครั้งเดียว

กฎ 3 อย่างที่บอกถึงลักษณะทั่วไปของ ที่อยู่ไอพีรุ่นที่หกคือ มีลักษณะเป็นเลขฐาน 16 ตัว เลขที่แสดงออกมามี 16 รูปแบบ ระหว่าง 0 ถึง F โดยจะมี 8 กลุ่ม จากเลขฐานสิบหก 4 ตัวอักษรทุกกลุ่มจะถูก แบ่งโดยโคลอน “:” อีก 2 กฎที่เหลือจะเทียบให้ดูตามด้านล่าง

ตัวอย่าง

ถ้านำเสนอรูปแบบของบิตที่อยู่จะได้รูปแบบดังตัวอย่าง

2001:0db8:4004:0010:0000:0000:6543:0ffd

ถ้าใส่เครื่องหมายกำมปูรอบที่อยู่จะได้ที่อยู่ตามจริง

[2001:0db8:4004:0010:0000:0000:6543:0ffd]

ถ้านำกฎลำดับที่สี่มาใช้ จะอนุญาตให้ย่อรูปเลข 0 ด้านซ้ายสุดของทุกกลุ่มออก จะได้

2001:db8:4004:10:0:0:6543:ffd

ถ้านำกฎลำดับที่ห้ามาใช้จะอนุญาตให้ย่อรูปกลุ่มที่มีแต่ 0 และอยู่ติดกันโดยใช้ “::” จะได้

2001:db8:4004:10::6543:ffd

ระยะเวลาใช้การย่อรูปและการขยายรูปของที่อยู่ไอพีรุ่นที่หก การประมวลผลสามารถนำ กลับมาเป็นแบบเดิมได้ ซึ่งเป็นปกติที่จะเกิดความผิดพลาด ตัวอย่างเช่น 2001:db8:a:0:0:12:0:80 สามารถย่อรูป โดยการใส่ “::” ได้สองทางเลือก

1) 2001:db8:a::12:0:80

2) 2001:db8:a:0:0:12::80

ทั้งคู่นั้นเป็นที่อยู่ไอพีรุ่นที่หกที่ถูกต้องแต่ที่อยู่ 2001:db8:a::12::80 นั้นผิด ซึ่งไม่ได้ทำตาม กฎของการย่อรูปด้านบนปัญหาคือการย่อรูปที่ไม่ดีทำให้เราไม่แน่ใจว่าจะขยายอย่างไร ซึ่งไม่ชัดเจนสามารถ ขยายแล้วจะเป็น 2001:db8:a:0:12:0:0:80 หรือเป็น 2001:db8:a:0:0:12:0:80

Network prefix ของไอพีรุ่นที่หก

สุดท้ายต้องรู้เกี่ยวกับแนวคิดของ Network Prefix ซึ่งจะบอกถึงบิตที่บังคับและบิตที่ไม่ระบุ ที่สามารถสร้าง Prefix ย่อยใหม่ หรือใช้บอกถึงที่อยู่ไอพีรุ่นที่หกที่มอบให้โฮส

ดังตัวอย่าง

1. เน็ตเวิร์ค Prefix 2001:db8:1::/48 (2001:0db8:0001:0000:0000:0000:0000:0000) บอกถึง 48 บิต ที่จะต้องเหมือนเดิม (2001:0db8:0001) ซึ่งสามารถใช้ 80 bits ด้วยอื่นๆได้ ตัวอย่างเช่น เพื่อจัด 2 Prefix ที่ เล็กกว่า 2001:0db8:1:a::/64 และ 2001:db8:1:b::/64

2. ถ้านำ 1 ใน Prefix เล็กด้านบน 2001:db8:1:b::/64 ที่ 64 บิตแรกจะถูกกำหนดไว้ พวกเราจะมี 64 บิตทางด้านขวาเพื่อกำหนด ตัวอย่างเช่น ไอพีรุ่นที่หกที่แสดงในโฮส 200001:db8:1:b:1:2:3:4 *ตัวอย่างนี้อนุญาตให้เราเสนอพื้นฐานของ ไอพีรุ่นที่หก * A/64 มักจะเป็น Prefix ที่ถูกใช้ใน LAN 64 บิตด้านขวา จะถูกเรียกว่า Interface Identifier (IID) เพราะว่ามีเพียงหนึ่งเดียวซึ่งสามารถระบุโฮสใน Local Network ถูกกำหนดโดย /64 จะได้ Prefix ตามรูปภาพนี้



รูปที่ 2.5 เน็ตเวิร์คไอดีและอินเทอร์เฟซไอดี

2.3.5 ใช้ไอพีรุ่นที่หกเพื่ออะไร

อย่างที่เห็นว่าไอพีรุ่นที่หกมีคุณสมบัติที่ช่วยเพิ่มความสะดวก เช่น Global Addressing และ Auto Configuration เพราะ ไอพีรุ่นที่หก นั้นมีที่อยู่มากมายพอให้ได้ใช้ไปอีกหลายร้อยปี พวกเราสามารถให้ Global unicast address ไอพีรุ่นที่หกกับทุกอุปกรณ์ที่พวกเราจะคิดถึงได้ ซึ่งจะนำกลับมาถึงการเริ่มต้นของรูปแบบอินเทอร์เน็ตที่ทุกๆ อุปกรณ์ไอพี สามารถสื่อสารกันได้. การเชื่อมต่อแบบปลายทางไปยังปลายทางจะอนุญาตให้เชื่อมต่อแบบสองทิศทางบนเครือข่ายระหว่างอุปกรณ์ไอพีใดๆก็ได้ ซึ่งเป็นผลลัพธ์ของแอปพลิเคชันที่ร่วมมือกัน การเก็บข้อมูลรูปแบบใหม่ๆ การส่งและเข้าถึงข้อมูล

ในเนื้อหาจะได้เห็นถึงตัวอย่างการเก็บข้อมูลเซ็นเซอร์ของไอพีรุ่นที่หกรอบโลกการส่งและการเข้าถึงข้อมูลจากสถานที่ต่างๆเพื่อสร้างเครือข่ายของการวัดข้อมูลบนพื้นที่จริง การเก็บข้อมูลและประมวลผล

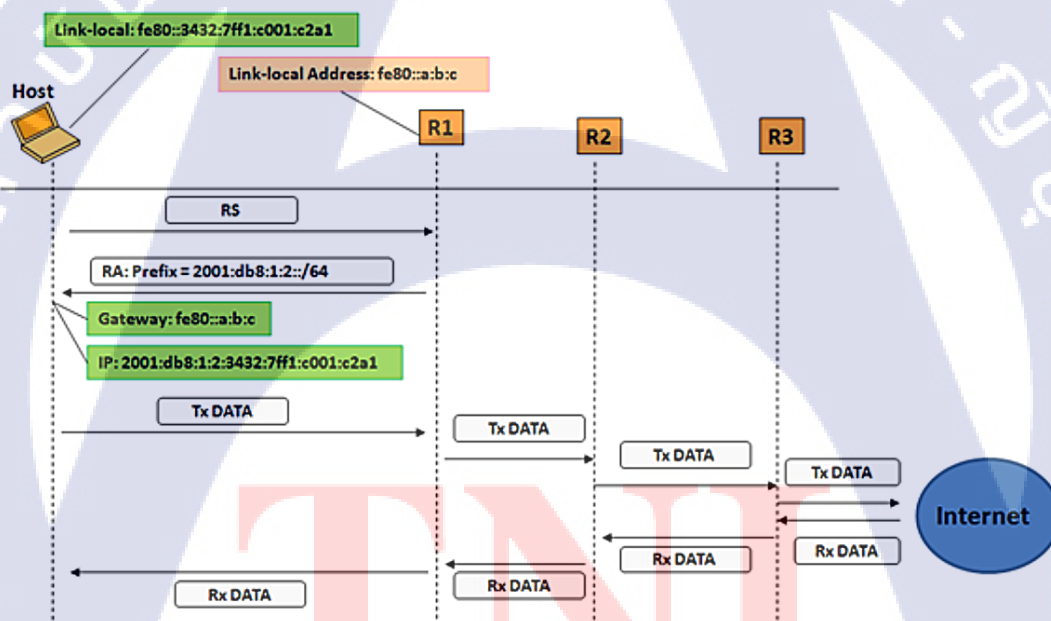
การมีอยู่ของที่อยู่จำนวนมหาศาล ทำให้เกิดวิธีใหม่ที่เรียกว่า Stateless Address Configuration (SLAAC) ซึ่งไม่มีในไอพีรุ่นที่สี่ซึ่ง โดยจะสรุปวิธีต่างๆที่ใช้ตั้งค่าที่อยู่ของไอพีรุ่นที่หก

Statically คุณสามารถที่จะมอบที่อยู่ให้อุปกรณ์ไอพีจากนั้นตั้งค่าเองในอุปกรณ์โดยใช้วิธีใดก็ได้ เช่น เว็บบรรทัดคำสั่งและอื่นๆ โดยปกติจะต้องตั้งค่าอื่น เช่น เกตเวย์เพื่อให้ส่งแพ็กเก็ตไปยังนอกเครือข่ายได้

DHCPv6 (Dynamic Host Configuration Protocol for IPv6) [RFC3315] นำมาจากวิธีที่คล้ายกับที่มีอยู่ในไอพีรุ่นที่สี่โดยจำเป็นจะต้องให้เครื่องแม่ข่ายที่ติดต่อเพื่อมอบที่อยู่ให้อุปกรณ์ไอพี DHCPv6 จะอนุญาตให้อุปกรณ์ไอพีตั้งค่าอัตโนมัติจึงเป็นเพราะว่าทำไมถึงชื่อว่า Stateful address Configuration เพราะ DHCPv6 Server จะเป็นคนรักษาสถานะและแจกที่อยู่

SLAAC Stateless Address Auto Configuration [RFC4862] เป็นวิธีใหม่นำเสนอด้วยไอพีรุ่นที่หกที่อนุญาตให้จัดการตั้งค่า ตัวแปรในเครือข่ายบนอุปกรณ์ไอพี โดยให้เราเตอร์มอบการเชื่อมต่อกับเครือข่ายให้

ข้อดีของ SLAAC จะทำให้การตั้งค่าอุปกรณ์จำนวนมากทำได้ง่ายขึ้น เช่น เซ็นเซอร์ กล้อง หรืออุปกรณ์พลังงานต่ำอื่นๆ ไม่ต้องตั้งค่าเครือข่ายในอุปกรณ์ไอพีเพื่อ เพียงเสียบปลั๊กและเชื่อมต่ออินเทอร์เน็ต ซึ่งง่ายต่อโครงสร้างเครือข่ายที่ใช้ไอพีรุ่นที่หกเพราะไม่จำเป็นต้องเพิ่มอุปกรณ์หรือเครื่องแม่ข่าย ใช้เราเตอร์ตัวเดิมที่ใช้ส่งแพ็กเก็ตไปนอกเครือข่าย เพื่อตั้งค่าอุปกรณ์ไอพีโดยที่เราไม่ต้องเข้าไปถึงรายละเอียดแต่จำเป็นต้องรู้ว่าอยู่ในเครือข่าย LAN เชื่อมต่ออินเทอร์เน็ตโดยใช้เราเตอร์ โดยเราเตอร์จะดูแลส่วนของการส่งข้อมูลที่จำเป็นต้องตั้งค่าไปยังโฮสโดยใช้ข้อความ RA (Router Advertisement) โดยเราเตอร์จะส่ง RAs เป็นช่วงๆเพื่อกระตุ้นการประมวลผล โฮสสามารถส่งข้อความ RS (Router Solicitation) เมื่อเชื่อมต่อกับเครือข่าย เราเตอร์จะส่ง RA ทันทีเพื่อตอบมายัง RS ตามขั้นตอนดังต่อไปนี้



รูปที่ 2.6 การแลกเปลี่ยนแพ็กเก็ตในไอพีรุ่นที่หก

1. R1 เราเตอร์มอบการเชื่อมต่อไปยังโฮสใน LAN และคอยส่ง RAs เป็นช่วงๆ
2. ทั้ง R1 และ โฮสจะมี Link-Local address ในการเชื่อมต่อไปยังโฮสของที่อยู่ใน LAN จะตั้งค่าอัตโนมัติเมื่อการเชื่อมต่อพร้อมใช้งาน โฮสจะสร้าง Link-Local address โดยรวม 64 บิตด้านซ้ายของ link-

local Prefix (fe80::/64) และ 64 บิตด้านขวาจะสร้าง IID (:3432:7ff1:c001:c2a1) Link-Local address เหล่านี้สามารถใช้ใน LAN เพื่อแลกเปลี่ยนแพ็กเก็ตแต่จะไม่สามารถส่งแพ็กเก็ตไปนอกเครือข่าย LAN ได้

3. โฮสจำเป็นต้องมีพื้นฐาน 2 อย่างเพื่อส่งข้อมูลไปนอกเครือข่ายคือ Global IPv6 Address และที่อยู่ของเกตเวย์ เช่น เราเตอร์ที่จะส่งแพ็กเก็ตจะต้องมีการหาเส้นทางเพื่อส่งไปนอกเครือข่าย

4. R1 ส่ง RAs เป็นช่วงๆ (ทุกๆหลายวินาที) เมื่อโฮสรับการเชื่อมต่อและตั้งค่า Link-Local Address และส่ง RS ไปยัง R1 และ R1 จะตอบด้วย RA ซึ่งบรรจุ 2 อย่างทันที

(1) Global prefix of length 64 บิต เพื่อสำหรับ SLAAC โฮสจะใช้รับ Prefix และมอบ IID เฉพาะมักเป็นส่วนที่ใช้สำหรับ Link-Local address ส่วนทาง Global IPv6 Address จะถูกตั้งค่าใน โฮสและสามารถสื่อสารบนเครือข่าย ไอพีรุ่นที่หก

(2) รวมไปถึง Link-local address ของ R1 เพราะเป็นที่อยู่ต้นทางของ RA โฮสจะสามารถใช้ ที่อยู่นี้ตั้งค่าดั้งเดิม เกตเวย์ส่งแพ็กเก็ต ออกไปโดยอัตโนมัติเพื่อไปยังโฮสไอพีรุ่นที่หกบนเครือข่าย

5. ทั้งเกตเวย์และไอพีรุ่นที่หกจะถูกตั้งค่า โฮสจะรับและส่งข้อมูล จากรูปภาพมีบางอย่างส่ง (Tx Data) ไปยังโฮสบนเครือข่าย เพื่อสร้างแพ็กเก็ต ไอพีรุ่นที่หกด้วยที่อยู่ปลายทางของผู้รับและต้นทางจะถูก auto-configured global address ที่ส่งมายังเกตเวย์ Link-local address ของ R1 โฮสปลายทางจะสามารถตอบข้อมูลด้วย บางข้อมูล (Rx Data)

2.3.6 ข้อดีของ ไอพีรุ่นที่หก เมื่อทำงานร่วมกับ IoT

2.3.6.1 ไอพีรุ่นที่หกจะถูกยอมรับไปตามเวลา

การเชื่อมต่ออินเทอร์เน็ตจำเป็นต้องมีโปรโตคอลเครือข่ายซึ่งจะบอกที่อยู่ในการส่งข้อมูลจากเว็บ ไอพีรุ่นที่สี่นั้นมีจำนวนที่อยู่ไอพี ที่จำกัด ซึ่งจะต้องเปลี่ยนเป็นไอพีรุ่นที่หกอย่างหลีกเลี่ยงไม่ได้ การสำรวจของกูเกิลพบว่าไอพีรุ่นที่หกได้รับการยอมรับเพิ่มขึ้นเป็นเท่าตัวทุกๆ 9 เดือน

2.3.6.2 ขยายเครือข่ายได้ง่าย

ไอพีรุ่นที่หกให้ที่อยู่กับอุปกรณ์ได้มากถึง 2^{128} เลขหมาย ซึ่งเพียงพอต่อปัจจุบันและอนาคตสำหรับอุปกรณ์การสื่อสารทุกชิ้นบนโลก

2.3.6.3 แก้ปัญหา NAT บังไอพีอุปกรณ์นอกเครือข่าย

เมื่อไอพีรุ่นที่สี่ถึงขีดจำกัด การขยายตัวอย่างไม่มีจำกัดอินเทอร์เน็ตจึงต้องยอมรับวิธี NAT ซึ่งจะอนุญาตให้ผู้ใช้งานแบ่งไอพีสาธารณะใช้ ซึ่งมีข้อเสียถึงสองอย่าง

(1) ผู้ใช้งาน NAT จะยืมและแบ่ง ที่อยู่ไอพีกับคนอื่น ส่งผลให้ไม่มีที่อยู่ไอพีสาธารณะ ไอพีจะกลายมาเป็นผู้ใช้งานอินเทอร์เน็ตที่ไม่มีไอพีจริง โดยสามารถใช้งานอินเทอร์เน็ตได้ แต่ไม่สามารถถูกเข้าถึงจากอินเทอร์เน็ตได้

(2) ทำให้การเชื่อมต่อแบบปลายทางไปยังปลายทางโดยตรงไม่ได้ และช้าลง มีความปลอดภัยต่ำ

2.3.6.4 มีความปลอดภัยมาก

ไอพีรุ่นที่หกครอบคลุมการเชื่อมต่อปลายทางไปยังปลายทางซึ่งมีหลายกลไกการหาเส้นทาง และไอพีรุ่นที่หกรองรับชุมชนผู้ใช้งานจำนวนมาก ผู้วิจัยที่จะช่วยเพิ่มความปลอดภัย รวมไปถึง IP Security

2.3.6.5 กองซ้อนมีขนาดเล็ก

ไอพีรุ่นที่หกและ IoT ถูกวิจัยมาหลายปี ผู้พัฒนาได้พัฒนาการย่อขนาดไอพีรุ่นที่หกลงซึ่งก็คือ 6LoWPAN ซึ่งง่ายและมีประสิทธิภาพต่อกฎเพื่อลดขนาดที่อยู่ลงเพื่ออุปกรณ์ที่มีข้อจำกัด อย่างไรก็ตามขอบของเครือข่ายจะต้องแปลงการบีบอัดให้กลับเป็นไอพีรุ่นที่หก กองซ้อนขนาดเล็กถูกพัฒนาเช่นเดียวกับ Contiki ซึ่งกินขนาดน้อยกว่า 11.5 กิโลไบต์

2.3.6.6 เครือข่ายของอุปกรณ์

ไอพีรุ่นที่หกมีที่จำนวนโพรโทคอลเครือข่ายมาก สามารถให้บริการทุกๆอุปกรณ์ได้ การทดลองพิสูจน์ได้ว่าไอพีรุ่นที่หกมีมากพอสำหรับเซ็นเซอร์บ้านอัจฉริยะ เมืองอัจฉริยะ COAP เป็นโพรโทคอลที่อนุญาต ให้บริการเว็บเซิร์ฟเวอร์ได้ง่ายที่จะทำตาม สถาปัตยกรรม REST

2.3.6.7 ความรวดเร็ว

ไอพีรุ่นที่หกรองรับการทำงานที่รวดเร็วของโหนดปลายทางและการหาเส้นทางของโหนดที่รวดเร็ว

2.3.6.8 การตั้งค่าที่อยู่ด้วยตัวเอง

ไอพีรุ่นที่หก รองรับการจัดค่าอุปกรณ์ด้วยตัวเอง โดยโหนดสามารถกำหนดที่อยู่ของตนเองได้ ซึ่งจะช่วยลดค่าใช้จ่ายในการตั้งค่าได้

2.3.6.9 ทำตามอินเทอร์เน็ตอย่างสมบูรณ์

ไอพีรุ่นที่หกส่งผลให้พัฒนาเครือข่าย อุปกรณ์สามารถทำงานร่วมกับอุปกรณ์ได้อย่างแท้จริงครอบคลุมทั่วโลก

ทั้ง 9 ปีที่ผ่านมาทำให้ไอพีรุ่นที่หกมีความจำเป็นต่อ IoT เป็นอย่างมากแต่การจะนำ IoT มาใช้งานร่วมกับไอพีรุ่นที่หกเลยเป็นเรื่องยากเพราะไอพีรุ่นที่หกเองยังใช้พลังงานที่สูงและ IoT จะต้องใช้สถาปัตยกรรมซอฟต์แวร์ที่มีความสามารถที่จะจัดการกับข้อมูลจำนวนมาก การจัดการ การคำนวณ การประมวลผลใหม่ๆ การถ่ายทอดสด การกรองข้อมูล การรวมข้อมูล และทำเหมืองข้อมูล ทั้งหมดนี้ถูกทำงานบน HTTP และไอพีซึ่งแตกต่างจากธรรมชาติของ IoT ที่ใช้พลังงานน้อยและต้องการสื่ออื่นๆเพื่อให้ต่ออินเทอร์เน็ตได้ขณะใช้งานแบตเตอรี่พลังงานจะถูกใช้กับการส่งข้อมูลที่ไม่จำเป็น หลายโพรโทคอลไม่เหมาะสมกับรูปแบบการสื่อสาร ซึ่งจำเป็นจะต้องเชื่อมต่ออินเทอร์เน็ต อินเทอร์เน็ตโพรโทคอลที่มีอยู่เช่น HTTP TCP ไม่เหมาะสมกับการสื่อสารที่ใช้พลังงานต่ำ เนื่องจากทั้งข้อมูลและส่วนหัวต้องการความน่าเชื่อถือของแพ็กเก็ตในชั้นที่สูงขึ้น ซึ่งเป็นอุปสรรคต่อการปรับโพรโทคอลที่มีอยู่ให้เข้ากับการทำงานของเครือข่ายนั้นๆ เพื่อที่จะเชื่อมต่อกัน เช่นเดียวกับอินเทอร์เน็ตเชื่อมต่อหลายอุปกรณ์ IoT เช่น RFID Sensor และอื่นๆ ซึ่งใช้พลังงานน้อย มีความน่าเชื่อถือ และเข้าถึงอินเทอร์เน็ตได้

2.4 เครือข่ายส่วนบุคคลไร้สายที่ใช้พลังงานต่ำสำหรับไอพีรุ่นที่หก

เครือข่ายส่วนบุคคลไร้สายที่ใช้พลังงานต่ำสำหรับไอพีรุ่นที่หก (IPv6 over Low power Wireless Personal Area Networks, 6LoWPAN) คือ ส่วนขับเคลื่อนของ IoT ซึ่งจะเชื่อมต่อกันผ่านเครือข่ายไร้สายโดยสร้างช่องทางการเชื่อมต่อและรับส่งข้อมูลกัน การนำเทคโนโลยีเครือข่ายไร้สายมาใช้จะเพิ่มอุปกรณ์ที่เชื่อมต่อกัน ส่งผลให้เกิดข้อจำกัดในเรื่องของการใช้งาน อายุแบตเตอรี่ การใช้พลังงาน และระยะห่างระหว่างอุปกรณ์ เทคโนโลยีใหม่และโพรโทคอลควรรับมือสถานะใหม่ถูกเรียกว่า Low power and Lossy network (LLNs) โดยมีลักษณะดังต่อไปนี้

1. จำเป็นต้องรองรับอุปกรณ์มากกว่าที่อยู่บน LAN ในปัจจุบัน
2. จำกัดโค้ดและการใช้แรมในอุปกรณ์
3. ข้อจำกัดในเรื่องของสื่อสารด้านระยะห่างของเครือข่าย พลังงาน และการประมวลผล
4. ทุกองค์ประกอบควรทำงานด้วยกันและใช้พลังงานและแบนด์วิธที่เหมาะสม

ปัจจัยอื่นที่จะถูกนำมาใช้ภายใน IoT คือการใช้ไอพีเป็นโพรโทคอล การใช้ไอพีมีข้อดีหลายอย่าง เพราะเป็นมาตรฐานที่เปิดให้ใช้อย่างแพร่หลาย นำไปใช้ได้ง่าย ทำงานร่วมกันได้ดีและง่ายต่อการนำไปพัฒนา

การใช้มาตรฐานที่เหมือนกัน เช่น การเชื่อมต่อจากปลายทางไปยังปลายทางบนพื้นฐานไอพีซึ่งแก้ปัญหการทำงานร่วมกันไม่ได้

สำหรับเทคโนโลยีการสื่อสารแบบไร้สาย IEEE 802.15.4 เป็นความหวังของเลเยอร์ชั้นด้านล่าง (ระดับชั้น Data Link และระดับชั้น Physical Layer) แม้ว่าจะมีการใช้ ทางเลือกอื่นๆ เช่น Low Power WiFi, Bluetooth พลังงานต่ำ, DECT Ultra Low Energy, ITU-T G.9959 และ NFC (Near Field Communication)

IETF แต่างจาก Working groups (WGs) ที่จะคอยพัฒนามาตรฐานเพื่อ WSN

1. 6LoWPAN เป็นมาตรฐานสำหรับไอพีรุ่นที่หกสื่อสารบน IEEE 802.15.4 เทคโนโลยีเครือข่ายไร้สาย 6LoWPAN ทำหน้าที่ระดับชั้นในการปรับตัวระหว่างมาตรฐานของไอพีรุ่นที่หกสื่อสารด้วยพลังงานต่ำและเครือข่ายไร้สายที่ถูกบีบอัด เป็นสื่อกลางโดย IEEE 802.15.4 ซึ่งสามารถใช้งานกับแคไอพีรุ่นที่หกเท่านั้นไม่รองรับไอพีรุ่นที่สี่

2. Roll (Routing Over Low power and Lossy networks) จำเป็นในการใช้วิธีเส้นทางที่เหมาะสม เพราะไม่สามารถใช้โปรโตคอลในการหาเส้นทางแบบที่มีอยู่ได้ WGs เน้นไปที่การแก้ปัญหการหาเส้นทาง ทุกแอปพลิเคชันที่ที่เป็นไปได้ของ LLNs (อุตสาหกรรม บ้าน สิ่งก่อสร้างและเครือข่ายเซ็นเซอร์ในเมือง) และโปรโตคอลถูกออกแบบให้เหมาะสมกับแอปพลิเคชัน WGs เน้นไปที่กรอบความคิดของสถาปัตยกรรมการหาเส้นทางบนไอพีรุ่นที่หก

3. 6lo (IPv6 over Networks of Resource-Constrained Nodes) WGs จัดการกับไอพีรุ่นที่หกให้เชื่อมต่อกับโหนดบนเครือข่ายที่มีข้อจำกัด ด้วยการขยายการทำงานของ 6LoWPAN WGs นิยาม IPv6-Over-Foo ที่ในชั้นของการปรับตัวใช้ 6LoWPAN สำหรับชั้น Data Link ในโหนดบนเครือข่ายที่มีข้อจำกัด

จากที่กล่าวมา 6LoWPAN เป็นจุดเริ่มต้นของงานที่ดำเนินการตามมาตรฐานของ IETF เพื่อสื่อสารบนโหนดที่มีข้อจำกัดด้านทรัพยากรใน LLNs โดยใช้ไอพีรุ่นที่หก การทำงานบน 6LoWPAN สำเร็จเรียบร้อย และสำเร็จมากยิ่งขึ้นด้วย Roll WGs ตอบสนองความต้องการของการหาเส้นทางและ 6lo เพื่อขยายมาตรฐาน 6LoWPAN ไปยังระดับชั้นในการเชื่อมต่ออื่น ตามมาด้วยรายละเอียดเพิ่มเติมที่เกี่ยวกับ 6LoWPAN ขั้นตอนแรกคือไอพีรุ่นที่หกบน WSN/IoT 6LoWPAN และมาตรฐานที่ใกล้เคียงต่างเป็นกังวลเกี่ยวกับการนำไอพีเพื่อเชื่อมต่ออุปกรณ์ซึ่งไม่เหมาะสมกับระดับชั้นด้านบน ยกเว้นเฉพาะ UDP ในชั้นการขนส่งข้อมูล

2.4.1 ภาพรวมของ LoWPAN

Low-power and lossy Network (LLNs) โดยทั่วไปจะหมายถึงเครือข่ายที่สร้างจากโหนดที่มีข้อจำกัดมาก (หน่วยประมวลผลที่มีข้อจำกัด หน่วยความจำ พลังงาน) เชื่อมต่อกันด้วยลักษณะ “ลดทอน” ผ่านคลื่นวิทยุพลังงานต่ำ มีลักษณะความเร็วต่ำ ประสิทธิภาพต่ำ มีราคาถูก และมีการเชื่อมต่อที่ไม่เสถียร

LoWPAN เฉพาะกรณีของ LLNs นั้นสร้างขึ้นโดยอุปกรณ์ที่ทำตามมาตรฐาน IEEE 802.15.4

ตัวอย่างคุณลักษณะของอุปกรณ์ LoWPAN

(1) มีความสามารถในการประมวลผลที่จำกัด ต่างรุ่นก็มีความเร็วสัญญาณนาฬิกาที่ต่างกัน โดยเริ่มจาก 8 บิต

(2) มีหน่วยความจำขนาดเล็ก แรมและรอมมีขนาดเล็ก ซึ่งคาดว่าจะมากกว่านี้ในอนาคต จึงต้องพยายามใช้น้อยที่สุดเท่าที่จำเป็น

(3) มีพลังงานต่ำ: เพื่อการใช้พลังงานที่ประหยัด

(4) มีระยะสั้น POS (The Personal Operating Space) ได้นิยามโดย IEEE 802.15.4 ว่าสามารถใช้ได้ในระยะ 10 เมตร สำหรับการใช้งานจริงและมากกว่า 100 เมตรสำหรับทางที่ไม่มีสิ่งกีดขวาง

(5) มีต้นทุนที่ต่ำขับเคลื่อนไปด้วยลักษณะ เช่น การประมวลผลที่ช้า หน่วยความจำต่ำและอื่นๆ

ทุกข้อจำกัดบนโหนดจะเปลี่ยนในอนาคตเหมือนเทคโนโลยีที่พัฒนาขึ้น แต่เปรียบกับปัจจัยอื่น คาดว่า LoWPANs นั้นสามารถใช้งานได้ในอุปกรณ์ที่มีข้อจำกัดอยู่มาก เพื่อรองรับอุปกรณ์ราคาถูก ใช้ได้นานและข้อจำกัดสมบัติอื่นๆด้วย

โดยปกติ LoWPAN จะรวมไปถึงอุปกรณ์ที่ทำงานร่วมกันเพื่อเชื่อมต่อสภาพแวดล้อมจริงไปยังแอปพลิเคชันที่ใช้จริง เช่น เซ็นเซอร์ไร้สาย แม้ว่า LoWPAN ไม่จำเป็นต้องประกอบด้วยโหนดเซ็นเซอร์เพียงอย่างเดียว ซึ่งอาจจะมีตัวกระตุ้นอื่นๆ

นอกจากนี้สิ่งสำคัญของลักษณะของ LoWPAN เพราะเป็นข้อจำกัดที่จะชี้นำทางในด้านเทคนิค

(1) แพ็กเก็ตขนาดเล็ก ขนาดเฟรมที่สูงสุดของ physical layer คือ 127 ไบต์ ผลลัพธ์สูงสุดในเฟรมที่ระดับชั้น MAC (Media Access Control) คือ 102 Octet Link-layer มีความปลอดภัยที่เพิ่มขึ้น ที่เพิ่มเข้ามาสำหรับข้อมูลแพ็กเก็ต สูงสุด 81 Octet

(2) IEEE 802.15.4 นิยามที่อยู่ไว้หลายรูปแบบที่อนุญาตให้ใช้อย่างใดอย่างหนึ่ง IEEE 64 บิต ถูกเสริมเข้ามา หรือหลังจากการรวมตัวที่อยู่ 16 บิต เฉพาะเพื่อใช้ใน PAN (Personal Area Network)

(3) แบนด์วิดท์ อัตราการส่งข้อมูลขนาด 2,510 Kbps, 40 Kbps , และ 20 Kbps ที่ระดับชั้น Physical กำหนดไว้คือ (2.4 GHz, 915 MHz และ 868 MHz ตามลำดับ)

(4) โทโพโลยีแบบดาวและตาข่าย

(5) อุปกรณ์จำนวนมากที่จะถูกล่อยออกไปตลอดการใช้งาน ตำแหน่งของอุปกรณ์ไม่ต้องถูกกำหนดล่วงหน้า มักถูกใช้งานในรูปแบบ ad-hoc บางครั้งอุปกรณ์อาจถูกวางไว้ในจุดที่เข้าถึงยาก หรือง่ายต่อการเคลื่อนย้ายก็ได้

(6) อุปกรณ์ภายใน LoWPAN มีแนวโน้มที่จะไม่น่าเชื่อถือหลากหลายเหตุผล การเชื่อมต่อด้วยคลื่นวิทยุที่ไม่แน่นอน การใช้พลังงาน อุปกรณ์ค้าง การโค่นปลอมข้อมูล และอื่นๆ

(7) โหมดนิทรา อุปกรณ์อาจจะพักเป็นช่วงเวลานานเพื่อประหยัดพลังงานทำให้ไม่สามารถสื่อสารได้ขณะพัก

2.4.2 การใช้ไอพิน 6LoWPAN

จากที่ได้กล่าวมาแล้วดูเหมือนว่าการใช้ไอพีโดยเฉพาะไอพีรุ่นที่หก ถูกปรับใช้ไปในวงกว้างเพราะมีข้อดีที่หลากหลาย 6LoWPAN ก็คือไอพีรุ่นที่หกบนเครือข่าย 6LoWPAN ในส่วนที่จะพูดถึงประโยชน์และปัญหาที่เพิ่มเข้ามาเมื่อใช้ไอพิน 6LoWPAN แอปพลิเคชันของเทคโนโลยีไอพีโดยเฉพาะเครือข่ายไอพีรุ่นที่หกถูกยอมรับว่ามีประโยชน์ต่อ 6LoWPAN ตามนี้

(1) ธรรมชาติของไอพีคือการมีอยู่ที่แพร่หลายโดยใช้ประโยชน์จาก โครงสร้างที่มีอยู่

(2) เทคโนโลยีบนไอพีมีอยู่แล้วอย่างที่รู้กันดีว่าถูกใช้งานอย่างแพร่หลาย ซึ่งช่วยให้ใช้งานได้ง่าย ง่ายต่อการปรับปรุงทำงานร่วมกันได้ดีและยังง่ายต่อการพัฒนาแอปพลิเคชัน

(3) เทคโนโลยีเครือข่ายไอพีถูกเปิดให้ใช้งานอย่างอิสระตามข้อกำหนด ซึ่งผู้คนสามารถเข้าใจได้ง่ายกว่าเลือกวิธีที่มีเจ้าของกรรมสิทธิ์

(4) เครื่องมือสำหรับไอพินั้นมีอยู่แล้ว

(5) อุปกรณ์ที่ใช้ไอพีสามารถเชื่อมต่อไปยังเครือข่ายอื่นได้ง่ายโดยที่ไม่จำเป็นต้องผ่านตัวกลาง เช่น ผ่านการแปลงโปรโตคอล เกตเวย์ หรือพร็อกซี

(6) การใช้ไอพีรุ่นที่หกช่วยให้มีที่อยู่ใช้งานเป็นจำนวนมากซึ่งง่ายต่อการตั้งค่าตัวแปรด้วยการตั้งค่าอัตโนมัติ (SLAAC) ซึ่งเป็นเรื่องสำคัญสำหรับ 6LoWPAN ที่มีรองรับการทำงานของอุปกรณ์จำนวนมาก

ในทางกลับกันการใช้ไอพีเพื่อสื่อสารบน 6LoWPAN นำมาซึ่งหลายปัญหาที่ต้องจำไว้

(1) เชื่อมต่อด้วยไอพี หนึ่งในลักษณะของ 6LoWPAN คือมีขนาดแพ็กเก็ตที่จำกัดซึ่งส่วนที่เพิ่มเข้ามาสำหรับส่วนหัวของไอพีรุ่นที่หก และเลขอ์ด้านบนจะต้องถูกบีบอัดเท่าที่เป็นไปได้

(2) โทโพโลยี 6LoWPAN รองรับหลากหลายโทโพโลยีรวมไปถึงแบบดาข่ายและแบบดาว โทโพโลยีแบบดาข่ายจะใช้ การกระโดดข้าม (multi-hop) เพื่อไปยังปลายทางตามที่ต้องการ ดังตัวอย่าง อุปกรณ์ตัวกลางจะทำหน้าที่ส่งต่อแพ็กเก็ตที่ระดับชั้น Data Link โทโพโลยีแบบดาวจะจัดการเตรียมอุปกรณ์ที่มีความสามารถในการส่งต่อข้อมูล ซึ่งเพิ่มเข้ามาใน IEEE 802.15.4 โดยอุปกรณ์เหล่านี้ใช้วิธีการอื่นในการเชื่อมต่อเครือข่าย เช่น อีเทอร์เน็ตหรือ 802.11 โดยเป้าหมายคือการรวมเครือข่ายบนเทคโนโลยีที่แตกต่างกันได้อย่างลงตัว ซึ่งแน่นอนว่าเป็นแรงขับเคลื่อนสำคัญที่เริ่มต้นด้วยไอพี

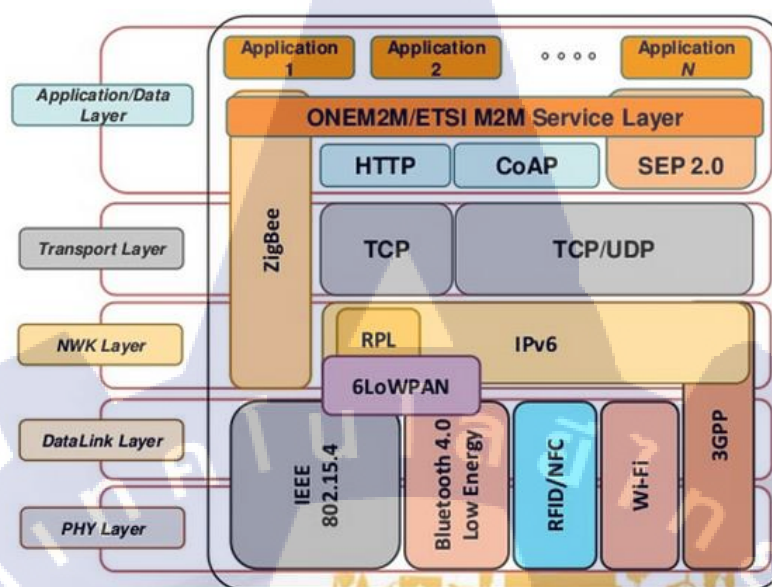
(3) การจำกัดขนาดแพ็กเก็ต แอปพลิเคชันใน 6LoWPAN นั้นคาดหวังให้มีขนาดแพ็กเก็ตที่เล็ก ขั้นที่เพิ่มเข้ามาสำหรับไอพีจะส่งผ่านในเฟรมเดียวโดยที่ไม่ต้องแบ่งส่วนและประกอบใหม่มากเกินไป นอกจากนี้โปรโตคอลจะต้องถูกออกแบบและเป็นส่วนตัว “Control/Protocol packet” ให้พอดีใน 802.15.4 เฟรมเดียว

(4) ข้อจำกัดในการตั้งค่าและการจัดการ อุปกรณ์ใน LoWPAN คาดว่าจะถูกปล่อยใช้งานอย่างมากมาย นอกจากนี้คาดว่าจะมีข้อจำกัดในเรื่องของการแสดงผลและการรองรับกระแสไฟฟ้า นอกจากนี้สถานที่ตั้งของอุปกรณ์อาจยากที่จะเข้าถึงตามมาด้วยโปรโตคอลที่ใช้ใน LoWPANs ควรตั้งค่าง่ายโดยเฉพาะ “out of the box” ง่ายต่อการออกแบบ อุปกรณ์เหล่านี้เมื่อเข้าถึงเครือข่ายจะมีลักษณะที่สามารถรักษาตนเองเมื่อได้รับการสืบทอดที่ผิดพลาด

(5) การค้นหาบริการ LoWPANs จำเป็นต้องใช้ Simple Service Discovery Protocol เพื่อค้นหาและควบคุมจัดการบริการ โดยอุปกรณ์

(6) ความปลอดภัย IEEE 802.15.4 บังคับให้ Data Link layer รักษาความปลอดภัยบน AES โดยที่จะยกเว้นรายละเอียดเกี่ยวกับ bootstrapping, key management และความปลอดภัยในระดับชั้นที่สูงกว่า ซึ่งแน่นอนว่าเป็นทางออกของการรักษาความปลอดภัยสำหรับอุปกรณ์ LoWPAN ซึ่งจะต้องเลือกแอปพลิเคชันอย่างระมัดระวัง

2.4.3 ชั้นการทำงานของ 6LoWPAN



รูปที่ 2.7 ชั้นการทำงานของ 6LoWPAN

จากที่กล่าวมาจะเห็นถึงลักษณะของ 6LoWPAN ไปแล้วโดยการทำงานของ 6LoWPAN จะอยู่ที่ชั้น Data Link และชั้น Network ซึ่งจะทำงานร่วมกับ RPL โพรโตคอลซึ่งจะกล่าวถึงภายหลัง

2.5 ระบบปฏิบัติการ Contiki

Contiki เป็นระบบปฏิบัติการแบบโอเพนซอร์สสำหรับ Internet of Things เพื่อเชื่อมต่ออุปกรณ์ขนาดเล็กราคาถูก หรือไมโครคอนโทรลเลอร์ที่มีพลังงานต่ำไปยังอินเทอร์เน็ต

Contiki รองรับการใช้พลังงานต่ำเพื่อสื่อสารบนอินเทอร์เน็ต รองรับมาตรฐานไอพีรุ่นที่สี่และไอพีรุ่นที่หก พร้อมทั้งมาตรฐานเครือข่ายไร้สายพลังงานต่ำเช่น 6LoWPAN RPL Coap ContikiMAC ของ Contiki และ sleepy routers แม้ว่าทำงานเป็นเราเตอร์ไร้สายสามารถใช้งานได้ด้วยแบตเตอรี่

ด้วย Contiki ง่ายต่อการพัฒนาเพราะแอปพลิเคชันถูกเขียนด้วยภาษาซีและมี Cooja simulator ซึ่งสามารถจำลองเครือข่ายก่อนติดตั้งบนอุปกรณ์จริงและ Instant Contiki ที่สามารถเข้าถึงการพัฒนา ด้วยการดาวน์โหลดเพียงครั้งเดียว

2.5.1 การติดตั้ง Contiki

มีหลายวิธีในการติดตั้ง Contiki โดยติดตั้งจากซอร์สโค้ดหรือใช้การจำลองระบบปฏิบัติการก็ได้ขึ้นอยู่กับความชื่นชอบและเวลาที่มี เพื่อใช้งาน Contiki นั้นจำเป็นต้องมี 3 อย่าง

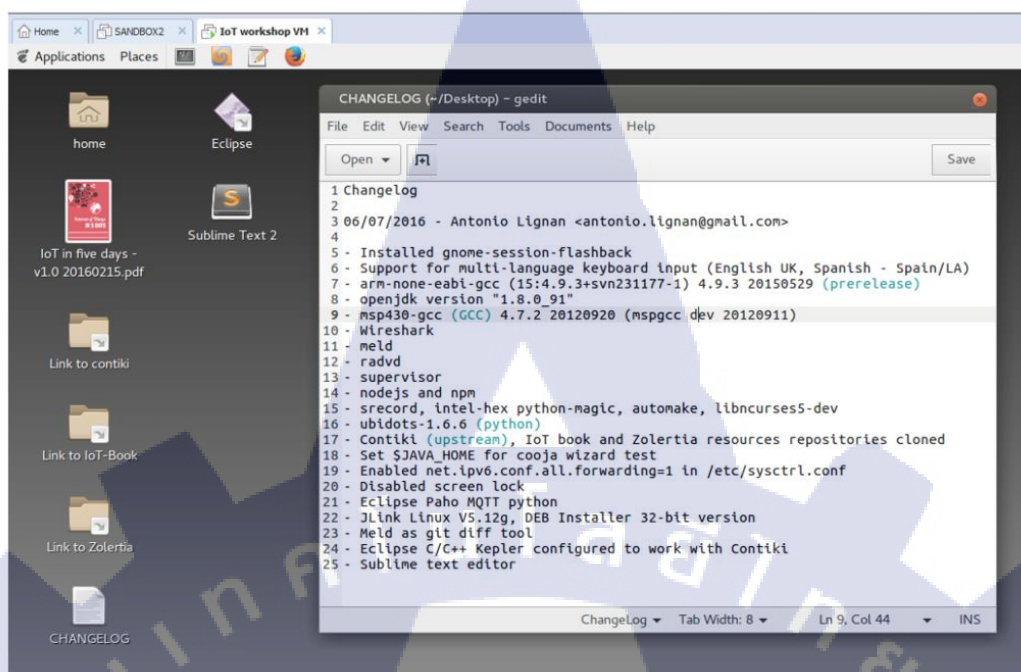
- 1.ซอร์สโค้ดของ Contiki
- 2.แพลตฟอร์มเป้าหมาย (เป็นการจำลองแพลตฟอร์มหรืออุปกรณ์จริงก็ได้)
- 3.เครื่องมือสำหรับแปลซอร์สโค้ดสำหรับแพลตฟอร์มเป้าหมาย

2.5.2 เครื่องมือจำลอง Instant contiki

Instant Contiki นั้นรวมถึงสิ่งจำเป็นทั้งหมดของ Contiki เพื่อใช้พัฒนาในการดาวน์โหลดเพียงครั้งเดียว ทำงานบนเครื่องจำลอง Ubuntu Linux และมีระบบปฏิบัติการ Contiki พร้อมทั้งเครื่องมือที่ใช้พัฒนาเครื่องมือที่ใช้แปลโปรแกรม และตัวจำลองถูกติดตั้งไว้เรียบร้อยแล้ว สามารถดาวน์โหลด Instant Contiki ได้จากเว็บไซต์ Contiki ได้โดยตรง

โดยใช้ VMware เปิดไปที่ไฟล์ Instant_Contiki_Ubuntu_12.04_32-bit.vmx ถ้ามีการแจ้งเตือนเกี่ยวกับ VMware ให้เลือก I copied It และรอการจำลองของ Ubuntu linux ขึ้นขึ้น เข้าสู่ระบบ Instant Contiki โดยผู้ใช้งานและรหัสผ่านคือ user และอย่าเพิ่งปรับรุ่นในตอนนี้อย่าอัปเดต Contiki ก่อนแล้วค่อยปรับรุ่น

เครื่องมือจำลองอย่างเป็นทางการของ IoT 5 days คือ IoT Workshop โดยจะมีทุกอย่างที่มีใน Instant Contiki แต่จะสร้างขึ้นบน Ubuntu Server 16.04 LTS (xenial) แทนเพิ่มเติมด้วยหลายโปรแกรมสำเร็จรูป เพื่อใช้พัฒนาได้ง่ายรวมไปถึง Eclipse IDE Workspace ที่ตั้งค่าไว้เรียบร้อยแล้ว



รูปที่ 2.8 IoT workshop VM

Folder	Description	Zolertia files
examples	Ready to build examples	examples/zolertia, examples/cc2538-common
app	Contiki applications	-
cpu	Specific MCU files	msp430, cc2538
dev	External chip and devices	cc2420, cc1200
platform	Specific files and platform drivers	z1, zoul
core	Contiki core files and libraries	-
tools	Tools for flashing, debugging, simulating, etc.	zolertia, sky
doc	Self-generated doxygen documentation	-
regression-tests	nightly regression tests	-

รูปที่ 2.9 โครงสร้าง Contiki

ถ้าหากว่าดึงข้อมูลเลียนแบบมาจากกึ่งของ IoT-Workshop (หรือว่าใช้เครื่องจำลองของ IoT in five days) จะพบเพิ่มนี้ [contiki/examples/zolertia/tutorial](https://github.com/contiki/examples/zolertia/tutorial) เนื้อหาของ Tutorial จะมี

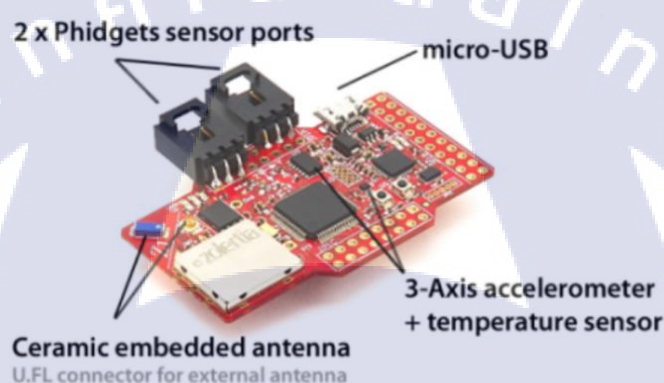
01-basics พื้นฐานของ Contiki (เวลา, GPIO, LEDs)

02-ipv6 เครือข่ายไร้สาย, พื้นฐานสัญญาณวิทยุ, การใช้ IPv6/6LoWPAN(UDP/TCP) และ
ขอบของเครือข่าย

03-coap ตัวอย่างการใช้งาน CoAP server

04-mqtt ตัวอย่างการใช้งาน MQTT client

2.6 Zolertia Z1



รูปที่ 2.10 Z1 motes

Z1 แพลตฟอร์มมีเป้าหมายเพื่อพัฒนาแพลตฟอร์มสำหรับเครือข่ายเซ็นเซอร์ไร้สาย (WSN) ออกแบบมาสำหรับนักวิจัย นักพัฒนา ผู้ที่สนใจหรือเป็นงานอดิเรก แพลตฟอร์มนี้สามารถทำงานร่วมกับ Tmote™ family motes ได้อย่างสมบูรณ์ ด้วยบางอย่างที่ถูกปรับให้ดีขึ้นจึงทำให้มีประสิทธิภาพในบางด้านสูงกว่าเดิมประมาณ 2 เท่า

Z1 motes ใช้รุ่นที่สองของ MSP430F2617 ใช้พลังงานต่ำ 16 บิต RISC หน่วยประมวลผล 16 MHz MCU, มีแรมขนาด 8 kB และรอมขนาด 92 kB และใช้ CC2420 อุปกรณ์รับส่งข้อมูลวิทยุ ทำงานร่วมกับ IEEE 802.15.4 ได้ ซึ่งทำงานบนคลื่น 2.4 GHz ส่งข้อมูลได้ถึง 250 kbps

Zolertia Z1 สามารถทำงานบนระบบปฏิบัติการ Tiny, ระบบปฏิบัติการ Contiki, ระบบปฏิบัติการ OpenWSN และระบบปฏิบัติการ RIOT ถูกนำมาใช้งานอย่างต่อเนื่องกว่า 5 ปีในมหาวิทยาลัย การวิจัย และศูนย์พัฒนา เป็นสินค้าในเชิงพาณิชย์มากกว่า 43 ประเทศ และมีงานตีพิมพ์ทางวิทยาศาสตร์ออกมากกว่า 50 งาน

ลักษณะของ Z1 แพลตฟอร์ม เป็นไมโครคอนโทรลเลอร์ที่ใช้พลังงานต่ำมากและตัวรับส่งสัญญาณ 2.4GHz มี 2 คิยิตัลเซ็นเซอร์พร้อมใช้งาน (อุณหภูมิและความเร่ง 3 แกน) ใช้ USB สำหรับติดตั้งโปรแกรมได้ง่าย จ่ายพลังงานได้หลายวิธี สามารถใช้ถ่าน (2xAA, 2xAAA) ถ่านนาฬิกา (ได้ถึง 3.6V) ผ่าน USB ต่อสายสองเส้น โดยตรงมายังแหล่งพลังงานเลข USB VCC และ GND คิยิตัลสามารถต่อพินขยายช่องได้ สามารถต่อไปยังพิน แหล่งพลังงานจาก 4V ไป 5.25V ซึ่งจะถูกรับเป็น 5V และ 3V

2.7 เซ็นเซอร์

เซ็นเซอร์คือตัวแปลงสัญญาณ โดยมีวัตถุประสงค์เพื่อตรวจจับหรือวิเคราะห์ลักษณะของ สภาพแวดล้อมโดยให้ผลลัพธ์ที่สอดคล้องกัน โดยทั่วไปจะเป็นสัญญาณไฟฟ้าที่เกี่ยวข้องกับตัวแปรที่วัด

2.7.1 อนาล็อกเซ็นเซอร์

อนาล็อกเซ็นเซอร์โดยทั่วไปจะเชื่อมต่อไปยัง ADC (Analogue to Digital Converter) เพื่อ แปลงสัญญาณอนาล็อกที่ต่อเนื่องให้อ่านค่าเป็นคิยิตัลในมิลลิโวลต์ คุณภาพและผลลัพธ์ในการวัดจะขึ้นอยู่กับ ทั้ง ADC (ได้ถึง 12 บิต ใน Z1 และ Re-mote) และ Sampling Frequency

Z1 mote และ RE-Mote เซ็นเซอร์แรงดันไฟฟ้าให้เป็นช่องรับสัญญาณ ADC เตรียมไว้แล้ว ตลอดจนนำไปใช้ในการอ่านค่าอนาล็อกเซ็นเซอร์ภายนอกได้

อนาล็อกเซ็นเซอร์นั้นสามารถเชื่อมต่อ Z1 ผ่านพอร์ทอนาล็อก

อนาล็อกเซ็นเซอร์ที่ทดลองแล้วสามารถใช้งานได้

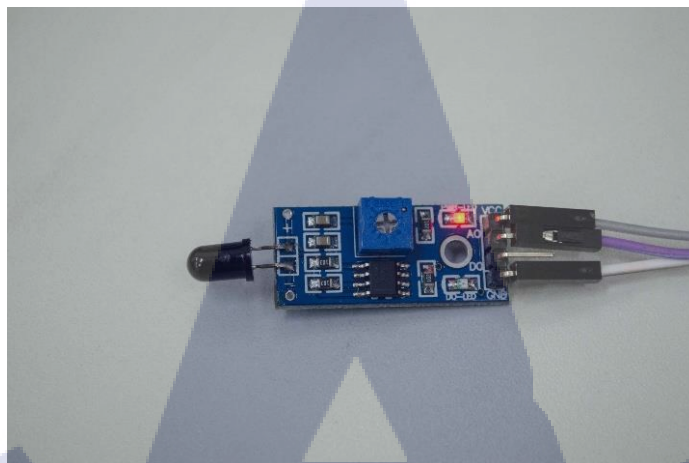
PIR Motion Sensor ใช้ในการตรวจสอบการเคลื่อนไหว

Grove - Light Sensor ใช้ในการวัดค่าของแสง

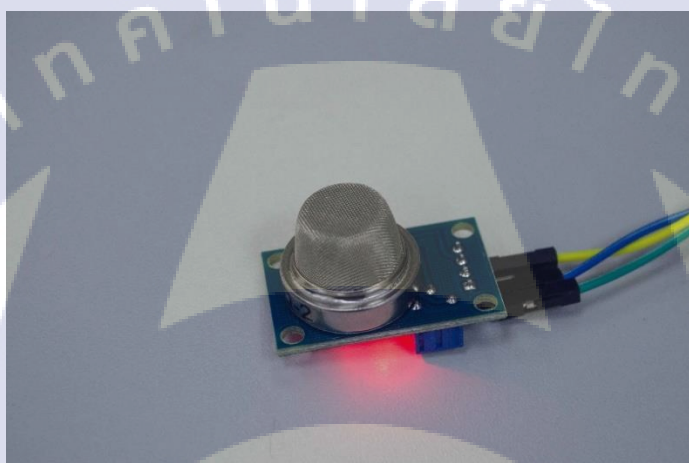
Grove - Rotary Angle Sensor ใช้วัดค่าของการหมุน

Grove - Loudness Sensor ใช้วัดความดังของเสียง

โดยเซ็นเซอร์ไฟและแก๊สที่นำมาใช้ในงานนี้มีลักษณะเป็นอนาล็อกเซ็นเซอร์โดยค่าที่แสดง ออกมานั้นมีลักษณะที่ต่างกัน คือ ค่าปกติของเซ็นเซอร์ไฟจะอยู่ที่ประมาณ 4,000 และหากตรวจพบค่าจะลดลง มา ส่วนค่าปกติของเซ็นเซอร์แก๊สจะอยู่ที่ 300 หากตรวจพบแก๊ส จะขึ้นไปถึงประมาณ 3,000



รูปที่ 2.11 เซ็นเซอร์ไฟ



รูปที่ 2.12 เซ็นเซอร์แก๊ส

2.7.2 ดิจิทัลเซ็นเซอร์

ดิจิทัลเซ็นเซอร์โดยทั่วไปจะจัดการผ่าน Digital communication protocol เช่น I2C, SPI, 1-Wire, Serial หรือขึ้นอยู่กับผู้ผลิต โปรโตคอลที่เป็นกรรมสิทธิ์ โดยทั่วไปตามสัญญาอนุญาต ดิจิทัลเซ็นเซอร์นั้นจะมีคำสั่งที่เพิ่มเข้ามา เช่น เปิด ปิด หรือจัดการข้อผิดพลาด

Z1 mote มีดิจิทัลเซ็นเซอร์ 2 อย่างติดตั้งมาแล้วคืออุณหภูมิและความเร่ง 3 แกน

ADXL345 ใช้วัดความเร่งมีตัวอย่างที่ชื่อ test-adxl345.c ใน examples/zolertia/z1 ที่แสดงถึงการทำงานของเซ็นเซอร์

TMP102 เซ็นเซอร์อุณหภูมิแบบดิจิทัล ซึ่งติดตั้งมาบน Z1 แพลตฟอร์มมีความแม่นยำ ± 0.5 เซลเซียส และสามารถวัดอุณหภูมิได้ตั้งแต่จาก -25 เซลเซียส ถึง 85 เซลเซียส ซึ่งตัวอย่าง examples/zolertia/z1/test-102.c จะแสดงถึงการทำงานของเซ็นเซอร์

External digital sensor สามารถเชื่อมต่อกับ Z1 ได้

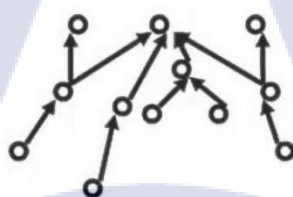
SHT25 Humidity and Temperature ใช้วัดอุณหภูมิและความชื้นด้วยเซ็นเซอร์ตัวเดียว

2.8 โพรโทคอลหาเส้นทางสำหรับอุปกรณ์พลังงานต่ำและลดทอนเครือข่าย

โพรโทคอลหาเส้นทางสำหรับอุปกรณ์พลังงานต่ำและลดทอนเครือข่าย (Routing Protocol for Low power and Lossy Networks , RPL) คือโพรโทคอลหาระยะของเส้นทาง (Distance vector routing protocol) ที่หาเส้นทางบน Destination or DoDAGs

ก่อนที่จะทำความเข้าใจเกี่ยวกับ RPL ต้องรู้ก่อนว่า DODAGs นั้นมีรูปแบบเป็นอย่างไร ต้องเข้าใจถึงศัพท์เฉพาะทางและอะไรคือหัวใจหลักของโพรโทคอลการหาเส้นทางนี้

DAG (Directed Acyclic Graph) เป็นกราฟที่กระบวนการไม่สมบูรณ์พวกเราจะเห็นกราฟจำพวกนี้ใน spanning tree ตามรูปภาพที่ 2.13



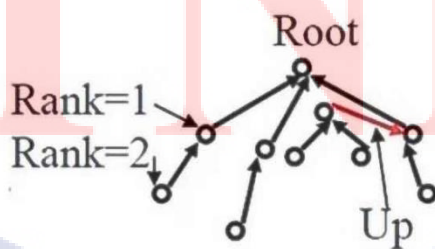
รูปที่ 2.13 DAG

Root เป็นปลายทางของโหนดใน DAG จะไม่มีขาออก

Up เป็นขอบใดก็ได้ที่ส่งข้อมูลโดยตรงไปยัง Root

Down เป็นขอบใดก็ได้ที่ส่งข้อมูลออกจาก Root

Destination Oriented DAG (DODAG) เป็น DAG ชนิดพิเศษที่ทุกโหนดจะต้องส่งไปให้ถึงปลายทางเดียวกันตามภาพที่ 2.14



รูปที่ 2.14 DODAG

Objective Function จะช่วยตัดสินใจว่าอยู่ใกล้หรือไกลจาก Root Objective function จะตัดสินใจตามโปรแกรมหรือคนออกแบบเป็นสิ่งที่ต้องการให้มีขนาดเล็ก สามารถเป็นพลังงานหรือแฝงอยู่และถ้าตัดสินใจว่าต้องการจะย่อให้เล็กลง ซึ่งจะมอบหมายเลขลำดับให้

Rank ตามรูปภาพที่ 2.15 แสดงระยะห่างจาก Root

RPL instance เมื่อมี 1 หรือมากกว่า 1 DODAGs ทุกๆ DODAGs จะเป็น instance ตามรูปภาพที่ 2.13 แสดงถึง Two RPL Instances



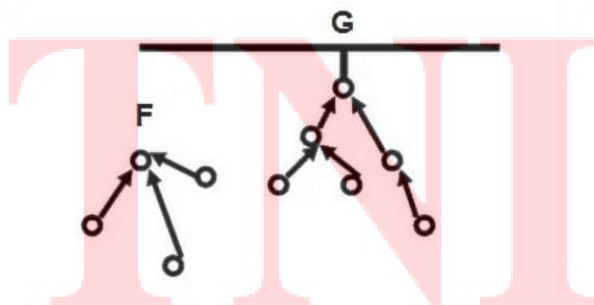
รูปที่ 2.15 Two RPL Instances

DODAG ID ทุกๆ DODAG จะมี IPv6 ID 128 บิต จะถูกมอบจาก Root เท่านั้นและตราที่ Root ไม่เปลี่ยน ID ก็จะไม่เปลี่ยน

DODAG version ทุกๆการเปลี่ยนรูปของ DODAG หมายถึงการเปลี่ยนรูปแบบ

GOAL คือที่ DODAG ต้องการจะไปให้ถึง สามารถเป็นเครือข่ายแบบมีสาย Goal จะเปลี่ยนไปตาม Objective Function ใน Objective Function เล็กไปถึงการทำให้มีขนาดเล็ก อย่างไรก็ตาม GOAL นั้นหมายถึงที่ๆต้องการจะไป

Floating เมื่อ DODAG ไม่เชื่อมต่อหรือยังไม่ถึง Goal จะถูกเรียกว่า Floating ในรูปภาพที่ 2.16 แสดงถึง Floating DODAG



รูปที่ 2.16 Floating และ Grounded

Parent คือที่ที่ลูกศรชี้ไปและ Child คือที่ๆลูกศรออกมาโดย Parents สามารถมีได้หลาย Childs เช่นเดียวกับ Child ที่สามารถมีได้หลาย Parents

Sub-DODAG เป็น Subtree ใดก็ได้ที่ถูกมอบโดย DODAG

Storing โหนดจะเก็บตารางการหาเส้นทางไว้ทั้งหมด โดยจะรู้การเดินทางจากโหนดหนึ่งไปยังโหนดอื่นๆ

Non-Storing โหนดที่ไม่รู้ตารางการหาเส้นทางโดยจะรู้แค่ Parent

DODAG ที่สมบูรณ์จะถูกยกเว้นจาก Root เพื่อรักษารูปร่างโดยจะต้องมีทั้ง Storing และ Non-Storing โดย Root จะต้องเป็น Storing เสมอ

RPL Control-messages มี Control Message 5 รูปแบบใน Spanning tree

1. DODAG information Object (DIO) ข้อความนี้จะมัลติแคสต์ลงด้านล่าง จะให้โหนดใน DODAG มัลติแคสต์ข้อความนี้เพื่อให้โหนดอื่นๆได้รู้ ไม่ว่าจะเป็น Grounded หรือไม่ หรือเป็น Storing หรือ non-storing ก็ตามโดยจะประกาศไปยังโหนดอื่นๆว่า “ถ้าสนใจจะเข้าร่วมให้ตอบกลับมา”

2. DODAG Information Solicitation (DIS) เมื่อไม่ได้รับการประกาศมา ถ้าโหนดต้องการเข้าร่วม DODAG จะส่ง Control Message เพื่อรู้การมีอยู่ของ DODAG ข้อความที่ส่งมาจะมีความหมายว่า “มี DODAG อยู่ไหม”

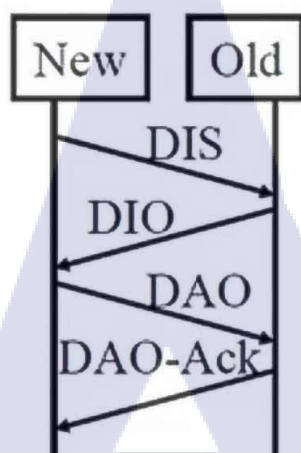
3. DODAG Advertisement Object (DAO) เป็นการร้องขอที่ส่งจาก Child ไปยัง Parent หรือ Root เป็นข้อความเพื่อขออนุญาตจาก Child เข้าร่วม DODAG

4. DAO-ACK เป็นการตอบกลับจาก Root ไป Parent หรือ Child โดยคำตอบเป็นได้ทั้งได้และไม่ได้

5. Consistency check เกี่ยวกับเรื่องของความปลอดภัย

ในภาพที่ 2.17 แสดงถึง Control Message รูปแบบต่างๆ

TNI

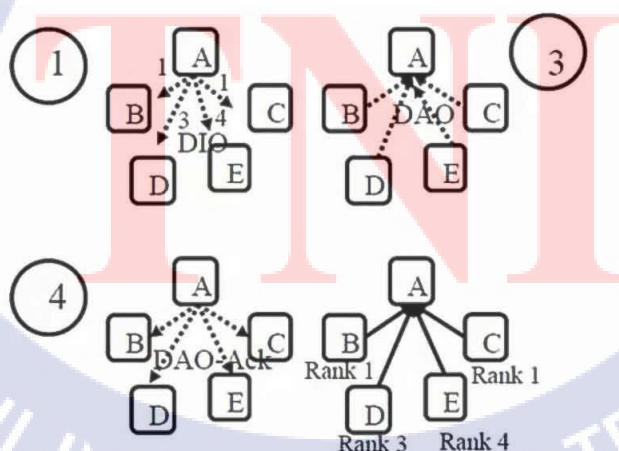


รูปที่ 2.17 Control Message

รูปแบบของ DODAG

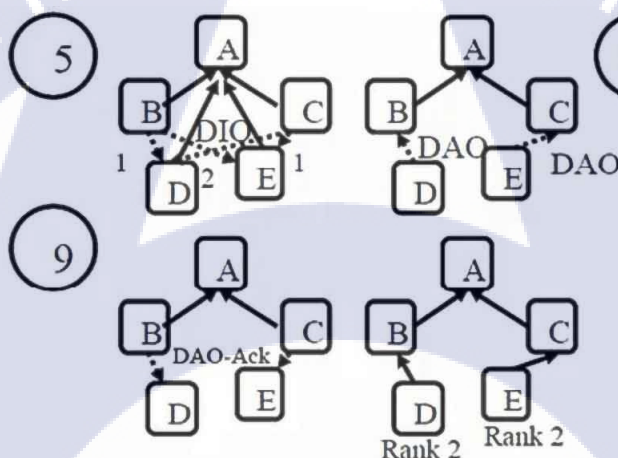
Root ใน DODAG นั้นเป็นโหนดชนิดพิเศษ โหนดทุกตัวไม่สามารถเป็น ROOT ได้ ซึ่งจะต้องถูกโปรแกรมเฉพาะทางไว้สมมุติว่ามี 5 โหนด A, B, C, D, E โดยต้องการจะสร้าง DODAG ตามรูปแบบนี้

1. โหนด A มัลติแคส DIO ออกไป
2. โหนดทั้งหมดได้รับ DIO (B, C, D, E) จะพยายามเข้าร่วมโดยไม่คำนึงถึงระยะห่าง จากนั้นจะรู้ระยะห่างจากโหนด A ว่าเป็น 1, 1, 3, 4 ตามลำดับ
3. โหนด B, C, D, E จะส่ง DAOs มาที่โหนด A
4. โหนด A ขอมรับและตอบกลับด้วย DAO-ACK



รูปที่ 2.18 รูปแบบ DODAG

5. เริ่มรอบใหม่ โดยเริ่มจากโหนดที่ใกล้ที่สุดของ Root ตั้งแต่โหนด B และโหนด C ที่อยู่ Rank 1 จะส่ง DIOs
6. โหนด D รับ DIOs และคำนวณระยะห่างจากทางโหนด B และโหนด C คือ 2 และ 1 ตามลำดับ
7. เช่นเดียวกับโหนด E ที่รับ DIOs มาและคำนวณระยะทางจากโหนด B และโหนด C คือ 2 และ 1 ตามลำดับ
8. โหนด D ที่อยู่ใกล้โหนด B มากกว่าและโหนด E ที่อยู่ใกล้โหนด C มากกว่าเพราะฉะนั้นโหนด D จะส่ง DAO กลับไปยังโหนด B และโหนด E จะส่ง DAO กลับไปยังโหนด C
9. เพื่อให้ DODAG สมบูรณ์โหนด C และโหนด B ตอบ DAO-ACK กลับไปยังโหนด D และโหนด E ตามรูป 2.19



รูปที่ 2.19 รูปแบบ DODAG (2)

2.9 IPv6 Tunnel Broker

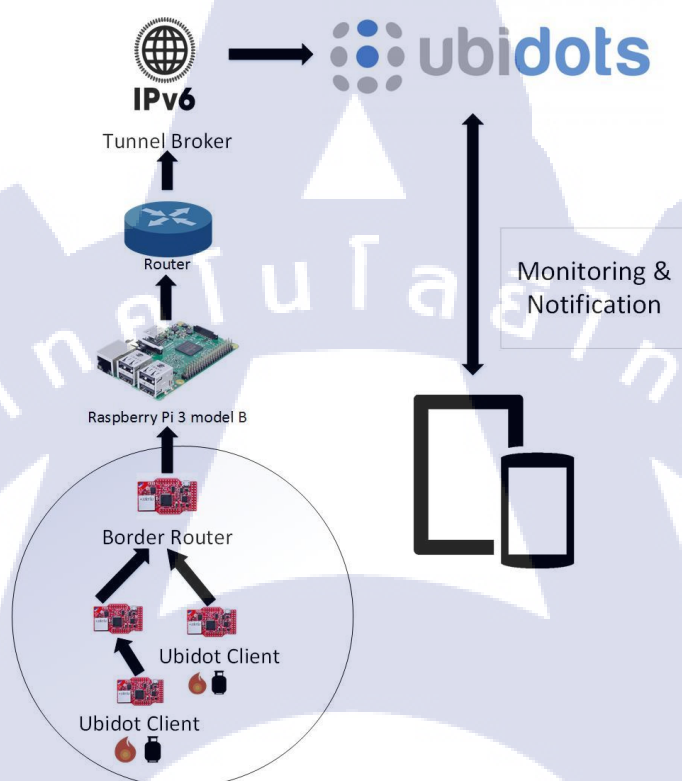
Zolertia Z1 เป็นอุปกรณ์ที่ทำการสื่อสารบนไอพีรุ่นที่หกซึ่งภายในห้องทดลองของสถาบันเทคโนโลยีไทย-ญี่ปุ่นยังใช้ไอพีรุ่นที่สี่จึงจำเป็นต้องใช้บริการของ Tunnel Broker ในการแปลงเป็นไอพีรุ่นที่หกเพื่อการสื่อสารโดยในการใช้งานมี 2 เงื่อนไขสำคัญคือ

1. ต้องเปิดพอร์ตสำหรับ ICMP เพื่อให้ทางเซิร์ฟเวอร์ตอบกลับมาได้
2. เปิดไอพีโปรโตคอล 41 สำหรับการเพิ่มพื้นที่บนส่วนหัวของไอพีรุ่นที่สี่เพื่อห่อหุ้มแพ็กเก็ต ไอพีรุ่นที่หกเข้าไป

เมื่อทำตาม 2 ข้อนี้นี้จะทำให้สามารถใช้บริการ Tunnel Broker เพื่อใช้งานไอพีรุ่นที่หกได้

2.10 Ubidots

Ubidots คือ IoT คลาวด์แพลตฟอร์มที่จะช่วยในการสร้างแอปพลิเคชันซึ่งเก็บข้อมูลจากการตรวจวัดมาใช้ให้เกิดประโยชน์โดยมีคุณสมบัติที่หลากหลาย

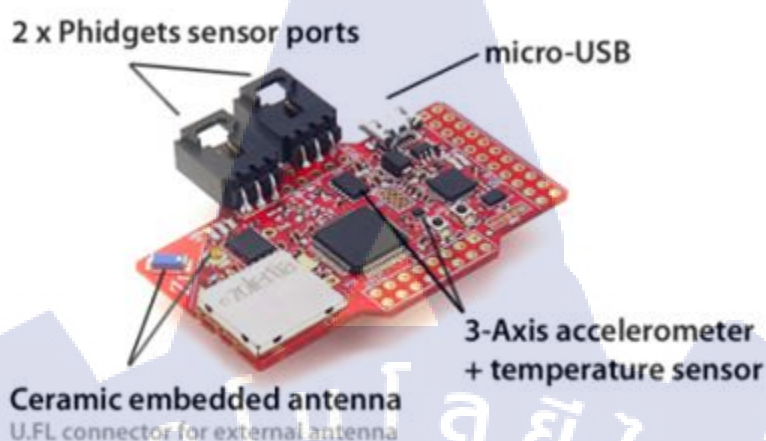


รูปที่ 2.20 ตัวอย่างการทำงาน

รูปภาพที่ 2.20 แสดงถึงการทำงานโดยเริ่มจากเมื่อ Z1 ที่ติดตั้งโปรแกรม Ubidots Client ได้รับความแจ้งเตือนจากเซ็นเซอร์ส่งค่ามายัง

Z1 ที่ติดตั้งโปรแกรม Border Router ซึ่งจะส่งค่าผ่าน USB มายังคอมพิวเตอร์โดยจากคอมพิวเตอร์จะส่งออกไปยังเราเตอร์ด้วยไอพีรุ่นที่สี่โดยส่งค่าไปยัง Tunnel Broker เพื่อใช้บริการ Tunnel แปลงเป็นไอพีรุ่นที่หกและส่งค่าไปยัง Ubidots โดยสามารถดูจากโทรศัพท์หรือเมื่อเกิดเหตุการณ์จะสามารถแจ้งเตือนมายังอีเมลหรือเบอร์โทรศัพท์ได้

2.11 อุปกรณ์ Zolertia Z1



<http://zolertia.sourceforge.net/wiki/index.php/Z1>

รูปที่ 2.21 อุปกรณ์ Zolertia Z1

อุปกรณ์ Zolertia Z1 เป็นเซ็นเซอร์เครือข่ายไร้สายพลังงานต่ำ (Wireless Sensor Network, WSN) เป็นโมดูลสำหรับการพัฒนางานแบบอนกประสงค์สำหรับนักพัฒนา ผู้ที่สนใจ และมีมือสมัครเล่น

อุปกรณ์ Zolertia Z1 เป็นอุปกรณ์ที่มีความยืดหยุ่นสูง สามารถเชื่อมต่อกับแหล่งพลังงาน เซ็นเซอร์ (sensor) และช่องทางการเชื่อมต่อเป็นจำนวนมาก และรองรับระบบปฏิบัติการแบบเปิดที่มีการทำงานของ WSN เช่น TinyOS และ Contiki โดยมีการรองรับสแต็กของระบบเครือข่าย ซึ่งรวมถึงซิกโลวแพน (6LowPAN), Texas Instrument's SimplicTI, Z-Stack และในอนาคตจะรองรับสแต็กของ ซิกบี (Zigbee) แบบเปิด

การทำงานของอุปกรณ์ Zolertia Z1 จะเป็นโมดูลไร้สายประหยัดพลังงานที่ทำงานบนมาตรฐาน IEEE 802.15.4 และโพรโทคอล Zigbee ซึ่งจะช่วยให้นักพัฒนาเซ็นเซอร์เครือข่ายไร้สาย สามารถทดสอบและพัฒนาโปรแกรมสำเร็จรูปของตัวเองและสร้างตัวต้นแบบที่ดีในระหว่างการพัฒนาและการรองรับฮาร์ดแวร์ได้อย่างยืดหยุ่น

ตัวอย่างโปรแกรมประยุกต์

- 1) การตรวจสอบสุขภาพส่วนบุคคล
- 2) การตรวจสอบด้านสิ่งแวดล้อม
- 3) เครื่องตรวจจับเวลาเกิดเหตุฉุกเฉิน
- 4) อุปกรณ์เพื่อความปลอดภัยและการกู้ภัย
- 5) การตรวจสอบแบบอัตโนมัติแบบระยะยาว

- 6) การวัดค่าพลังงาน
- 7) การตรวจสอบทางการเกษตร

ลักษณะของผลิตภัณฑ์

- 1) เป็นแพลตฟอร์มการพัฒนาตัวต้นแบบของเซ็นเซอร์เครือข่ายไร้สาย ได้อย่างรวดเร็ว
- 2) มีคุณภาพระดับอุตสาหกรรม รองรับอุณหภูมิตั้งแต่ -40 องศาเซลเซียส ถึง 85 องศาเซลเซียส
- 3) เพิ่มช่องทางการเชื่อมต่อเป็น 52 พิน
- 4) ใช้หน่วยประมวลผล MSP430 รุ่นที่สอง แบบประหยัดพลังงานมากเป็นพิเศษ
- 5) มีเซ็นเซอร์วัดความเคียงแบบ 3 ทิศทาง และระบบเน็ตเวิร์กใช้ 2.4 GHz มาตรฐาน IEEE 802.15.4, 6LowPAN และรองรับ Zigbee
- 6) มีเซ็นเซอร์วัดอุณหภูมิดิจิทัลที่ใช้พลังงานต่ำ (ความแม่นยำ ± 0.5 องศาเซลเซียส (ในช่วง -25 ถึง 85 องศาเซลเซียส)
- 7) มีช่องต่อเสาสัญญาณเสริม
- 8) มีช่องต่อ Micro-USB สำหรับการจ่ายพลังงานและการเขียนโปรแกรม

2.12 อุปกรณ์ Raspberry Pi



<https://www.raspberrypi.org/weekly/connected/>

รูปที่ 2.22 อุปกรณ์ Raspberry Pi

Raspberry Pi เป็นเครื่องคอมพิวเตอร์ขนาดเล็กที่สามารถทำงานได้เหมือนกับคอมพิวเตอร์เกือบทุกอย่าง รองรับ HDMI มีช่องต่อยูเอสบี สำหรับการนำเมาส์และคีย์บอร์ดมาต่อเพิ่ม และระบบการจ่ายไฟใช้หัว Mini-USB เหมือนกับสายชาร์จมือถือ และมีพอร์ต General-purpose input/output (GPIO) สำหรับการเชื่อมต่อเซ็นเซอร์และอุปกรณ์ควบคุมต่างๆ เช่น เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion sensor)

ระบบปฏิบัติการที่ Raspberry Pi รองรับมีหลากหลาย เช่น Raspbian, Ubuntu MATE, Windows 10 IOT Core, OSMC, LIBREELEC, PiNET, Weather Station, Snappy Ubuntu, RISC OS เป็นต้น ซึ่งแต่ละระบบปฏิบัติการก็จะมีข้อดีแตกต่างกันไป ซึ่งที่ใช้นานนี้คือ Raspbian

Raspbian เป็นระบบปฏิบัติการที่มาจากผู้ผลิตเอง ซึ่งจะมาพร้อมกับโปรแกรมเขียนโค้ดภาษาต่างๆ เช่น ภาษา Python, ภาษา Java, โปรแกรม Scratch และโปรแกรม Sonic Pi เป็นต้น

2.13 เว็บไซต์ TunnelBroker.net



<https://www.tunnelbroker.net/>

รูปที่ 2.23 โลโก้เว็บไซต์ tunnelbroker.net

Hurricane Electric เป็นเว็บไซต์ที่ให้บริการในการทำอุโมงค์ไอพีรุ่นที่หก (IPv6 Tunnel Broker) สำหรับผู้ที่มีไอพีรุ่นที่สี่ (IPv4) อยู่ และต้องการจะใช้บริการไอพีรุ่นที่หก (IPv6) ด้วย โดยการทำอุโมงค์ (Tunnel) เป็นการทำให้ไอพีรุ่นที่หก สามารถวิ่งผ่านการเชื่อมต่อเครือข่ายไอพีรุ่นที่สี่ได้ ฟังก์ชันของ TunnelBroker.net นั้น จะมีเราเตอร์ (Router) ทำการแจกไอพีรุ่นที่หกมาให้ ส่วนทางฝั่งของเครื่องลูกข่ายก็ต้องมีเราเตอร์ ที่รองรับการทำงานของไอพีรุ่นที่หก และต้องเชื่อมต่ออินเทอร์เน็ตผ่านไอพีรุ่นที่สี่ จึงจะสามารถใช้งานอุโมงค์ไอพีรุ่นที่หกได้

2.14 ไอพีรุ่นที่สี่ (Internet Protocol version 4, IPv4)

ไอพีรุ่นที่สี่ เป็นอินเทอร์เน็ตโพรโทคอล (Internet Protocol, IP) รุ่นที่สี่ โดยไอพีจะอยู่ในระดับเน็ตเวิร์กเลเยอร์ (Network Layer) ทำหน้าที่จัดการเกี่ยวกับแอดเดรสและข้อมูล และควบคุมการส่งข้อมูลที่ใช้ใน

การหาเส้นทางของแพ็กเก็ต (Package) ซึ่งการหาเส้นทางของไอพี จะทำการหาเส้นทางที่ดีที่สุด และสามารถเปลี่ยนแปลงเส้นทางของข้อมูลในระดับดาต้าลิงก์เลเยอร์ (Data Link Layer)

การเชื่อมต่อไอพี เพื่อทำการส่งข้อมูลจะเป็นแบบ connectionless หรือเกิดการเชื่อมต่อข้อมูลในทุกๆ ครั้งของการส่งข้อมูล 1 ดาต้าแกรม (datagram) โดยจะไม่ทราบถึงข้อมูลดาต้าแกรมที่ส่งก่อนหน้าหรือส่งตามมา แต่การส่งข้อมูลใน 1 ดาต้าแกรม อาจเกิดการส่งได้หลายครั้งในกรณีที่มีการแบ่งข้อมูลออกเป็นส่วนย่อยๆ (fragmentation) และจะถูกนำไปรวมเป็นดาต้าแกรมเดิมเมื่อถึงปลายทาง

ไอพีแอดเดรสของไอพีรุ่นที่สี่ เป็นระบบ 32 บิต หรือสามารถระบุตัวเลขได้ตั้งแต่ 0.0.0.0 ถึง 255.255.255.255 ไอพีแอดเดรสของไอพีรุ่นที่สี่ จะเขียนโดยใช้เลขฐานสิบจำนวนจำนวนสี่หลักโดยใช้จุดคั่นระหว่างแต่ละหลัก โดยไอพีรุ่นที่สี่ทั้งหมดจะถูกแบ่งออกเป็นคลาสชนิดต่างๆ เพื่อจุดประสงค์การใช้งานต่างๆ ดังต่อไปนี้

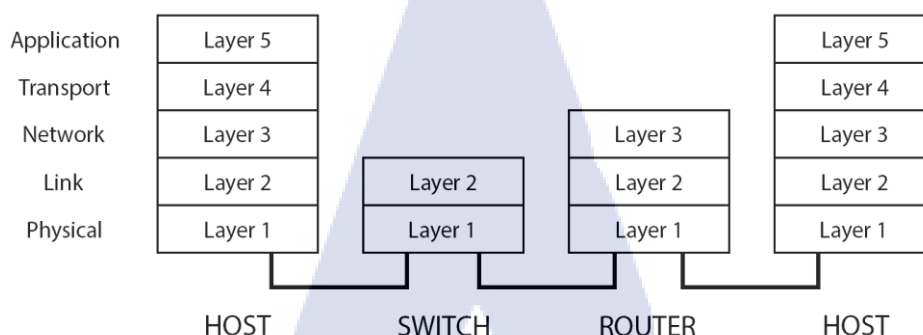
ตารางที่ 2.1 ไอพีแอดเดรสและโฮสต์ในแต่ละคลาส

คลาส	ไอพีเริ่มต้น	ไอพีสิ้นสุด	เน็ตไอดี(บิต)	โฮสต์ไอดี(บิต)
A	0.0.0.0	127.255.255.255	8	24 = 16777216
B	128.0.0.0	191.255.255.255	16	16 = 65536
C	192.0.0.0	223.255.255.255	24	8 = 256
D	224.0.0.0	239.255.255.255	-	มัลติแคสต์
E	240.0.0.0	247.255.255.255	-	สำรอง

2.15 ไอพีรุ่นที่หก (Internet Protocol version 6, IPv6)

2.15.1 แนวคิดของ IPv6

เราจำเป็นต้องรู้จักฐานล่าสุดของอินเทอร์เน็ตโพรโทคอล เพื่อที่จะเข้าใจว่าทำไมไอพีรุ่นที่หกถึงมีประโยชน์กับ Internet of Things และไอพีรุ่นที่หกเกี่ยวข้องกับโพรโทคอลอื่นๆ อย่างไร เช่น 6LoWPAN ถ้าหากมีความคุ้นเคยมาก่อนเกี่ยวกับ บิต, ไบต์, สแต็กของเครือข่าย, แพ็กเกจ, ส่วนหัวของไอพี และอื่นๆ คุณควรเข้าใจว่าไอพีรุ่นที่หก มีโพรโทคอลที่ต่างกัน และไม่เข้ากันกับไอพีรุ่นที่สี่



รูปที่ 2.24 สเต็กของอินเทอร์เน็ตโปรโตคอล

ไอพีรุ่นที่หกทำงานในชั้นที่สามที่เรียกว่า เน็ตเวิร์คเลเยอร์ (network layer) ส่วนของข้อมูลที่ถูกจัดการในชั้นที่สามเรียกว่า แพ็กเกจ อุปกรณ์ที่เชื่อมต่อกับเป็นได้ทั้งโฮส (host) และเราเตอร์

2.15.2 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks)

6LoWPAN เป็นมาตรฐานในการสื่อสารของไอพีรุ่นที่หก บนมาตรฐานเทคโนโลยีการสื่อสารไร้สาย IEEE 802.15.4 โดยจะทำหน้าที่เป็นชั้นในการปรับเปลี่ยนระหว่างมาตรฐานไอพีรุ่นที่หก และการใช้พลังงานต่ำและการสื่อสาร ไร้สายแบบถูกลดทอน ที่ถูกนำเสนอโดย IEEE 802.15.4

Low power and Lossy networks (LLNs) เป็นคำที่ใช้อ้างอิงถึงระบบเครือข่ายที่สร้างจากอุปกรณ์ที่มีข้อจำกัดสูง (ข้อจำกัดของหน่วยประมวลผล, หน่วยความจำ และการใช้พลังงาน) ที่เชื่อมต่อกันด้วยความประหยัที่มีหลากหลายอย่างผ่านการเชื่อมต่อวิทยุแบบพลังงานต่ำ ซึ่งลักษณะความเร็วต่ำ, ลดคุณภาพลง, ประหยัดต้นทุน และการเชื่อมต่อที่เสถียรภาพต่ำ

LoWPAN เป็นกรณีของ LLN ที่เกิดขึ้นจากอุปกรณ์ที่ใช้งานร่วมกับมาตรฐาน IEEE 802.15.4

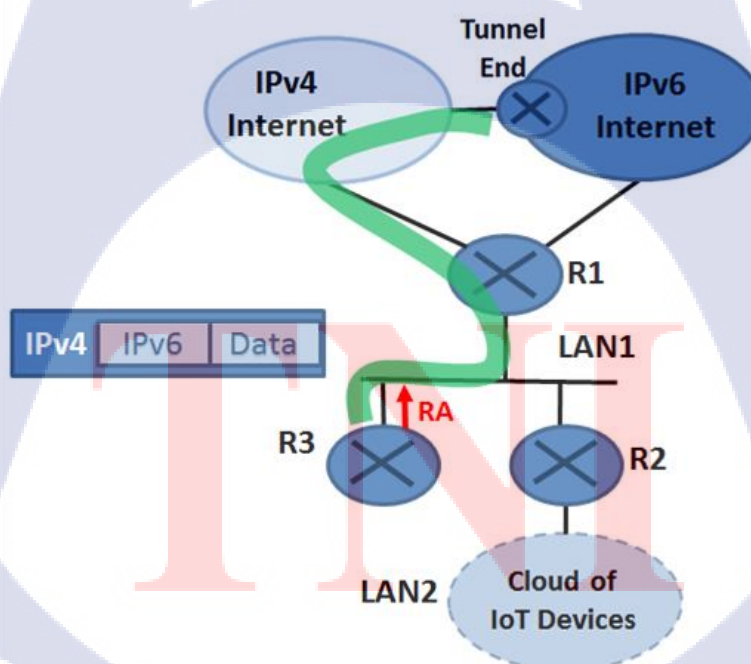
ลักษณะที่สำคัญของอุปกรณ์ที่ใช้ LoWPAN คือ

- 1) การรองรับการประมวลผลที่จำกัด คือ รูปแบบที่ต่างกัน และรอบความเร็วที่ใช้ในการประมวลผล โดยจะเริ่มตั้งแต่ 8 บิต
- 2) การรองรับหน่วยความจำขนาดเล็ก คือ จากหน่วยความจำชั่วคราวไม่กี่กิโลไบต์ กับหน่วยความจำถาวรที่มีไม่กี่สิบกิโลไบต์ และคาดว่าจะมีการเพิ่มขึ้นอีกในอนาคต แต่ก็ยังคงความจำเป็นขั้นต่ำเอาไว้
- 3) ใช้พลังงานต่ำ ใช้พลังงานเพียงไม่กี่สิบมิลลิแอมแปร์

- 4) สื่อสารระยะสั้น The Personal Operating Space (POS) กำหนดให้มาตรฐาน IEEE 802.15.4 หมายถึงมีระยะการใช้งานได้ถึง 10 เมตร และสำหรับการใช้งานจริงจะสามารถไปไกลได้กว่า 100 เมตรในระยะสายตา
- 5) ประหยัดต้นทุน เพราะอุปกรณ์เหล่านี้ทำงานด้วยลักษณะเฉพาะ เช่น การประมวลผลต่ำ หน่วยความจำน้อย และอื่นๆ

2.15.3 IPv6 Transition Mechanism (6in4)

ถ้าไม่ได้เชื่อมต่อกับไอพีรุ่นที่หกและเราเตอร์ยังไม่รองรับไอพีรุ่นที่หก ในสถานการณ์นี้ยังหมายถึงการที่ไม่ได้รับไอพีรุ่นที่หกจากผู้ให้บริการอินเทอร์เน็ต (Internet Service Provider) เพื่อแก้ปัญหาการขาดแคลนไอพีรุ่นที่หกจากผู้ให้บริการ เราสามารถใช้ อูโมงค์ 6in4 ได้ และวิธีการนี้ยังสามารถแก้ปัญหาเราเตอร์ที่ไม่รองรับไอพีรุ่นที่หกได้อีกด้วย วิธีการแก้คือการเพิ่มเราเตอร์ใหม่ที่รองรับทั้งไอพีรุ่นที่หกและไอพีรุ่นที่สี่ และสร้างอูโมงค์ 6in4 ขึ้นจากเราเตอร์นี้กับปลายอูโมงค์แห่งหนึ่งในอินเทอร์เน็ตของไอพีรุ่นที่หก ซึ่งเป็นวิธีที่นิยมและง่ายต่อการทำมากที่สุด โดยหลักการคือ สร้างอูโมงค์ตั้งค่าด้วยตัวเอง เชื่อมต่อแบบจุดต่อจุด ซึ่งอูโมงค์นี้จะทำการห่อหุ้มแพ็กเกจไอพีรุ่นที่หกไว้ในไอพีรุ่นที่สี่



รูปที่ 2.25 Network Diagram 6in4 Tunneling

ในสถานการณ์นี้สมมุติให้เราเตอร์ R3 ทำการสร้างอุโมงค์ 6in4 เชื่อมต่อกับปลายอุโมงค์ที่ทำหน้าที่เป็นเกตเวย์ (gateway) ของไอพีรุ่นที่หก ทำการส่ง RAs ไปยังอุปกรณ์ไอพีรุ่นที่หกที่ทำการตั้งค่าอัตโนมัติในช่อง LAN1 และทำการห่อหุ้มแพ็กเก็ตที่เครื่องต้นทาง และจะไปคลายแพ็กเก็ตที่เครื่องปลายทาง ซึ่งอุโมงค์ 6in4 นี้จะต้องใช้ไอพีรุ่นที่สี่แบบสาธารณะ ดังนั้น R3 จะต้องมียไอพีรุ่นที่สี่แบบสาธารณะ

2.16 Ubidots Cloud Platform



<https://ubidots.com/>

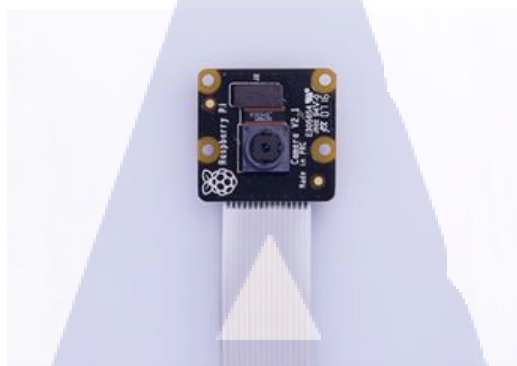
รูปที่ 2.26 โลโก้ Ubidots

Ubidots เป็นการออกแบบคลาวด์แพลตฟอร์มโดยไม่ต้องเขียนโค้ดเอง เพื่อช่วยในการสร้างตัวต้นแบบและขยายโครงการไอโอทีเพื่อการเพิ่มผลผลิต โดยตัว Ubidots เองจะเป็นคลาวด์แพลตฟอร์มที่ให้บริการกับนักพัฒนาไอโอทีทั่วไป รวมถึงองค์กรที่ต้องการความน่าเชื่อถือในการใช้งาน และการจัดการลูกค้า

ขั้นตอนการใช้ Ubidots เบื้องต้น

- 1) เชื่อมต่ออุปกรณ์ไอโอทีไปยังคลาวด์แพลตฟอร์มโดยใช้คลังเฟิร์มแวร์ของเรา
- 2) สร้างการแสดงผลแบบกำหนดเองแบบเรียบง่าย มีการแสดงผลตลอดเวลาบนหน้าแดชบอร์ด
- 3) สามารถส่งเอสเอ็มเอส อีเมลล์ หรือทำให้เกิดคำสั่งต่อไปได้บนพื้นฐานข้อมูลของคุณ
- 4) ปรับใช้ ปรับแต่งกับงานระดับธุรกิจหรือสร้างแอปพลิเคชันของตัวเอง

2.17 กล้อง Raspberry Pi NoIR Camera V2



<https://www.raspberrypi.org/products/pi-noir-camera-v2/>

รูปที่ 2.27 กล้อง Raspberry Pi NoIR Camera v2

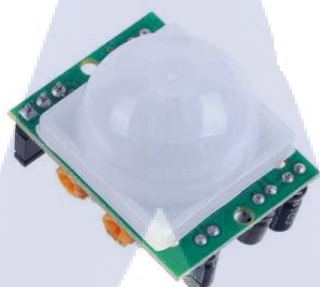
กล้องราสเบอร์รี่พายอินฟราเรดโมดูลรุ่นที่สองมาแทนที่กล้องราสเบอร์รี่พายอินฟราเรดรุ่นเริ่มต้น โดยใช้เซ็นเซอร์ โซนี่ IMX219 แพลตฟอร์มฟิคเซล ในเดือนเมษายน ปี 2016 แทนรุ่นเก่า (เมื่อเทียบกับเซ็นเซอร์ OmniVision OV5647 หัวล้านฟิคเซลของกล้องรุ่นเริ่มต้น).

กล้องราสเบอร์รี่พายอินฟราเรดสามารถทำได้ทุกอย่างเหมือนกับกล้องราสเบอร์รี่พายธรรมดา แต่มีหนึ่งสิ่งที่แตกต่างกันคือ กล้องราสเบอร์รี่พายอินฟราเรดจะไม่ได้ติดตั้งตัวกรองอินฟราเรด (NoIR = No Infrared.) ซึ่งหมายความว่า รูปที่ถ่ายในเวลากลางวันจะมีลักษณะซีด แต่ก็มีความสามารถในการมองเห็นในตอนกลางคืนแทนโดยใช้ร่วมกับแสงอินฟราเรด

ในแพ็คเกจของกล้องราสเบอร์รี่พายอินฟราเรด จะมีแผ่นเจลสีน้ำเงินสีเหลืองเล็กๆแถมมาพร้อมกับกล้องอินฟราเรด สำหรับใช้ในการตรวจสอบสุขภาพของพืช กล้องราสเบอร์รี่พายอินฟราเรดเป็นที่นิยมมากสำหรับมือสมัครเล่นที่ชอบในการส่องสัตว์ โดยใช้ร่วมกับหลอดไดโอดเปล่งแสงอินฟราเรด (LED Infrared) คุณสามารถตรวจสอบสัตว์ที่ออกหากินตอนกลางคืนได้ว่า พวกสัตว์มาทำอะไรในสวนของคุณโดยไม่ต้องรบกวนพวกสัตว์เหล่านั้น

กล้องราสเบอร์รี่พายอินฟราเรดสามารถใช้งานได้กับราสเบอร์รี่พายทุกรุ่น มันสามารถเข้าถึงได้ด้วย MMAL และ V4L APIs และสามารถเข้าถึงได้โดย third-party libraries ที่ถูกสร้างสำหรับมัน รวมถึง PiCamera Python library

2.18 เซ็นเซอร์ตรวจจับความเคลื่อนไหว (PIR motion sensor)



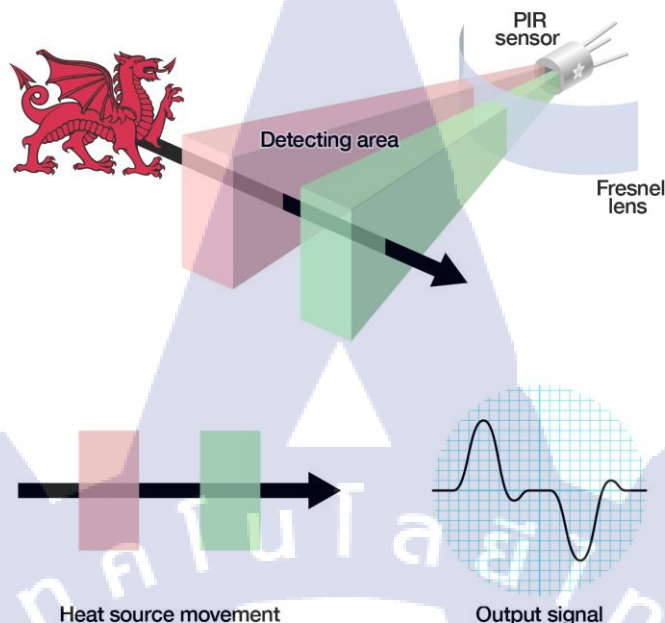
<http://thaisensormodule.com/index.php/light/pirsens1>

รูปที่ 2.28 PIR Motion Sensor

PIR ย่อมาจาก Passive Infrared เป็นอุปกรณ์ที่ใช้ตรวจจับความเคลื่อนไหวด้วยการวัดความร้อนในพื้นที่ที่ต้องการ โดยความร้อนสามารถวัดได้จากการเปลี่ยนแปลงระดับรังสีอินฟราเรดที่ปล่อยออกมาจากวัตถุ (สิ่งมีชีวิตทุกชนิดจะแผ่รังสีอินฟราเรดออกมาจากตัวเอง การแผ่รังสีเกิดจากการเคลื่อนของอิเล็กตรอนในอะตอม ปริมาณรังสีจะมีมากน้อยตามแต่โครงสร้างทางเคมีและอุณหภูมิของวัตถุหรือสิ่งมีชีวิตนั้นๆ) เมื่อวัตถุเคลื่อนที่จะทำให้แรงดันไฟฟ้ามีค่าสูงขึ้นและนำมาแปลงเป็นเสียงหรือแสง สำหรับใช้เป็นเซ็นเซอร์ได้ จึงมีการนำเอา PIR มาประยุกต์ใช้งานเพื่อเป็นตรวจจับการเคลื่อนไหวของสิ่งมีชีวิต หรือตรวจจับการบุกรุกในงานรักษาความปลอดภัย หรือที่เรียกกันว่า PIR motion sensor นั่นเอง

หลักการทำงานของ Passive Infrared sensor

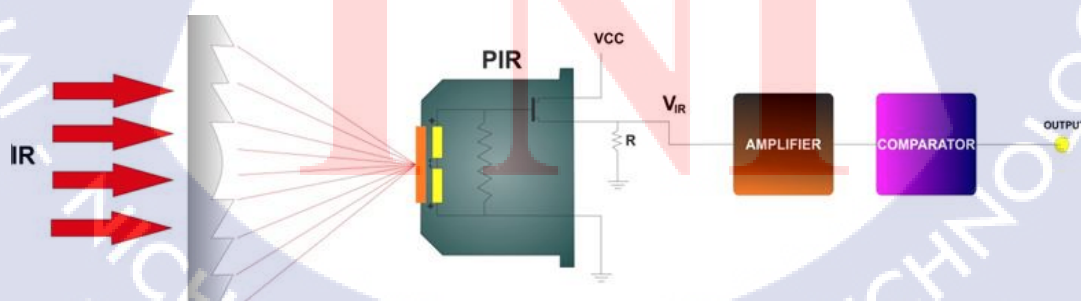
การทำงานของ PIR จะซับซ้อนกว่าเซ็นเซอร์อื่นๆ เพราะมีหลายตัวแปรที่มีผลต่อเอาต์พุตและอินพุตของเซ็นเซอร์ตัวนี้



<https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>

รูปที่ 2.29 การทำงานของ PIR Motion Sensor

ภายใน PIR จะมีอุปกรณ์ตรวจจับรังสีอินฟราเรดอยู่สองชุดด้วยกันดังรูป โดยในแต่ละรูของอุปกรณ์จะทำจากวัสดุพิเศษที่ไวต่อรังสีอินฟราเรดมาก ส่วนเลนส์(Fresnel lens) ที่ครอบอยู่ทำหน้าที่คล้ายเลนส์นูน กระจายแสงจากจุดโฟกัสด้านหนึ่งเป็นลำแสงขนานออกไปในระยะไกลได้ เมื่อมีคนหรือสัตว์ที่มีความอบอุ่นในร่างกายเคลื่อนที่ผ่านเข้ามาในพื้นที่ตรวจจับ ซึ่งในขณะที่เคลื่อนไหวย่อมมีรังสีความร้อนแผ่ออกมารอบๆตัวในปริมาณที่แน่นอนอยู่จำนวนหนึ่ง ทำให้อุณหภูมิในบริเวณนั้นเกิดการเปลี่ยนแปลง PIR สามารถตรวจจับคลื่นรังสีอินฟราเรดที่แผ่ออกมาจากสิ่งมีชีวิตได้ PIR จะส่งคลื่นรังสีมายังตัว Pyro Electric ซึ่งจะเปลี่ยนพลังงานความร้อนจากรังสีอินฟราเรดเป็นพลังงานไฟฟ้า ตามการเคลื่อนไหวยื่นออกมาทางขาเอาต์พุต แล้วจะถูกป้อนเข้าสู่ไอซีเพื่อทำการขยายสัญญาณต่อ



<http://www.thaieasyelec.com/article-wiki/review-product-article/pir-motion-sensor-getting-started.html>

รูปที่ 2.30 การทำงานของ PIR motion sensor เมื่อมีรังสีอินฟราเรดเปลี่ยนแปลง

ในตอนที่ยังไม่มีสิ่งมีชีวิตผ่านเซ็นเซอร์ไป ทั้งสองอุปกรณ์ตรวจจับคลื่นรังสีจะมีค่าแรงดันปกติ และเมื่อมีสิ่งมีชีวิตเคลื่อนที่ผ่าน อุปกรณ์ตรวจจับรังสีอินฟราเรดตัวที่หนึ่งจะได้สัญญาณเอาต์พุตออกมาสูงกว่าแรงดันปกติ เมื่อเทียบกับแรงดันจากอุปกรณ์อีกตัวหนึ่ง และเมื่อสิ่งมีชีวิตเคลื่อนออกจากพื้นที่ที่ตรวจจับ จะเกิดการย้อนกลับของแรงดัน ทำให้แรงดันเอาต์พุตต่ำกว่าแรงดันปกติ แล้วจึงกลับมาเท่าปกติในภายหลัง นั่นหมายความว่าสามารถตรวจจับสิ่งมีชีวิตที่เคลื่อนไหวผ่านเซ็นเซอร์นี้ได้แล้ว

2.19 ระบบปฏิบัติการ Contiki OS



<http://www.thewindowsclub.com/confiki-os-vs-windows-10-internet-things>

รูปที่ 2.31 ระบบปฏิบัติการ Contiki OS

ระบบปฏิบัติการ Contiki เป็นระบบปฏิบัติการแบบสาธารณะ ถูกออกแบบมาสำหรับ Internet of Things เพื่อเชื่อมต่ออุปกรณ์ขนาดเล็ก ไมโครคอนโทรลเลอร์ที่ใช้พลังงานต่ำไปยังอินเทอร์เน็ต

ระบบปฏิบัติการ Contiki รองรับการใช้พลังงานต่ำและสามารถเชื่อมต่ออินเทอร์เน็ตได้ รองรับมาตรฐานไอพีรุ่นที่สี่และไอพีรุ่นที่หก พร้อมทั้งมาตรฐานของระบบเครือข่ายไร้สายพลังงานต่ำด้วย เช่น 6LoWPAN, RPL, CoAP, ContikiMAC ของ Contiki และ Sleepy Routers ถึงแม้ว่าจะทำงานด้วยระบบเครือข่ายไร้สายพลังงานต่ำก็ตาม ก็ยังสามารถใช้งานได้ดีในขณะที่ใช้พลังงานจากแบตเตอรี่

ระบบปฏิบัติการ Contiki ง่ายต่อการนำไปพัฒนาต่อ เพราะแอปพลิเคชันถูกเขียนด้วยภาษาซี และยังสามารถใช้โปรแกรมจำลอง Cooja Simulator ที่สามารถจำลองระบบเครือข่ายได้ ก่อนที่จะนำไปใช้งานบนอุปกรณ์และสถานที่จริง และมี Instant Contiki ที่เป็นระบบปฏิบัติการ Contiki แบบสำเร็จรูป สามารถดาวน์โหลดมาแล้วใช้งานได้ทันที

ส่วนประกอบของระบบปฏิบัติการ Contiki OS มีโครงสร้างดังนี้

ตารางที่ 2.2 ตารางโครงสร้างของระบบปฏิบัติการ Contiki OS

แฟ้มข้อมูล	รายละเอียด	ข้อมูลของอุปกรณ์ Zolertia
Examples	เตรียมการเพื่อสร้างตัวอย่าง	examples/zolertia, examples/cc2538-common
App	แอปพลิเคชันของ Contiki	-
Cpu	แฟ้มข้อมูลเฉพาะของ MCU	mcp430, cc2538
Dev	หน่วยประมวลผลภายนอกและอุปกรณ์	cc2420, cc1200
Platform	แฟ้มข้อมูลเฉพาะและแพลตฟอร์มของตัวขับเคลื่อน	z1, zoul
Core	แฟ้มแกนหลักของ Contiki และคลังข้อมูล	-
Tools	เครื่องมือที่ใช้ในการล้าง, แก้ไขข้อบกพร่อง, การจำลอง และอื่นๆ	zolertia, sky
Docs	เอกสารต่างๆที่จัดทำขึ้นมาด้วยตัวเอง	-
regression-tests	การทดสอบการถดถอย	-

ถ้าหากเข้าไปในแฟ้มข้อมูล `contiki/examples/zolertia/tutorial/` จะพบกับคู่มือการใช้งานระบบต่างๆของระบบปฏิบัติการ Contiki บนอุปกรณ์ Z1 และ Zoul โดยเนื้อหาจะแบ่งตามแฟ้มข้อมูลดังนี้

- 01-basics เกี่ยวกับพื้นฐานของ Contiki (เวลา, GPIO, หลอดไฟ LED, โดดแปลงแสง)
- 02-ipv6 เกี่ยวกับระบบเครือข่ายไร้สาย, พื้นฐานของสัญญาณวิทยุ, การใช้ IPv6/6LoWPAN และการใช้ขอบของเราเตอร์
- 03-coap ตัวอย่างการใช้งานเซิร์ฟเวอร์ CoAP
- 04-mqtt ตัวอย่างการใช้งาน MQTT บนเครื่องลูกข่าย

2.20 คลาวด์แพลตฟอร์ม Amazon Simple Storage Service (Amazon S3)



<http://www.signiant.com/fast-file-transfer-to-amazon-s3/>

รูปที่ 2.32 โลโก้ Amazon Simple Storage Services

Amazon Simple Storage Service (Amazon S3) คือที่เก็บข้อมูลที่อยู่บนอินเทอร์เน็ต คุณสามารถใช้ Amazon S3 ในการจัดเก็บและเรียกกลับข้อมูลขนาดใดก็ได้ตลอดเวลา จากที่ไหนก็ได้ที่มีการเชื่อมต่อกับอินเทอร์เน็ต คุณสามารถทำสิ่งเหล่านี้ได้ด้วยวิธีการที่เรียบง่ายและเข้าใจได้กับหน้าเว็บอินเทอร์เน็ตเฟสหน้าด้านการจัดการของ AWS และคู่มือนี้จะเป็นการทำความรู้จักกับ Amazon S3 และวิธีการใช้หน้าด้านการจัดการของ AWS ให้สำเร็จด้วยการแสดงขั้นตอนการทำงานดังนี้



<http://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS3.html>

รูปที่ 2.33 ขั้นตอนการทำงานของ Amazon S3

- (1) สมัครเข้าใช้งาน Amazon S3
- (2) สร้างคลังเก็บข้อมูล
- (3) เพิ่มข้อมูลเข้าไปในคลัง
- (4) เรียกดูข้อมูลได้
- (5) สามารถเคลื่อนย้ายข้อมูลได้
- (6) สามารถข้อมูลหรือลบคลังออกได้

2.21 ชุดพัฒนาซอฟต์แวร์ Boto3

โบโทสาม(Boto3) เป็นชุดพัฒนาซอฟต์แวร์ของ Amazon Web Services (AWS) ที่ใช้ภาษาไพธอน (Python programming language) ซึ่งช่วยให้นักพัฒนาที่ใช้ภาษาไพธอนสามารถเขียนซอฟต์แวร์เรียกใช้บริการ

ของ Amazon ได้ เช่น Simple Storage Services (S3), Elastic Compute Cloud (EC2) ชุดพัฒนาไบนารีนั้น ง่ายต่อการใช้งาน ใช้ API ในงานเชิงวัตถุได้เช่นเดียวกับภาษาระดับล่าง สามารถเข้าถึงบริการได้โดยตรง

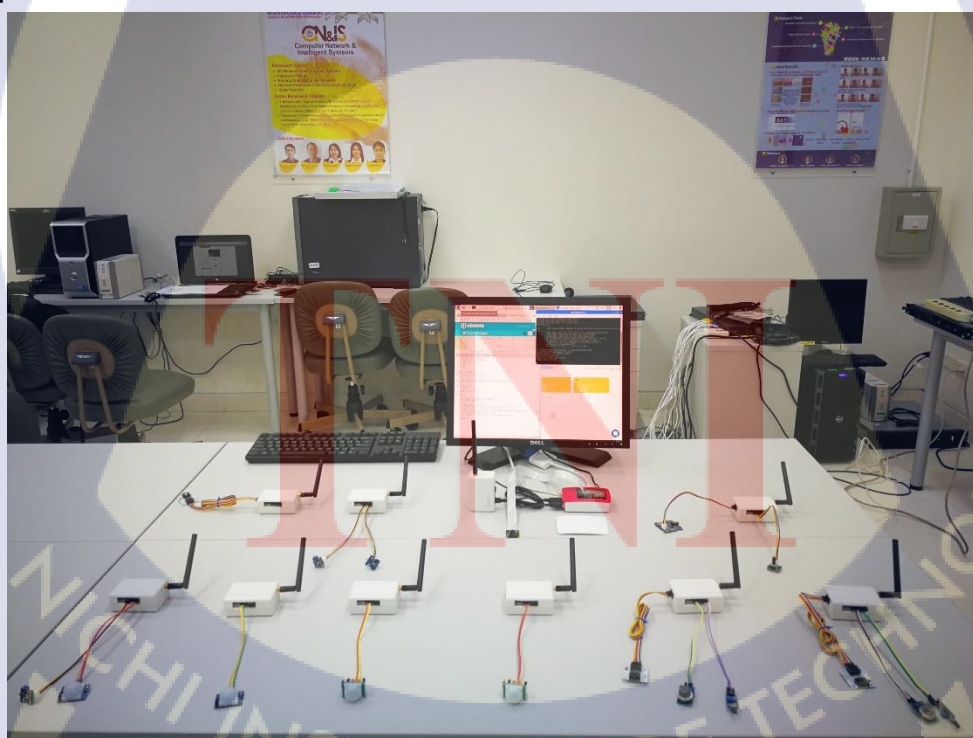
โบทสามถูกเขียนขึ้นมาใหม่จากโบท มันถูกใช้ในการส่งข้อมูลในการสร้างคลาสในระหว่างการทำงานโดยไฟล์ JSON ที่ถูกใช้ร่วมกันในระหว่างการพัฒนาซอฟต์แวร์ในภาษาต่างๆ ซึ่งรวมถึงการเขียนโปรแกรมในภาษาระดับสูง การเขียนในเชิงวัตถุที่มีอินเทอร์เฟซคล้ายกับหน้าตาของรุ่นก่อนๆ ของโบท

โบทสามประกอบด้วยคุณสมบัติที่สำคัญดังนี้

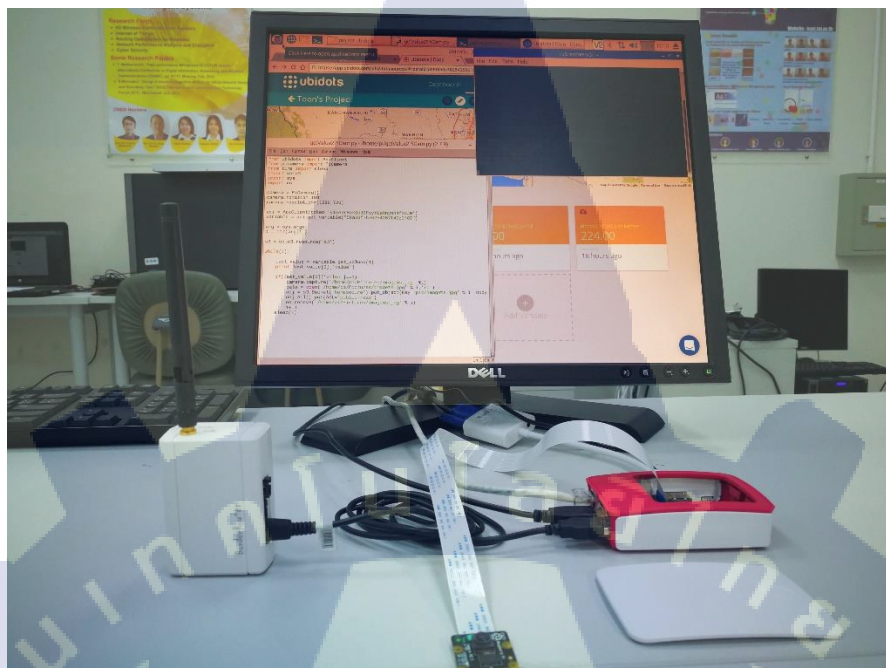
- 1) แหล่งข้อมูล - ใช้ภาษาระดับสูง, ออกแบบอินเทอร์เฟซในเชิงวัตถุ
- 2) คอลเลกชัน - ใช้เครื่องมือในการทำซ้ำและการจัดการกลุ่มของแหล่งข้อมูล
- 3) เครื่องลูกข่าย - เชื่อมต่อบริการในระดับล่าง
- 4) การจัดหน้า - การจัดหน้าอัตโนมัติทันทีที่มีการตอบสนอง
- 5) การให้บริการ - หาทางป้องกันจนกว่าจะมีสภาพที่แน่นอนที่จะไปถึงเป้าหมาย

พร้อมด้วยคุณสมบัติเหล่านี้ โบทสามได้มีการเตรียมการต่างๆ และการทดสอบชุดไบนารีรองรับ และการกำหนดค่าความปลอดภัยพื้นฐาน เช่น การยืนยันตัวตน พารามิเตอร์และการควบคุมการตอบสนอง มีระบบอีเว้นท์สำหรับการปรับ เป็นต้น

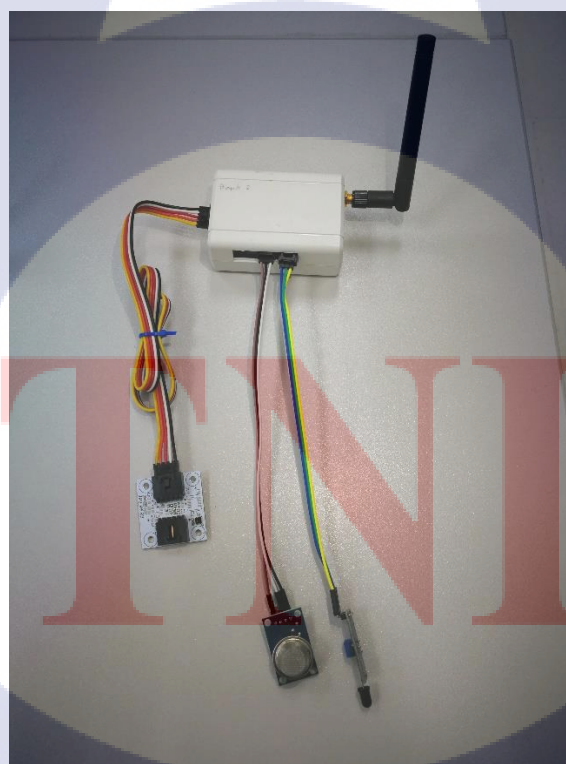
2.22 ชุดอุปกรณ์เซ็นเซอร์ไร้สายที่ใช้ในการวิจัย



รูปที่ 2.34 ภาพรวมของอุปกรณ์ที่ใช้นในโครงการวิจัย



รูปที่ 2.35 อุปกรณ์ Border Router และ Raspberry pi



รูปที่ 2.36 อุปกรณ์เซ็นเซอร์ไร้สาย Z1 และอุปกรณ์เซ็นเซอร์ต่อพ่วงต่าง ๆ

บทที่ 3

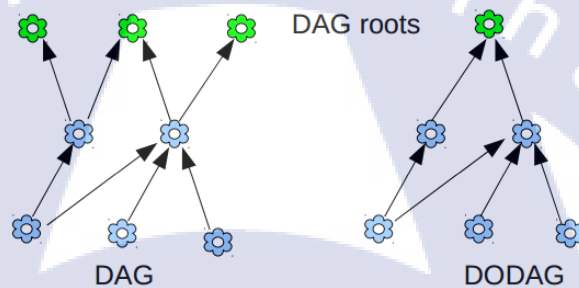
การออกแบบและดำเนินงานวิจัย

3.1 ออกแบบอัลกอริทึมเพื่อจัดเส้นทาง

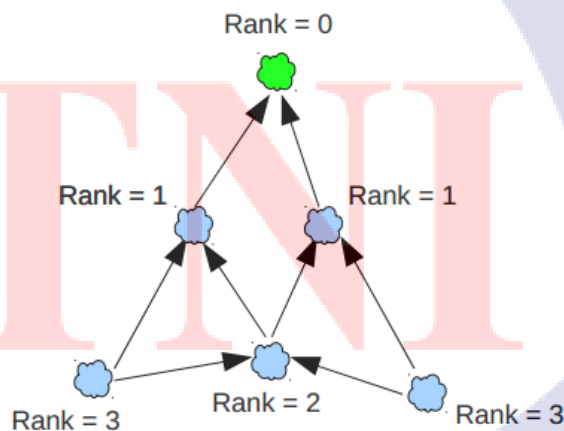
3.1.1 IPv6 Routing Protocol for LLNs (RPL) 1

กราฟอธิบายหลักการทำงานของ RPL ตามรูปที่ 3.1

- Directed Acyclic Graph (DAG) = กราฟที่ไม่มีการวนซ้ำ
- Destination Oriented Directed Acyclic Graph (DODAG) = กราฟ DAG ที่มีปลายทางที่เดียว



รูปที่ 3.1 RPL Node Rank เพื่อกำหนดค่า Rank สัมพันธ์กับ DODAG root



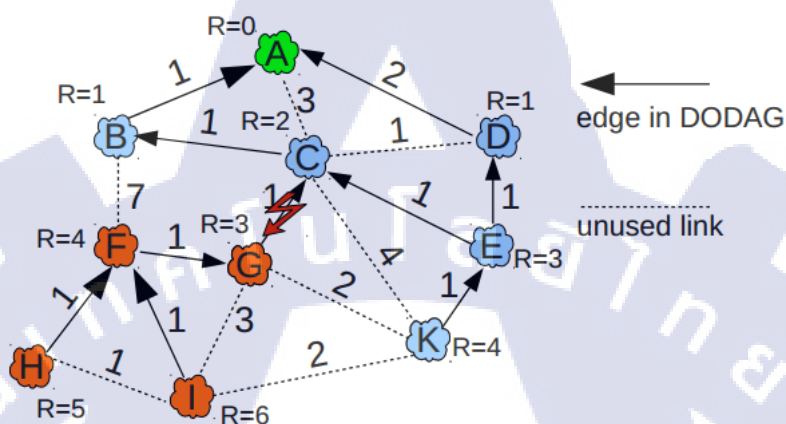
รูปที่ 3.2 DODAG Example กลุ่มของ parent node และโหนดไม่รับรู้ child node

3.1.2 IPv6 Routing Protocol for LLNs (RPL) 2

DODAG Repair ดึงรั้งระหว่าง G กับ C เสีย และเลือกโหนดแม่ที่มี Rank ต่ำกว่า

Global repair - ใช้หมายเลขลำดับใน DODAG

Local repair - ทำ sub-DODAG ด้วยการกำหนด Rank เป็น Infinity



รูปที่ 3.3 DODAG Repair

3.1.3 Low power and Lossy Networks (LLNs)

เครือข่ายที่มีการใช้พลังงานต่ำ การสูญเสียข้อมูลสูง คือเครือข่ายประเภทหนึ่งที่ เราเตอร์และการเชื่อมต่อถูกจำกัดไว้ เราเตอร์ในเครือข่าย LLN ปกติแล้วจะมีข้อจำกัดในเรื่องของความสามารถในการประมวลผล หน่วยความจำ และพลังงาน ส่วนการเชื่อมต่อระหว่างกันก็มีอัตราสูญเสียข้อมูลสูง อัตราการขนส่งข้อมูลต่ำ และขาดความมั่นคง LLN ประกอบด้วยเราเตอร์เพียงไม่กี่เครื่อง ไปจนถึงหลายพันเครื่อง รูปแบบการไหลของข้อมูลประกอบด้วยแบบ point-to-point (ระหว่างอุปกรณ์ใน LLN) point-to-multipoint (จากศูนย์กลางเครือข่ายไปสู่อุปกรณ์อื่นๆ) multipoint-to-point (จากอุปกรณ์รอบนอกไปสู่ศูนย์กลาง)

สรุปลักษณะของ LLNs

- LLNs ประกอบด้วยอุปกรณ์หลายสิบถึงหลายพันเครื่อง
- มีปัญหาจากอุปกรณ์มีข้อจำกัดเรื่องของการประมวลผล, หน่วยความจำ และพลังงาน (แบตเตอรี่)
- การเชื่อมต่อมีลักษณะ ข้ำ สัญญาณไม่มั่นคง และมีอัตราการสูญเสียข้อมูลสูง

3.1.4 Routing over Low power and Lossy networks (ROLL)

IETF working group พยายามแก้ไขปัญหาของการหาเส้นทางบน LLN

- ร่างข้อกำหนดต่างๆสำหรับใช้กับ LLN
- ประเมินโปรโตคอลที่มีอยู่ว่าเหมาะกับ LLN หรือไม่
- Routing Protocol for Low power and lossy networks (RPL)

ตารางที่ 3.1 จากการประเมินโปรโตคอลที่มีอยู่แล้ว

Protocol	State	Loss	Control	Link	Node
OSPF/IS-IS	fail	fail	fail	pass	fail
OLSRv2	fail	?	?	pass	pass
TBRPF	fail	pass	fail	pass	?
RIP	pass	fail	pass	?	fail
AODV	pass	fail	pass	fail	fail
DYMO	pass	?	pass	?	?
DSR	fail	pass	pass	fail	fail

ความหมาย

Protocol = Protocol ชื่อโปรโตคอลที่มีการสำรวจ

State = Routing State จำกัดหน่วยความจำ

Loss = Loss Response เกิดอะไรขึ้นเมื่อลิงก์ล้มเหลว

Control = Control Cost ข้อจำกัดในการควบคุมการสื่อสาร

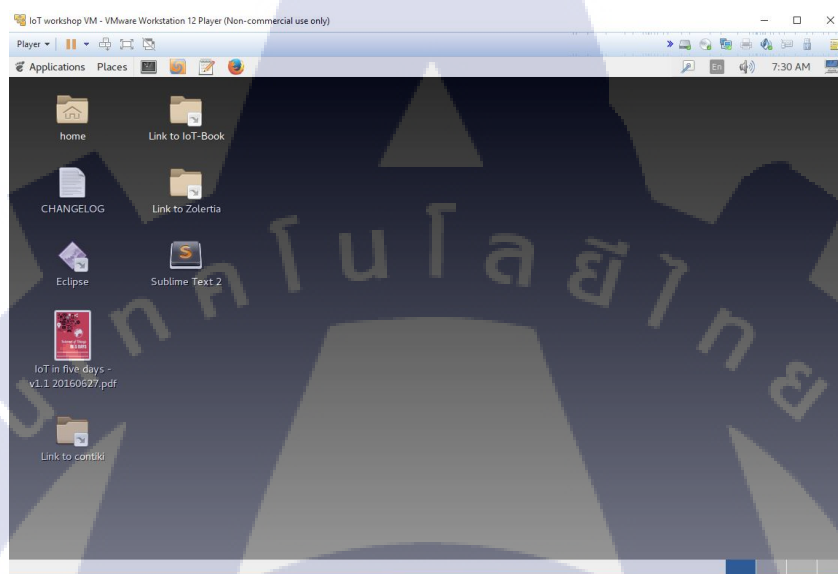
Link = Link Cost คุณสมบัติของลิงค์

Node = Node Cost คุณสมบัติของโหนด

3.2 ขั้นตอนการดำเนินงาน

3.2.1 ติดตั้ง VMware และ Contiki

สำหรับกรณีที่ใช้งานระบบปฏิบัติการวินโดวส์ ทำการติดตั้ง VMware Workstation player โดยสามารถดาวน์โหลดมาใช้งานและติดตั้งได้ฟรี จากนั้นดาวน์โหลด IoT Workshop VM เพื่อใช้งาน Contiki OS



รูปที่ 3.4 IoT Workshop และ VMware

สำหรับระบบปฏิบัติการ Linux ให้โหลด Contiki ผ่าน Terminal ด้วยคำสั่ง

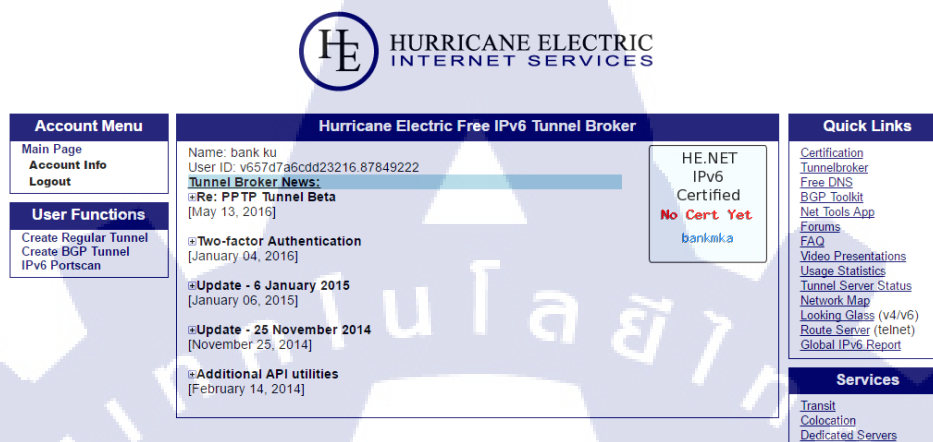
```
cd /home/user/contiki
git remote add iot-workshop https://github.com/alignan/contiki
git fetch iot-workshop
git checkout iot-workshop
```

เพื่อใช้ในการติดตั้ง Contiki จากนั้นให้ดาวน์โหลดเครื่องมือสำหรับแพลตฟอร์มโค้ดสำหรับ Z1 ด้วยคำสั่ง

```
sudo apt-get -y install gcc gcc-msp430
```

3.2.2 ทำ Tunnel จากไอพีรุ่นที่สี่เพื่อใช้ไอพีรุ่นที่หก

สมัครใช้งานเว็บ Hurricane Tunnel Broker เพื่อขอใช้งานไอพีรุ่นที่หกผ่านไอพีสาธารณะรุ่นที่สี่ที่ใช้งานอยู่ซึ่งจำเป็นต้องเปิดพอร์ต ICMP และโปรโตคอล 41 เพื่อให้แพ็กเก็ตที่ออกไปสามารถทำการ Tunnel เพื่อให้ใช้งานไอพีรุ่นที่หกได้



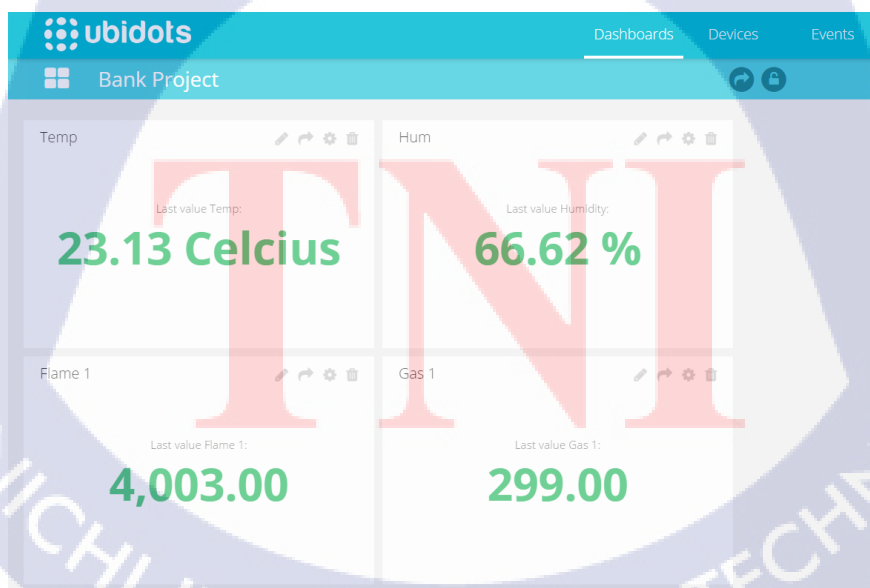
HURRICANE ELECTRIC
INTERNET SERVICES

Account Menu	Hurricane Electric Free IPv6 Tunnel Broker	Quick Links
Main Page Account Info Logout	Name: bank ku User ID: v657d7a6cdd23216.87849222 Tunnel Broker News: - Re: PPTP Tunnel Beta [May 13, 2016] - Two-factor Authentication [January 04, 2016] - Update - 6 January 2015 [January 06, 2015] - Update - 25 November 2014 [November 25, 2014] - Additional API utilities [February 14, 2014]	HE.NET IPv6 Certified No Cert Yet bankku
User Functions Create Regular Tunnel Create BGP Tunnel IPv6 Portscan		Services - Transit - Colocation - Dedicated Servers

รูปที่ 3.5 Tunnel Broker

3.2.3 สมัครใช้งานและตั้งค่า Ubidots เพื่อรับค่าและแจ้งเตือน

สมัคร Ubidots คลาวด์แพลตฟอร์มและตั้งค่าสำหรับเตรียมรับข้อมูลจากเซ็นเซอร์โดยวิธีการสมัครใช้งานและวิธีการตั้งค่าสามารถดูได้ที่ภาคผนวก ก.



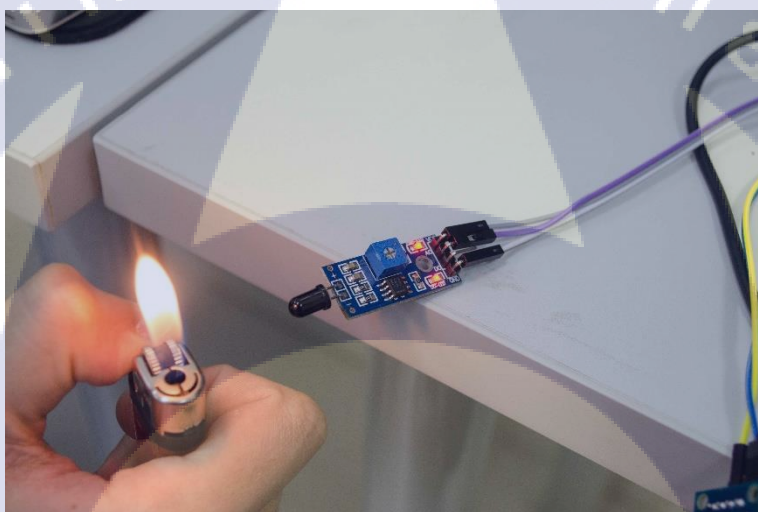
รูปที่ 3.6 Ubidots

3.2.4 ติดตั้งโปรแกรมบนอุปกรณ์ด้วยโปรแกรม Border Router และ Ubidots client บน Z1 ที่ต่อกับเซ็นเซอร์แก๊สและเซ็นเซอร์ไฟ

เตรียมอุปกรณ์ Z1 ที่ติดตั้งโปรแกรม Ubidots client ที่แก้ไขโค้ดให้ส่งค่าจากอนาล็อกเซ็นเซอร์โดยเชื่อมต่อกับเซ็นเซอร์ไฟและแก๊สไว้ ส่วน Z1 อีกหนึ่งอันติดตั้งโปรแกรม Border Router เพื่อเป็นทางผ่านข้อมูลโดยเชื่อมต่อกับ Raspberry pi

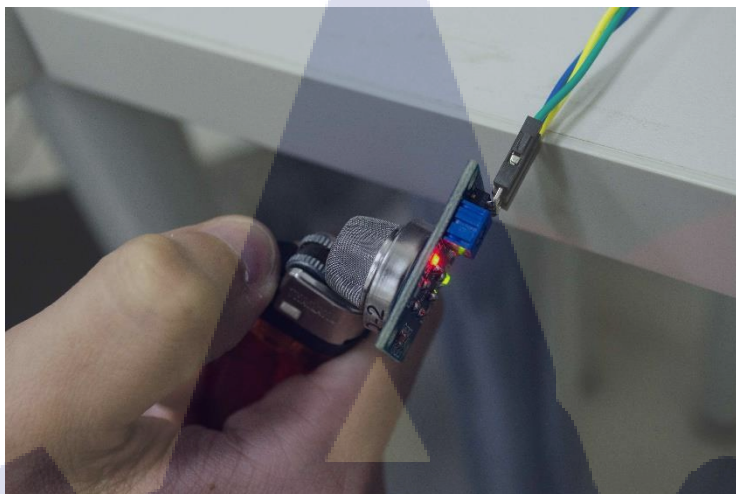
3.2.5 จุดไฟแช็คให้ติดไฟหรือแก๊สเพื่อทดลองกับเซ็นเซอร์

จุดไฟจากไฟแช็คขึ้นมาโดยสมมุติเหตุการณ์กรณีมีเปลวไฟภายในห้อง ซึ่งหากพบไฟภายในห้องเซ็นเซอร์จะแสดงสีแดงขึ้นมามีรูปที่ 3.4 โดยใช้งานร่วมกับเซ็นเซอร์ไฟพร้อมทั้งส่งค่าให้ขึ้น Ubidots ได้ซึ่งจะต้องมีค่าแตกต่างกันเมื่อจุดไฟและไม่จุดไฟ



รูปที่ 3.7 ทดสอบเซ็นเซอร์ไฟ

จุดแก๊สจากไฟแช็คโดยจ่อไปที่เซ็นเซอร์เพื่อทดลองกรณีเกิดแก๊สรั่ว ซึ่งจะมีสีเหลืองแสดงขึ้นมาบนเซ็นเซอร์ดังรูปที่ 3.5 โดยค่าที่แสดงบนคลาวด์จะต่างกันกรณีที่มีแก๊สและไม่มีแก๊ส

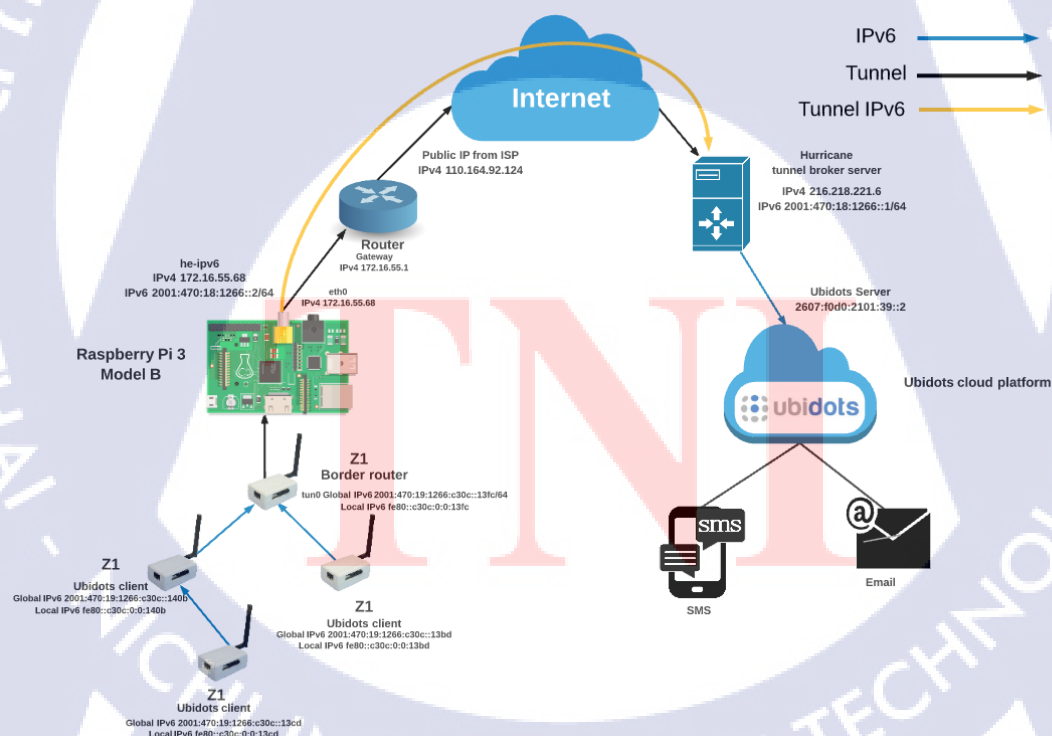


รูปที่ 3.8 ทดสอบเซ็นเซอร์แก๊ส

3.2.6 Ubidots แจ้งเตือนมายังโทรศัพท์ของผู้ใช้งาน

เมื่อตรวจพบแก๊สหรือไฟจะส่งข้อมูลมายังอีเมลและ SMS ที่ได้ตั้งค่าไว้เพื่อแจ้งเตือนผู้ใช้งาน เพื่อจำลองสถานการณ์ไฟไหม้จริง

3.3 ขั้นตอนการทำงาน



รูปที่ 3.9 ขั้นตอนการทำงาน

3.3.1 การตั้งค่าจาก Ubidot Clients (Z1) เพื่อส่งข้อมูลไปยัง Border Router (Z1)

เริ่มจากการติดตั้ง Ubidot Clients บนอุปกรณ์ที่ต้องการรับค่าโดยโค้ดที่ทำการติดตั้งถูกปรับปรุงจากเดิมที่สามารถรับค่าได้เพียงจากเซ็นเซอร์พอร์ทดิจิทัลและมีเวลาประมาณ 1 นาทีต่อการอัปเดตหนึ่งครั้งให้ส่งได้เร็วขึ้นประมาณ 3 วินาทีต่อ 1 ค่า และทำให้สามารถรับค่าจากเซ็นเซอร์พอร์ท Piglet ได้ และเปลี่ยนค่าของ VARIEABLE KEY ให้ตรงกับที่ได้ตั้งค่าไว้ใน Ubidots ในทุกๆอุปกรณ์ที่จะทำการติดตั้งโดยสามารถดูโค้ดได้ที่ภาคผนวก ก.

Z1 นั้นเป็นอุปกรณ์ที่ทำงานบนไอพีรุ่นที่หก จึงจำเป็นต้องใช้งานไอพีรุ่นที่หกเพื่อมอบไอพีให้กับอุปกรณ์แต่ในห้วงเวลานั้นไม่สามารถใช้งานไอพีรุ่นที่หกได้ แต่สามารถงานไอพีรุ่นที่สี่แบบสาธารณะได้

ทำการติดตั้ง Border Router ด้วยคำสั่ง `Make border-router.upload && make connect-router PREFIX=2001:470:19:1266::/64` ซึ่งเป็นไอพีที่ได้รับมาจากการทำ tunnel และติดตั้ง Ubidots Client ด้วยคำสั่ง `make ubidots-client.upload` ซึ่งจะต้องทำการปรับโค้ดในส่วนของ VARIEABLE KEY ทุกครั้งในทุกอุปกรณ์

เมื่อติดตั้งโปรแกรมเสร็จเรียบร้อยแล้วต้องรอเวลาสักครู่ เพื่อทำการแจกไอพีและอัปเดตตารางการของอุปกรณ์ด้วยวิธีการของ RPL

3.3.2 การส่งข้อมูลจาก Border Router (Z1) ไปยัง Raspberry pi

เมื่อติดตั้ง Border Router โปรแกรมจะเชื่อมต่อไปยัง Raspberry pi โดยเชื่อมต่อออกไปยังเครือข่ายและส่งข้อมูลผ่านพอร์ท USB บนอินเทอร์เน็ต `tun0` ซึ่งเชื่อมต่อกันระหว่าง Border Router กับ Raspberry pi โดย Border Router จะรู้ไอพีรุ่นที่หกของตนจากการกำหนด Prefix ซึ่งในกรณีนี้ Prefix จะได้มาจากการทำ Tunnel Broker เพื่อใช้งานไอพีรุ่นที่หก

3.3.3 การตั้งค่าบน Raspberry pi เพื่อส่งต่อข้อมูลไปยังเราเตอร์

บน Raspberry pi นั้นจะทำหน้าที่เป็นทางผ่านของข้อมูลจาก Zolertia Z1 ที่ติดตั้งโปรแกรม Border Router โดยก่อนอื่นนั้นต้องการทำการตั้งค่าบน Raspberry pi โดยบน Raspberry pi จะมี 3 อินเทอร์เฟซดังนี้

1. อินเทอร์เฟซ `eth0` ใช้สำหรับรับไอพี ที่ทางสถาบันฯทำการ NAT กับไอพีรุ่นที่สี่แบบสาธารณะ เพื่อใช้ออกสู่อินเทอร์เน็ต

2. อินเทอร์เฟซเสมือน tun0 ใช้สำหรับเชื่อมต่อเครือข่ายไอพีรุ่นที่หกบนเครือข่ายบุคคลไร้สายพลังงานต่ำของอุปกรณ์ Zolertia Z1 ที่ทำหน้าที่เป็นขอบอุปกรณ์จัดเส้นทาง เพื่อเชื่อมต่อกับอินเทอร์เฟซ eth0

3. อินเทอร์เฟซเสมือน he-ipv6 ใช้สำหรับเชื่อมโยงเครือข่ายไอพีรุ่นที่หกที่วิ่งผ่านเครือข่ายไอพีรุ่นที่สี่จากเว็บไซต์ tunnelbroker.net เพื่อให้ Raspberry pi สามารถออกสู่อินเทอร์เน็ตผ่านระบบเครือข่ายไอพีรุ่นที่หกได้

จากนั้นให้ตั้งค่าตารางไอพีเพื่อให้การส่งกลุ่มข้อมูลสามารถวิ่งไปถึงปลายทางที่กำหนดได้ โดยการตั้งค่าให้ ขอมรับ ในทุกช่องทาง ทั้งการนำเข้า การส่งต่อ และการนำออกของกลุ่มข้อมูล ตั้งค่าให้อินเทอร์เฟซ eth0 สามารถส่งต่อกลุ่มข้อมูลไปที่อินเทอร์เฟซเสมือน tun0 และตั้งค่าให้อินเทอร์เฟซเสมือน tun0 ส่งต่อกลุ่มข้อมูลกลับไปให้อินเทอร์เฟซ eth0 ได้เช่นกัน ตั้งค่าให้อินเทอร์เฟซ eth0 ทำหน้าที่จัดหาเส้นทางให้กับเครือข่ายภายใน โดยการพิมพ์บรรทัดคำสั่งลงบนโปรแกรมคำสั่งงานดังนี้

```
iptables -I INPUT -j ACCEPT
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -i eth0 -o tun0 -m state --state RELATED,ESTABLISHED -j
ACCEPT
iptables -A FORWARD -i tun0 -o eth0 -m state --state RELATED,ESTABLISHED -j
ACCEPT
```

ทำการเปิดทางให้ไอฟิรุ่นที่สี่และไอฟิรุ่นที่หกสามารถส่งต่อกลุ่มข้อมูลได้ โดยการเข้าถึงแฟ้มข้อมูล /etc/sysctl.conf และลบเครื่องหมาย หมายเหตุ ที่อยู่ข้างหน้าบรรทัดด้านล่างนี้

```
net.ipv4.ip_forward = 1 และ
```

```
net.ipv6.conf.default.forwarding=1
```

ตั้งค่าอินเทอร์เฟซ eth0 ให้เป็นค่าคงที่เพื่อให้เส้นทางของกลุ่มข้อมูลสามารถส่งได้ถูกต้อง โดยการเข้าถึงแฟ้มข้อมูล /etc/dhcpd.conf และทำการเพิ่มบรรทัดคำสั่งด้านล่างนี้

```
interface eth0
```

```
static ip_address=172.16.55.68/24
```

```
static routers=172.16.55.1
```

```
static domain_name_servers=172.16.10.10
```

สร้างอินเทอร์เฟซเสมือน he-ipv6 เพื่อให้ Raspberry pi สามารถใช้งานอินเทอร์เน็ทผ่านไอฟิรุ่นที่หกได้ โดยการเพิ่มคำสั่งดังนี้

```
auto he-ipv6
```

```
iface he-ipv6 inet6 v4tunnel
```

```
address 2001:470:18:126::2
```

```
netmask 64
```

```
endpoint 216.218.221.6
```

```
local 172.16.55.68
```

```
ttl 255
```

```
gateway 2001:470:18:1266::1
```

เมื่อทำการตามขั้นตอนนี้แล้วจะทำให้ Raspberry pi สามารถเป็นทางผ่านข้อมูลให้กับ Border Router ผ่าน tun0 และสามารถส่งข้อมูลต่อได้เพราะตั้งค่าบน Raspberry pi แล้ว โดยบนอินเทอร์เฟซ eth0

นั้นสามารถใช้งานไอฟี紗ธารณะทำให้สามารถใช้งานไอฟีรุ่นที่หกได้ผ่านการทำ tunnel ผ่าน tunnel broker บนอินเทอร์เน็ตซึ่งตั้งค่าไว้ที่อินเทอร์เน็ต he-ipv6

3.3.4 การตั้งค่าบนเราเตอร์ เพื่อส่งข้อมูลไปยัง Tunnel Broker

จากเราเตอร์ไปยัง Tunnel Broker นั้นจะเชื่อมต่อไปยังอินเทอร์เน็ตเครือข่ายภายนอกโดยต้องตั้งค่าที่เราเตอร์ให้เปิดใช้งานสองส่วนคือ

1. โพรโตคอล 41 ไว้เพื่อเป็นการเปิดอนุญาตสำหรับการเพิ่มพื้นที่บนส่วนหัวของไอฟีรุ่นที่สี่เพื่อห่อหุ้มแพ็กเก็ตไอฟีรุ่นที่หกเข้าไป
 2. พอร์ต ICMP เพื่อให้ทางเซิร์ฟเวอร์สามารถตอบกลับมายัง Raspberry pi ได้
- ซึ่งเมื่อทำตามทั้ง 2 อย่างนี้จะทำให้แพ็กเก็ตที่ออกไปสามารถใช้บริการของ Tunnel broker ได้และจะได้รับไอฟีรุ่นที่หกมาใช้งานซึ่งสามารถใช้เชื่อมต่อกับทุกเว็บ

3.3.5 จาก Tunnel broker ไปยัง Ubidots

ข้อมูลจาก Tunnel Broker นั้นเมื่อเชื่อมต่อไป Ubidots จะเชื่อมต่อไปด้วยไอฟีรุ่นที่หกออกมาจาก Tunnel broker ซึ่งข้อมูลของแพ็กเก็ตเกิดจาก Border Router จะถูกคลายออกในขั้นตอนนี้และส่งต่อไปด้วยไอฟีรุ่นที่หก

3.3.6 จาก Ubidots แจ้งเตือนไปยังผู้ใช้งาน

บน Ubidots สามารถตั้งค่าการแจ้งเตือนผู้ใช้งานได้โดยมีหลายฟังก์ชันให้เลือกแต่ที่ง่ายที่สุดคือแจ้งผ่านข้อความหรืออีเมล โดยตั้งเงื่อนไขให้กับค่าที่ได้รับมาและเมื่อค่าเป็นไปตามเงื่อนไขจะทำการส่งข้อมูล

3.3.7 การตั้งค่าบนอุปกรณ์ Zolertia Z1

3.3.7.1 ส่วนของ Border Router

แอปพลิเคชันขอบเราเตอร์ประกอบไปด้วยเว็บเซิร์ฟเวอร์ การแสดงผลข้อมูลเครือข่าย เช่น การแสดงผลของอุปกรณ์ที่เชื่อมต่ออยู่ และเส้นทางการเชื่อมต่อของระบบเครือข่าย

วิธีการใช้งาน

- 1) เข้าไปที่เพิ่มข้อมูล /home/pi/contiki/zolertia/tutorial/02-ipv6/02-border-router
- 2) เขียนแอปพลิเคชัน border-router.c ลงบนอุปกรณ์ Zolertia Z1 โดยใช้บรรทัดคำสั่งดังนี้

```
make clean && make border-router.upload && make connect-router
PREFIX=2001:470:19:1266::/64
```

คำอธิบายบรรทัดคำสั่ง

- 1) คำสั่ง make clean ใช้สำหรับลบการคอมไพล์โปรแกรมอ็อบเจกต์ก่อนหน้านี้
- 2) คำสั่ง make border-router.upload ใช้สำหรับเขียนแอปพลิเคชันขอบเราเตอร์ลงบนอุปกรณ์ Zolertia Z1
- 3) คำสั่ง make connect-router ใช้สำหรับตั้งค่าเลขชุดหน้าของไอพีรุ่นที่หก ถ้าไม่ได้ตั้งค่าในส่วนนี้ ค่าเดิมจะเป็น aaaa::1/64

```
pi@raspberrypi: ~
File Edit Tabs Help
rm border-router.ihex obj_z1/httpd-simple.o obj_z1/slip-bridge.o border-router.co obj_
z1/contiki-z1-main.o
using saved target 'z1'
sudo ./tools/tunslip6 2001:470:19:1266::/64
*****SLIP started on `/dev/ttyUSB0'
opened tun device `/dev/tun0'
ifconfig tun0 inet 'hostname' mtu 1500 up
ifconfig tun0 add 2001:470:19:1266::/64
ifconfig tun0 add fe80::470:19:1266:0/64
ifconfig tun0

tun0      Link encap:UNSPEC  Hwaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
inet addr:127.0.1.1  P-t-P:127.0.1.1  Mask:255.255.255.255
inet6 addr: 2001:470:19:1266::/64 Scope:Global
inet6 addr: fe80::470:19:1266:0/64 Scope:Link
UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:500
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

*** Address:2001:470:19:1266:: => 2001:0470:0019:1266
Got configuration message of type P
Setting prefix 2001:470:19:1266::
Server IPv6 addresses:
 2001:470:19:1266:c30c::13fc
 fe80::c30c:0:0:13fc
```

รูปที่ 3.10 หน้าต่างที่ปรากฏหลังใช้คำสั่ง make connect-router

3.3.7.2 ส่วนของ Ubidots Client

แอปพลิเคชัน Ubidots ถูกใช้โดยอุปกรณ์ Zolertia Z1 และทำการประกาศข้อมูลของตัวรับรู้โดยใช้ไอพีรุ่นหกเชื่อมต่อไปที่ things.ubidots.com เพื่อแสดงผลของข้อมูลบนหน้าแดชบอร์ดวิธีการใช้งาน

- 1) เข้าไปที่แฟ้มข้อมูล /home/pi/contiki/examples/zolertia/tutorial/99-apps/project-ubidots

- 2) เข้าไปแก้ไขไฟล์ project-conf.h ดังนี้

```
#define POST_PERIOD                (CLOCK_SECOND)

#define UBIDOTS_CONF_AUTH_TOKEN    “[ใส่ token key ของชื่อผู้ใช้จากเว็บไซต์
Ubidots]”

#define VARKEY_VARIABLE_ONE        “[ใส่ variable key ที่สร้างขึ้นจากเว็บไซต์
Ubidots]”

#define UBIDOTS_CONF_REMOTE_HOST “2607:f0d0:2101:39::2”
```

เมื่อแก้ไขเสร็จ กด Save เพื่อบันทึก

- 3) เขียนแอปพลิเคชัน ubidots-client.c ลงบนอุปกรณ์ Zolertia Z1 โดยใช้บรรทัดคำสั่งดังนี้

```
make clean && make ubidots-client.upload
```

คำอธิบายบรรทัดคำสั่ง

- 1) คำสั่ง make clean ใช้สำหรับลบการคอมไพล์โปรแกรมอ็อบเจกต์ก่อนหน้านี้
- 2) คำสั่ง make ubidots-client.upload ใช้สำหรับเขียนแอปพลิเคชัน ubidots-client ลงบนอุปกรณ์ Zolertia Z1

```
*project-conf.h
~/contiki/examples/zolertia/tutorial/99-apps/project-ubidots

/*-----*/
#ifndef PROJECT_CONF_H
#define PROJECT_CONF_H
/*-----*/

/* User configuration */
#define POST_PERIOD                (CLOCK_SECOND)
#define VARIABLE_BUF_LEN          16
#define UBIDOTS_CONF_AUTH_TOKEN    "yiAeGrBoXQdJOFsyXLaNnynN4fxoJM"
#define VARKEY_VARIABLE_ONE        "5899865d76254256212365d8" //Motion1
// #define VARKEY_VARIABLE_TWO      "5899865d762542561e2ab48e" //Motion2
// #define VARKEY_VARIABLE_THREE    "589986647625425620c3f2e5" //Motion3

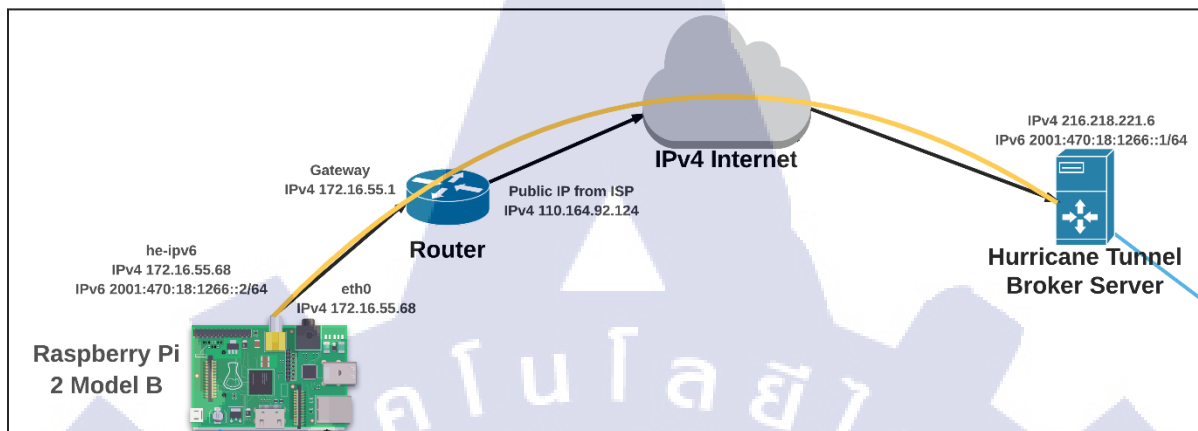
#define UBIDOTS_CONF_IN_BUFFER_SIZE 64
/*-----*/
/* IPv6 address of things.ubidots.com is "2607:f0d0:2101:39::2", leave
 * commented to resolve the host name. The NAT64 address is ":ffff:3217:7c4c"
 */
#define UBIDOTS_CONF_REMOTE_HOST    "2607:f0d0:2101:39::2"
/*-----*/

#undef NETSTACK_CONF_RADIO
#define NETSTACK_CONF_RADIO          cc2538_rf_driver
#define ANTENNA_SW_SELECT_DEF_CONF    ANTENNA_SW_SELECT_2_4GHZ
#define NETSTACK_CONF_RDC            nullrdc_driver

#undef IEEE802154_CONF_PANID
#define IEEE802154_CONF_PANID        0xABCD
/*-----*/
/* The following are Z1 specific */
#undef RF_CHANNEL
```

รูปที่ 3.11 หน้าต่างหลังแก้ไขไฟล์ project-conf.h

3.3.8 การตั้งค่าบนอุปกรณ์ Raspberry Pi



รูปที่ 3.12 การทำงานตั้งแต่อุปกรณ์ Raspberry Pi จนถึงเซิร์ฟเวอร์ของ tunnelbroker.net

- 1) อินเทอร์เฟซ eth0 ใช้สำหรับรับไอพี ที่ทางสถาบันฯทำการ NAT กับไอพีรุ่นที่สี่แบบสาธารณะ เพื่อให้ออกสู่อินเทอร์เน็ต
- 2) อินเทอร์เฟซเสมือน tun0 ใช้สำหรับเชื่อมต่อเครือข่ายไอพีรุ่นที่หกบนเครือข่ายบุคคลไร้สาย พลังงานต่ำของอุปกรณ์ Zolertia Z1 ที่ทำหน้าที่เป็นขอบเราเตอร์ เพื่อเชื่อมต่อกับ อินเทอร์เฟซ eth0
- 3) อินเทอร์เฟซเสมือน he-ipv6 ใช้สำหรับเชื่อมโยงเครือข่ายไอพีรุ่นที่หกที่วิ่งผ่านเครือข่ายไอพีรุ่นที่สี่จากเว็บไซต์ tunnelbroker.net เพื่อให้ตัว Raspberry Pi 3 สามารถออกสู่อินเทอร์เน็ตผ่านระบบเครือข่ายไอพีรุ่นที่หกได้

ตั้งค่าตารางไอพีเพื่อให้การส่งกลุ่มข้อมูลสามารถวิ่งไปถึงปลายทางที่กำหนดได้ โดยการตั้งค่าให้ยอมรับ ในทุกช่องทาง ทั้งการนำเข้า การส่งต่อ และการนำออกของกลุ่มข้อมูล ตั้งค่าให้อินเทอร์เฟซ eth0 สามารถส่งต่อกลุ่มข้อมูลไปที่อินเทอร์เฟซเสมือน tun0 และตั้งค่าให้อินเทอร์เฟซเสมือน tun0 ส่งต่อกลุ่มข้อมูลกลับไปสู่อินเทอร์เฟซ eth0 ได้เช่นกัน ตั้งค่าให้อินเทอร์เฟซ eth0 ทำหน้าที่จัดหาเส้นทางให้กับเครือข่ายภายใน โดยการพิมพ์บรรทัดคำสั่งลงบนโปรแกรมคำสั่งงานดังนี้

```

iptables -I INPUT -j ACCEPT

iptables -F

iptables -X

iptables -t nat -F

iptables -t nat -X

iptables -t mangle -F

iptables -t mangle -X

iptables -P INPUT ACCEPT

iptables -P FORWARD ACCEPT

iptables -P OUTPUT ACCEPT

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

iptables -A FORWARD -i eth0 -o tun0 -m state --state RELATED,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i tun0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT

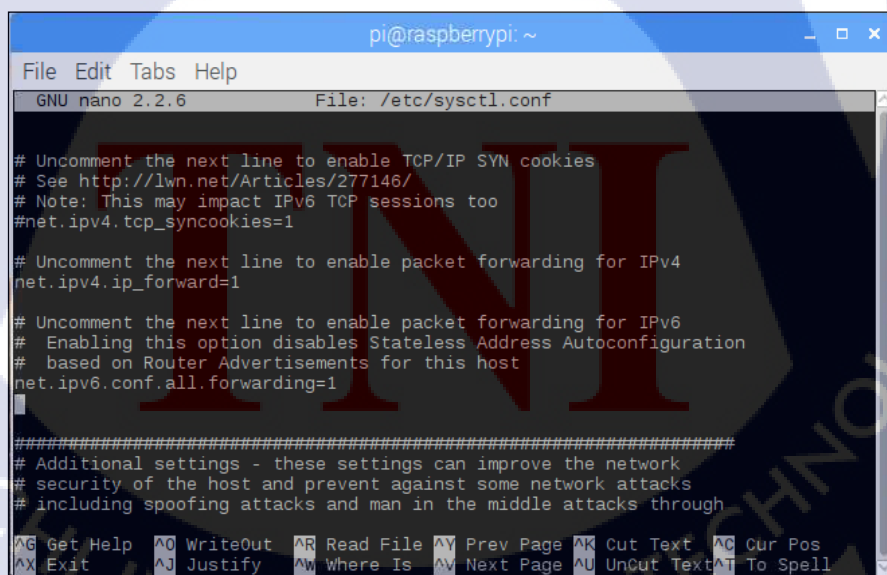
```

ทำการเปิดทางให้อิพีรุ่นที่สี่และไอพีรุ่นที่หกสามารถส่งต่อกลุ่มข้อมูลได้ โดยการเข้าถึงแฟ้มข้อมูล `/etc/sysctl.conf` และลบเครื่องหมาย หมายเหตุ (#) ที่อยู่ข้างหน้าบรรทัดด้านล่างนี้

```

net.ipv4.ip_forward = 1 และ
net.ipv6.conf.default.forwarding=1

```



รูปที่ 3.13 หน้าต่างการแก้ไขไฟล์ `sysctl.conf`

ตั้งค่าอินเทอร์เฟซ eth0 ให้เป็นค่าคงที่เพื่อให้เส้นทางของกลุ่มข้อมูลสามารถส่งได้ถูกต้อง โดยการเข้าถึงแฟ้มข้อมูล /etc/dhcpd.conf และทำการเพิ่มบรรทัดคำสั่งด้านล่างนี้

```
interface eth0
static ip_adress=172.16.55.68/24
static routers=172.16.55.1
static domain_name_servers=172.16.10.10
```

สร้างอินเทอร์เฟซเสมือน he-ipv6 เพื่อให้ Raspberry Pi สามารถใช้งานอินเทอร์เน็ตผ่านไอพีรุ่นที่หกได้ โดยการเพิ่มบรรทัดคำสั่งดังนี้

```
auto he-ipv6
iface he-ipv6 inet6 v4tunnel
    address 2001:470:18:126::2
    netmask 64
    endpoint 216.218.221.6
    local 172.16.55.68
    ttl 255
    gateway 2001:470:18:1266::1
```

3.3.8.1 วิธีการทำ Tunneling แบบ 6in4 โดยใช้ Hurricane Electric IPv6 Tunnel Broker

- 1) เข้าไปที่ www.tunnelbroker.net ทำการสมัครสมาชิก และ Login เพื่อเข้าใช้งาน
- 2) ให้กดที่ Create Regular Tunnel
- 3) ให้เลือก Server Tunnel ปลายทางที่อยู่ใกล้มากที่สุด แล้วใส่หมายเลข IPv4 ของเราลง (ดูได้จากบรรทัดด้านล่างตรง you are viewing from xxx.xxx.xxx) แล้วกดปุ่ม Create Tunnel ที่อยู่ด้านล่าง
- 4) เมื่อสร้างเสร็จแล้ว ก็จะขึ้นหน้าแสดงรายละเอียดต่าง ๆ ของ Tunnel จากนั้นก็ไปที่แท็บ

Example Configurations

- 5) แล้วก็เลือกระบบปฏิบัติการที่เราใช้งานอยู่ ในที่นี้ใช้ Raspberry Pi ซึ่งระบบปฏิบัติการคือ Debian

6) ถ้าหาก IP เครื่องที่เราใช้อยู่ ไม่ใช่ Public IP จริง ก็ต้องเปลี่ยนตรงบรรทัด local ให้เป็น IP ของเครื่องเรา ในที่นี้คือเปลี่ยนจาก

```
local 160.18.2.109
```

เป็น

```
local 172.16.55.68
```

7) จากนั้นก็นำค่าที่ได้มาใส่ในเครื่อง เพื่อทำการตั้งค่า โดยการเปิด Terminal ขึ้นมาแล้วพิมพ์

```
sudo nano /etc/network/interfaces
```

แล้วกด Enter

8) เมื่อเปิดโปรแกรมขึ้นมาแล้ว ให้เลื่อนมาที่บรรทัดล่างสุด แล้วพิมพ์ตามคำสั่งในเว็บ

```
tunnelbroker.net
auto he-ipv6
iface he-ipv6 inet6 v4tunnel
    address [Client IPv6 Address]
    netmask 64
    endpoint [Server IPv4 Address]
    local [IP ของเครื่องเรา]
    ttl 255
    gateway [Server IPv6 Address]
```

แล้วกด Ctrl+X >>> กด Y >>> แล้วกด Enter

จากนั้นก็พิมพ์

```
sudo nano /etc/dhcpd.conf
```

แล้วกด Enter

9) เลื่อนลงมาที่บรรทัดล่างสุด ถ้าต่ออินเทอร์เน็ตผ่านสายแลนให้ใช้ eth0 แต่ถ้าต่อผ่านไวไฟให้ใช้ wlan0 แล้วใช้คำสั่งตามนี้

```
interface eth0
    static ip_address=[IP ของเครื่องเรา/netmask]
    static routers=[Default Gateway]
    static domain_name_servers=[DNS]
```

เมื่อพิมพ์เสร็จแล้วให้กด Ctrl+X >>> กด Y >>> แล้วกด Enter

จากนั้นลองใช้คำสั่ง ping6 ipv6.google.com ถ้าสามารถ ping ไปหา IPv6 ของ Google ได้ ก็แสดงว่าทำสำเร็จ

HURRICANE ELECTRIC INTERNET SERVICES

Account Menu
Main Page
Account Info
Logout

User Functions
Create Regular Tunnel
Create BGP Tunnel
IPv6 Portscan

Tunnel Details
IPv6 Tunnel | Example Configurations | Advanced

Tunnel ID: 367917
Creation Date: Oct 13, 2016
Description: [Empty field]

IPv6 Tunnel Endpoints
Server IPv4 Address: 216.218.221.6
Server IPv6 Address: 2001:470:18:1266::1/64
Client IPv4 Address: 110.164.92.124
Client IPv6 Address: 2001:470:18:1266::2/64

Routed IPv6 Prefixes
Routed /64: 2001:470:19:1266::/64
Routed /48: Assign /48

DNS Resolvers
Anycast IPv6 Caching Nameserver: 2001:470:20::2
Anycast IPv4 Caching Nameserver: 74.82.42.42

rDNS Delegations
rDNS Delegated NS1:
rDNS Delegated NS2:
rDNS Delegated NS3:
rDNS Delegated NS4:
rDNS Delegated NS5:

Quick Links
Certification
Tunnelbroker
Free DNS
BGP Toolkit
Net Tools App
Forums
FAQ
Video Presentations
Usage Statistics
Tunnel Server Status
Network Map
Looking Glass (v4/v6)
Route Server (telnet)
Global IPv6 Report

Services
Transit
Colocation
Dedicated Servers

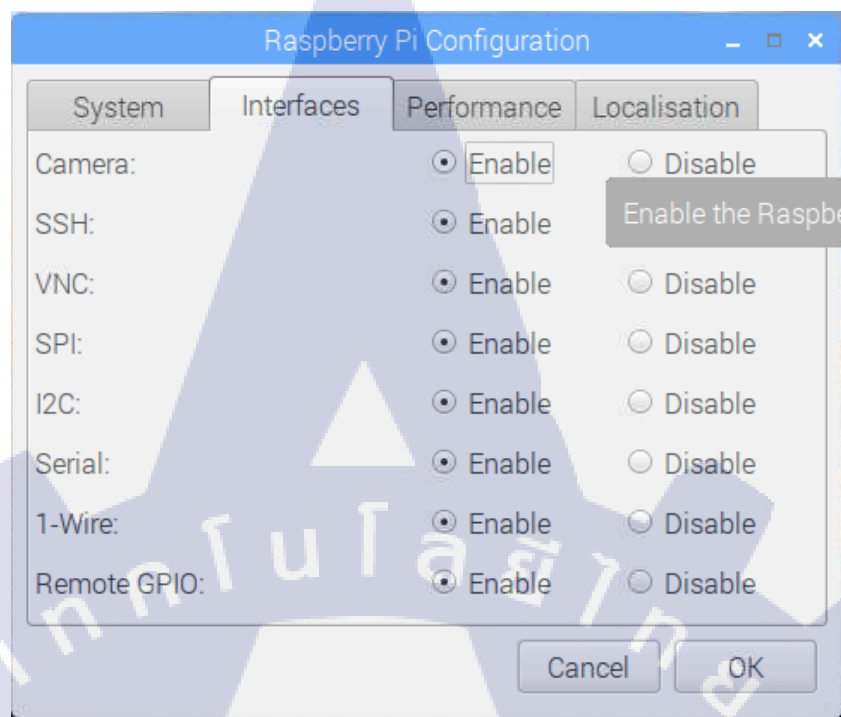
v4 Exhaustion
IPv4 & IPv6 Statistics
RIR v4 IPs Left
AfriNIC 18,062,610
APNIC 6,891,757
ARIN 0
LACNIC 38,366
RIPE 12,549,131
v6 ASNs

รูปที่ 3.14 หน้าต่างหลังสร้างอุโมงค์ที่เว็บไซต์ tunnelbroker.net สำเร็จ

3.3.8.2 การต่อกล้อง Raspberry Pi NoIR Camera กับการเปิดใช้งาน

ในแพ็คเกจของกล้อง Raspberry Pi NoIR Camera V2 จะมีตัวกล้องติดมาพร้อมกับสายแพ ให้เรานำสายแพมาเสียบกับช่อง Camera ของอุปกรณ์ Raspberry Pi

จากนั้นให้เข้าไปตั้งค่าที่ Raspberry Pi Configuration แล้วไปเลือก Enable ที่หัวข้อ Camera ที่แท็บ Interface



รูปที่ 3.15 หน้าต่างการตั้งค่าของอุปกรณ์ Raspberry Pi

3.3.8.3 การติดตั้ง Boto3 และ AWS CLI เพื่อใช้งาน Amazon Simple Storage Services

การติดตั้ง Boto3 ให้พิมพ์คำสั่งลงใน Command-line ดังนี้

```
sudo pip install boto3
```

การติดตั้ง AWS CLI ให้พิมพ์คำสั่งลงใน Command-line ดังนี้

```
sudo pip install awscli
```

จากนั้นพิมพ์คำสั่งเพื่อยืนยันตัวตนดังนี้

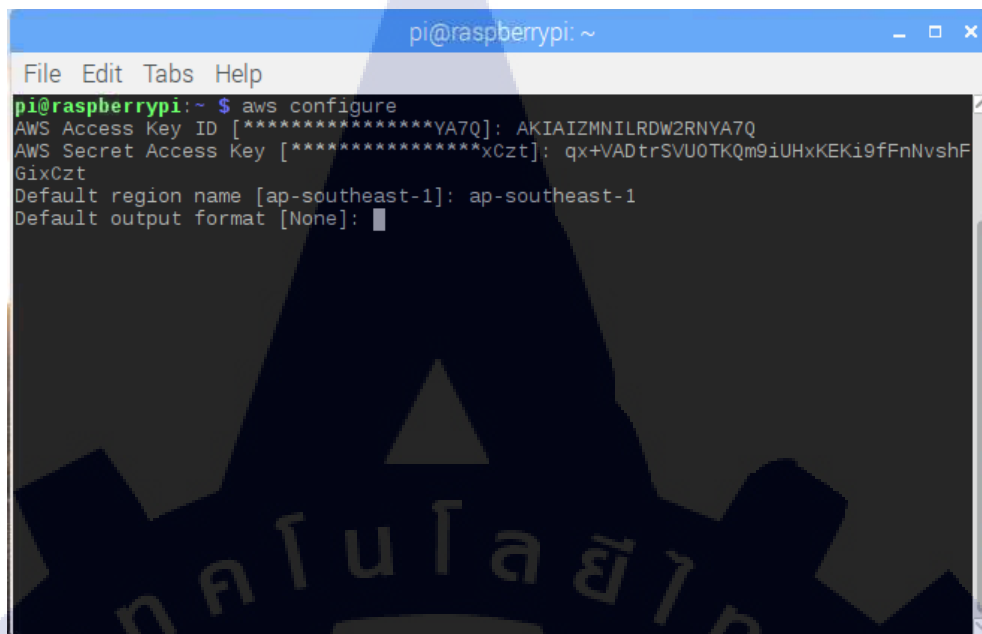
```
aws configure
```

AWS Access Key ID : [ใส่ key id ที่ได้มาจาก AWS]

AWS Secret Access Key : [ใส่ secret key ที่ได้มาจาก AWS]

Default Region Name : ap-southeast-1

Default output format : [เว้นว่าง]



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ aws configure
AWS Access Key ID [*****YA7Q]: AKIAIZMNILRDW2RNYA7Q
AWS Secret Access Key [*****xCzt]: qx+VADtrSVU0TKm9iUHxkEKi9fFnNvshF6ixCzt
Default region name [ap-southeast-1]: ap-southeast-1
Default output format [None]:

```

รูปที่ 3.16 หน้าต่างการตั้งค่าของ AWS CLI

3.3.8.4 เขียนโปรแกรมให้กล้องถ่ายรูปเมื่อตรวจจับความเคลื่อนไหวได้ และเมื่อถ่ายรูปแล้ว

ให้อัปโหลดรูปขึ้น Amazon Simple Storage Services

ใช้คำสั่งเพื่อติดตั้งไลบรารีของ Ubidots เพื่อใช้ติดต่อกับเว็บไซต์ ดังนี้

```
sudo pip install ubidots
```

จากนั้น เขียนโปรแกรมด้วยภาษาไพธอน ดังนี้

```

from ubidots import ApiClient
from picamera import PiCamera
from time import sleep
import boto3
import sys
import os

camera = PiCamera()
camera.rotation = 180
camera.resolution = (1920,1080)
api = ApiClient(token='[ใส่ token key ที่ได้จากเว็บไซต์ ubidots]')

```

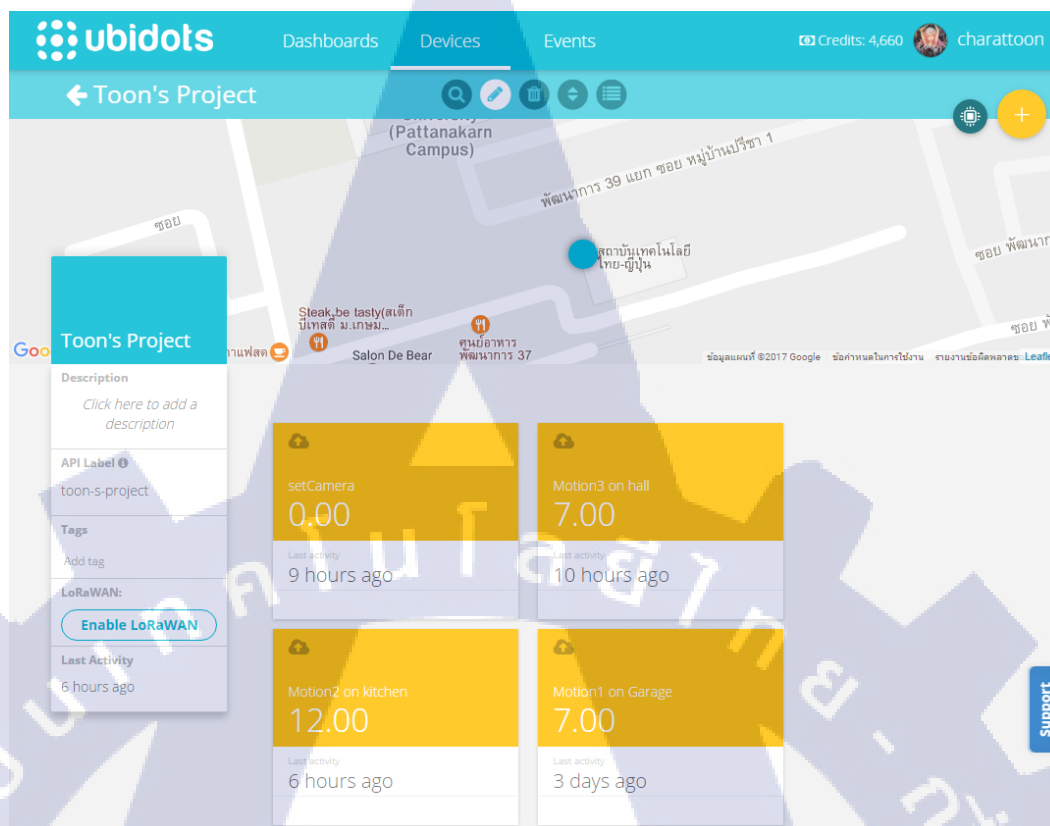
```

variable = api.get_variable("[ใส่ variable key ที่สร้างจากเว็บไซต์ ubidots]")
arg = sys.argv
i = int(arg[1])
s3 = boto3.resource('s3')
while(1):
    last_value = variable.get_values(1)
    print last_value[0]['value']
    if(last_value[0]['value'] == 1):
        camera.capture('/home/pi/Pictures/image%s.jpg' %i)
        data=open('/home/pi/Pictures/image%s.jpg' %i, 'rb')
        obj=s3.bucket('homesecond').put_object(Key='pic/image%s.jpg' %i,
            Body=data)
        obj.Acl.put(ACL='public-read')
        os.remove('/home/pi/Pictures/image%s.jpg' %i)
    sleep(1)

```

3.3.9 การตั้งค่าบนเว็บไซต์ Ubidots

- 1) เข้าไปที่ <https://app.ubidots.com/> และทำการสมัครสมาชิก
- 2) ไปที่แท็บ Devices กดเครื่องหมาย + เพื่อสร้าง Device
- 3) เมื่อสร้างเสร็จ คลิกเข้าไปใน Device แล้วกด + เพื่อสร้าง Variable key สำหรับรับ-ส่งค่าจากเซ็นเซอร์



รูปที่ 3.17 หน้าต่างการสร้างตัวแปรเพื่อรับค่าจากเซ็นเซอร์

4) คลิกที่แท็บ Events กด + เพื่อสร้างการแจ้งเตือนเมื่อเกิดการเคลื่อนไหวขึ้น

-เลือก Device

-เลือก Variable

-เลือก greater มากกว่า 1000

-จากนั้นเลือกให้ส่งทางอีเมลและข้อความมือถือ หรือจะให้ส่งค่ากลับไปยังอุปกรณ์

Raspberry Pi เพื่อทำการถ่ายรูป แล้วกด Finish

Select a Device

Filter Devices

Select a Variable

Filter Variables

My Data Source Toon's Project Bank's Project

setCamera Motion3 on hall Motion2 on kitchen Motion1 on Garage

if

Motion1 on Garage

Value

less greater less or equal greater or equal equal

than

1000

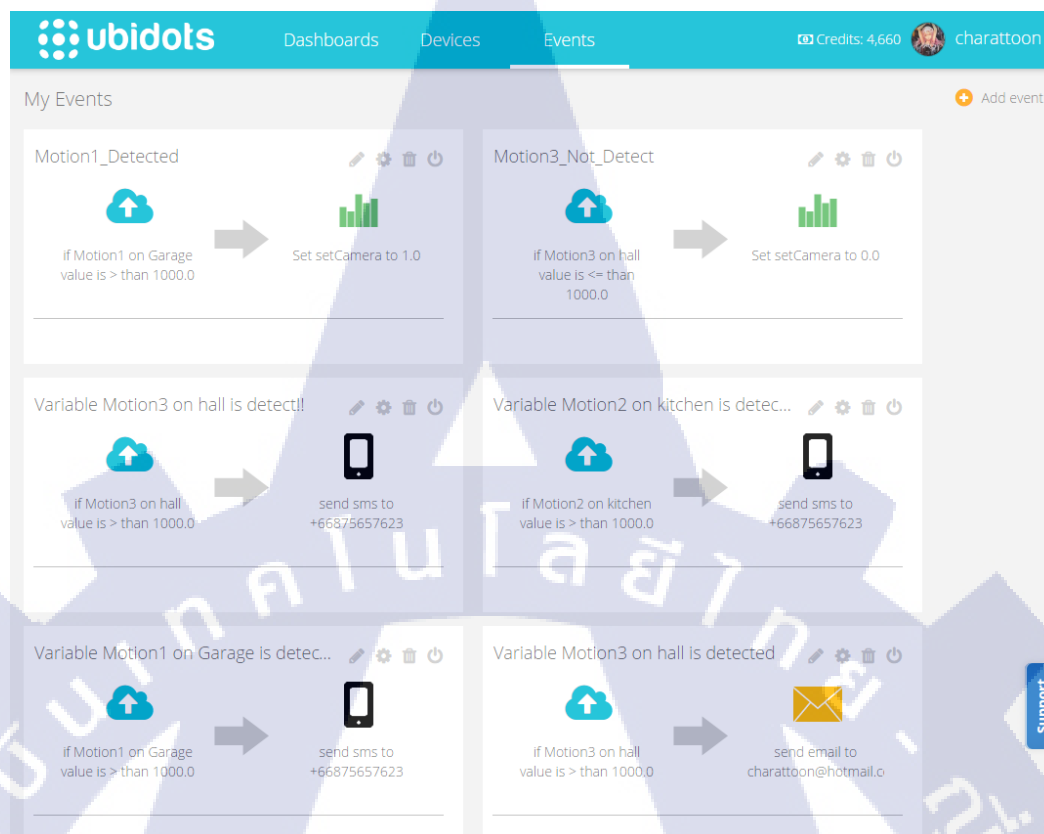
Continue

Send e-mail Send SMS Send Telegram WebHook Set a variable

Email charattoon@hotmail.com

Add email

รูปที่ 3.18 หน้าต่างการตั้งค่า Events ในเว็บไซต์ Ubidots

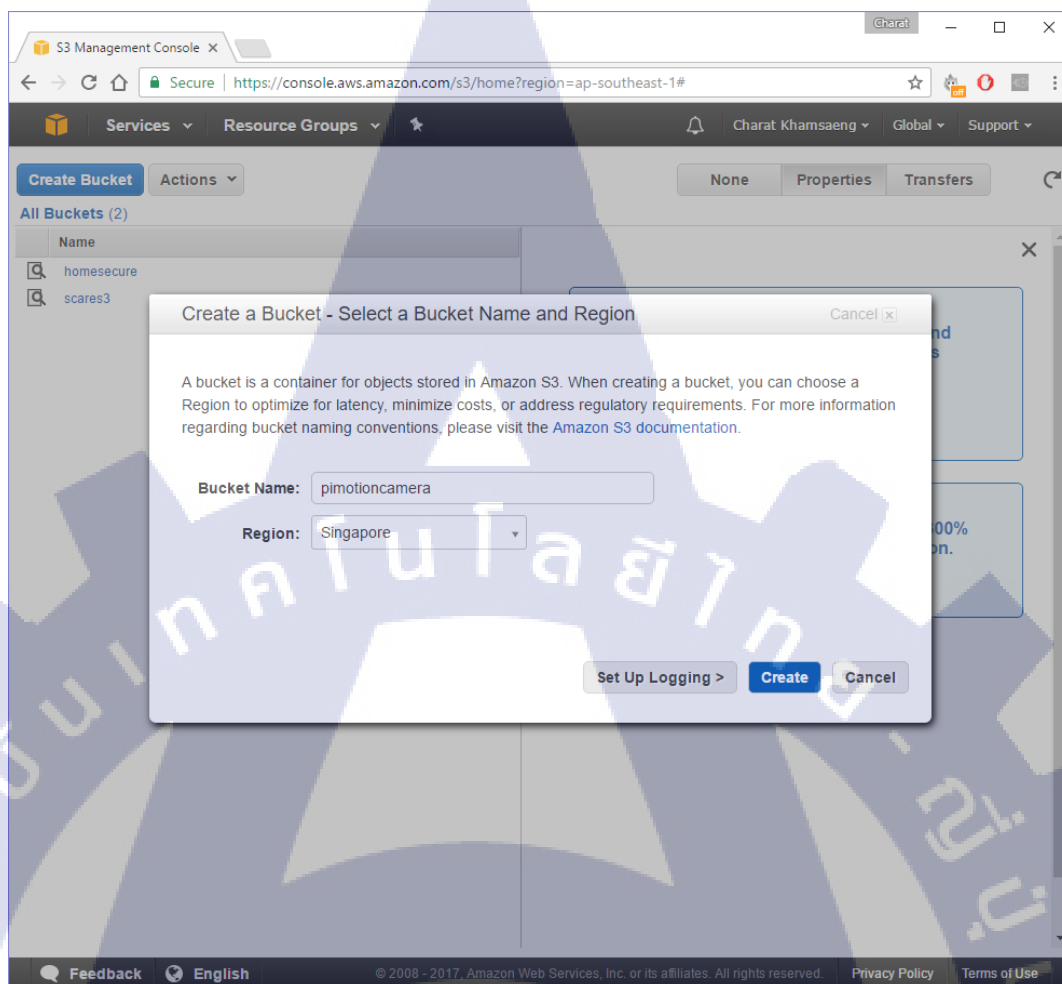


รูปที่ 3.19 หน้าต่างเมื่อสร้าง Events สำเร็จ

เพียงเท่านี้ เมื่อเกิดการเคลื่อนไหว ระบบก็จะทำการแจ้งเตือนไปยังอีเมลและข้อความมือถือ

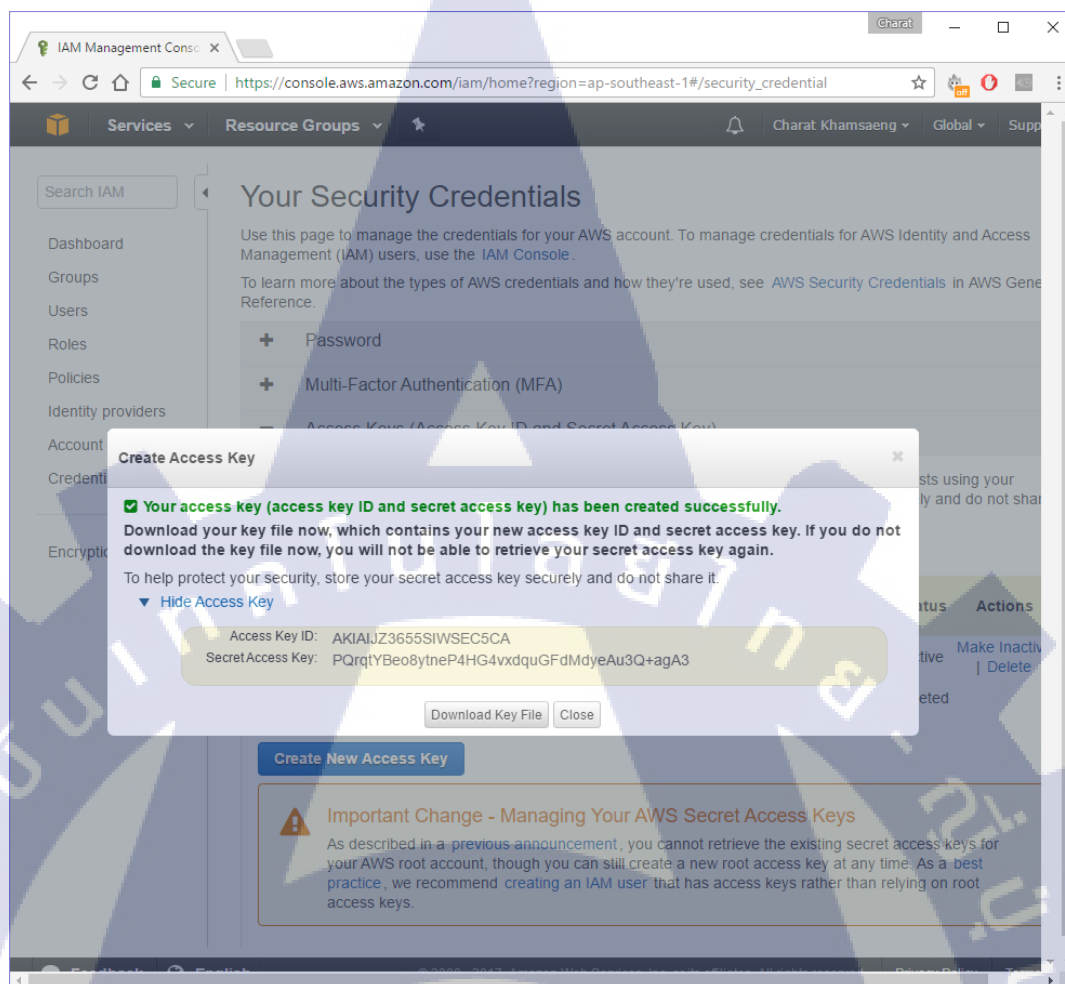
3.3.10 การตั้งค่าบนเว็บไซต์ Amazon Simple Storage Services

- 1) เข้าไปที่ <https://console.aws.amazon.com/console/home> แล้วทำการสมัครสมาชิก
- 2) เลือกบริการ Storage S3
- 3) กด Create Bucket > ใส่ชื่อของ Bucket แล้วเลือกภูมิภาค จากนั้นกด Create



รูปที่ 3.20 ขั้นตอนการสร้าง Bucket

- 4) กดที่ชื่อข้างบนขวา แล้วเลือก My Security Credentials
- 5) คลิกที่ Access Keys (Access Key ID and Secret Access Key) แล้วกด Create New Access Key



รูปที่ 3.21 หน้าต่างการขอ Access Key และ Secret Access Key

- 6) จะปรากฏหน้าต่าง Access Key และ Secret Access Key ขึ้นมา ให้กด Download Key File ไว้ใช้ยืนยันตัวตนบนอุปกรณ์ Raspberry Pi

บทที่ 4

วิเคราะห์ผลการวิจัย

4.1 ประสิทธิภาพของ M-RPL อัลกอริทึม

4.1.1 ตัวอย่างการออกแบบ

การออกแบบอัลกอริทึมเพื่อจัดเส้นทางที่สามารถรองรับการสื่อสารทั้งแบบยูนิคาสต์ มัลติคาสต์และแอนนีกาสต์ของเครือข่ายเซ็นเซอร์ไร้สายด้วยอินเทอร์เน็ตโพรโตคอลรุ่นที่ 6 โดยให้ชื่อว่า M-RPL อัลกอริทึม โดยใช้เครื่องมือ NARVAL toolbox ของโปรแกรม Scilab Simulation ที่เป็นโปรแกรมฟรีแวร์ที่สามารถออกแบบอัลกอริทึมของอุปกรณ์เครือข่ายไร้สาย จำนวน 100 โหนด ที่มีการส่งข้อมูลทั้งยูนิคาสต์และมัลติคาสต์ที่มีความจุขนาด 10 Kbps.

4.1.2 M-RPL Algorithm “Pseudo Code”

ผู้วิจัยได้ออกแบบและกำหนดค่าต่าง ๆ เพื่อทดสอบอัลกอริทึม M-RPL ตาม Pseudo Code ดังต่อไปนี้

```
n=100; //network size
```

```
l=1000; //network squared area side
```

```
d=100; //Locality radius
```

```
[g]=NL_T_LocalityConnex(n,L,dmax); //generation of a topology in respect with the Locality method
```

```
i=NL_F_RandInt1n(length(g.node_x)); //selection of the source node
```

```
dw=2; //display parameter
```

```
ind=1; //window index
```

```
g.node_diam(i)=50; //node diameter
```

```
g.node_border(i)=10; //node border
```

```
g.node_color(i)=5; //node color
```

```
[f]=NL_G_ShowGraphN(g,ind); //graph visualization
```

```
[dist,pred]=NL_R_Dijkstra(g,i); //application of NL_R_Dijkstra
```

```
ETX=5;
```

```
[v] = NL_F_RandVector0nminus1(length(g.head),ETX); //update of weighth
```

```

v=v+1;

g.edge_weight=g.edge_length;

g.edge_length=v;

xc=l/2; //area center

yc=l/2;

[s]=NL_G_NodeClose2XY(g,xc,yc); //root node

c=5; //5 possible routes

[pred,dist,ra,DAG,DIO]=NL_R_RPL(g,s,c); //application of NL_R_RPL

[go]=NL_R_RPLPlot(g,pred); //highlight RPL tree

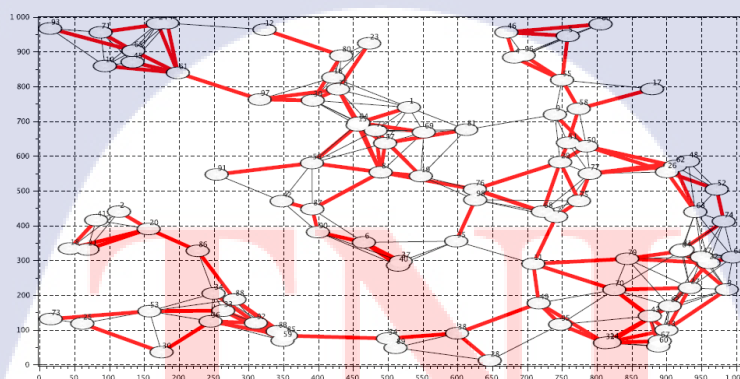
ind=1; //window index

f=NL_G_ShowGraphN(go,ind); //graph visualization

```

4.1.3 กราฟที่ได้จากการทดสอบ

จากรูปที่ 4.1 แสดงเส้นทางการสื่อสารทั้งแบบยูนิคาสต์และมัลติคาสต์ ที่ได้จาก M-RPL อัลกอริทึม ที่ผู้วิจัยได้จำลองบนโปรแกรมจำลองเส้นทาง Scilab Simulation ของเซ็นเซอร์ไร้สายทั้ง 100 โหนด



รูปที่ 4.1 กราฟเส้นทางของการสื่อสารภายในเครือข่ายเซ็นเซอร์ไร้สาย

4.1.4 เปรียบเทียบค่าน้ำหนักในการจัดเส้นทาง

ตามตารางที่ 4.1-4.5 แสดงค่าน้ำหนักของการจัดเส้นทางด้วย RPL อัลกอริทึม เมื่อเปรียบเทียบกับ M-RPL อัลกอริทึม ที่มีการสื่อสารแบบยูนิคาสต์ที่เข้าออกภายในโหนด ที่ 64, 128, 256, 512 and 1024 kbps. และการสื่อสารแบบมัลติคาสต์ที่กำหนดตามตาราง ดังนี้

ตารางที่ 4.1 คำนวณน้ำหนักในการจัดเส้นทางมีการสื่อสารแบบยูนิคาสต์ที่เข้าออกภายในโหนด ที่ 64 KBPS.

Multicast Traffic	RPL	M-RPL
100 node $\times$ 128 Kbps	13430	11481

ตารางที่ 4.2 คำนวณน้ำหนักในการจัดเส้นทางมีการสื่อสารแบบยูนิคาสต์ที่เข้าออกภายในโหนด ที่ 128 KBPS.

Multicast Traffic	RPL	M-RPL
100 node $\times$ 128 Kbps	14726	12584

ตารางที่ 4.3 คำนวณน้ำหนักในการจัดเส้นทางมีการสื่อสารแบบยูนิคาสต์ที่เข้าออกภายในโหนด ที่ 256 KBPS.

Multicast Traffic	RPL	M-RPL
100 node $\times$ 128 Kbps	15674	13962

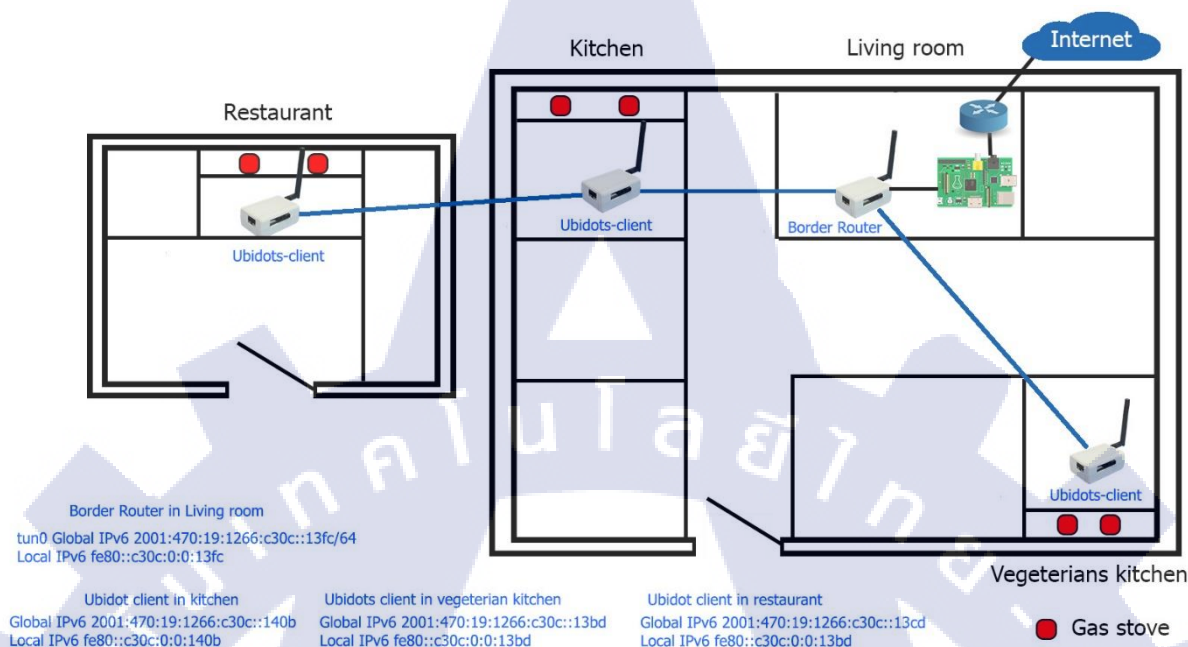
ตารางที่ 4.4 คำนวณน้ำหนักในการจัดเส้นทางมีการสื่อสารแบบยูนิคาสต์ที่เข้าออกภายในโหนด ที่ 512 KBPS.

Multicast Traffic	RPL	M-RPL
100 node $\times$ 128 Kbps	18385	16638

ตารางที่ 4.5 คำนวณน้ำหนักในการจัดเส้นทางมีการสื่อสารแบบยูนิคาสต์ที่เข้าออกภายในโหนด ที่ 1024 KBPS.

Multicast Traffic	RPL	M-RPL
100 node $\times$ 128 Kbps	20068	18753

4.2 ผลการวิจัยผ่านอุปกรณ์เซ็นเซอร์ไร้สาย กรณีที่ 1

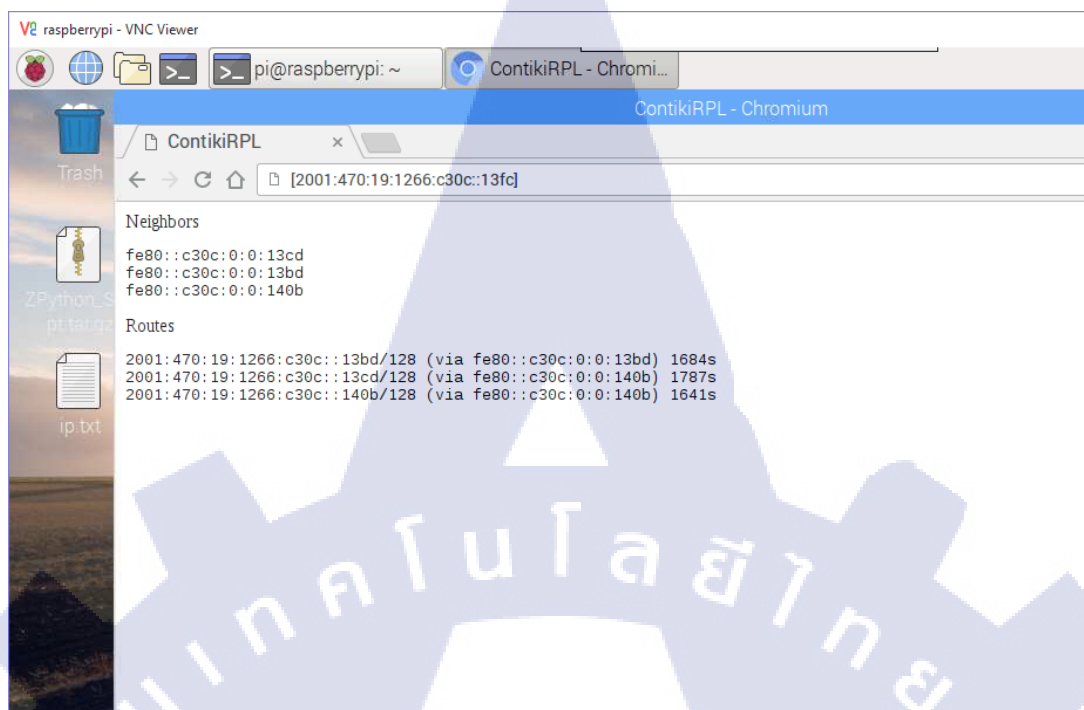


รูปที่ 4.2 กรณีศึกษาสถานการณ์จำลองระบบแจ้งเตือนอัคคีภัยในครัวเรือน

การนำระบบแจ้งเตือนอัคคีภัยในครัวเรือนด้วยเทคโนโลยี IoT6 มาติดตั้งในบ้าน โดยจากรูป 4.2 แสดงถึงบ้านขนาดใหญ่มีหลายห้อง มีสองห้องครัวโดยห้องครัวแรกเป็นห้องครัวสำหรับอาหารมังสวิรัตและห้องที่สองเป็นห้องครัวสำหรับอาหารทั่วไป โดยมีพื้นที่ขนาดเล็กข้างบ้านเปิดกิจการเป็นร้านอาหารขนาดเล็กซึ่งทั้งสามห้องนั้นมีการใช้งานเตาแก๊ส

ในส่วนของห้องนั่งเล่นนั้นเป็นตำแหน่งที่ใช้วางเราเตอร์และเชื่อมต่อกับ Raspberry pi 3 โดยติดตั้ง Border Router บน Z1 ในห้องนั่งเล่น

ในส่วนของห้องห้องครัวสำหรับอาหารมังสวิรัต ห้องครัวสำหรับอาหารทั่วไป และร้านอาหารข้างๆ นำ Z1 ที่ติดตั้งโปรแกรม Ubidots Client พร้อมติดเซ็นเซอร์ไฟและแก๊สไว้



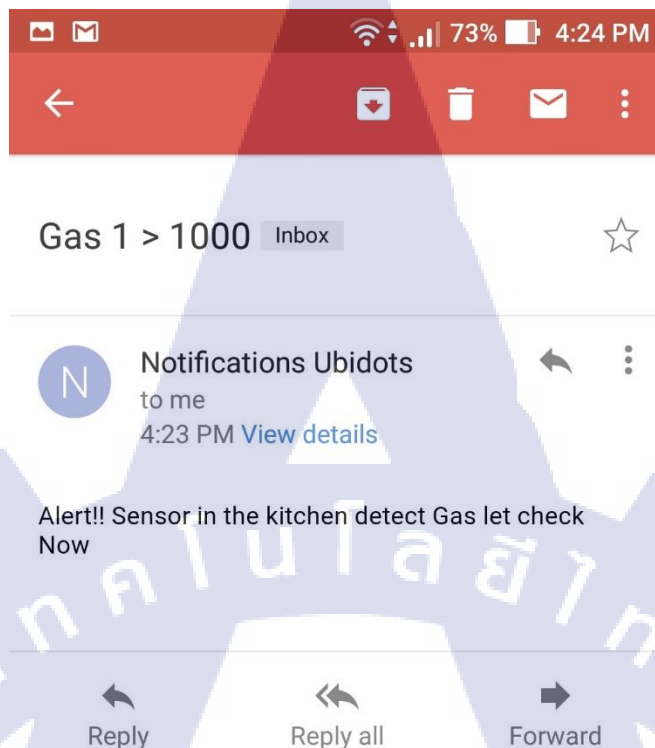
รูปที่ 4.3 ตารางการหาเส้นทางของ Border Router

โดยเมื่อวางอุปกรณ์ตามรูปที่ 4.2 จากนั้นให้เปิดเว็บเบราว์เซอร์บน Raspberry pi จากนั้นใส่หมายเลขไอพีเข้าไปที่ช่อง URL จะพบดังรูปที่ 4.3 แสดงถึงตารางการหาเส้นทางบน Border Router โดย Border Router จะถูกตั้งในห้องนั่งเล่นและเชื่อมต่อกับ Z1 ในห้องสำหรับครัวทั่วไปและ Z1 ในห้องครัวสำหรับอาหารมังสวิรัตได้โดยตรง โดยดูจากไอพี ::13bd และ ::140b ว่าเชื่อมต่อกับไอพีภายในตัว Z1 ของตัวเอง แต่สำหรับ Z1 ในร้านอาหารข้างๆบ้านนั้นเชื่อมต่อมายัง Border Router ผ่าน Z1 ในห้องครัวทั่วไปโดยดูจากไอพี ::13cd ที่เชื่อมต่อผ่าน ::140b

จากการทดลองใช้งาน Z1 เซ็นเซอร์ไฟและแก๊สสามารถทำงานร่วมกันได้ โดยสามารถแจ้งเตือนไปยังผู้ใช้งานได้โดยจะแบ่งออกเป็นกรณีดังนี้

4.2.1 ห้องครัวสำหรับอาหารมังสวิรัตเกิดแก๊สรั่ว

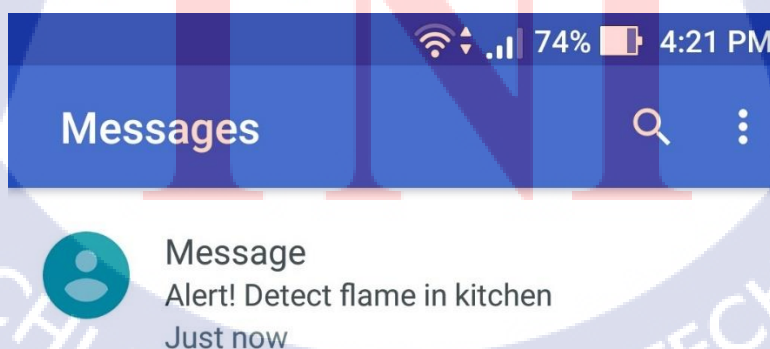
เมื่อเกิดกรณีแก๊สรั่วออกมาจากถังแก๊ส เซ็นเซอร์แก๊สในห้องครัวสำหรับอาหารมังสวิรัตจะตรวจพบแก๊สรั่วออกมาจากถังแก๊สและส่งค่าไปยัง Border Router ในห้องนั่งเล่นและส่งค่าผ่าน Raspberry pi และเราเตอร์ไปยัง Ubidots โดยค่าที่ถูกส่งไปจะถูกแสดงใน Ubidots และเมื่อค่าเกิดการเปลี่ยนแปลงซึ่งเป็นเรื่องสำคัญว่าตรวจพบแก๊สจะทำการแจ้งเตือนมายังผู้ใช้งานด้วยเมลและข้อความมายังเบอร์โทรศัพท์ที่ได้ตั้งค่าไว้ใน Ubidots



รูปที่ 4.4 เมลแจ้งเตือนจาก Ubidots

4.2.2 ห้องครัวสำหรับอาหารทั่วไปตรวจพบไฟ

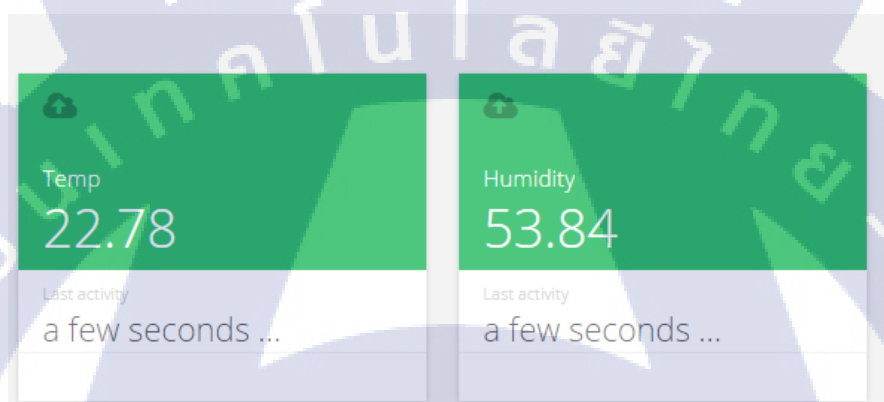
เมื่อเกิดกรณีเกิดเหตุเพลิงไหม้ในห้องอันเนื่องมาจากเหตุผลใดๆ เช่น เซอร์โไฟในห้องครัวสำหรับอาหารทั่วไปจะตรวจพบไฟในห้อง และจะส่งค่าไฟที่รับมาไปยัง Border Router ในห้องนั่งเล่นและส่งค่าผ่าน Raspberry pi และเราเตอร์ไปยัง Ubidots โดยค่าที่ถูกส่งไปจะถูกแสดงใน Ubidots และเมื่อค่าเกิดการเปลี่ยนแปลงซึ่งเป็นสัญญาณว่าตรวจพบไฟจะทำการแจ้งเตือนมายังผู้ใช้งานด้วยข้อความมายังเบอร์โทรศัพท์ที่ได้ตั้งค่าไว้ใน Ubidots



รูปที่ 4.5 ข้อความแจ้งเตือนจาก Ubidots

4.2.3 ตรวจสอบค่าความชื้นและอุณหภูมิในร้านอาหารข้างบ้าน

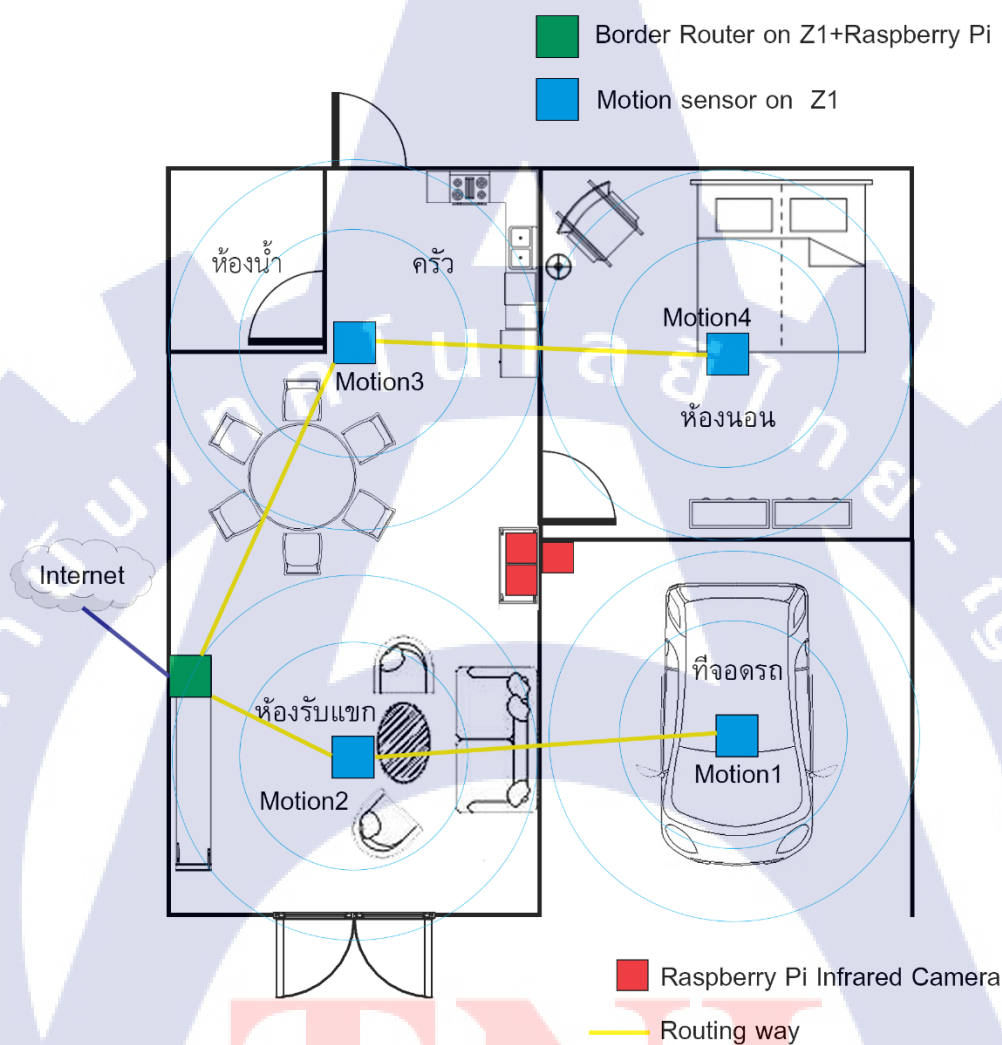
เมื่อเกิดกรณีแก๊สรั่วออกมาจากถังแก๊ส เช่น เซอร์แก๊ส ในร้านอาหารข้างบ้านจะตรวจพบแก๊สรั่วออกมาจากถังแก๊สและส่งค่าไปยัง Border Router ในห้องนั่งเล่น ผ่าน Z1 ในห้องครัวสำหรับอาหารทั่วไป ซึ่ง Z1 ในห้องครัวสำหรับอาหารทั่วไปจะทำการส่งต่อข้อมูลไปให้ Border Router และส่งค่าผ่าน Raspberry pi และเราเตอร์ไปยัง Ubidots โดยค่าที่ถูกส่งไปจะถูกแสดงใน Ubidots และเมื่อค่าเกิดการเปลี่ยนแปลงซึ่งเป็นสัญญาณว่าตรวจพบแก๊สจะทำการแจ้งเตือนมายังผู้ใช้งานด้วยอีเมลและข้อความมายังเบอร์โทรศัพท์ที่ได้ตั้งค่าไว้ใน Ubidots



รูปที่ 4.6 การแสดงข้อมูลบน Ubidots

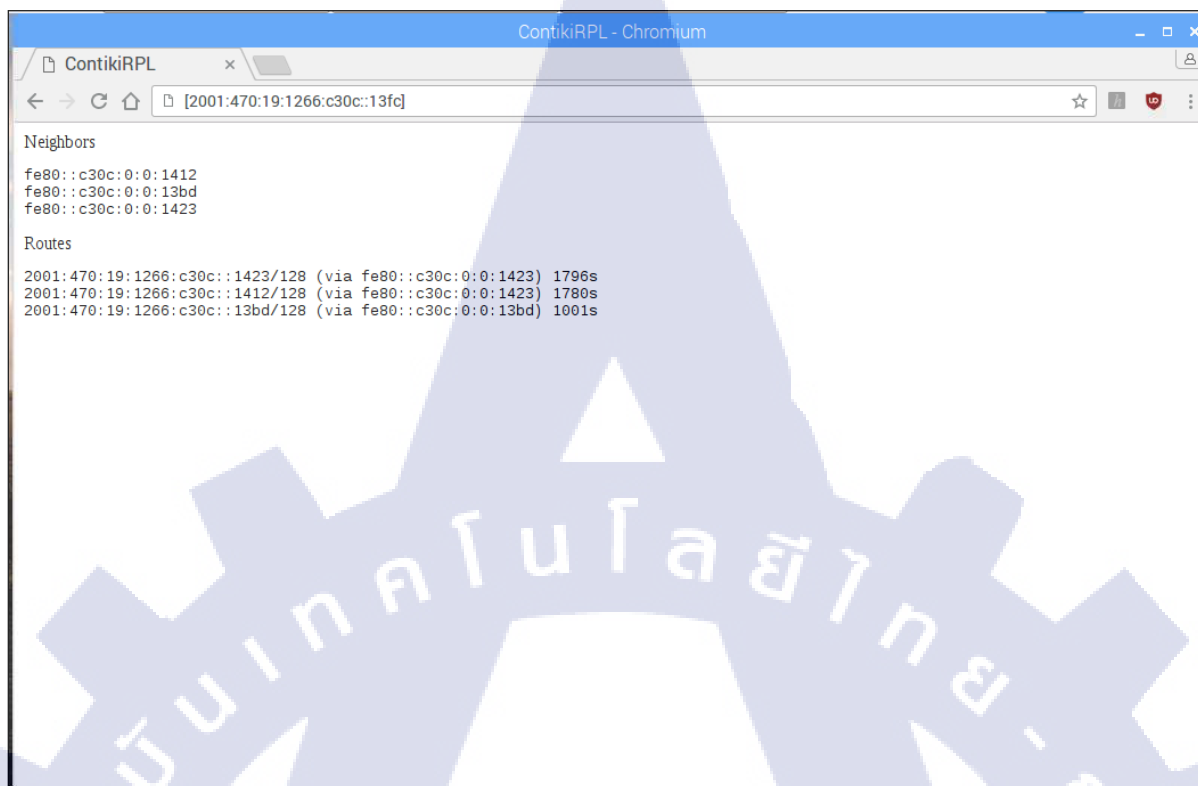
4.3 ผลการวิจัยผ่านอุปกรณ์เซ็นเซอร์ไร้สาย กรณีที่ 2

4.3.1 การติดตั้งในสถานที่จริง



รูปที่ 4.7 ผังการติดตั้งอุปกรณ์ภายในบ้าน

การติดตั้งระบบป้องกันการบุกรุกบ้าน จากรูปที่ 4.7 จะเห็นได้ว่าการติดตั้ง Motion Sensor ไว้ที่ห้องรับแขก ห้องครัว ห้องนอน รวมถึงที่จอดรถด้วย โดยตัว Motion4 จะมีการส่งข้อมูล hop-by-hop ผ่านตัว Motion3 ไปยัง Border Router และตัว Motion1 ส่งข้อมูลผ่านตัว Motion2 ไปยัง Border Router เพื่อออกสู่อินเทอร์เน็ต และมีการติดตั้งกล้อง Raspberry Pi NoIR Camera ไว้ที่ห้องรับแขก ห้องครัว และที่จอดรถ



รูปที่ 4.8 หน้าต่างแสดงตารางการเชื่อมต่ออุปกรณ์ผ่าน RPL โพรโทคอล

โดยจะสามารถตรวจสอบการเชื่อมต่อได้จากการเปิดเว็บเบราว์เซอร์ขึ้นมา แล้วใส่ IPv6 ของตัว Border Router ก็จะขึ้นหน้าต่างแสดงตารางของอุปกรณ์ที่เชื่อมต่ออยู่ผ่าน RPL โพรโทคอล ดังรูปที่ 4.1.2 จะเห็นว่าอุปกรณ์ที่มีไอพีรุ่นที่หก หมายเลข ::1412 ได้เชื่อมต่อกับ Border Router ผ่านอุปกรณ์ที่มีไอพีรุ่นที่หก หมายเลข ::1423

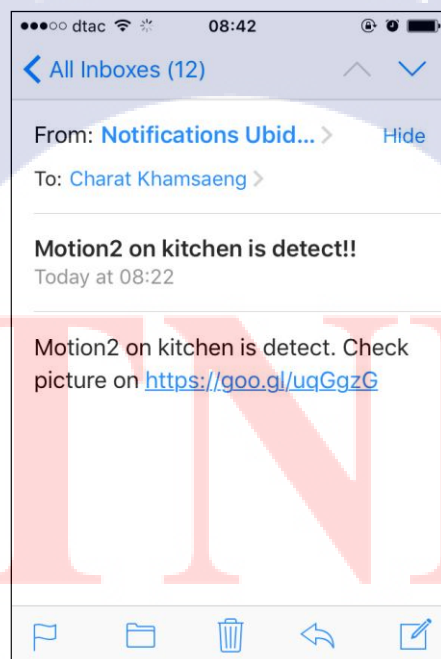
4.3.2 สามารถแจ้งเตือนผ่านอีเมลและข้อความมือถือ

การทดสอบอุปกรณ์ในสถานการณ์จำลอง โดยใช้ห้องวิจัย A604 ชั้น 6 อาคาร A สถาบันเทคโนโลยีไทย-ญี่ปุ่น เป็นสถานที่ทดสอบ ได้มีการติดตั้ง Motion Sensor ไว้ที่ประตูของห้องวิจัยทั้ง 2 บาน และใช้กล้อง Raspberry Pi NoIR Camera ในการถ่ายรูป โดยรูปที่ถ่ายได้จะเห็นประตูครบทั้ง 2 บาน โดยบานแรกใช้ Motion1 ในการตรวจจับ และบานที่ 2 ใช้ Motion2 ในการตรวจจับ

ทดสอบโดยเริ่มจากเดินผ่านประตูที่ 2 ตัว Motion2 ก็จับความเคลื่อนไหวได้และส่งทั้งอีเมลและข้อความมือถือมายังผู้ใช้ และเมื่อเดินผ่านประตูที่ 1 ตัว Motion1 ก็จับความเคลื่อนไหวได้เช่นกัน และมีการส่งอีเมลและข้อความมือถือมายังผู้ใช้ด้วยเช่นกัน ดังรูปที่ 4.9 และรูปที่ 4.10



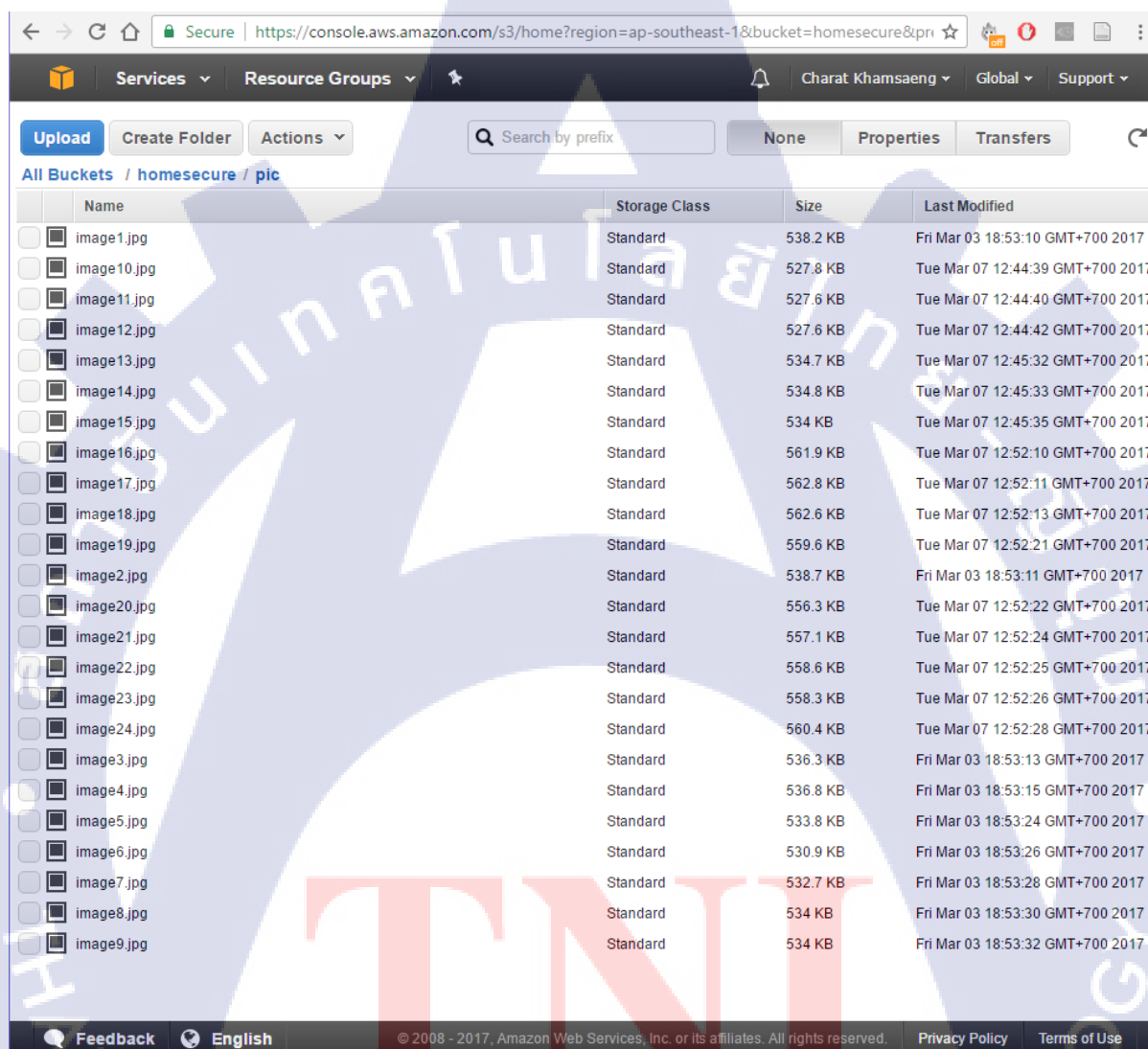
รูปที่ 4.9 ข้อความมือถือที่ถูกส่งมาเมื่อมีการตรวจจับความเคลื่อนไหว



รูปที่ 4.10 อีเมลที่ถูกส่งมาเมื่อมีการตรวจจับความเคลื่อนไหว

4.3.3 สามารถอัปโหลดรูปถ่ายขึ้น Amazon Simple Storage Services

เมื่อถ่ายรูปแล้ว ก็ทำการอัปโหลดรูปภาพขึ้นไปยัง Amazon Simple Storage Services เพื่อทำการเก็บไว้ แล้วจึงลบรูปภาพออกจากอุปกรณ์ Raspberry Pi ซึ่งสามารถอัปโหลดรูปถ่ายได้ และสามารถลบรูปภาพหลังจากอัปโหลดได้เช่นกัน



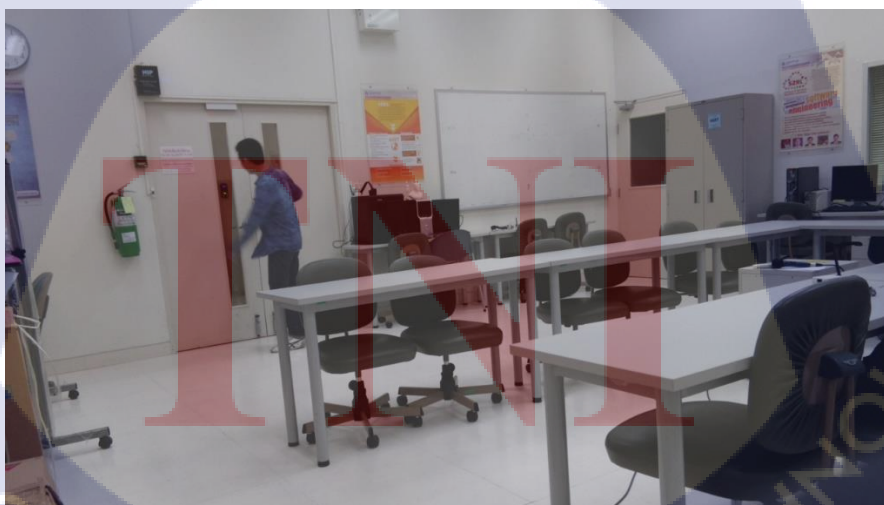
รูปที่ 4.11 รูปถ่ายที่ถูกอัปโหลดขึ้นไปที่ Amazon Simple Storage Services

4.3.4 แสดงการถ่ายรูปของกล้อง Raspberry Pi NoIR Camera

กล้อง Raspberry Pi NoIR Camera จะถ่ายรูปก็ต่อเมื่อ Motion Sensor ตรวจจับความเคลื่อนไหวได้ แล้วส่งค่าขึ้นไปเว็บไซต์ Ubidots จากนั้น เว็บไซต์ Ubidots จะส่งค่ากลับมายัง Raspberry Pi เพื่อสั่งให้กล้องถ่ายรูป โดยจากสถานการณ์จำลอง กล้องสามารถถ่ายรูปได้ตามที่ต้องการดังรูปที่ 4.12 และรูปที่ 4.13



รูปที่ 4.12 ภาพที่ถูกถ่ายหลังจาก Motion2 ตรวจจับความเคลื่อนไหวได้



รูปที่ 4.13 ภาพที่ถูกถ่ายหลังจาก Motion1 ตรวจจับความเคลื่อนไหวได้

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงานวิจัย

การศึกษาวิจัยในโครงการนี้ได้ทำการออกแบบเกณฑ์วิธีเพื่อจัดเส้นทางของการสื่อสารภายในเครือข่ายเซ็นเซอร์ไร้สายที่มีการสื่อสารแบบยูนิคาสต์ มัลติคาสต์และแอนนิกาสต์ ด้วยอินเทอร์เนตโพรโทคอลรุ่นที่ 6 โดยให้มีชื่อว่า M-PRL อัลกอริทึม โดยอัลกอริทึมดังกล่าวผู้วิจัยได้พัฒนาต่อขยายมาจาก RPL อัลกอริทึมที่มีการเรียกใช้การทำงานของ Trickle Multicast อัลกอริทึมร่วมกับ Prim-Dijkstra อัลกอริทึมในการจัดการเส้นทางของข้อมูลประเภทต่าง ๆ ภายในโครงสร้างข้อมูลแบบ Spanning Tree ที่ใช้จำนวน 100 โหนดทดลองในการออกแบบครั้งนี้ ผลปรากฏว่าค่าน้ำหนักที่ได้จาก M-PRL อัลกอริทึมที่ผู้วิจัยออกแบบสามารถจัดเส้นทางได้ดีกว่าค่าน้ำหนักที่ได้จาก RPL อัลกอริทึมเดิมในทุกกรณีที่ทำาทดลอง และผู้วิจัยยังได้ทดลองบนอุปกรณ์จริงก็สามารถรับส่งค่าภายในเครือข่ายเซ็นเซอร์ไร้สายได้เป็นอย่างดี

โดยอุปกรณ์สามารถทำงานร่วมกับเซ็นเซอร์ได้ดี สามารถส่งค่าขึ้น Ubidots ได้และแจ้งเตือนมายังสมาร์ทโฟนได้ตลอดเวลาตามวัตถุประสงค์ ซึ่งภายในห้องทดลองสามารถทำงานได้ดี แต่การจะนำไปใช้งานจริงจำเป็นต้องมีไอพีรุ่นที่สี่แบบสาธารณะในการใช้งานซึ่งการขอตามบ้านเรือนอาจจะเป็นไปได้ยากและยังต้องเปิดพอร์ต ICMP และโพรโทคอล 41 หากต้องการทำ Tunnel จึงทำให้ในตอนนี้สามารถทำการทดลองได้เพียงภายในแลปของสถาบันเทคโนโลยีไทย-ญี่ปุ่น เท่านั้น

และอุปกรณ์ Zolertia Z1 สามารถทำงานร่วมกับเซ็นเซอร์ได้ดี สามารถตรวจจับความเคลื่อนไหวได้ สามารถใช้งานไอพีรุ่นที่หกได้ โดยการทำโมดูลไอพีรุ่นที่หกให้วิ่งผ่านเครือข่ายไอพีรุ่นที่สี่ สามารถส่งค่าของเซ็นเซอร์ขึ้นไปยังเว็บไซต์ของ Ubidots ได้ สามารถทำการแจ้งเตือนด้วยอีเมลล์และข้อความมือถือได้ในทันทีที่ Motion Sensor ตรวจจับความเคลื่อนไหวได้ และสามารถใช้กล้อง Raspberry Pi NoIR Camera ถ่ายรูปเพื่อระบุตัวตนของผู้บุกรุกได้

5.2 ปัญหา

5.2.1 ไอพีรุ่นที่สี่แบบสาธารณะ

ในห้องทดลองสามารถใช้งานไอพีรุ่นที่สี่แบบสาธารณะได้ได้แต่ในการใช้งานตามบ้านเรือนจริงนั้นยังเป็นปัญหาอยู่ซึ่งหากทางตามบ้านเรือนสามารถใช้งานไอพีรุ่นที่สี่แบบสาธารณะ หรือใช้งานไอพีรุ่นที่หกได้จะสามารถแก้ปัญหาในส่วนนี้ได้

5.2.2 Z1 ต้องเชื่อมต่อ Micro USB

Z1 มีพอร์ตเชื่อมต่อกับเซ็นเซอร์แบบ 5V และ 3.3V ซึ่งหากเชื่อมต่อกับเซ็นเซอร์ด้วยพอร์ต 5V เพื่อใช้งานจะต้องเชื่อมต่อพอร์ต Micro USB เพื่อให้พลังงานเพียงพอ ซึ่งเซ็นเซอร์แก๊สนั้นจำเป็นต้องใช้เชื่อมต่อกับพอร์ต 5V หากต่อกับพอร์ต 3.3V จะส่งค่าผิด ดังนั้นในการทดลองจึงจำเป็นต้องต่อกับ Micro USB เพื่อทดลอง

5.2.3 ปัญหาที่ Ubidots

Ubidots เป็นบริการคลาวด์แพลตฟอร์มซึ่งในบางครั้งย่อมมีการปรับปรุงแก้ไขซึ่งหากเซิร์ฟเวอร์ทำการปิดบริการชั่วคราวจะส่งผลให้การส่งค่าขึ้นไปบน Ubidots จะหยุดชะงักไปชั่วคราว

5.3 ข้อเสนอแนะ

IoT6 ยังเป็นเทคโนโลยีที่ใหม่สามารถนำไปพัฒนาต่อไปได้เหมาะแก่การศึกษาเพื่อทดลองทำการวิจัยอย่างยิ่ง หรือนำไปใช้แก้ไขปัญหาคือ ปัญหาในชีวิตประจำวันที่ใช้งานได้จริง เรื่องของ IPv6 และอุปกรณ์ IoT ยังเป็นเรื่องใหม่ที่ยังสามารถเรียนรู้ได้อีกมาก และสามารถนำความรู้เหล่านั้นมาประยุกต์ใช้ในชีวิตประจำวันให้เกิดประโยชน์ได้อีกมากมาย

5.4 แผนการทำงานในอนาคต

5.4.1 พัฒนาคloudแพลตฟอร์มของตัวเอง

ในการใช้งาน Ubidots จะมีบางข้อจำกัดในการใช้งานบางคุณสมบัติหรือพบปัญหาจาก Ubidots ซึ่งไม่สามารถแก้ไขปัญหได้ในทันที การสร้างcloudแพลตฟอร์มของตัวเองจึงเป็นอีกหนึ่งในแผนการทำงานในอนาคต

5.4.2 เพิ่มเซ็นเซอร์

นำเซ็นเซอร์ เช่น เซ็นเซอร์แสง เซ็นเซอร์เสียง หรือเซ็นเซอร์อื่นๆ เข้ามาในระบบเพื่อเพิ่มคุณสมบัติการทำงานให้มากขึ้น

เอกสารอ้างอิง

- [1] ทวีปราชญ์ ปิยะชัย, “Application of Embedded Linux Base: เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion Sensor),” Application of Embedded Linux Base, 2553.
- [2] “Boto 3 Documentation — Boto 3 Docs 1.4.4 documentation.” [Online]. Available: <http://boto3.readthedocs.io/en/latest/index.html>. [Accessed: 12-Jan-2017].
- [3] “Configuring the AWS Command Line Interface - AWS Command Line Interface.” [Online]. Available: <http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html>. [Accessed: 12-Jan-2017].
- [4] “EdgeMAX - วิธีการสร้าง IPv6 Tunnel ด้วย Hurricane Electric TunnelBroker in Edgemax และตัวอย่างการตั้งค่าเบื้องต้น - Page 1.” [Online]. Available: <http://www.ubiquiticlub.com/edgemax/21/edgemax-ipv6-tunnel-hurricane-electric-tunnelbroker/206/>. [Accessed: 16-Dec-2016].
- [5] “Getting Started with Amazon Simple Storage Service - Amazon Simple Storage Service.” [Online]. Available: <http://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS3.html>. [Accessed: 12-Jan-2017].
- [6] “Getting started with picamera | Raspberry Pi Learning Resources.” [Online]. Available: <https://www.raspberrypi.org/learning/getting-started-with-picamera/>. [Accessed: 20-Jan-2017].
- [7] “How to give your Raspberry Pi a Static IP Address - UPDATE,” Cases for your Raspberry Pi. [Online]. Available: <https://www.modmypi.com/blog/how-to-give-your-raspberry-pi-a-static-ip-address-update>. [Accessed: 18-Dec-2016].
- [8] “Hurricane Electric Free IPv6 Tunnel Broker.” [Online]. Available: <https://tunnelbroker.net/>. [Accessed: 16-Dec-2016].
- [9] “IoT platform | Internet of Things | Ubidots.” [Online]. Available: <https://ubidots.com/>. [Accessed: 18-Dec-2016].
- [10] “IPv4,” Wikipedia. 16-Dec-2016.
- [11] “IPv6,” Wikipedia. 16-Dec-2016.
- [12] “networking - How to route traffic from a specific port on tun0 to eth0 (and back) - Raspberry Pi Stack Exchange.” [Online]. Available: <http://raspberrypi.stackexchange.com/questions/57380/how-to-route-traffic-from-a-specific-port-on-tun0-to-eth0-and-back>. [Accessed: 20-Dec-2016].

[13] “PIR Motion Sensor Overview.” [Online]. Available: <http://irish7.exteen.com/>. [Accessed: 12-Jan-2017].

[14] “RPi-Cam-Web-Interface - eLinux.org.” [Online]. Available: <http://elinux.org/RPi-Cam-Web-Interface>. [Accessed: 21-Jan-2017].

[15] “Worksheet - Infrared Bird Box | Raspberry Pi Learning Resources.” [Online]. Available: <https://www.raspberrypi.org/learning/infrared-bird-box/worksheet/>. [Accessed: 21-Jan-2017].

[16] “Zolertia/Resources,” GitHub. [Online]. Available: <https://github.com/Zolertia/Resources>. [Accessed: 18-Dec-2016].

[17] “-IPv4 และ IPV6 แตกต่างกันอย่างไรว - sawitree I’ngong.” [Online]. Available: <https://sites.google.com/site/sawitree1008/bthkhwam/-ipv4-laea-ipv6-taek-tang-kan-xyangri>. [Accessed: 24-Nov-2016].



TNI



ภาคผนวก

M-RPL: A Design Algorithm for WSNs with Mixed Traffic

Annop Monsakul

Abstract—A design of wireless sensor network to support traffic engineering for both unicast and multicast traffic is a very difficult problem. This paper proposes a design algorithm called M-RPL. It is IPv6 based routing protocol for low power, lossy Networks (LLNs) that concern routing of both types of traffic. However, since multicast traffic model could be employed in many situations and could be managed by various kinds of multicast routing protocols. The efficiency of M-RPL is evaluated for various traffic demands and networks of 100 node and the total unicast traffic in/out per node of 64, 128, 256, 512 and 1024 kbps, compared with RPL. The experimental results show that, in almost all cases, M-RPL give better performance in term of installation cost.

Index Terms—Traffic engineering, wireless sensor network, IPv6 routing protocol, spanning tree.

I. INTRODUCTION

Nowadays, a wireless sensor network (WSNs) are very popular technology, it is a wireless network consisting of spatially distributed autonomous devices using sensors to environmental conditions. Commonly monitored parameters are temperature, humidity, pressure, wind speed, sound intensity, power-line voltage, vital body functions, etc. WSN [1] appears as an essential platform for prominent concept of Internet of Things (IoT).

IPv6 Routing Protocol for Low power and Lossy Networks (RPL) is the standard IPv6 based routing protocol for low power, lossy Networks (LLNs) consist largely of constrained nodes with limited power, memory, and processing resources as proposed by IETF such as WSNs. It is a routing protocol [2] adapted for information routing with low power, low storage and processing sensor devices. RPL are supports traffic flows include Multipoint-to-Point (MP2P), Point-to-Multipoint (P2MP), and Point-to-Point (P2P) traffic. Thus, IP version 6 (IPv6) is an answer to solve this problem or IPv6 called Internet Protocol Next Generation (IPng), as IPv6 has 128-bit address space pushing the theoretical limit unique IPv6 nodes to roughly $3.4E1038$ unique addresses, which is more enough for new devices in the future. The first time Internet and Internet Protocol (IP) were designed to provide a best-effort, fair delivery service. Under a best-efforts scheme, Internet treats all packets equally, no guarantees, no special resources

allocation; and if it has congestion, packets are dropped to relieve the congestion.

RPL routing algorithm that constructs and maintains Destination Oriented Directed Acyclic Graph (DODAGs) [3] through DODAG Information Object (DIO) messages to transmit data from sensors to root over a single path. This information is exchanged as a new type of ICMPv6 message called the RPL Control Message. RPL is a gradient based routing protocol that builds the graph known as Directed Acyclic Graph (DAG) by using a set of metrics/constraints and an objective function.

RPL provides support for a large number of technologies and features that matches all service requirements reviewed in the introduction. One of the key characteristics of RPL is that the protocol is highly flexible and dynamic; it has been designed to operate in harsh environments with low-speed links potentially experiencing high error rates, while generating very low control plane traffic [4]. Routing metrics are used by routing protocols to compute shortest paths. Interior Gateway Protocols (IGPs) such as IS-IS and OSPF use static link metrics.

Presently, several important emerging multicast application such as distributed database systems, radio, television, video conferencing system, distance learning system, are becoming more and more popular. As a result, the portion of multicast traffic on the IP network in almost all organization is increasing rapidly. Therefore, IP network design process should also effectively route multicast traffic in addition to the traditional unicast traffic. A design that concern both unicast and multicast routing is a very difficult problem.

This study proposes a design routing algorithm for WSNs with mixed unicast and multicast traffic called M-RPL. The algorithm is obtained by modifying spanning tree building portion of RPL. M-RPL to construct core spanning tree instead of Prim-Dijkstra algorithm. The performances of networks design by M-RPL are evaluated in term of installation cost and compared with the networks design by original RPL for various traffic demands and networks with different number of nodes. The experimental results show that, in almost all cases, M-RPL gives better performance in term of installation cost.

The reminder of this paper is organized as follows. In Section II survey recent research works that deal with mobility support for RPL. Section III presents a detailed description of M-RPL. The Scilab/NARVAL [5] simulation results related to the comparison of M-RPL against the standard specification of RPL in WSNs are presented in Section IV. Section V concludes the paper and discusses future works.

Manuscript received March 20, 2016; revised May 19, 2016.

Annop Monsakul is with the Faculty of Information of Technology, Thai-Nichi Institute of Technology, 1771/1, Pattanakarn Rd, Suan Luang, Bangkok, 10250, Thailand (e-mail: annop@tni.ac.th).

II. RELATED WORK

There has been some research addressing the problem of extending RPL to support mobility.

F. Melakessou *et al.* in [6] propose to study the effectiveness of RPL compared to a shortest path algorithm such like the Dijkstra's algorithm. The authors analyzed peer-to-peer communications inside random wireless sensor network topologies. The authors have built a particular simulation environment named Network Analysis and Routing eVALuation (NARVAL). The authors finally performed all paths between each couple of two distinct sensor nodes and compared them to the corresponding shortest paths obtained by the Dijkstra's algorithm. This approach permits to retrieve some statistics on the path extension between RPL and the Dijkstra's algorithm. The authors also analyzed the impact of the sink position and the network size on this path extension.

In [7], the authors studied a Heuristic Load Distribution algorithm (HeLD) based on a braided multipath extension of the standard Routing Protocol for Low power and lossy networks (RPL), which tries to achieve a well-balanced traffic load and simultaneously maximizes the total packet throughput during the network lifetime.

In [8], the authors propose a new routing algorithm called Natural Gradient Routing (NGR), which uses sensed values in a gradient field as a rank, and a single forwarding rule to approach the Sink. With extensive simulation work they compare our proposal against two reference routing algorithms: Flooding, which represents the absence of predefined routes and AODV, a deterministic reactive shortest-path approach.

In [9] presented address this gap and propose Co-RPL as an extension to RPL based on the Corona mechanism to support mobility. They study the impact of node speed, packet transmission rate and number of Directed Acyclic Graphs (DAG) roots on network performance. The simulation results show that Co-RPL decreases packet loss ratio by 45%, average energy consumption by 50% and end-to-end delay by 2.5 seconds, in comparison with the standard RPL.

More recently, the paper in [10] presented specifies the Multicast Protocol for Low power and Lossy Networks (MPL), which provides IPv6 multicast forwarding in constrained networks. MPL avoids the need to construct or maintain any multicast routing topology, disseminating multicast messages to all MPL Forwarders in an MPL

Domain. By using the Trickle algorithm, MPL requires only small, constant state for each MPL device that initiates disseminations. The Trickle algorithm also allows MPL to be density-aware, allowing the communication rate to scale logarithmically with density.

III. OVERVIEW OF M-RPL

In this section, the author describe the M-RPL mechanism for mobile low power and lossy WSNs.

M-RPL starts with node clustering and select the backbone nodes in the same way as RPL. Next, since all multicast transmitters share the same tree to distribute the data, instead of building Prim-Dijkstra tree, the one or more root nodes shortest-path tree is build based on Trickle Multicast (TM) [11], [12] algorithm. The algorithm forms a shortest-path tree with a high rate set of receivers and then successively adds lower rate sets of receivers to the tree and, finally, adds nodes without receiver (zero rate receivers) to the tree. To guide the multicast traffic to flow only through the core shortest-path tree the multicast link weight should be set appropriately. One simple solution is to set the multicast link weight of all links in the shortest-path tree to 1 and that of the others to very large number.

After the author obtained a shortest-path tree, direct links are added and their unicast link weights are determined based on the unicast traffic in exactly the same way as in RPL. The capacity a link on the tree is determined by the sum of its own unicast traffic, the overflow unicast traffic from other routes and the multicast traffic that flow through it.

IV. PERFORMANCE EVALUATION

In this section, the author detail the performance evaluation of M-RPL using NARVAL toolbox, a widely-used and reliable sensor network simulator/emulator under Scilab simulation. In order to evaluate the efficiency of network design calculated by M-RPL algorithm, the author analyze the performances in term of installing cost.

A. Design Example

A WSNs composes of 100 nodes shown in Fig. 1. The unicast and multicast traffic random generate by Scilab simulation. The installation cost of link with 10 Kbps capacity.

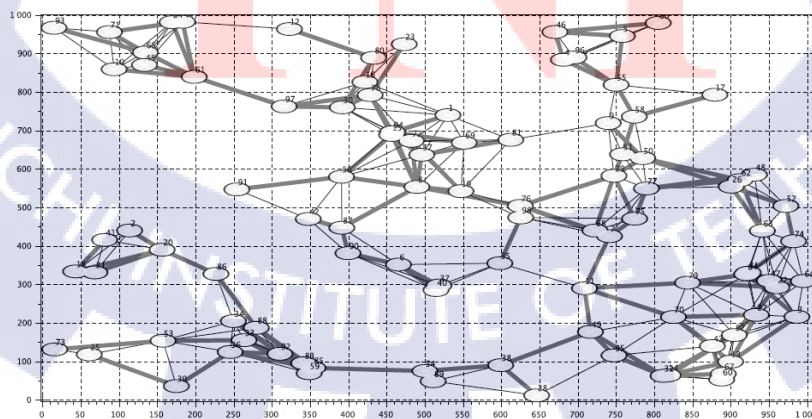


Fig. 1. Final forwarding topology built by M-RPL.

A. Essential Steps of M-RPL

- /*(1) Generate Topology
- /*(2) Update of Weight
- /*(3) Compute Root Node
- /*(4) Build M-RPL Tree
- /*(5) Install Traffic
- /*(6) Calculate Link Capacity
- /*(5) Graph Visualization

B. Pseudo Code

$$[P,D,R,DAG,DIO] = NL_R_RPL(G,S,N) \quad (1)$$

where arguments are G : Graph, S : Source node (root), N : Storage size, P : Predecessor vector, D : Distance vector, R : Rank vector, DAG : DAG vector and DIO : DIO vector. Shown in Fig. 2.

C. Network Design Results

Using Prim-Dijkstra algorithm as in M-RPL. Traffic load on each direction may compose of unicast and multicast traffic. Shown in Fig. 1.

```

n=100; //network size
l=1000; //network squared area side
d=100; //Locality radius
[g]=NL_T_LocalityConnex(n,L,dmax); //generation of a topology in
respect with the Locality method
i=NL_F_RandIntIn(length(g.node_x)); //selection of the source
node
dw=2; //display parameter
ind=1; //window index
g.node_diam(i)=50; //node diameter
g.node_border(i)=10; //node border
g.node_color(i)=5; //node color
[jf]=NL_G_ShowGraphN(g,ind); //graph visualization
[dist,pred]=NL_R_Dijkstra(g,i); //application of NL_R_Dijkstra
ETX=5;
[v]=NL_F_RandVector0nminus1(length(g.head),ETX) ;//update of
weight
v=v+1;
g.edge_weight=g.edge_length;
g.edge_length=v;
xc=l/2; //area center
yc=l/2
[s]=NL_G_NodeClose2XY(g,xc,yc); //root node
c=5; //5 possible routes
[pred,dist,ra,DAG,DIO]=NL_R_RPL(g,s,c); //application of
NL_R_RPL
[go]=NL_R_RPLPlot(g,pred); //highlight RPL tree
ind=1; //window index
f=NL_G_ShowGraphN(go,ind) ;//graph visualization

```

Fig. 2. M-RPL algorithm “Pseudo code”.

D. Installation Cost

Tables I-V present the installation cost for 100 nodes networks obtained by M-RPL and the total unicast traffic in/out per node of 64, 128, 256, 512 and 1024 kbps. What the author have modified is that all multicast traffics are forced to flow through the spanning tree of the RPL network. And thus the link capacities of the spanning tree are determined by the summation of the unicast and multicast traffic flow through them. The design results obtained by M-RPL. In case, it presented in the tables by order to compare with of network.

TABLE I: INSTALLATION COST OF 100 NODE NETWORK WITH TOTALUNICAST TRAFFIC IN/OUT OF 64 KBPS

Multicast Traffic	RPL	M-RPL
100 node $\times$ 128 Kbps	13430	11481

TABLE II: INSTALLATION COST OF 100 NODE NETWORK WITH TOTALUNICAST TRAFFIC IN/OUT OF 128 KBPS

Multicast Traffic	RPL	M-RPL
100 node $\times$ 128 Kbps	14726	12584

TABLE III: INSTALLATION COST OF 100 NODE NETWORK WITH TOTALUNICAST TRAFFIC IN/OUT OF 256 KBPS

Multicast Traffic	RPL	M-RPL
100 node $\times$ 128 Kbps	15674	13962

TABLE IV: INSTALLATION COST OF 100 NODE NETWORK WITH TOTALUNICAST TRAFFIC IN/OUT OF 512 KBPS

Multicast Traffic	RPL	M-RPL
100 node $\times$ 128 Kbps	18385	16638

TABLE V: INSTALLATION COST OF 100 NODE NETWORK WITH TOTALUNICAST TRAFFIC IN/OUT OF 1024 KBPS

Multicast Traffic	RPL	M-RPL
100 node $\times$ 128 Kbps	20068	18753

V. CONCLUSION

This study proposes design routing algorithm for WSNs called M-RPL that support both unicast and multicast traffic simultaneously. However, since multicast traffic model could be employed in many situations and could be managed by various kinds of multicast routing protocols. M-RPL is a modified version of RPL that uses Trickle Multicast (TM) algorithm, rather than Prim-Dijkstra Algorithm, to construct spanning tree. An example of 100 backbone nodes network design is given.

The efficiency of M-RPL is evaluated in term of network installation cost. The installation cost of 100 nodes networks and the total unicast traffic in/out per node of 64, 128, 256, 512 and 1024 kbps. M-RPL are calculated and compared with that of RPL with various design parameters and various conditions of mixed traffics. It is shown that, all most all cases, M-RPL networks give lowest installation cost.

REFERENCES

- [1] Q. Le, M. T. Ngo-Quynh, and T. Magedanz, "RPL-based multipath routing protocols for internet of things on wireless sensor networks," in *Proc. 2014 International Conference on Advanced Technologies for Communications (ATC)*, 2014, pp. 424–429.
- [2] I. E. Korbi, M. B. Brahim, C. Adjih, and L. A. Saidane, "Mobility enhanced RPL for wireless sensor networks," in *Proc. 2012 Third International Conference on the Network of the Future (NOF)*, 2012, pp. 1–8.
- [3] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. Alexander, "RFC 6550: RPL: IPv6 routing protocol for low-power and lossy networks," *Internet Engineering Task Force (IETF) Request For Comments*, March 2012.
- [4] Cisco White Papers. (January 2014). A Standardized and Flexible IPv6 Architecture for Field Area Networks. [Online]. Available: <http://www.cisco.com/c/en/us/products/collateral/routers/1000-series-connected-grid-routers/white-paper-c11-730860.pdf>
- [5] Scilab homepage. [Online]. Available: <http://www.scilab.org/>
- [6] F. Melakessou and T. Engel, "Path extension analysis of peer-to-peer communications in small 6LoWPAN/RPL sensor networks," in *Proc. 2013 IEEE 21st International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*, 2013, pp. 303–307.
- [7] M. N. Moghadam and H. Taheri, "High throughput load balanced multipath routing in homogeneous wireless sensor networks," in *Proc. the 22nd Iranian Conference on Electrical Engineering (ICEE 2014)*, 2014, pp. 1516–1521.
- [8] C. Buchhorsts and D. Dujovne, "Natural gradient routing: Sink convergence using data as guide," in *Proc. 2013 IEEE Latin-America Conference on Communications (LATINCOM)*, 2013, pp. 1–6.
- [9] O. Gaddour, A. Koubaa, R. Rangarajan, O. Cheikhrouhou, E. Tovar, and M. Abid, "Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism," in *Proc. 2014 9th IEEE International Symposium on Industrial Embedded Systems (SIES)*, 2014, pp. 200–209.
- [10] J. Hui and P. Thubert, "Multicast forwarding using trickle," *Internet Draft*, Apr. 2011.
- [11] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A selfregulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proc. First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, 2004, pp. 15–28.
- [12] P. Levis, T. H. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," RFC 6206, Mar. 2011.



Annop Monsakul received the bachelor degree of industrial technology in power electronics technology from King Mongkut's Institute of Technology North Bangkok, Thailand, in 2000; the graduate diploma of engineering in nuclear technology from Chulalongkorn University, Bangkok, Thailand in 2001; the M.Sc. degree in telecommunications and computer networks from Rangsit University, Pathumthani, Thailand, in 2005; and the Ph.D. degree in information technology, Rangsit University, Pathumthani, Thailand, in 2014. He is currently working as a lecturer in the Faculty of Information of Technology, Thai-Nichi Institute of Technology, his research interests are in the areas of data network topology design algorithm, traffic engineering and the optimization of engineering problems.

ระบบความปลอดภัยบ้านอัจฉริยะด้วยอินเทอร์เน็ตสำหรับทุกสิ่ง ก้าวสู่อินเทอร์เน็ตยุคหน้าที่ยอมรับอินเทอร์เน็ตโพรโตคอลรุ่นที่หก

Smart Home Security Systems with the Internet of Things to the Future Internet through IPv6 support

ศุภโชค เกียรติทิติกุล¹ ชรัตน์ ขำแสง² และ อรรณพ หมั่นสกุล³

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีไทย-ญี่ปุ่น

1771/1 ซ.พัฒนาการ 37 ถนนพัฒนาการ แขวงสวนหลวง เขตสวนหลวง กรุงเทพฯ 10250

ki.suppachok_st@tni.ac.th¹ kh.charat_st@tni.ac.th² and annop@tni.ac.th³

บทคัดย่อ

งานวิจัยนี้ได้นำเทคโนโลยี IoT6 (Internet of Things with IPv6) เพื่อออกแบบและประยุกต์การใช้งานภายในที่อยู่อาศัยที่สามารถรักษาความปลอดภัยและแจ้งเตือนอัคคีภัยได้ โดยได้นำอุปกรณ์ที่ทันสมัยในปัจจุบัน ได้แก่ อุปกรณ์เซ็นเซอร์ไร้สายรุ่น Zolertia Z1 อุปกรณ์ Raspberry Pi 3 อุปกรณ์กล้อง Pi NoIR อุปกรณ์เซ็นเซอร์เปลวไฟ อุปกรณ์เซ็นเซอร์แก๊ส และอุปกรณ์เซ็นเซอร์ตรวจจับความเคลื่อนไหว โดยอุปกรณ์เซ็นเซอร์ไร้สายรุ่น Zolertia Z1 จะมีการอ้างอิงตำแหน่งด้วย IPv6 ซึ่งถือว่าเป็นโพรโตคอลรุ่นใหม่ที่มีข้อดีหลายประการและสามารถรับส่งข้อมูลของอุปกรณ์เซ็นเซอร์ต่างๆ ผ่านอุปกรณ์ Raspberry Pi 3 เพื่อส่งต่อไปยังคลาวด์แพลตฟอร์ม (ที่มีชื่อว่า Ubidots) โดยเมื่อเกิดเหตุฉุกเฉิน ไฟไหม้ แก๊สรั่ว หรือตรวจพบผู้บุกรุก ระบบจะทำการถ่ายภาพและอัปโหลดภาพขึ้นไปยัง Amazon Simple Storage Services เพื่อทำการแจ้งเตือนไปยังผู้ใช้งานผ่านอีเมลหรือข้อความสั้นได้ทันที

คำสำคัญ: ระบบความปลอดภัยบ้านอัจฉริยะ, เซ็นเซอร์ไร้สาย, อินเทอร์เน็ตสำหรับทุกสิ่ง, อินเทอร์เน็ตโพรโตคอลรุ่นที่หก, คลาวด์แพลตฟอร์ม

ABSTRACT

This research using IoT6 Technology for design and implement in the house. With the innovation technology, it can improve security and inform the user when an accident happened with smart devices such as Wireless Sensor Zolertia Z1, Raspberry Pi 3, Pi NoIR camera, fire sensor, gas sensor and motion sensor. The Zolertia z1 is located by IPv6 that have many advantages and transmit data from various sensors through Raspberry Pi 3 and then forwarding data to a Ubidots cloud platform. Therefore, this paper has tested the system in the event of an emergency happened such as fire, gas leak or detect the thief. The system will take a picture and upload to Amazon Simple Storage Services for immediately inform the user by email or short message.

Keywords: Smart Home Security Systems, Wireless Sensor Network, Internet of Things, Internet Protocol version 6, Cloud Platform

1) บทนำ

ในปัจจุบันเทคโนโลยี IoT6 เป็นอีกเทคโนโลยีที่มีความทันสมัยและน่าสนใจ เพราะในอุปกรณ์ที่มีขนาดเล็กแต่สามารถเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตบนไอพีรุ่นที่หกและส่งข้อมูลที่ได้รับการตรวจวัดจากเซ็นเซอร์ต่างๆ มากมายหลายชนิด ที่ทำการอัปโหลดขึ้นไปบนอินเทอร์เน็ตได้หลากหลายแพลตฟอร์ม โดยสามารถเข้าถึงข้อมูลที่ได้ตรวจวัดได้ทุกที่ทุกเวลา

งานวิจัยนี้ได้นำอุปกรณ์ IoT6 มาติดตั้งภายในบ้านและนำมาช่วยแจ้งเตือนผู้ใช้งานเมื่อเกิด 2 เหตุการณ์หลักๆ ดังนี้ เหตุการณ์แรกคือกรณีที่เกิดเหตุการณ์แก๊สรั่ว ซึ่งเมื่อ Z1 ตรวจพบแก๊สรั่วออกมา จะทำการแจ้งเตือนไปยังผู้ใช้งานผ่านอีเมลและข้อความสั้น เหตุการณ์ที่สองคือ เมื่อ Z1 ติดตั้งเซ็นเซอร์การเคลื่อนไหว สามารถแจ้งเตือนหากเกิดผู้บุกรุกภายในบ้านในเวลาที่ไม่อยู่บ้าน ซึ่งจะทำการถ่ายภาพและอัปโหลดภาพขึ้นไปยังคลาวด์เพื่อให้ผู้ใช้งานสามารถตรวจเช็คได้ในทันที

2) เครื่องมือและเทคโนโลยี

2.1) Zolertia Z1

Zolertia Z1^[1] มีเป้าหมายเพื่อพัฒนาแพลตฟอร์มสำหรับเครือข่ายเซ็นเซอร์ไร้สาย (WSN, Wireless Sensor Network) สื่อสารบนโพรโตคอล 6LoWPAN ซึ่งทำให้อุปกรณ์ขนาดเล็กที่ประหยัดพลังงานในการทำงานและสามารถสื่อสารด้วย IPv6 ที่ทำงานแบบ Hop-by-hop มีเซ็นเซอร์อุณหภูมิและเซ็นเซอร์วัดความเอียงอยู่ในตัวอุปกรณ์และสามารถเชื่อมต่อเซ็นเซอร์ภายนอกอื่น ๆ เพื่อตรวจจับค่าต่าง ๆ จากสภาพแวดล้อมจริงได้

2.2) เซ็นเซอร์ชนิดต่าง ๆ

ในงานวิจัยนี้จะนำเซ็นเซอร์เปลวไฟ เซ็นเซอร์แก๊ส เซ็นเซอร์อุณหภูมิและความชื้น และเซ็นเซอร์ตรวจจับความเคลื่อนไหวมาใช้งาน

2.2.1) เซ็นเซอร์เปลวไฟ

เซ็นเซอร์ตรวจจับเปลวไฟโดยใช้การทำงานของแสงอินฟราเรด สามารถแสดงผลการทำงานได้ทั้งในรูปแบบสัญญาณดิจิตอล



(ให้ค่า 0 หรือ 1 เพื่อแสดงค่าที่ตรวจจับได้) และสัญญาณแอนะล็อก (ให้ค่าสัญญาณเพื่อใช้ประมวลผลต่อ)

2.2.2) เซ็นเซอร์แก๊ส MQ-2

MQ-2 เป็นเซ็นเซอร์ตรวจสอบปริมาณ ก๊าซไวไฟและควัน เช่น แอลกอฮอล์ มีเทน แอลกอฮอล์ ไฮโดรเจน และควันในอากาศ ซึ่งเมื่อเราเริ่มจ่ายพลังงานให้ MQ-2 จะทำให้เกิดความร้อนขึ้นที่ขดลวด เมื่อก๊าซไวไฟต่างๆ เข้ามาทำปฏิกิริยาจะทำให้ค่าความต้านทานลดลง ซึ่งค่าจะวัดได้จากความต้านแปรผกผันกับปริมาณของแก๊สที่เกิดขึ้น

2.2.3) เซ็นเซอร์อุณหภูมิและความชื้น

เป็นอุปกรณ์ที่ใช้สำหรับวัดอุณหภูมิ โดยจะเปลี่ยนระดับอุณหภูมิเป็นสัญญาณไฟฟ้า และค่าความชื้น โดยความชื้นนี้มาจากความชื้นสัมพัทธ์ ซึ่งหมายถึง อัตราส่วนของปริมาณไอน้ำที่มีอยู่จริงในอากาศต่อปริมาณไอน้ำที่จะทำให้อากาศอิ่มตัว ณ อุณหภูมิเดียวกัน ซึ่งค่าความชื้นสัมพัทธ์จะแสดงในรูปของร้อยละ (%) มีหน่วยเป็น %RH

2.2.4) เซ็นเซอร์ตรวจจับความเคลื่อนไหว

เป็นอุปกรณ์ที่ใช้ตรวจจับความเคลื่อนไหวด้วยการวัดความร้อนจากการเปลี่ยนแปลงรังสีอินฟราเรดที่ปล่อยออกมาจากวัตถุ นั้นๆ แล้วนำมาแปลงเป็นค่าสัญญาณไฟฟ้าเพื่อใช้งาน

2.3) Ubidots

เป็นคลาวด์แพลตฟอร์มชนิดหนึ่ง สำหรับนักพัฒนาที่จะช่วยในการสร้างแอปพลิเคชัน ซึ่งเก็บข้อมูลจากการตรวจวัดของเซ็นเซอร์มาใช้ให้เกิดประโยชน์ โดยมีคุณสมบัติที่หลากหลาย สามารถแสดงค่าบนหน้าเว็บ แจ้งเตือนผ่านข้อความสั้นและอีเมลได้

2.4) Amazon Simple Storage Services

เป็นคลาวด์แพลตฟอร์มชนิดหนึ่งที่ใช้สำหรับเก็บรูปถ่ายไว้บนอินเทอร์เน็ต โดย Raspberry Pi 3 จะอัปโหลดรูปถ่ายขึ้นไปทันทีที่มีการถ่ายรูปจากกล้อง Pi NoIR

2.5) Boto3

เป็นชุดพัฒนาซอฟต์แวร์ของ Amazon Web Services ที่ช่วยในการใช้งานบริการต่างๆ ของ Amazon ทำให้สามารถใช้งาน Amazon Simple Storage Services ได้ง่ายขึ้น สามารถใช้ภาษาไพธอนในการแก้ไขปรับปรุงซอฟต์แวร์ให้สามารถทำงานได้

2.6) Raspberry Pi 3

อุปกรณ์ขนาดเล็กสามารถติดตั้งระบบปฏิบัติการและใช้งานได้เหมือนคอมพิวเตอร์ทั่วไป สามารถใช้ระบบปฏิบัติการ Raspbian ซึ่งเป็นระบบปฏิบัติการที่มาจากผู้ผลิตเอง

2.7) กล้อง Pi NoIR

กล้องสำหรับเชื่อมต่อกับ Raspberry Pi 3 มีความละเอียด 8 ล้านพิกเซล แบบไม่ตัดตัวกรองแสงอินฟราเรดออก ทำให้สามารถถ่ายรูปได้ในตอนกลางคืน โดยต้องใช้ร่วมกับแสงอินฟราเรด สามารถเขียนโปรแกรมบน Raspberry Pi 3 เพื่อส่งงานได้

2.8) ระบบปฏิบัติการ Contiki

ระบบปฏิบัติการแบบโอเพนซอร์สสำหรับ Internet of Things โดยรองรับการใช้พลังงานต่ำเพื่อสื่อสารบนอินเทอร์เน็ตยุคหน้าที่สามารถรองรับมาตรฐานไอพีรุ่นที่สี่และไอพีรุ่นที่หก พร้อมทั้งมาตรฐานเครือข่ายไร้สายพลังงานต่ำ เช่น 6LoWPAN, RPL และ CoAP สามารถนำไปพัฒนาต่อได้ง่าย เนื่องจากระบบถูกพัฒนามาจากภาษาซีและยังมี

โปรแกรมจำลอง Cooja Simulator ที่สามารถทำการจำลองระบบเครือข่ายเซ็นเซอร์ไร้สายได้

2.9) ไอพีรุ่นที่หก

การเชื่อมต่ออินเทอร์เน็ตจำเป็นต้องมีโพรโทคอลเครือข่ายซึ่งจะบอกที่อยู่ในการส่งข้อมูลจากเว็บไซด์ ไอพีรุ่นที่สี่นั้นมีจำนวนที่อยู่อ้างอิงไอพีที่จำกัด ซึ่งจะต้องเปลี่ยนเป็นไอพีรุ่นที่หกอย่างหลีกเลี่ยงไม่ได้ การสำรวจของกูเกิลพบว่าไอพีรุ่นที่หกได้รับการยอมรับเพิ่มขึ้นเป็นเท่าตัวทุกๆ 9 เดือน

2.10) IPv6 Transition Mechanism (6in4)

เป็นกระบวนการเปลี่ยนผ่านไอพี เพื่อแก้ปัญหาการขาดแคลนไอพีรุ่นที่หก โดยผู้วิจัยเลือกใช้วิธีการทำอุโมงค์ 6in4 ซึ่งวิธีการนี้ยังสามารถแก้ปัญหาเราเตอร์ที่ไม่รองรับไอพีรุ่นที่หกได้อีกด้วย วิธีการก็คือการเพิ่มเราเตอร์ที่รองรับทั้งไอพีรุ่นที่หกและไอพีรุ่นที่สี่ และสร้างอุโมงค์ 6in4 ขึ้นจากเราเตอร์กับปลายทางอุโมงค์แห่งหนึ่งในอินเทอร์เน็ตของไอพีรุ่นที่หก ซึ่งเป็นวิธีที่นิยมและง่ายต่อการทำมากที่สุด

2.11) <http://tunnelbroker.net>

เป็นเว็บไซต์ที่ให้บริการในการทำอุโมงค์ไอพีรุ่นที่หก สำหรับผู้ที่มีแต่ไอพีรุ่นที่สี่ แล้วต้องการจะใช้ไอพีรุ่นที่หก การทำอุโมงค์จะทำให้ไอพีรุ่นที่หกสามารถวิ่งผ่านเครือข่ายที่เป็นไอพีรุ่นที่สี่ได้

โดย Zolertia Z1 เป็นอุปกรณ์ที่ทำการสื่อสารบนไอพีรุ่นที่หก ซึ่งภายในห้องวิจัยเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีไทย-ญี่ปุ่น ยังใช้ไอพีรุ่นที่สี่อยู่ จึงจำเป็นต้องใช้บริการของ <http://tunnelbroker.net> ในการใช้งานไอพีรุ่นที่หก เพื่อรองรับการทำงานของระบบที่ออกแบบด้วยไอพีรุ่นที่หก

3) ขั้นตอนการทำงาน

3.1) การเตรียมการ

เตรียมการอุปกรณ์และตั้งค่าพื้นฐานสำหรับใช้งานอุปกรณ์ Zolertia Z1 ในการส่งค่าไปยัง Ubidots

3.1.1) ติดตั้ง Contiki

ดาวน์โหลดระบบปฏิบัติการ Contiki และติดตั้งโปรแกรม msp430 บน Raspberry Pi 3 เพื่อใช้ในการติดตั้งโปรแกรมบน Zolertia Z1

3.1.2) การทำอุโมงค์ไอพีรุ่นที่หกให้เชื่อมต่อมาที่ Raspberry Pi

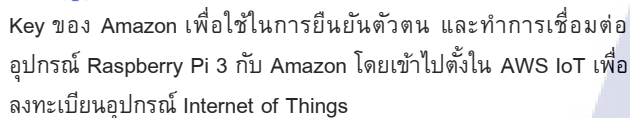
สมัครใช้งานเว็บไซต์ <http://tunnelbroker.net> เพื่อขอใช้งานไอพีรุ่นที่หกผ่านไอพีสาธารณะรุ่นที่สี่ที่ใช้งานอยู่ ซึ่งจำเป็นต้องเปิดพอร์ต ICMP และโพรโทคอล 41 เพื่อให้แพ็กเก็ตไอพีรุ่นที่หกสามารถวิ่งผ่านเครือข่ายที่เป็นไอพีรุ่นที่สี่ได้

3.1.3) สมัครใช้งานและตั้งค่า Ubidots

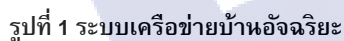
สมัคร Ubidots คลาวด์แพลตฟอร์ม สร้างตัวแปรสำหรับเตรียมรับข้อมูลจากเซ็นเซอร์ สร้างหน้าต่างการแสดงผลของเซ็นเซอร์ที่หน้า Dashboard และทำการตั้งค่า Events สำหรับการแจ้งเตือนข้อความสั้นและอีเมลไปยังผู้ใช้งาน

3.1.4) สมัครใช้งาน Amazon Simple Storage Services

สมัคร Amazon Simple Storage Services สร้างคลังเก็บข้อมูลเพื่อเก็บข้อมูลรูปภาพที่ถ่ายรูปจากกล้อง Pi NoIR สร้าง Access



ทำการตั้งค่าอุปกรณ์และเชื่อมต่อจตุรภาพภายในห้องวิจัย
เทคโนโลยีสารสนเทศ จตุรภาพที่ 1



เริ่มจากการติดตั้ง Ubidots Client บนอุปกรณ์ที่ต้องการ
รับค่าโดยการแก้ไขค่าของ VARIEABLE KEY ให้ตรงกับที่ได้ตั้งค่าไว้
ในเว็บไซต์ Ubidots ในทุกอุปกรณ์ที่จะทำการติดตั้ง

ติดตั้ง Border Router โดยเชื่อมต่อกับ Raspberry Pi 3 ผ่านพอร์ต USB เพื่อเป็นทางผ่านของข้อมูลและจ่ายพลังงานให้กับ Zolertia Z1

ตัว Border Router จะทำหน้าที่ค้นหา Z1 ตัวอื่นๆ และ จัดหาเส้นทางในการเชื่อมต่อ โดย Border Router จะรู้ไอพีรุ่นที่หกของ ตนจากการกำหนด Prefix ให้ ซึ่งในกรณีนี้จะใช้ IPv6 Prefix Routes ที่ ได้มาจากการทำอุโมงค์ไอพีรุ่นที่หกบนเว็บไซต์ <http://tunnelBroker.net> และทำหน้าที่ส่งข้อมูลผ่านสาย USB ไปยัง Raspberry Pi 3 เพื่อทำการ เชื่อมต่อออกไปยังเครือข่ายอินเทอร์เน็ตและส่งข้อมูลขึ้นไปยังเว็บไซต์ Ubidots

บน Raspberry Pi 3 นั้นจะทำหน้าที่เป็นทางผ่านของข้อมูลจาก Zolertia Z1 ที่ติดตั้งโปรแกรม Border Router อย่างแรกนั้นต้องทำการตั้งค่าบน Raspberry Pi 3 โดยบน Raspberry Pi 3 จะมีทั้งหมดสามอินเตอร์เฟซ ดังนี้

1. อินเทอร์เน็ตเฟส eth0 ใช้สำหรับรับไอพี ที่ทางสถาบันฯ ทำการ NAT กับไอพีรุ่นที่สี่แบบสาธารณะ เพื่อใช้ออกสู่อินเทอร์เน็ต
2. อินเทอร์เน็ตเฟสเสมือน tun0 ใช้สำหรับเชื่อมต่อเครือข่ายไอพีรุ่นที่หกบนเครือข่ายบุคคลไร้สายพลังงานต่ำของอุปกรณ์ Zolertia Z1 ที่ทำหน้าที่เป็นขอบ Border Router เพื่อเชื่อมต่อกับอินเทอร์เน็ตเฟส eth0
3. อินเทอร์เน็ตเฟสเสมือน he-ipv6 ใช้สำหรับเชื่อมโยง

เครือข่ายไอพีรุ่นที่หกที่วิ่งผ่านเครือข่ายไอพีรุ่นที่สี่จากเว็บไซต์ <http://tunnelBroker.net> เพื่อให้ Raspberry Pi 3 สามารถออกสู่อินเทอร์เน็ตผ่านระบบเครือข่ายไอพีรุ่นที่หกได้

จากนั้นให้ตั้งค่าตารางไอพีเพื่อให้การส่งแพ็กเก็ตสามารถวิ่งไปถึงปลายทางที่กำหนดได้ โดยการตั้งค่าให้ ยอมรับ ในทุกช่องทาง ทั้งการนำเข้า การส่งต่อ และการนำออกของแพ็กเก็ต ตั้งค่าให้อินเทอร์เฟซ eth0 สามารถส่งต่อแพ็กเก็ตไปที่อินเทอร์เฟซเสมือน tun0 และตั้งค่าให้อินเทอร์เฟซเสมือน tun0 ส่งต่อกลุ่มข้อมูลกลับไปให้อินเทอร์เฟซ eth0 ได้เช่นกัน ตั้งค่าให้อินเทอร์เฟซ eth0 ทำหน้าที่จัดหาเส้นทางให้กับเครือข่ายภายใน และเข้าไปทำการแก้ไขไฟล์ระบบให้สามารถส่งต่อแพ็กเก็ตไอพีที่ทักได้

จากเราเตอร์ไปยัง <http://tunnelbroker.net> นั้นจะเชื่อมต่อไปยังอินเทอร์เน็ตผ่านเครือข่ายไอพีพูนที่สี่ โดยต้องตั้งค่าที่เราเตอร์ให้เปิดการใช้งานสองส่วนคือ

1. โปรโตคอล 41 ไว้เพื่อเป็นการเปิดอนุญาตสำหรับการเพิ่มพื้นที่บนส่วนหัวของไอพีรุ่นที่สี่เพื่อห่อหุ้มแพ็กเกจไอพีรุ่นที่หก
 2. พอร์ต ICMP เพื่อให้งานเซิร์ฟเวอร์สามารถตอบกลับมา
- ยัง Raspberry Pi 3 ได้

ซึ่งเมื่อทำตามทั้ง 2 อย่างนี้จะสามารถใช้บริการของ <http://tunnelbroker.net> ได้ และจะได้รับไอพีรุ่นที่หกมาใช้งาน ซึ่งสามารถเข้าใช้งานได้กับทุกเว็บไซต์ที่รองรับไอพีรุ่นที่หก

3.2.5) จากเว็บไซต์ <http://tunnelbroker.net> ไปยัง Ubidots
ข้อมูลจากเว็บไซต์ <http://tunnelbroker.net> นั้นเมื่อเชื่อมต่อไป Ubidots จะเชื่อมต่อไปด้วยไอฟีร์นที่หูกที่ออกมาจากอุโมงค์ไอฟีร์นที่หูก ซึ่งข้อมูลของแพ็กเกจจาก Border Router จะถูกคลายออกในขั้นตอนนี้และส่งต่อไปด้วยเครือข่ายไอฟีร์นที่หูก

3.2.6) Ubidots ส่งค่ากลับยัง Raspberry Pi 3
เมื่อเกิดกรณีที่ต้องการจะถ่ายภาพ Ubidots จะส่งค่า
กลับมายัง Raspberry Pi 3 ซึ่งทางฝั่ง Raspberry Pi 3 จะต้องรัน
โปรแกรมเพื่อรับค่าที่เปลี่ยนแปลงเพื่อเป็นเงื่อนไขของการถ่ายรูป
3.2.7) Raspberry Pi 3 ส่งให้กล้องถ่ายรูป และอัปโหลด
รูปภาพขึ้นไปยัง Amazon Simple Storage Services

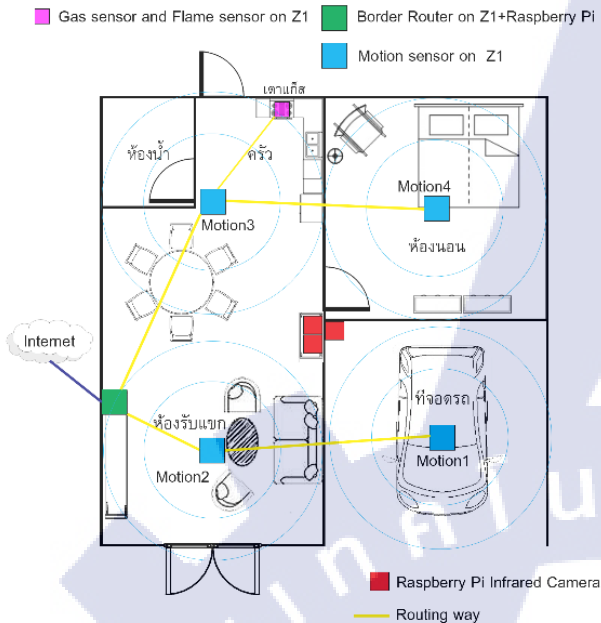
ทำการเชื่อมต่อกล้อง Pi NoIR เข้ากับตัว Raspberry Pi 3 และเข้าไปตั้งค่าที่ระบบปฏิบัติการให้เปิดการใช้งานกล้อง Pi NoIR และเขียนโปรแกรมด้วยภาษาไพธอน ส่งให้รับค่าของตัวแปรที่จะใช้ในการถ่ายรูปบนเว็บไซต์ Ubidots เมื่อมีค่าที่เปลี่ยนไป ตัวโปรแกรมจะส่งให้กล้อง Pi NoIR ถ่ายรูปในทันที ซึ่งเมื่อถ่ายเสร็จแล้ว Raspberry Pi 3 จะทำการอัปโหลดภาพถ่ายขึ้นไปยัง Amazon Simple Storage Services ในทันที หลังจากนั้นจึงส่งไฟล์รูปภาพออกจากเครื่อง เพื่อไม่ให้หน่วยความจำภายในเครื่องเต็ม และผู้ใช้สามารถเข้ามาตรวจสอบรูปภาพได้ตลอดเวลา

3.2.8) เว็บไซต์ Ubidots แจ้งเตือนมายังผู้ใช้งานบนเว็บไซต์ Ubidots ทำการสร้าง Events โดยการกำหนดค่าจากตัวแปร เมื่อถึงค่าที่กำหนด จะทำการแจ้งเตือนในรูปแบบของข้อความสั้นและอีเมลมายังผู้ใช้งานได้



4) ผลการดำเนินงาน

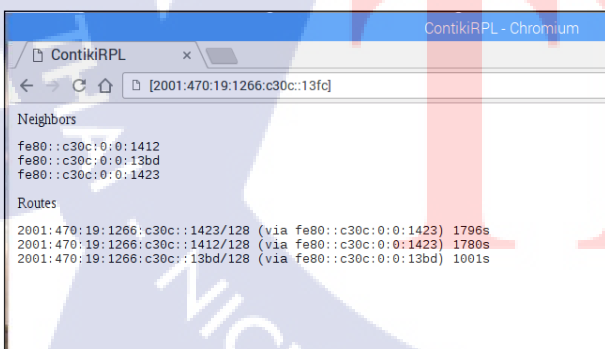
นำอุปกรณ์ตั้งค่าไว้ให้เรียบร้อยและนำไปวางในบ้านจำลองเพื่อเป็นกรณีศึกษา ดังรูปที่ 2



รูปที่ 2 แผนผังการวางอุปกรณ์ภายในบ้านจำลอง

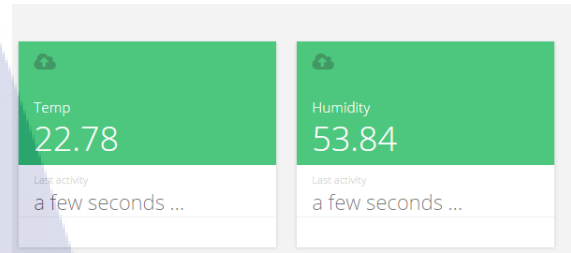
โดยจะทำการติดตั้งเซ็นเซอร์ตรวจจับความเคลื่อนไหวไว้ที่ห้องรับแขก ห้องครัว ห้องนอน และที่โรงจอดรถ เพื่อตรวจจับความเคลื่อนไหวตอนที่ไม่มีคนอยู่บ้าน ทำการติดตั้ง Pi NoIR ให้ถ่ายรูปที่ห้องรับแขกจนถึงประตูหน้าบ้าน ห้องครัวจนถึงประตูหลังบ้าน และที่โรงจอดรถ และติดตั้งเซ็นเซอร์เปลวไฟและแก๊สไว้ที่เตาแก๊สในห้องครัว โดย Zolertia Z1 มีการทำงานแบบ Hop-by-hop ตามเส้นสีเหลือง แล้วเชื่อมต่อไปที่ Border Router เพื่อส่งค่าออกสู่อินเทอร์เน็ต ไปที่เว็บไซต์ Ubidots

4.1) สามารถทำการเชื่อมต่อระบบเครือข่ายไอพีรุ่นที่หกแบบ Hop-by-hop ได้สำเร็จ ดังรูปที่ 3



รูปที่ 3 การเชื่อมต่อแบบ hop-by-hop

4.2) สามารถตรวจสอบค่าของเซ็นเซอร์ได้ตลอดเวลา และสามารถตรวจสอบสถานะของค่าผ่านเว็บไซต์ Ubidots ได้ตลอดเวลา ดังรูปที่ 4

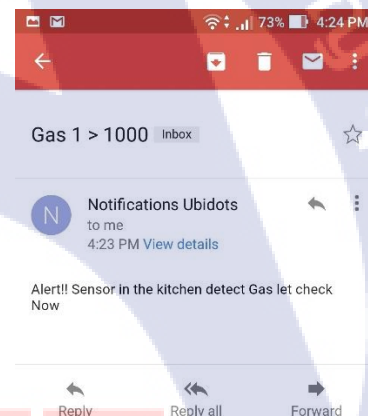


รูปที่ 4 ตรวจสอบสถานะค่าจากเซ็นเซอร์

4.3) สามารถแจ้งเตือนผ่านอีเมลและข้อความมือถือเมื่อเกิดเหตุฉุกเฉิน เช่น ไฟไหม้หรือแก๊สรั่ว สามารถแจ้งเตือนไปยังผู้ใช้งานได้ในทันทีผ่านข้อความสั้นหรืออีเมล ดังรูปที่ 5-6



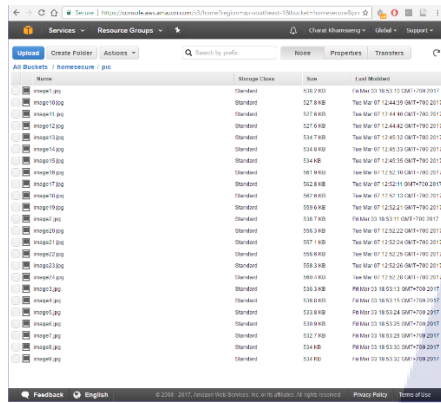
รูปที่ 5 การแจ้งเตือนมายังโทรศัพท์



รูปที่ 6 การแจ้งมายังอีเมล

4.4) สามารถอัปโหลดรูปถ่ายขึ้น Amazon Simple Storage Services

เมื่อถ่ายรูปแล้ว ก็ทำการอัปโหลดรูปภาพขึ้นไปยัง Amazon Simple Storage Services เพื่อทำการเก็บไว้ แล้วจึงลบรูปภาพออกจากอุปกรณ์ Raspberry Pi 3 เพื่อไม่ให้หน่วยความจำในอุปกรณ์เต็ม ซึ่งสามารถอัปโหลดรูปถ่ายได้ และสามารถลบรูปภาพหลังจากอัปโหลดได้เช่นกัน ดังรูปที่ 7



Name	Storage Class	Size	Last Modified
image1.jpg	Standard	538.2 KB	Thu Mar 15 18:51:11 GMT+07:00 2017
image2.jpg	Standard	527.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image3.jpg	Standard	527.8 KB	Tue Mar 14 07:14:44:38 GMT+07:00 2017
image4.jpg	Standard	527.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image5.jpg	Standard	534.7 KB	Tue Mar 14 07:11:45:35 GMT+07:00 2017
image6.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image7.jpg	Standard	534.8 KB	Tue Mar 14 07:14:44:38 GMT+07:00 2017
image8.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image9.jpg	Standard	534.8 KB	Tue Mar 14 07:11:45:35 GMT+07:00 2017
image10.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image11.jpg	Standard	534.8 KB	Tue Mar 14 07:14:44:38 GMT+07:00 2017
image12.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image13.jpg	Standard	534.8 KB	Tue Mar 14 07:11:45:35 GMT+07:00 2017
image14.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image15.jpg	Standard	534.8 KB	Tue Mar 14 07:14:44:38 GMT+07:00 2017
image16.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image17.jpg	Standard	534.8 KB	Tue Mar 14 07:11:45:35 GMT+07:00 2017
image18.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image19.jpg	Standard	534.8 KB	Tue Mar 14 07:14:44:38 GMT+07:00 2017
image20.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image21.jpg	Standard	534.8 KB	Tue Mar 14 07:11:45:35 GMT+07:00 2017
image22.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image23.jpg	Standard	534.8 KB	Tue Mar 14 07:14:44:38 GMT+07:00 2017
image24.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image25.jpg	Standard	534.8 KB	Tue Mar 14 07:11:45:35 GMT+07:00 2017
image26.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image27.jpg	Standard	534.8 KB	Tue Mar 14 07:14:44:38 GMT+07:00 2017
image28.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017
image29.jpg	Standard	534.8 KB	Tue Mar 14 07:11:45:35 GMT+07:00 2017
image30.jpg	Standard	534.8 KB	Tue Mar 14 07:12:44:38 GMT+07:00 2017

รูปที่ 7 Amazon Simple Storage Services

4.5) รูปภาพที่ถูกถ่ายมีความคมชัดและสามารถระบุตัวตน

ได้ว่าเป็นใครที่เข้ามาในห้อง ดังรูปที่ 8-9



รูปที่ 8 หลังจาก Motion1 สามารถตรวจจับความเคลื่อนไหว



รูปที่ 9 หลังจาก Motion2 สามารถตรวจจับความเคลื่อนไหว

5) สรุปการดำเนินงาน

งานวิจัยนี้ใช้การอ้างอิงตำแหน่งอุปกรณ์ต่าง ๆ ด้วย IPv6 โดยอุปกรณ์สามารถทำงานร่วมกับเซ็นเซอร์ได้ดี ได้แก่ สามารถตรวจจับความเคลื่อนไหวได้ สามารถตรวจจับแก๊สที่รั่วและเปลวไฟได้ สามารถอัปโหลดค่าขึ้นเว็บไซต์ Ubidots ได้และแจ้งเตือนมายังสมาร์ตโฟนผ่านข้อความสั้นและอีเมลได้ตลอดเวลา รวมถึงการถ่ายรูปและอัปโหลดรูปภาพไปยัง Amazon Simple Storage Services ก็สามารถทำงานได้ดีโดยไม่มีปัญหาเช่นกัน

ข้อจำกัดการจะนำไปใช้งานจริงจำเป็นต้องมีไอพีรุ่นที่สี่แบบสาธารณะ ซึ่งการใช้งานตามบ้านเรือนทั่วไปยังมีความยุ่งยาก เนื่องจากต้องเปิดพอร์ต ICMP และโปรโตคอล 41 เมื่อมีความต้องการทำอุโมงค์

ไอพีรุ่นที่หก ซึ่งเป็นข้อจำกัดของผู้ให้บริการอินเทอร์เน็ตที่ยังไม่สามารถให้บริการได้ในปัจจุบัน

6) กิตติกรรมประกาศ

งานวิจัยนี้สำเร็จลุล่วงไปได้ด้วยดีด้วยความกรุณาจากผู้ช่วยศาสตราจารย์ ดร.อรรถพร หมั่นสกุล ที่ได้เขียนโครงการเพื่อขอทุนวิจัยสำหรับการซื้ออุปกรณ์ Z1 เพื่อศึกษาค้นคว้าในงานวิจัยนี้ รวมถึงคอยให้คำปรึกษาเสนอแนวคิดต่างๆ ช่วยเหลือแก้ไขข้อบกพร่อง และความช่วยเหลือในทุกๆ ด้าน ซึ่งเป็นประโยชน์ต่องานวิจัยนี้อย่างยิ่ง และขอแสดงความขอบคุณต่อคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีไทย-ญี่ปุ่น ที่เอื้อเฟื้อสถานที่เพื่อใช้ในการวิจัยและทดลองใช้งานอุปกรณ์ในงานวิจัยนี้สำเร็จลุล่วงด้วยดี

7) เอกสารอ้างอิง

- [1] "marcozennaro/IPv6-WSN-book," GitHub. [Online]. Available: <https://github.com/marcozennaro/IPv6-WSN-book>.
- [2] "Zolertia/Resources," GitHub. [Online]. Available: <https://github.com/Zolertia/Resources>.
- [3] "IoT platform | Internet of Things | Ubidots." [Online]. Available: <https://ubidots.com/>.
- [4] "Hurricane Electric Free IPv6 Tunnel Broker." [Online]. Available: <https://tunnelbroker.net/>.
- [5] "IPv6 for IoT | IoT6.eu." [Online]. Available: http://iot6.eu/ipv6_for_iot.
- [6] "AKIF," AKIF. [Online]. Available: <https://zaidmufti.wordpress.com/>.
- [7] "IoT Standards & Protocols Guide | 2017 Comparisons on Network, Wireless Comms, Security, Industrial." [Online]. Available: internet-of-things-protocols/.
- [8] "How to give your Raspberry Pi 3 a Static IP Address - UPDATE," Cases for your Raspberry Pi. [Online]. Available: <https://www.modmypi.com/blog/how-to-give-your-raspberry-pi-a-static-ip-address-update>.
- [9] "networking - How to route traffic from a specific port on tun0 to eth0 (and back) - Raspberry Pi 3 Stack Exchange." [Online]. Available: <http://raspberrypi.stackexchange.com/questions/57380/how-to-route-traffic-from-a-specific-port-on-tun0-to-eth0-and-back>.
- [10] "EdgeMAX - วิธีการสร้าง IPv6 Tunnel ด้วย Hurricane Electric TunnelBroker in Edgemax และตัวอย่างการตั้งค่าเบื้องต้น - Page 1." [Online]. Available: <http://www.ubiquiticlub.com/edgemax/21/edgemax-ipv6-tunnel-hurricane-electric-tunnelbroker/206/>.
- [11] "Getting Started with Amazon Simple Storage Service - Amazon Simple Storage Service." [Online]. Available: <http://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS3.html>.
- [12] "Boto 3 Documentation — Boto 3 Docs 1.4.4 documentation." [Online]. Available: <http://boto3.readthedocs.io/en/latest/index.html>.
- [13] "PIR Motion Sensor Overview." [Online]. Available: <http://irish7.exteen.com/>.
- [14] ปราโมทย์ บิยะชัย, "Application of Embedded Linux Base: เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion Sensor)," Application of Embedded Linux Base, 2553.
- [15] "Getting started with picamera | Raspberry Pi 3 Learning Resources." [Online]. Available: <https://www.raspberrypi.org/learning/getting-started-with-picamera/>.
- [16] "RPI-Cam-Web-Interface - eLinux.org." [Online]. Available: <http://elinux.org/RPI-Cam-Web-Interface>.