

การศึกษาประสิทธิภาพการทำงานของโปรโตคอลสำหรับการค้นหาเส้นทางใน
เครือข่ายที่มีพลังงานต่ำและมีอัตราการสูญหายของข้อมูลสูงในการใช้งาน
ภายในอาคาร

A STUDY OF IPV6 ROUTING PROTOCOLS FOR LOW POWER
AND LOSSY NETWORK (RPL) PERFORMANCE IN IN-DOOR
ENVIRONMENT



T146235



เลขที่ 146235
ลงทะเบียน
วัน เดือน ปี 25 10 2560

.b. 128A000 x
.i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาประสิทธิภาพการทำงานของโปรโตคอลสำหรับการค้นหาเส้นทางใน
เครือข่ายที่มีพลังงานต่ำและมีอัตราการสูญหายของข้อมูลสูงในการใช้งาน
ภายในอาคาร

**A STUDY OF IPV6 ROUTING PROTOCOLS FOR LOW POWER
AND LOSSY NETWORK (RPL) PERFORMANCE IN IN-DOOR
ENVIRONMENT**



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**A STUDY OF IPV6 ROUTING PROTOCOLS FOR LOW POWER
AND LOSSY NETWORK (RPL) PERFORMANCE IN IN-DOOR
ENVIRONMENT**



**KANNAPHAT KHUNAKORNPATTANAKARN
WISANU TONGPRADIT**

**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ **2/2015** เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2016

FACULTY OF INFORMATION TECHNOLOGY

เอกสารนี้เป็นทรัพย์สินทางปัญญาของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งได้รับการคุ้มครอง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญาโท ประจำปีการศึกษา 2558

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การศึกษาประสิทธิภาพการทำงานของโปรโตคอลสำหรับการค้นหา
เส้นทางในเครือข่ายที่มีพลังงานต่ำและมีอัตราการสูญหายของข้อมูลสูง
ในการใช้งานภายในอาคาร

A STUDY OF IPV6 ROUTING PROTOCOLS FOR LOW
POWER AND LOSSY NETWORK (RPL) PERFORMANCE IN
IN-DOOR ENVIRONMENT

ผู้จัดทำ

1. นางสาว กัญญ์ณพัชร คุณากรพัฒนาการ รหัสนักศึกษา 55070004
2. นาย วิษณุ ทองประดิษฐ์ รหัสนักศึกษา 55070111

.....อาจารย์ที่ปรึกษา

(ดร. กัทธ ประดิษฐ์ทัศนีย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการ	การศึกษาประสิทธิภาพการทำงานของ โพรโตคอลสำหรับการค้นหาเส้นทางในเครือข่ายที่มีพลังงานต่ำและมีอัตราการสูญหายของข้อมูลสูงในการใช้งานภายในอาคาร	
นักศึกษา	นางสาว กัญญณ์พัชร์ คุณากรพัฒนาการ	รหัสนักศึกษา 55070004
	นายวิษณุ ทองประดิษฐ์	รหัสนักศึกษา 55070111
ปริญญา	วิทยาศาสตรบัณฑิต	
สาขาวิชา	เทคโนโลยีสารสนเทศ	
ปีการศึกษา	2558	
อาจารย์ที่ปรึกษา	ดร. ลภัส ประดิษฐ์ทัศนีย์	

บทคัดย่อ

IPv6 Routing Protocols for Low Power and Lossy Network (RPL) เป็น โพรโตคอลในการค้นหาเส้นทางที่ออกแบบมาเพื่อใช้งานสำหรับอุปกรณ์ในระบบเครือข่ายที่มีพลังงานต่ำและการรับส่งข้อมูลบนเครือข่ายที่มีโอกาสสูญหายของข้อมูลสูง จากลักษณะคุณสมบัติที่น่าสนใจดังกล่าว โครงการนี้จึงได้นำ RPL มาทำการศึกษาประสิทธิภาพในการทำงานของ RPL ภายใต้สภาพแวดล้อมภายในอาคาร โดยในโครงการนี้จะใช้โพรโตคอล RPL ที่มีมากับระบบปฏิบัติการ Contiki ซึ่งการทดสอบประสิทธิภาพในการทำงานของ RPL นั้นจะทำทั้งในโปรแกรมจำลอง Cooja simulator และการนำไปใช้กับอุปกรณ์จริง เพื่อศึกษาและวิเคราะห์ประสิทธิภาพของ RPL ในการใช้งานภายในอาคาร ทั้งในกรณีที่มีการรบกวนจากระบบภายนอกต่ำ และมีการรบกวนจากระบบภายนอกสูง ทั้งนี้เพื่อนำผลการทดลองที่ได้มาวิเคราะห์เพื่อเสนอแนะแนวทางในการปรับปรุงค่าพารามิเตอร์ในการทำงานของ RPL ให้มีประสิทธิภาพดียิ่งขึ้นและหวังว่าจะเป็นส่วนช่วยสำหรับผู้สนใจเพื่อนำไปต่อยอดและเกิดการพัฒนางานวิจัยและนำไปประยุกต์ใช้งานต่อไป

Project Title A study of IPv6 routing protocols for low power and lossy network (RPL) performance in in-door environment

Student Ms. Kannaphat Khunakornpattanakarn Student ID 55070004
Ms. Wisanu Tongpradit Student ID 55070111

Degree Bachelor of Science

Program Information Technology Academic

Year 2015

Advisor Dr. Lapas Pradittasnee

ABSTRACT

IPv6 Routing Protocols for Low Power and Lossy Network (RPL) is a Routing Protocol that designed for use in networks with low power and the traffic on the network has a high opportunity of data loss. From attractive features this project has brought RPL was studied and developed further in order to provide higher efficiency limits on energy and the physical environment. By which this project is to improve the RPL is based on the Contiki operating system and Cooja Simulator to simulate the properties of data loss. And make improvements in the performance parameters of RPL have better. And presents the results of the updated value of RPL. Finally, this project hopes to contribute for those who are interested. In order to bring this project continue development research and to be applied in the future.

กิตติกรรมประกาศ

งานวิจัยฉบับนี้สำเร็จลงได้ด้วยดี เนื่องจากได้รับความกรุณาอย่างสูงจากอาจารย์ที่ปรึกษา
งานวิจัยคือ ดร. ลภัส ประดิษฐทัศนีย์ ที่กรุณาให้คำแนะนำปรึกษา ตลอดจนปรับปรุงแก้ไข
ข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างดียิ่ง ผู้วิจัยตระหนักถึง ความตั้งใจจริงและความทุ่มเท
ของอาจารย์และขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณ รองศาสตราจารย์ ดร. โขติพัทธ์ ภรณ์วลัย ซึ่งเป็นผู้ทรงคุณวุฒิที่ให้ความ
อนุเคราะห์ อุปกรณ์ Z1 Zolertia แก่ผู้วิจัยและคำแนะนำ รวมถึงขอขอบพระคุณที่เอื้อสถานที่ให้แก่
ผู้วิจัยใช้ในการทำงานวิจัยจาก ผู้ช่วยศาสตราจารย์ ดร. สุเมธ ประภาวัต จนทำให้งานวิจัยนี้สำเร็จ
ลุล่วงไปด้วยดี

อนึ่ง ผู้วิจัยหวังว่า งานวิจัยฉบับนี้จะมีประโยชน์อยู่ไม่น้อย จึงขอมอบส่วนดี ทั้งหมดนี้
ให้แก่เหล่าคณาจารย์ที่ได้ประสิทธิ์ประสาทวิชาจนทำให้ผลงานวิจัยเป็น ประโยชน์ต่อผู้ที่เกี่ยวข้อง
และขอแสดงความกตัญญูตเวทิตาคุณ แต่บิดา มารดา และผู้มี พระคุณทุกท่าน สำหรับข้อบกพร่อง
ต่าง ๆ ที่อาจจะเกิดขึ้นนั้น ผู้วิจัยขออภัยผู้เดียว และยินดีที่จะรับฟังคำแนะนำจากทุกท่าน
ที่ได้เข้ามาศึกษา เพื่อเป็นประโยชน์ใน การพัฒนางานวิจัยต่อไป

กัญญ์ณพัชร คุณากรพัฒนาการ
วิษณุ ทองประดิษฐ

สารบัญ

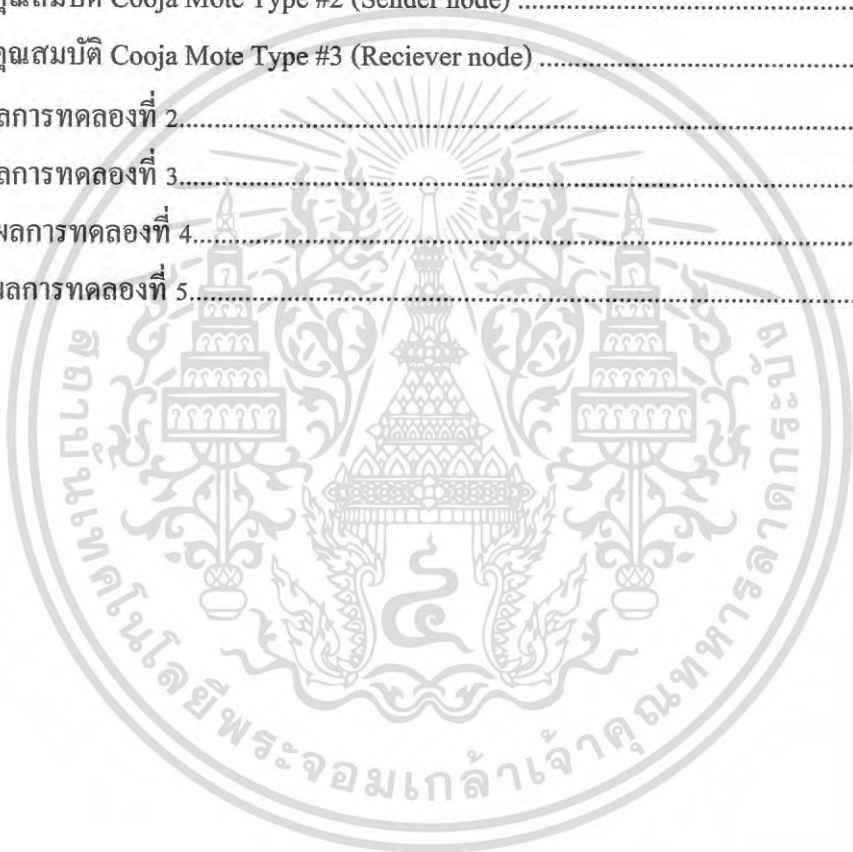
	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการวิจัย.....	1
1.3 สมมติฐานของการศึกษา.....	2
1.4 ทฤษฎีแนวคิดที่ใช้ในโครงการ.....	2
1.5 ขอบเขตการวิจัย.....	2
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง.....	4
2.1 IPv6 over Low power WPAN (6LoWPAN).....	4
2.2 Internet of Thing (IoT).....	7
2.3 IPv6 Routing Protocol for Low Power and Lossy Networks (RPL).....	8
2.4 Wireshark (version 1.7.2).....	11
2.5 Z1 Zolertia.....	12
2.6 ContikiMAC.....	14
บทที่ 3 วิธีการดำเนินงานวิจัย.....	15
3.1 ศึกษาแอปพลิเคชันที่ใช้ในการจำลอง.....	15
3.1.1 Cooja Simulator.....	15
3.1.2 Eclipse IDE for C/C++ Developers Version Mars Release (4.5.0).....	22
3.2 การนำอุปกรณ์ Z1 มาใช้ในการวิจัย.....	23

สารบัญ (ต่อ)

บทที่ 4 ผลการทดลองและสรุป.....	31
4.1 การทดลองใน Cooja Simulator.....	31
4.1.3 การทดลองที่ 1 เครื่องข่าย 6 โหนดและผลการทดลอง.....	32
4.1.4 การทดลองที่ 2 เครื่องข่าย 10 โหนดและผลการทดลอง.....	39
4.1.5 สรุปการทดลองใน Cooja Simulator	46
4.2 การทดลองกับอุปกรณ์ Z1	47
4.2.3 การทดลองที่ 3 เครื่องข่าย 2 โหนด.....	53
4.2.4 การทดลองที่ 4 เครื่องข่าย 3 โหนด.....	58
บทที่ 5 แบบเสนอแนวทางการปรับปรุง.....	60
บรรณานุกรม.....	62
ภาคผนวก ก.....	64
ประวัติผู้เขียน.....	73

สารบัญตาราง

ตารางที่	หน้า
4.1 แสดงคุณลักษณะ Cooja Mote Type #1 (RPL root node)	33
4.2 แสดงคุณสมบัติ Cooja Mote Type #2 (Sender node)	33
4.3 แสดงคุณสมบัติ Cooja Mote Type #3 (Receiver node).....	34
4.4 สรุปผลการทดลองที่ 1.....	38
4.5 แสดงคุณลักษณะ Cooja Mote Type #1 (RPL root node)	39
4.6 แสดงคุณสมบัติ Cooja Mote Type #2 (Sender node)	40
4.7 แสดงคุณสมบัติ Cooja Mote Type #3 (Receiver node)	40
4.8 สรุปผลการทดลองที่ 2.....	46
4.9 สรุปผลการทดลองที่ 3.....	56
4.10 สรุปผลการทดลองที่ 4.....	57
4.11 สรุปผลการทดลองที่ 5.....	59



สารบัญรูป

รูปที่	หน้า
2.1 อินเทอร์เน็ตในชีวิตประจำวัน	4
2.2 แสดงเฟรมของ 6LoWPAN มีขนาด 127 ไบต์	5
2.3 6LoWPAN IPv6 & UDP packet compression	6
2.4 Internet of Things	7
2.5 DAG Graph , DODAO Graph	8
2.6 DODAG Rank	8
2.7 RPL Instance 1 , RPL Instance 2	9
2.8 The DODAG Information Object (DIO) Base Object.....	9
2.9 The DODAG Information Solicitation (DIS) Base Object.....	10
2.10 The DAO Base Object.....	10
2.11 ภาพจำลองสถานการณ์บน Wireshark	11
2.12 Z1 Zolertia.....	12
2.13 ตัวอย่างอัตราสิ้นเปลืองไฟฟ้าของวงจร Z1.....	13
2.14 ContikiMAC.....	14
3.1 แสดงถึง Sky mote sensor ที่ใช้งานบน Cooja Simulator.....	15
3.2 ภาพจำลองสถานการณ์บน Cooja Stimulator	16
3.3 หน้า Login ของ Instant Contiki	17
3.4 การเข้า Cooja simulator	17
3.5 การตั้งชื่อ simulation	18
3.6 การสร้าง โหนด	18
3.7 การเพิ่มจำนวน โหนด	19
3.8 การเริ่ม simulation	20
3.9 ตัวอย่างการ simulation	21
3.10 ภาพจำลองสถานการณ์บน Eclipse	22
3.11 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	23
3.12 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	24

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.13 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	24
3.14 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	25
3.15 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	25
3.16 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	26
3.17 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	26
3.18 การเชื่อมต่อระหว่าง Z1 กับ คอมพิวเตอร์	27
3.19 ตรวจสอบการเชื่อมต่อ Z1.....	27
3.20 ตรวจสอบการเชื่อมต่อ Z1.....	28
3.21 การ reboot virtual machine	29
3.22 การอัปเดตข้อมูล หลังจาก reboot virtual machine.....	29
3.23 การอัปเดตข้อมูล หลังจาก reboot virtual machine.....	30
4.1 การทดลองที่ 1 เครื่องข่าย 6 โหนด.....	35
4.2 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 1 เมื่อครบ 5 นาที.....	35
4.3 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 1 Remove mote 3 ในนาทีที่ 5.....	36
4.4 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 1 Remove mote 4 ในนาทีที่ 10.....	36
4.5 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 1 เมื่อครบ 20 นาที	37
4.6 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 1 เมื่อเวลาผ่านไปอีก 5 นาที.....	37
4.7 การทดลองที่ 2 เครื่องข่าย 10 โหนด.....	42
4.8 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 2 เมื่อครบ 5 นาที.....	42
4.9 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 2 Remove mote 3 ในนาทีที่ 5.....	43
4.10 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 2 Remove mote 5 ในนาทีที่ 10.....	43
4.11 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 2 Remove mote 9 ในนาทีที่ 15.....	44
4.12 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 2 เมื่อครบ 25 นาที.....	44
4.13 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 2 เมื่อปล่อยไว้อีก 5 นาที.....	45
4.14 การทำงานของ Server.....	49
4.15 การทำงานของ Client.....	50

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.13 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	24
3.14 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	25
3.15 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	25
3.16 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	26
3.17 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki.....	26
3.18 การเชื่อมต่อระหว่าง Z1 กับ คอมพิวเตอร์	27
3.19 ตรวจสอบการเชื่อมต่อ Z1.....	27
3.20 ตรวจสอบการเชื่อมต่อ Z1.....	28
3.21 การ reboot virtual machine	29
3.22 การอัป โหลดข้อมูล หลังจาก reboot virtual machine.....	29
3.23 การอัป โหลดข้อมูล หลังจาก reboot virtual machine.....	30
4.1 การทดลองที่ 1 เครื่องข่าย 6 โหนด.....	35
4.2 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 1 เมื่อครบ 5 นาที.....	35
4.3 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 1 Remove mote 3 ในนาทีที่ 5.....	36
4.4 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 1 Remove mote 4 ในนาทีที่ 10.....	36
4.5 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 1 เมื่อครบ 20 นาที	37
4.6 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 1 เมื่อเวลาผ่านไปอีก 5 นาที.....	37
4.7 การทดลองที่ 2 เครื่องข่าย 10 โหนด.....	42
4.8 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 2 เมื่อครบ 5 นาที.....	42
4.9 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 2 Remove mote 3 ในนาทีที่ 5.....	43
4.10 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 2 Remove mote 5 ในนาทีที่ 10.....	43
4.11 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 2 Remove mote 9 ในนาทีที่ 15.....	44
4.12 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 2 เมื่อครบ 25 นาที.....	44
4.13 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 2 เมื่อปล่อยไว้อีก 5 นาที.....	45
4.14 การทำงานของ Server.....	49
4.15 การทำงานของ Client.....	50

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.16 การทดลองที่ 3 ระยะ45เมตร ทางโค้ง โดยใช้ 2 mote.....	51
4.17 การทดลองที่ 4 บนห้อง Common Room ใช้ 2 mote.....	52
4.18 การทดลองที่ 4 ใช้ 3 mote โดยให้ Client1 รับและส่งข้อมูล.....	52
4.19 การทดลองที่ 3 อัตราการส่งข้อมูล 10 วินาที.....	53
4.20 การทดลองที่ 3 อัตราการส่งข้อมูล 5 วินาที.....	53
4.21 การทดลองที่ 3 อัตราการส่งข้อมูล 1 วินาที.....	54
4.22 การทดลองที่ 4 อัตราการส่งข้อมูล 10 วินาที.....	54
4.23 การทดลองที่ 4 อัตราการส่งข้อมูล 5 วินาที.....	55
4.24 การทดลองที่ 4 อัตราการส่งข้อมูล 1 วินาที.....	55
4.25 การทดลองที่ 5 อัตราการส่งข้อมูล 10 วินาที.....	58
4.26 การทดลองที่ 5 อัตราการส่งข้อมูล 5 วินาที.....	58
4.27 การทดลองที่ 5 อัตราการส่งข้อมูล 1 วินาที.....	59

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันอุปกรณ์เซนเซอร์แบบไร้สาย (Wireless Sensor Devices) เป็นเทคโนโลยีที่มีการใช้กันอย่างกว้างขวาง ไม่ว่าจะเป็นทางด้านอุตสาหกรรม ด้านเกษตรกรรม หรือในชีวิตประจำวัน สามารถติดตั้งอุปกรณ์เซนเซอร์และเข้ามาเป็นส่วนหนึ่งของระบบเครือข่ายได้ เพื่อเป็นตัวช่วยอำนวยความสะดวกในด้านต่างๆ เช่น ตู้เย็น, เครื่องใช้ไฟฟ้าต่าง ทำให้อินเทอร์เน็ตโพรโทคอลเวอร์ชัน 4 (Internet Protocol Version 4) ไม่เพียงพอต่อการใช้งาน และต้องใช้ อินเทอร์เน็ตโพรโทคอลเวอร์ชัน 6 (Internet Protocol Version 6) และเทคโนโลยีนี้ก็มีข้อจำกัดที่ไม่อาจหลีกเลี่ยงได้คือ ข้อจำกัดด้านพลังงานที่มีอยู่อย่างจำกัด และข้อจำกัดด้านสภาพแวดล้อมเช่น อากาศ ความร้อน ความชื้น การรบกวนจากเครือข่ายไร้สายชนิดอื่น เป็นต้น ซึ่งส่วนแล้วแต่มีผลต่อการเกิดการสูญหายของข้อมูลได้ ซึ่งในเทคโนโลยีเครือข่ายไร้สาย นี้บนรูปแบบมาตรฐานอินเทอร์เน็ตโพรโทคอลเวอร์ชัน 6 (Internet Protocol Version 6) ที่มีการบริหารจัดการ โดยใช้พลังงานต่ำสอดคล้องกับซิกซ์โลว์พาวเวอร์ (6LoWPAN) นั้น RPL (IPv6 Routing Protocol for Low Power and Lossy Network) เป็นตัวเลือกหนึ่งใน Routing Protocol ที่สามารถตอบโจทย์ในข้อจำกัดของเทคโนโลยีเซนเซอร์แบบไร้สาย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการวิจัย

- มีความเข้าใจในในหลักการการทำงานของโพรโทคอล RPL ตามที่กำหนดในเอกสาร rfc-6550
- ศึกษาการทำงานของ Contiki OS และ โปรแกรม Cooja simulator สำหรับการใช้ทดสอบประสิทธิภาพการทำงานของโพรโทคอล RPL
- วิเคราะห์ผลที่ได้จากการทดสอบประสิทธิภาพของโพรโทคอล RPL และนำเสนอแนวทางการปรับปรุงพารามิเตอร์หรือกระบวนการในการทำงานของโพรโทคอล RPL ให้ทำงานได้ดีถึงแม้มีการรบกวนจากระบบไร้สายอื่น

1.3 สมมติฐานของการศึกษา

ศึกษาการทำงานของ IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) สำหรับการใช้งานภายในอาคาร โดยให้ความสนใจกับประสิทธิภาพในการทำงานภายใต้การรบกวนจากระบบอื่นที่ใช้งานอยู่ในอาคารเดียวกัน เช่น ระบบเครือข่ายไร้สาย IEEE 802.11 หรือเครือข่าย Bluetooth

1.4 ทฤษฎีแนวคิดที่ใช้ในโครงการ

นำหลักการของ IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) มาศึกษาเพื่อใช้ในเสนอแนวทางการปรับปรุงการค้นหาเส้นทางอย่างมีประสิทธิภาพ ด้วยการใช้เครื่องมือ Cooja Simulator บน Contiki-OS ควบคู่ไปกับการทดสอบประสิทธิภาพเมื่อนำโปรโตคอล RPL มาติดตั้งบน Z1

1.5 ขอบเขตการวิจัย

- ศึกษาและติดตั้งโปรโตคอล RPL บนโปรแกรมจำลอง Cooja Simulator เพื่อทำการสร้างและทดสอบประสิทธิภาพการทำงานของโปรโตคอล RPL
- ศึกษาและติดตั้งโปรโตคอล RPL บนตัว Z1 mote รวมทั้งการติดตั้งและใช้งานโปรแกรมค้นหาค่าข้อมูล เพื่อใช้ทดสอบประสิทธิภาพการทำงานของโปรโตคอล RPL ในระบบเครือข่ายที่ใช้งานจริง
- ศึกษาและวิเคราะห์ เพื่อเสนอแนวทางในการปรับปรุงโปรโตคอล RPL เพื่อให้มีประสิทธิภาพดีขึ้นในกรณีที่มีการรบกวนจากระบบเครือข่ายไร้สายอื่น

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- เพื่อนำไปต่อยอดในการพัฒนาการค้นหาเส้นทางบนเครือข่ายไร้สายให้เป็นไปอย่างมีประสิทธิภาพตามความต้องการจากตัวแอปพลิเคชันที่กำหนดในอนาคต
- ทำให้ได้รับความเข้าใจและความรู้เพิ่มเติมจากการค้นคว้าความรู้ใหม่ๆ
- ฝึกฝนการศึกษาและทำงานในเชิงวิจัย
- เรียนรู้การทำงานที่เป็นระบบแบบแผน

1.7 ขั้นตอนการดำเนินงาน

- ศึกษาหลักการทำงานของ IPv6 Routing Protocol for Low Power and Lossy Networks (RPL)
- ศึกษาหลักการทำงานของเครื่องมือ Cooja Stimulator บน Contiki-OS
- ศึกษาภาษา Programming C
- ศึกษาทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง
- ดำเนินการทำงานทดลองบนเครื่องมือ Cooja Stimulator
- ดำเนินการทำงานทดลองบนอุปกรณ์ Z1
- บันทึกผลการวิจัย เพื่อนำผลที่ได้ไปพัฒนาต่อให้มีประสิทธิภาพมากขึ้น
- สรุปปัญหาที่พบ และศึกษาปัญหาที่พบ
- ดำเนินการแก้ไขปัญหาที่พบ
- สรุปผลการวิจัยที่ได้



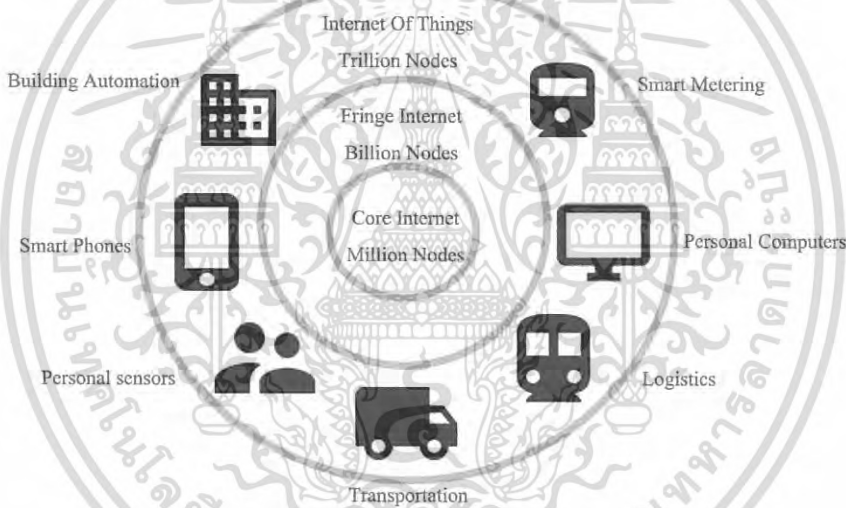
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

2.1 IPv6 over Low power WPAN (6LoWPAN)

IPv6 over Low power WPAN หรือ 6LoWPAN เป็นชุดของมาตรฐานที่กำหนดโดย Internet Engineering Task Force (IETF) ซึ่งถูกสร้างให้เป็นไปตามสถาปัตยกรรมตามแบบอินเทอร์เน็ตโปรโตคอล โดยที่มาตรฐาน 6LoWPAN ช่วยให้มีการใช้งาน Internet protocol version 6 (IPv6) ที่มีประสิทธิภาพในอุปกรณ์ฝังตัว (Embedded devices) ที่มีความเรียบง่ายและปรับขึ้นให้เข้ากับโปรโตคอลที่เกี่ยวข้องเพื่อทำให้มีประสิทธิภาพ แนวคิดของ 6LoWPAN มาจากความคิดที่ว่า “สามารถนำโปรโตคอลอินเทอร์เน็ตมาใช้ได้แม้กระทั่งอุปกรณ์ที่มีขนาดเล็กที่สุด”



รูปที่ 2.1 อินเทอร์เน็ตในชีวิตประจำวัน

IETF 6LoWPAN working group เริ่มทำงานเป็นทางการในปี 2005 แม้ว่าประวัติของ Embedded IP นั้นจะมีมาตั้งแต่สมัย 1990 ซึ่งถูกสันนิษฐานว่ามีความเจริญก้าวหน้าตามกฎของมัวร์จนทำให้อุปกรณ์ประเภท Embedded นั้นสามารถใช้โปรโตคอล IP จนกลายเป็น Internet of Things (IoT) ถึงแม้ว่าความเจริญก้าวหน้าจะเป็นไปอย่างรวดเร็วแต่ก็ยังไม่สามารถบอกได้ว่าราคานั้นถูกไม่ใครคอนโทรลเลอร์ที่ใช้พลังงานต่ำและมีเทคโนโลยีคลื่นวิทยุแบบไร้สายนั้นส่วนใหญ่วิทยุแบบเรียบง่ายนั้นยังคงใช้เป็นลักษณะแบบ 8-bit และแบบ 16-bit ทำให้มีข้อจำกัดในด้านหน่วยความจำและพลังงานที่ต่ำซึ่งส่งผลให้มีการจำกัดอัตราการส่งข้อมูล ขนาดของเฟรม และ duty cycles เช่นมาตรฐาน IEEE 802.15.4 ซึ่งในช่วงแรกได้มีการนำ uIP จาก Swedish Institute of Computer

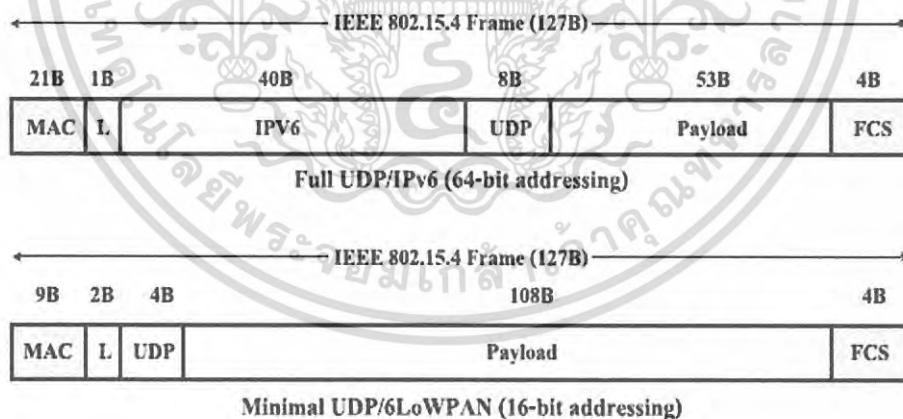
Science และ NanoIP จาก Centre for Wireless Communications เพื่อนำมาใช้งาน จนในปี 2003 ได้มีราคาไม่แพงเกินไป ทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีการปล่อยมาตรฐาน IEEE 802.15.4 เพื่อนำไปสู่มาตรฐาน 6LoWPAN และข้อกำหนดแรกได้ถูกปล่อยตัวในปี 2007 และเป็นครั้งแรกบน RFC4944

6LoWPAN ได้มีการปรับ Layer เพื่อให้เข้ากับการส่งและการรับ packets ของ IPv6 และยังใช้ 802.15.4 เพื่อทำการ unslotted CSMA/CA มีการทำ Fragmentation และ reassembly เพื่อให้ packets เข้ากับเทคโนโลยีที่แตกต่างกันได้ อีกทั้งยังมีการทำการ encapsulation ของ IPv6 และยังมีบีบอัดในส่วนของ header และในส่วนของ UDP/ICMP และสนับสนุน Mesh routing อีกด้วย

คุณสมบัติหลักของ 6LoWPAN

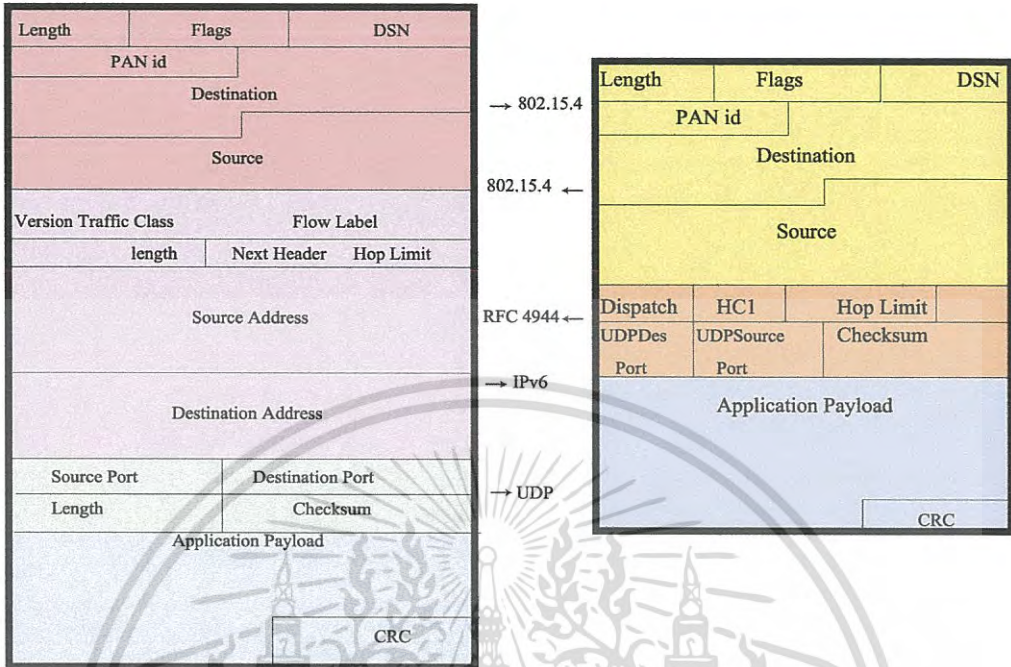
- มีขนาดแพ็กเก็ตขนาดเล็ก มีขนาดเพียง 127 ไบต์ ใน Physical Layer
- สามารถเข้า Sleep Mode ได้เพื่อประหยัดพลังงานอุปกรณ์
- ใช้แบนด์วิดธ์ต่ำ (250/40/20 kbps)
- รองรับการใช้งานในอุปกรณ์พลังงานต่ำ เช่น การใช้พลังงานจากแบตเตอรี่
- ค่าใช้จ่ายราคาถูก
- รองรับการใช้งานแบบ ad-hoc ได้
- รองรับ MAC Address ที่ขนาด 16 bit และ 64 bit
- รองรับการทำ AES block cypher (AES-CCM-64 mandatory)



รูปที่ 2.2 แสดงเฟรมของ 6LoWPAN มีขนาด 127 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบโครงสร้างของ Packet และ รูปการบีบอัดของ IPv6 และ UDP เป็นดังรูปที่ 2.3



รูปที่ 2.3 6LoWPAN IPv6 & UDP packet compression

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Internet of Thing (IoT)

Internet of Things (IoT) หรือ “อินเทอร์เน็ตในทุกสิ่ง” คือ วัตถุทางกายภาพหรือสิ่งต่างๆ ที่ฝังอยู่กับอุปกรณ์ อิเล็กทรอนิกส์, ซอฟต์แวร์, เซ็นเซอร์ และการเชื่อมต่อเครือข่าย ถูกเชื่อมโยงทุกสิ่งทุกอย่างเข้าสู่โลกอินเทอร์เน็ต ที่ช่วยวัตถุเหล่านี้ในการเก็บรวบรวมและแลกเปลี่ยนข้อมูล IoT จะช่วยให้วัตถุมีความรู้สึกในการตอบสนองและควบคุมจากระยะไกลผ่านเครือข่ายโครงสร้างพื้นฐานที่มีอยู่ เพื่อให้โลกทางกายภาพและระบบคอมพิวเตอร์ที่ใช้มีประสิทธิภาพที่ดีขึ้น ทั้งในเรื่องของความถูกต้องและผลประโยชน์ทางเศรษฐกิจ เครื่องจักรในโรงงานอุตสาหกรรม อาคาร บ้านเรือน เครื่องใช้ในชีวิตประจำวันต่างๆ ผ่านเครือข่ายอินเทอร์เน็ต เป็นต้น

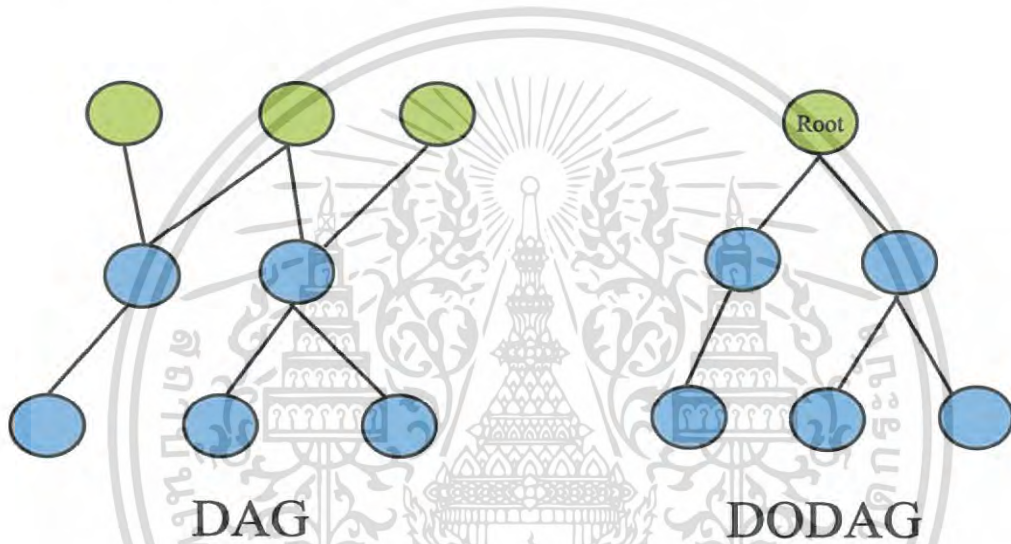


รูปที่ 2.4 Internet of Things

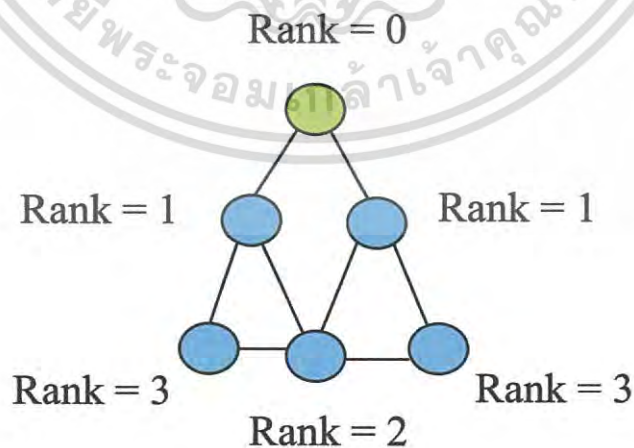
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 IPv6 Routing Protocol for Low Power and Lossy Networks (RPL)

RPL เป็น Routing Protocol ที่ถูกพัฒนามาเป็นพิเศษสำหรับเพื่อประยุกต์ใช้งานกับเครือข่ายที่มีการสูญหายของข้อมูลสูงและใช้งานกับอุปกรณ์ที่มีพลังงานต่ำ (LLNs) สอดคล้องกับ 6LoWPAN โดยมีการใช้อัลกอริทึมในการค้นหาเส้นทางแบบ Distance Vector มีพื้นฐานโครงสร้าง Topology ที่มีลักษณะเป็นแบบลำดับขั้นที่เรียกว่า DAG (Directed Acyclic Graph) โดยแต่ละกลุ่มจะสร้างเครือข่ายที่เชื่อมต่อกันเป็นแบบ DODAG (Destination Oriented DAG) คือจะมีโหนดหลัก (Root node) ได้เพียงโหนดเดียว เดียวดังรูปที่ 2.5



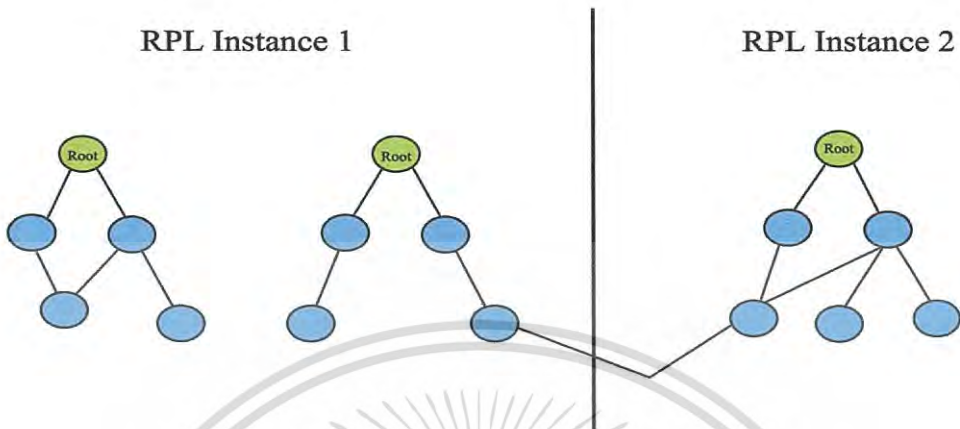
รูปที่ 2.5 DAG Graph , DODAO Graph



รูปที่ 2.6 DODAG Rank

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.6 แสดงถึง ความสัมพันธ์ในแต่ละโหนดของ DODAG ภายใน Instance อาจประกอบไปด้วยหลายๆ DODAG และ มี Root Node ที่ไม่เหมือนกัน ดังรูปที่ 2.7



รูปที่ 2.7 RPL Instance 1 , RPL Instance 2

RPL ได้กำหนด Control messages สำหรับการสร้างและบำรุงรักษาโครงสร้างของ RPL และตารางเส้นทาง มีสามประเภทดังต่อไปนี้

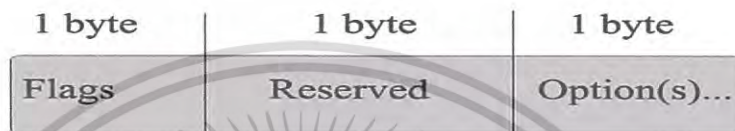
1. DAG Information Object (DIO): เป็น message เพื่อใช้ในการค้นพบกราฟ DODAG ซึ่งเริ่มโดยโหนด root (SINK) และดำเนินการกระจายข้อมูลเพื่อให้โหนดข้างเคียงทำการเรียนรู้ค่าพารามิเตอร์เพื่อใช้ในการเลือก parent และบำรุงรักษา DODAG โดย DIO message นั้นจะถูกส่งเป็น link-local multicast ไปยังโหนดใกล้เคียงในทิศทาง downward และเปิดการทำงานแบบ Point-to-Multipoint traffic ในทิศ upward ซึ่งใน DIO message จะประกอบไปด้วยข้อมูล routing metrics, rank, objective function และ DODAG-ID ซึ่ง DIO message นี้จะถูกส่งออกเป็นระยะตามกระบวนการทำงานของการเลือก parent

1 byte		1 byte		2 bytes	
RPLInstanceID		Version Number		Rank	
G	0	MOP (3bit)	DTSN (3bit)	DTSN	Flags Reserved
DODAGID (128bit)					
Option(s)...					

รูปที่ 2.8 The DODAG Information Object (DIO) Base Object

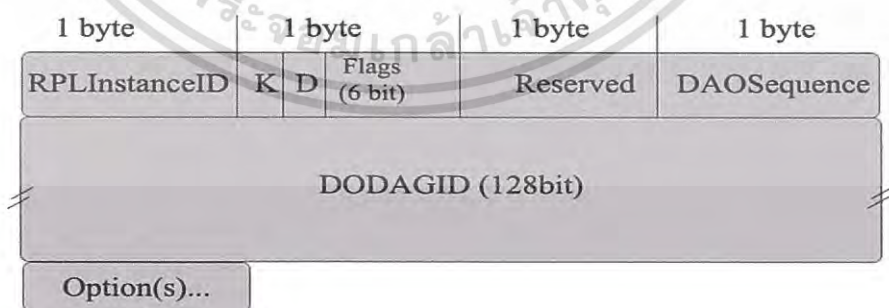
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. DAG Information Solicitation (DIS): เป็น message เพื่อนำมาใช้เพื่อก่อให้เกิดการส่ง DIO message จากโหนดเพื่อนบ้าน โดย message นี้จะถูกส่งเมื่อต้องการเข้าร่วมเครือข่าย กล่าวคือ เมื่อโหนดต้องการเข้าร่วมเครือข่ายโหนดจะทำการ broadcast DIS message ไปยังโหนดใกล้เคียงเพื่อทำการร้องขอให้มีการส่ง DIO message จากโหนดใกล้เคียงนั้น



รูปที่ 2.9 The DODAG Information Solicitation (DIS) Base Object.

3. Destination Advertisement Object (DAO): เป็น Message ที่ถูกส่งโดยโหนดในทิศทางขึ้นเพื่อเปิดการทำงานแบบ point-to-multipoint traffic ในทิศทาง downward โดย Message นี้สามารถส่ง unicast จาก โหนด child ไปยังโหนด parents ที่เลือกไว้ หรือส่ง unicast ไปยังโหนด Root (SINK) ซึ่งใน DAO message จะมีข้อมูล DAO sequence, RPLInstanceID, DAO rank, DAO lifetime, Route tag, Destination Prefix และ Reverse RouteStack



รูปที่ 2.10 The DAO Base Object.

เมื่อเริ่มสร้างโทโพโลยีขึ้น ผู้ดูแลระบบเครือข่ายจะกำหนดค่าโหนดหนึ่งมาเป็น Root Node จะเริ่มสร้างแพ็กเก็ตชื่อว่า DAG Information Object (DIO) แล้วเริ่มการส่งแบบ link local multicast เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

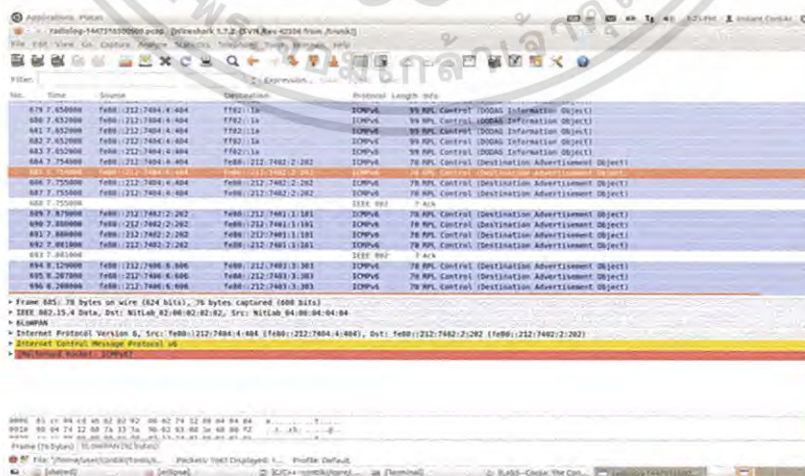
ไปยังทุก Childs Node ซึ่งบาง Childs Node ไม่ได้รับDIOจึงร้องขอ DIO จาก Root Node โดยการส่ง DIS ไปยัง Root Node และ Root Node ส่ง DIOกลับมาทันทีที่ได้รับ DIS Childs Node ที่ได้รับ DIO จาก Root Node จะดำเนินการเลือก Parent Node โดยใช้ OF เป็นพื้นฐานในการตัดสินใจ

2.4 Wireshark (version 1.7.2)

Wireshark เป็นฟรีแวร์ที่ใช้สำหรับแก้ไขปัญหาและวิเคราะห์แพ็กเก็ตเกิดบนระบบเครือข่ายที่ติดต่อสื่อสารกันผ่าน โพรโทคอล เดิม Wireshark นั้นมีชื่อว่า Ethereal และถูกเปลี่ยนในปี 2006 เนื่องจากปัญหาเครื่องหมายการค้า ซึ่ง Wireshark จะใช้ pcap เพื่อจับแพ็กเก็ตซึ่งทำงานบน Linux, OSX, BSD, Solaris และ Microsoft Windows นอกจากนี้ยังมีในลักษณะแบบ Non-GUI หรือที่เรียกว่า TShark ซึ่งจะทำงานบน Terminal อีกด้วย

ลักษณะพิเศษของ Wireshark ดังต่อไปนี้

- สามารถดักจับข้อมูลผ่านสายที่ทำการเชื่อมต่อหรืออ่านจากไฟล์ที่มีการจับไว้แล้วได้
- สามารถอ่านข้อมูลจากเครือข่ายประเภทต่างๆ รวมทั้ง Ethernet ,IEEE 802.11 , PPP และ Loopback
- ไฟล์ที่มีการจับมาสามารถแก้ไขได้โดยการใช้ editcap
- สามารถตัวกรอง(filter)เพื่อแสดงผลตามที่ต้องการได้
- มี Plug-in เพื่อใช้สำหรับการสร้างและดัดแปลง โพรโทคอลชั้นใหม่ได้
- การจับทราฟฟิกประเภท VoIP ก็สามารทำได้
- สามารถจับข้อมูลที่ผ่าน USB ก็ได้เช่นกัน
- สามารถจับข้อมูลที่เป็นการเชื่อมต่อแบบ ไร้สายก็ได้เช่นกัน



รูปที่ 2.11 ภาพจำลองสถานการณ์บน Wireshark

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Z1 Zolertia



รูปที่ 2.12 ภาพอุปกรณ์ Z1 Zolertia

Z1 เป็นแพลตฟอร์มในการพัฒนาสำหรับ เครือข่ายเซ็นเซอร์ไร้สายพลังงานต่ำ (WSN) ตามมาตรฐาน IEEE 802.15.4 ออกแบบมาเพื่อทำปฏิบัติการในงานวิจัย และผู้ที่ชื่นชอบในการพัฒนาสามารถทำงานร่วมกันได้ระหว่างตระกูลของ Tmote™ อีกด้วยนอกจากนี้

สถาปัตยกรรมหลักของ Z1 อยู่บนพื้นฐานของ MSP430 กับ CC2420 ของไมโครคอนโทรลเลอร์และส่งสัญญาณวิทยุโดย Texas Instruments ทำให้มันเข้ากันได้กับ mote ที่ใช้ในสถาปัตยกรรมเดียวกัน อย่างไรก็ตาม MCU ที่มีอยู่ใน Z1 เป็น MSP430F2xxx แทนที่จะเป็น MSP430F1xxx อย่างที่เป็นใน motes อื่นๆ เช่น Crossbow's TelosB, Moteiv's Tmote ทำให้สร้างความแตกต่าง เนื่องจากการเปลี่ยนแปลงภายในระหว่างอุปกรณ์ F1xxx และ F2xxx

โมดูลนี้จะได้รับการคิดค้นจากจุดเริ่มต้นที่มีสองเป้าหมายดังต่อไปนี้

- มีความเข้ากันได้ได้อย่างสูงสุดในตระกูล Tmote™ ขณะที่การปรับปรุงประสิทธิภาพการทำงานในหลายๆด้าน
- มีความยืดหยุ่นสูงสุดและรองรับการขยายเกี่ยวกับการจัดเตรียมในเรื่องพลังงาน, การเชื่อมต่อกันของเซนเซอร์

นอกจากนั้นยังสนับสนุนสำหรับระบบปฏิบัติการแบบเปิดที่กำลังมาในปัจจุบันที่มาโดย WSN community เช่น TinyOS 2.x และ Contiki รองรับการใช้งานระบบเครือข่ายที่เรียงทับซ้อนกัน (network stacks) ที่รวมถึง 6LoWPAN

การประยุกต์ใช้งาน

- ประยุกต์อุปกรณ์ลงใน Internet of Things (IoT)
- การตรวจสอบการดูแลสุขภาพส่วนบุคคล
- การตรวจสอบสิ่งแวดล้อม

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และใช้เพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- อุปกรณ์ความปลอดภัยและการกู้ภัย
- การตรวจสอบแบบอัตโนมัติในระยะยาว
- ระบบวัดค่าการใช้พลังงานไฟฟ้า
- การตรวจสอบทางการเกษตร

คุณสมบัติของ Z1

- แพลตฟอร์มการพัฒนาที่เหมาะสมสำหรับการสร้างต้นแบบอย่างรวดเร็ว / การปรับ WSN
- รองรับช่วงอุณหภูมิของระดับอุตสาหกรรม (-40°C- 5°C)
- รองรับการเชื่อมต่อเพิ่มเติมอีก 52-pin
- มี MSP430™ รุ่นที่ 2 ที่มีการใช้พลังงานต่ำเป็นพิเศษ และมี 16-bit MCU 16MHz
- 2.4GHz IEEE 802.15.4, มาตรฐาน 6LoWPAN และ ZigBee™
- 3-Axis, $\pm 2/4/8/16$ g digital accelerometer (เซ็นเซอร์วัดความเอียง)
- เป็นเซ็นเซอร์วัดอุณหภูมิแบบดิจิทัลที่มีความแม่นยำ $\pm 0.5^{\circ}\text{C}$ (ในช่วง $-25^{\circ}\text{C} \sim 85^{\circ}\text{C}$) ที่ใช้พลังงานต่ำ
- เสออากาศภายนอกตัวเลือกการเชื่อมต่อผ่าน U.FL
- เชื่อมต่อ micro-USB สำหรับการชาร์จและแก้จุดบกพร่อง

คุณสมบัติทางไฟฟ้า

Z1 เป็นโมดูลที่ได้มีการกำหนดขีดอุณหภูมิตามอุตสาหกรรม ในปกติแล้วควรใช้พลังงาน (Power) ที่ 3V ถึงแม้ว่าอาจจะทำงานบางส่วนก็ใช้พลังงานที่ 1.8V (w/o radio) หรือ 2.1V (w/ radio) ซึ่งคำอธิบายต่างๆของวงจรถูกแสงไว้ตามรูป 2.13 ซึ่งช่วยให้นักพัฒนาคำนวณหรือประมาณการของการใช้พลังงานของในแต่ละ IC ได้

IC	Operating Range	Current Consumption	Notes
MSP430F2617	1.8V to 3.6V	0.1 μA	OFF Mode
		0.5 μA	Standby Mode
		0.5mA	Active Mode @1MHz
		< 10mA	Active Mode @16MHz
CC2420	2.1V to 3.6V	<1 μA	OFF Mode
		20 μA	Power Down
		426 μA	IDLE Mode
		18.8mA	RX Mode
ADXL345	1.8V to 3.6V	0.1 μA	Standby
		40 μA to 145 μA	Active Mode
M25P16	2.7V to 3.6V	1 μA	Deep Power Down
		4mA to 15mA	A Active Mode
TMP102	1.4V to 3.6V	1 μA	Shutdown Mode
		15 μA	Active Mode

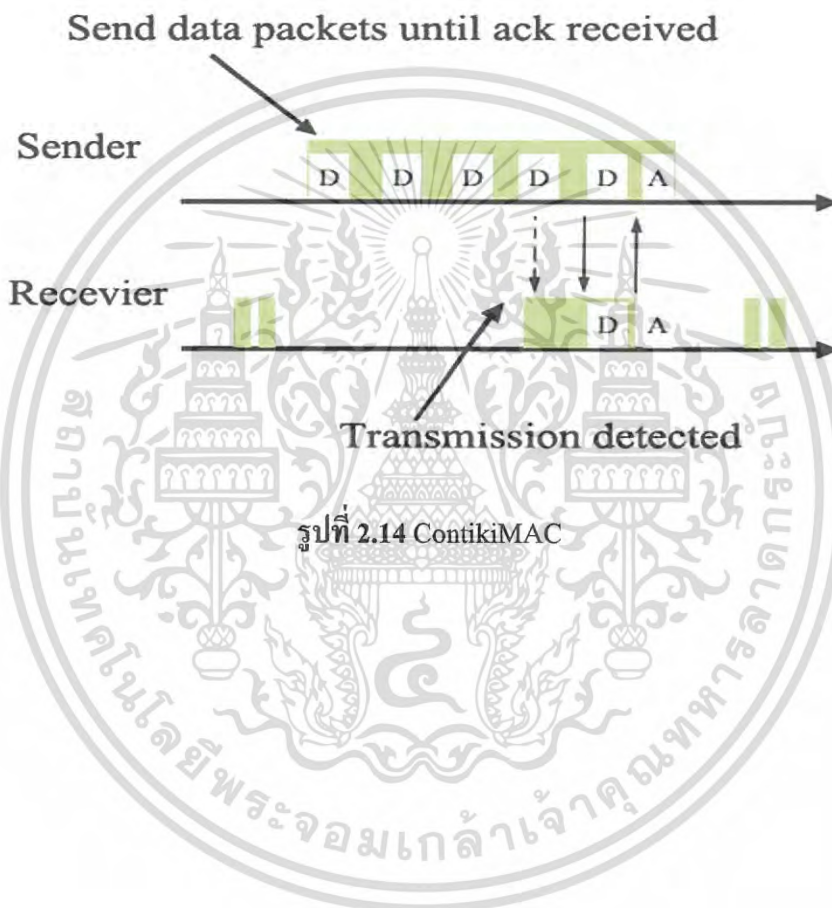
รูปที่ 2.13 ตัวอย่างอัตราสิ้นเปลืองไฟฟ้าของวงจร Z1

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือสงวนลิขสิทธิ์ในชื่อของเจ้าของข้อมูล เมื่อผู้ใดนำข้อมูลนี้ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ContikiMAC

ContikiMAC เป็น โพรโตคอลชนิด radio duty cycling ที่มีช่วงเวลาตื่นขึ้นมารับแพ็กเก็ตเกิดจากโหนดที่อยู่ติดกันที่อยู่ใกล้กัน หากช่วงเวลาการส่งแพ็กเก็ตเกิดเป็นช่วงเวลาที่มีโหนดตื่น ตัวรับก็ยังคงที่จะได้รับแพ็กเก็ต เมื่อได้รับแพ็กเก็ตเรียบร้อยแล้ว ตัวรับจะทำการส่ง acknowledgment การส่งแพ็กเก็ตของผู้ส่งจะส่งซ้ำๆจนกว่าจะได้รับ acknowledgment จากตัวรับ ว่ามันได้รับแพ็กเก็ตแล้ว

ดังรูปที่ 2.13



บทที่ 3

วิธีการดำเนินงานวิจัย

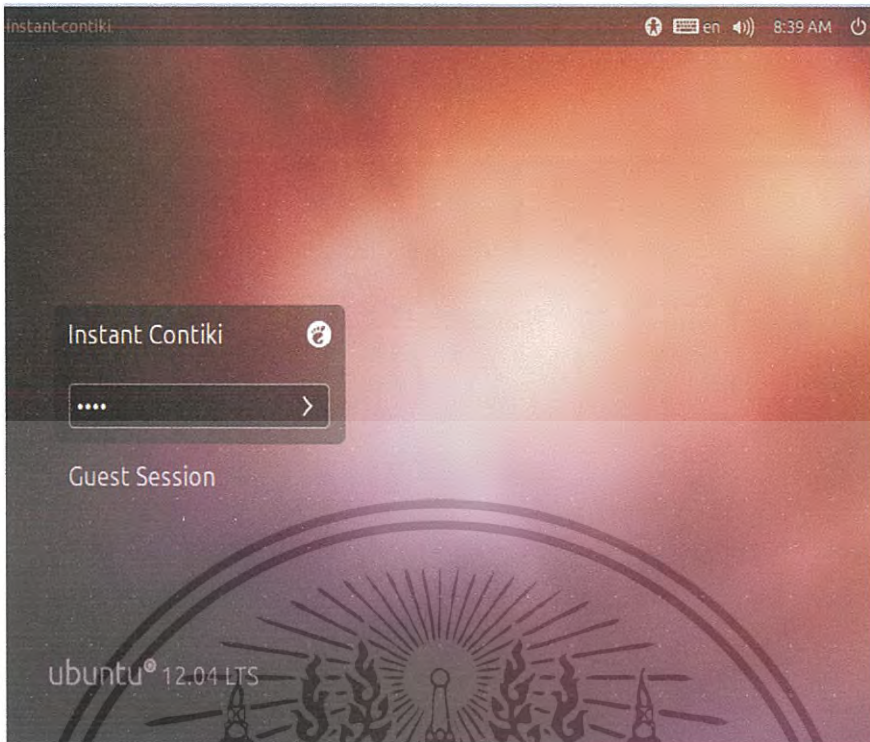
3.1 ศึกษาแอปพลิเคชันที่ใช้ในการจำลอง

3.1.1 Cooja Simulator

เป็นแอปพลิเคชันที่มาพร้อมกับตัว Contiki OS ใช้ในการจำลองสภาพแวดล้อมของระบบระบบเครือข่าย และยังสามารถจำลอง mote ไม่ว่าจะเป็น Sky mote ,Z1 Mote ,CC430 mote , cooja mote เป็นต้นออกมาได้ และ Firmware นั้นถูกพัฒนาด้วยภาษา C ที่สามารถแก้ไขตัดแปลงต่างๆเองของตัว mote ได้อีกด้วยที่ช่วยให้นักพัฒนาได้เห็นถึงการทำงานของสภาพเครือข่าย และรายละเอียดต่างๆ โดย Cooja Simulator ยังเป็น Open Source สำหรับ Internet of Things รับรองการสื่อสารที่ใช้พลังงานต่ำ อีกทั้งยังรองรับ IPv4 และ IPv6 รองรับมาตรฐาน 6LoWPAN , RPL , CoAP อีกด้วย คิวโวลดซอร์ฟแวร์ได้ที่ <http://sourceforge.net/projects/contiki/files/Instant%20Contiki/>

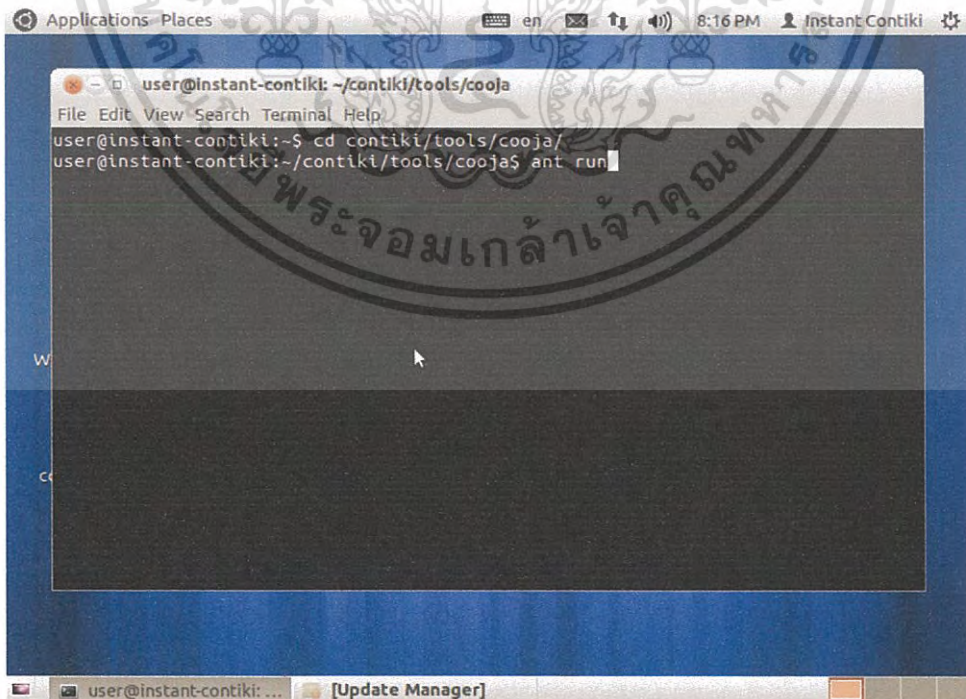


รูปที่ 3.1 ตัวอย่าง Sky mote sensor ที่ใช้งานบน Cooja Simulator



รูปที่ 3.3 หน้า Login ของ Instant Contiki

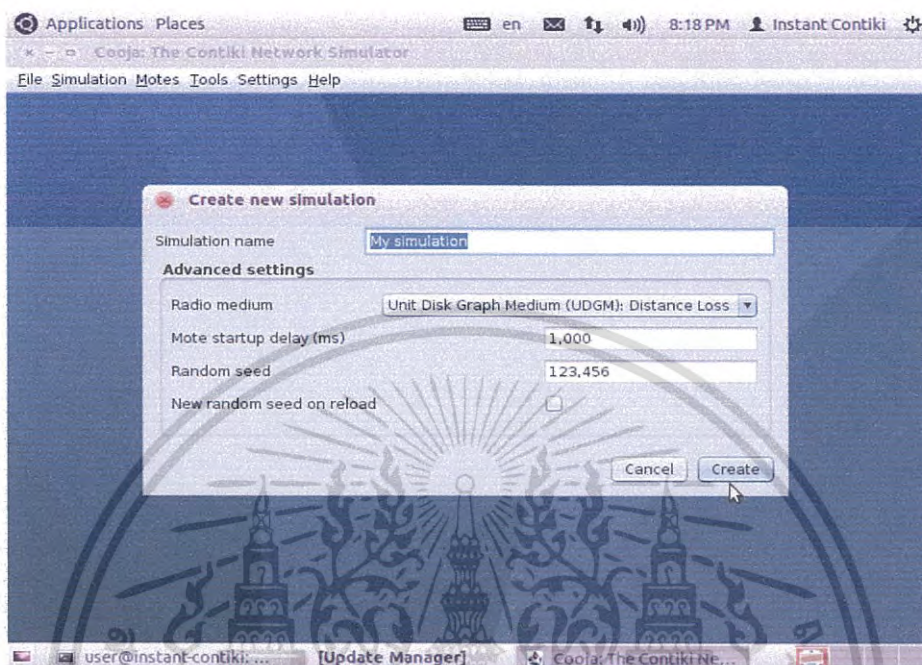
4. บนหน้า Desktop จะมี Icon ที่ชื่อว่า cooja อยู่เข้าทาง Desktop ได้เลย หรือเข้าโดยผ่าน terminal ก็คือ `cd contiki/tools/cooja ant run`



รูปที่ 3.4 การเข้า Cooja simulator

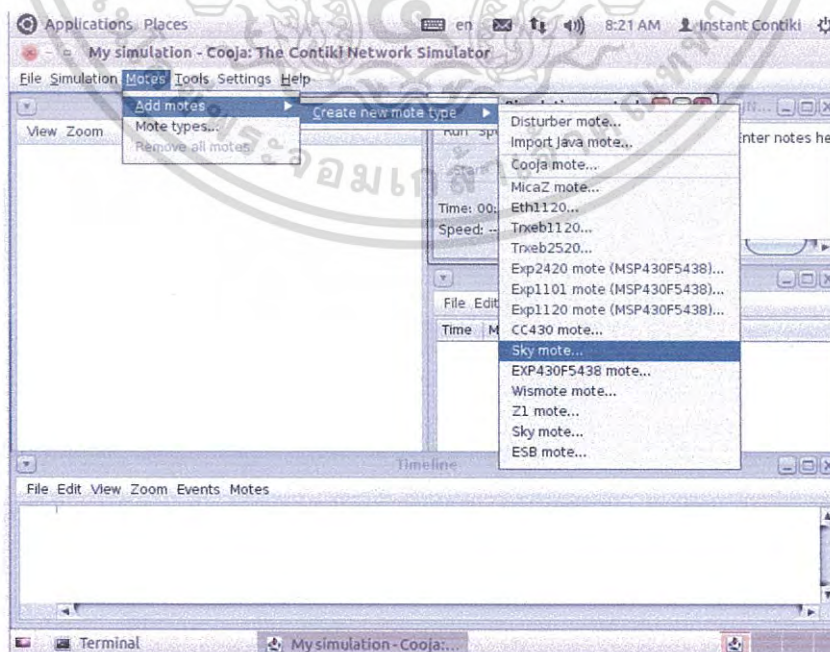
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เมื่อเข้ามาดูหน้าต่าง cooja ให้ทำการสร้าง Simulation ขึ้นมาใหม่ โดยไปที่ File > New Simulation... (Ctrl+N) จะมีหน้าต่างขึ้นมาให้ใส่ชื่อของการ Simulation



รูปที่ 3.5 การตั้งชื่อ Simulation

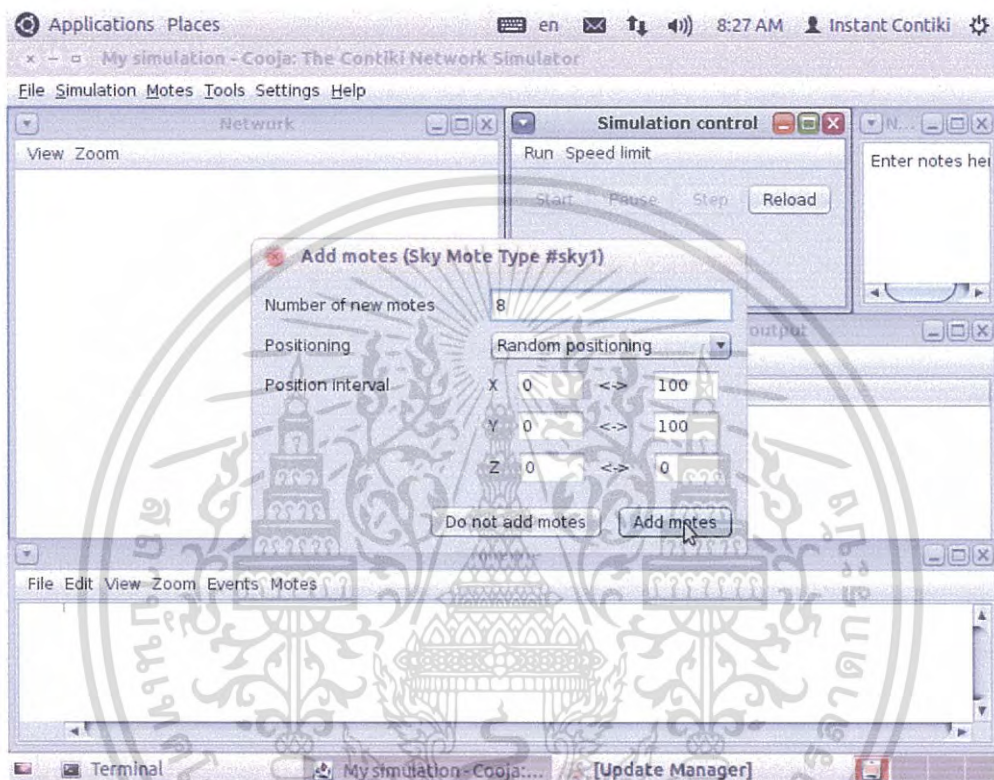
6. ทำการเพิ่ม Mote โดยไปที่ Motes > Create new mote type > Sky mote โดยในส่วนนี้เราสามารถเลือกการจำลอง mote ขึ้นมาได้ตามอุปกรณ์ที่เราต้องการซึ่งมาให้เลือกมากมาย



รูปที่ 3.6 การสร้างโหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

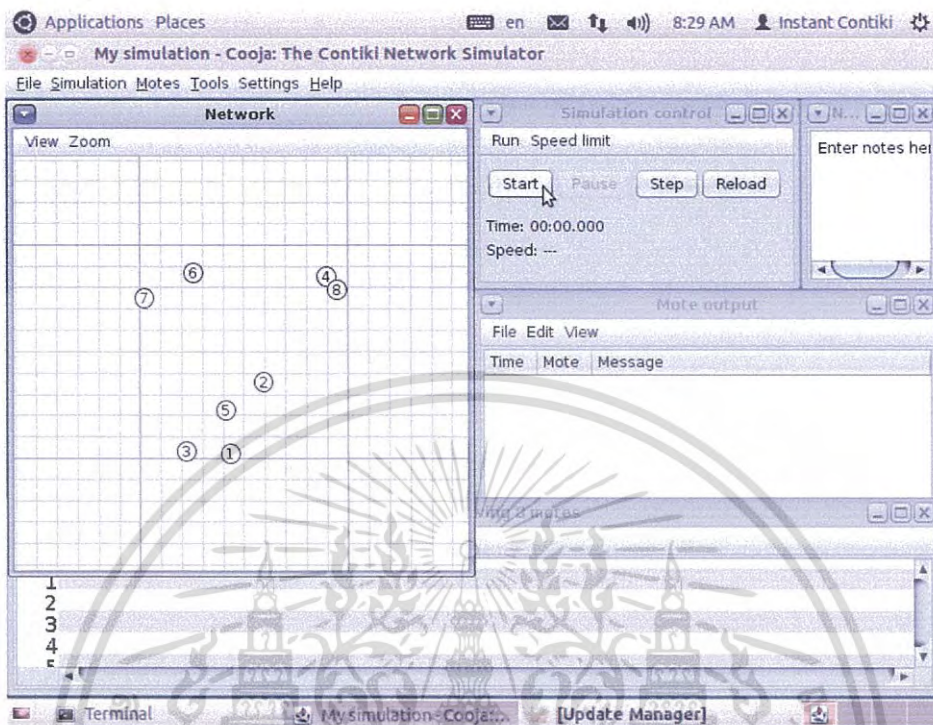
7. จากนั้นสามารถตั้ง คำอธิบาย mote ได้ และทำการคลิก Browse เพื่อทำการเลือก Firmware โดยเลือกที่ /home/user/contiki/examples/ipv6/simple-udp-rpl/broadcast-example.c และทำการคลิก compile และคลิก create จากนั้นจะมีหน้าต่างต่างมา ให้ระบุจำนวน mote อาจจะระบุไปเป็น 8 mote จากนั้นคลิก Add notes



รูปที่ 3.7 การเพิ่มจำนวน โหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

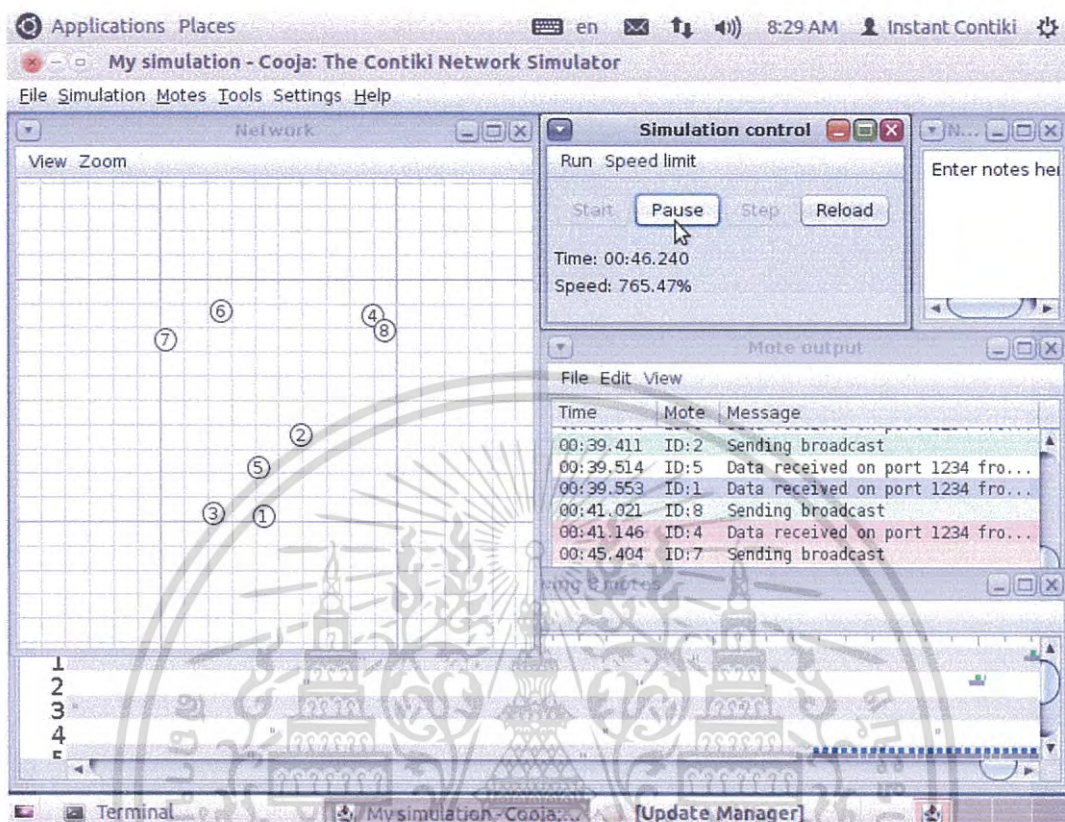
8. ที่หน้าต่าง Simulation control จะมีปุ่ม Start ให้คลิกก็จะเป็นการเริ่ม Simulation



รูปที่ 3.8 การเริ่ม simulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ที่หน้าต่าง Simulation control จะมีปุ่ม Pause ให้คลิกก็จะเป็นการหยุดชั่วคราว Simulation



รูปที่ 3.9 ตัวอย่างการ simulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 Eclipse IDE for C/C++ Developers Version Mars Release (4.5.0)

เป็นโปรแกรมที่ใช้สำหรับพัฒนาภาษา C และ C++ และเป็นซอฟต์แวร์ประเภท OpenSource ส่วนประกอบของโปรแกรม Eclipse IDE for C/C++ Developers Version Mars Release (4.5.0) ประกอบไปด้วย

- Eclipse C/C++ Development Tools
- Eclipse Git Team Provider
- Mylyn Task List
- Remote System Explorer

```

1236 PRINTF("RPL: Received consistent DIO\n");
1237 if(dag->joined) {
1238     instance->dio_counter++;
1239 }
1240 } else {
1241     p->rank=dio->rank;
1242 }
1243 }
1244
1245 PRINTF("RPL: preferred DAG ");
1246 PRINTGADDR(&instance->current_dag->dag_id);
1247 PRINTF(", rank %u, min rank %u,
1248     instance->current_dag->rank, instance->current_dag->min rank);
1249 PRINTF("parent rank %u, parent etx %u, link metric %u, instance etx %u\n",
1250     p->rank, -1/p->mc.obj.etx, p->link metric, instance->mc.obj.etx);
1251
1252 /* We have allocated a candidate parent; process the DIO further. */
1253
1254 #if RPL DAG MC != RPL DAG MC NONE
1255 memcpy(&p->mc, &dio->mc, sizeof(p->mc));
1256 #endif /* RPL DAG MC != RPL DAG MC NONE */
1257 if(rpl_process_parent_event(instance, p) == 0) {
1258     PRINTF("RPL: The candidate parent is rejected\n");
1259     return;
1260 }
1261
1262 /* We don't use route control, so we can have only one official parent. */
  
```

รูปที่ 3.10 ภาพจำลองสถานการณ์บน Eclipse

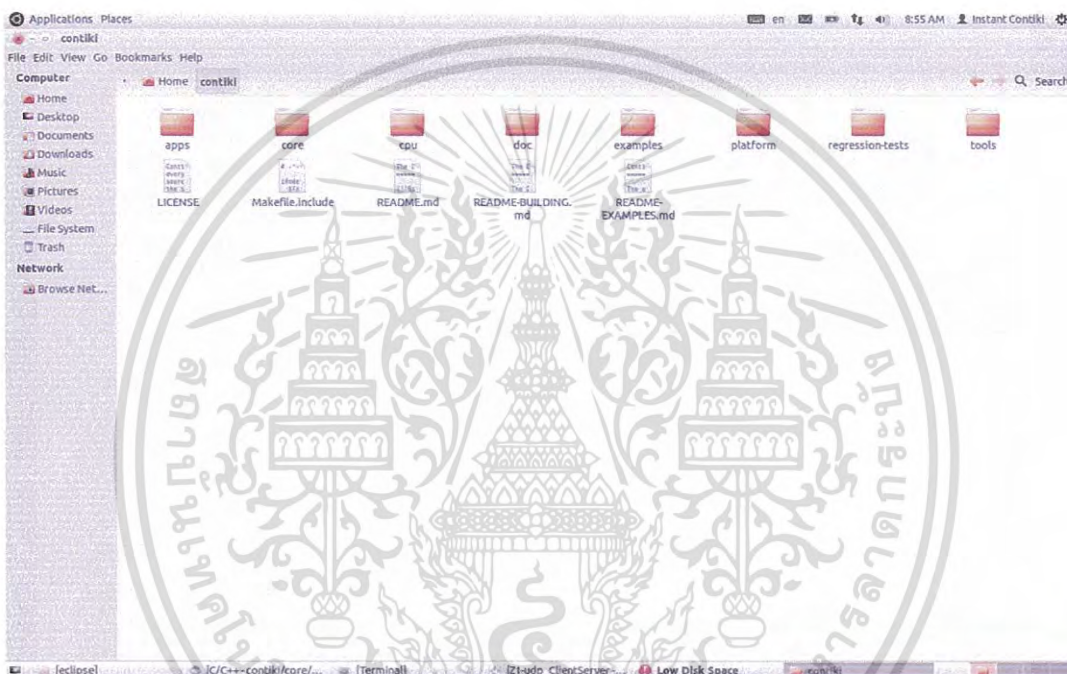
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การนำอุปกรณ์ Z1 มาใช้ในการวิจัย

ขั้นตอนการใช้งาน Z1 เบื้องต้น

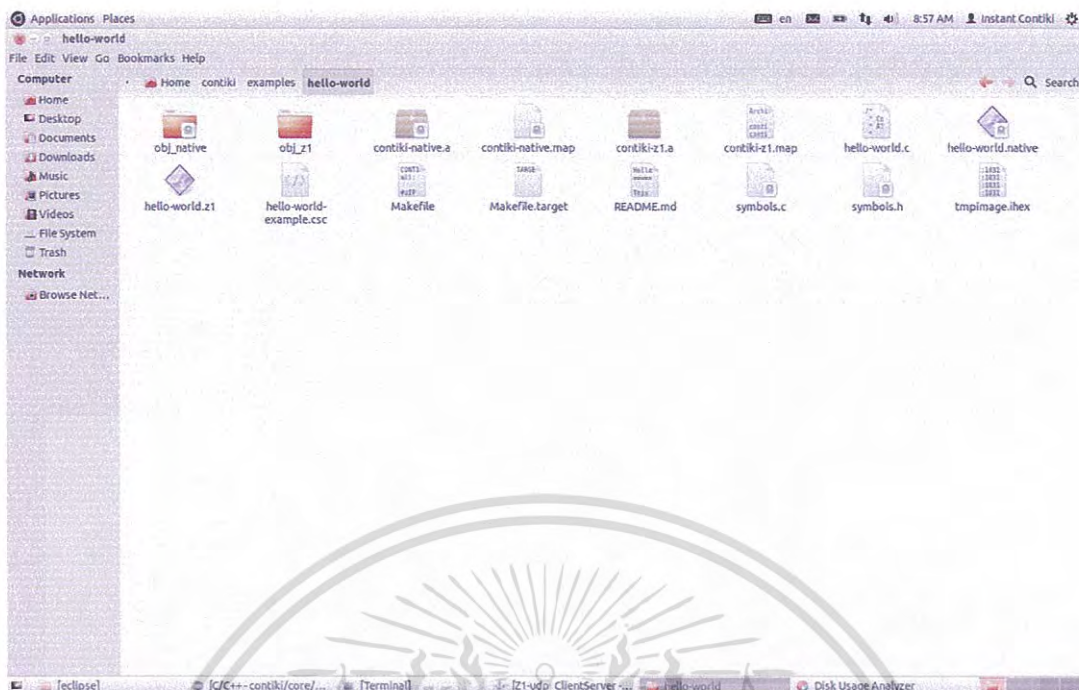
ตรวจสอบความพร้อม

- ตรวจสอบว่า ระบบปฏิบัติการ Contiki มีอยู่ใน Instant Contiki โดยระบบปฏิบัติการ Contiki จะอยู่ใน /Home และจะมีลักษณะโครงสร้างต่างๆ ดังรูปที่ 3.11



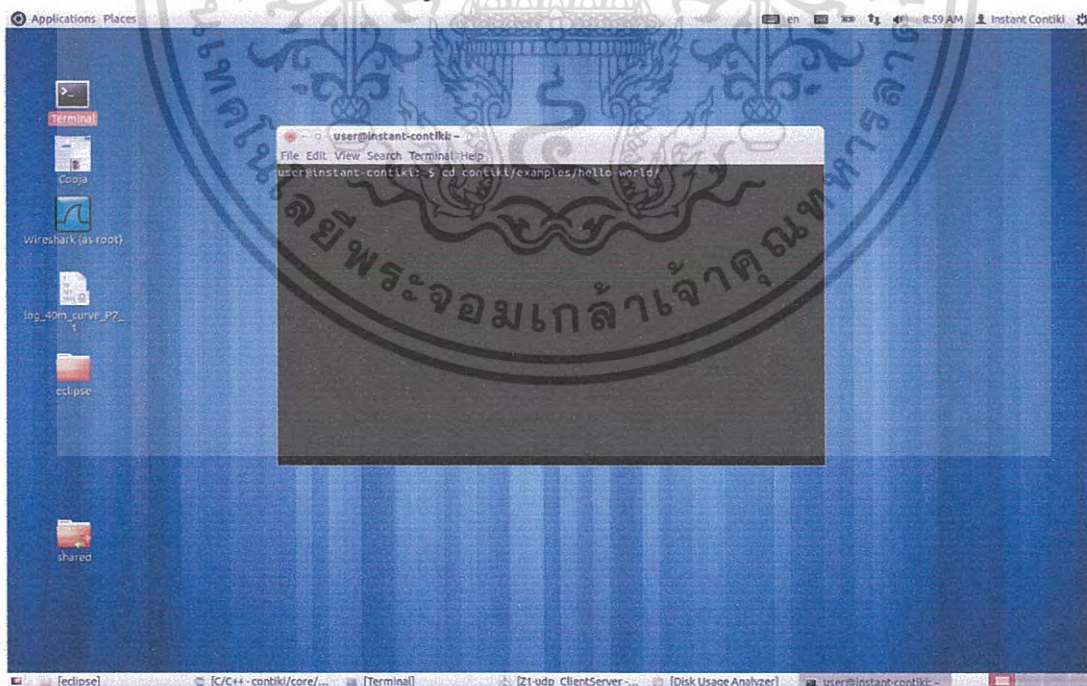
รูปที่ 3.11 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki

- ตรวจสอบโปรแกรมที่จะทำการติดตั้งลงบน Z1 ในที่นี้จะใช้ตัวโปรแกรม hello-world.c ที่มีให้ใน /Home/contiki/examples/hello-world โดยมีลักษณะโครงสร้างดังรูปที่ 3.12



รูปที่ 3.12 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki

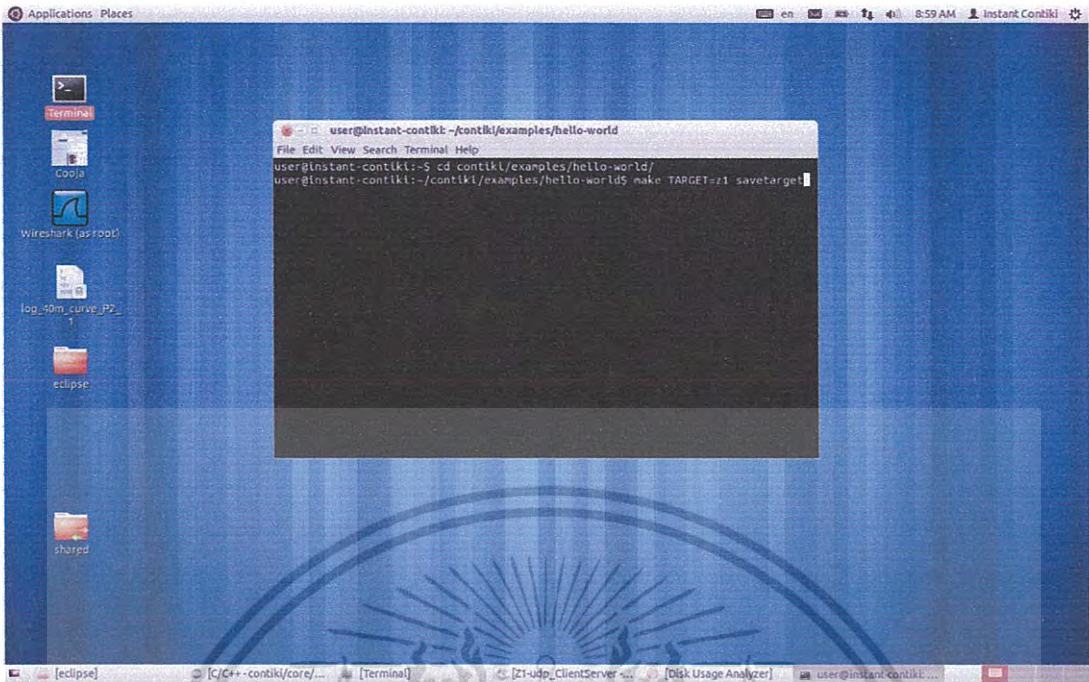
จากนั้นทำการลอง compile โปรแกรมดังกล่าว โดยไปที่ตำแหน่งของโปรแกรม
cd examples/hello-world ดังรูปที่ 3.13



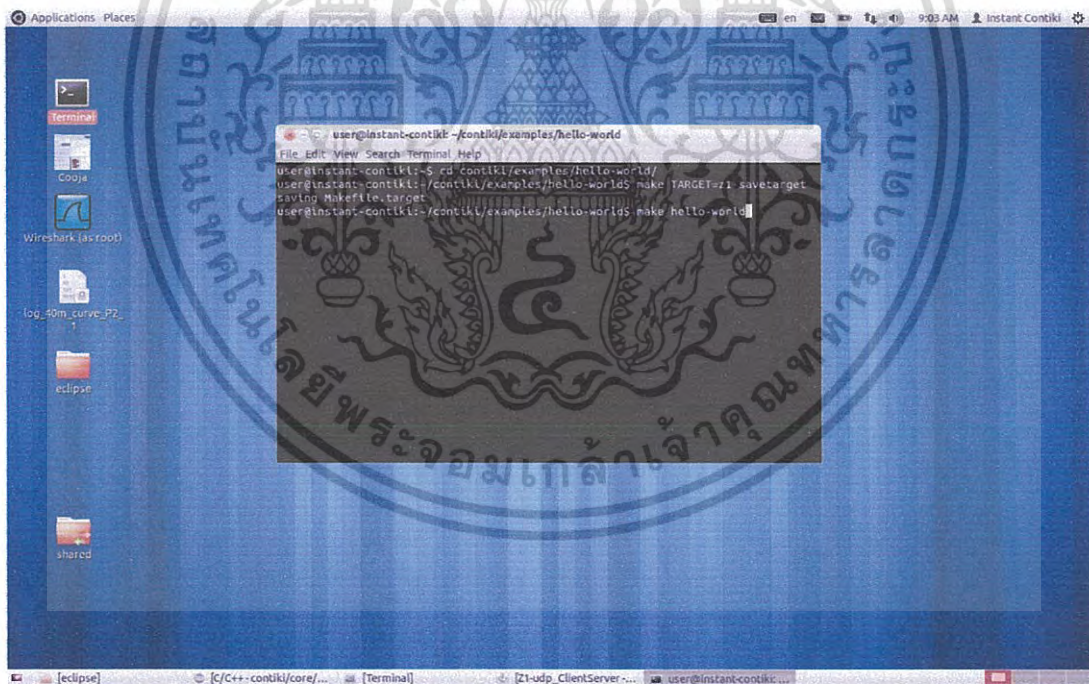
รูปที่ 3.13 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki

ตั้งค่าการ compile ไปที่ Z1 make TARGET=z1 savetarget ดังรูปที่ 3.14 – 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



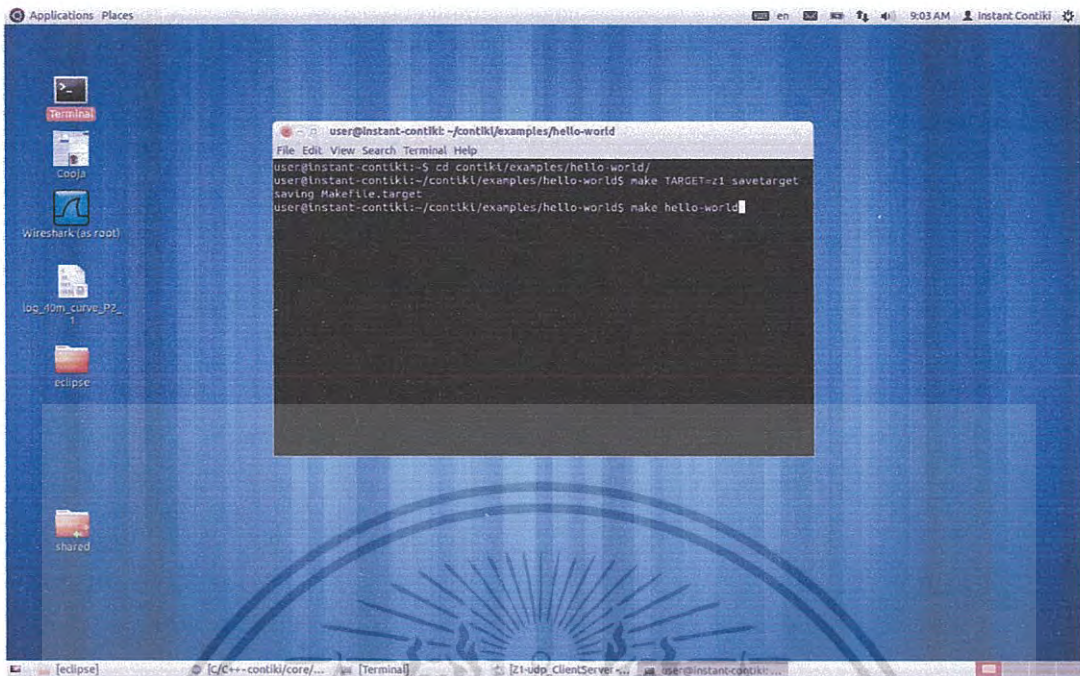
รูปที่ 3.14 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki



รูปที่ 3.15 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki

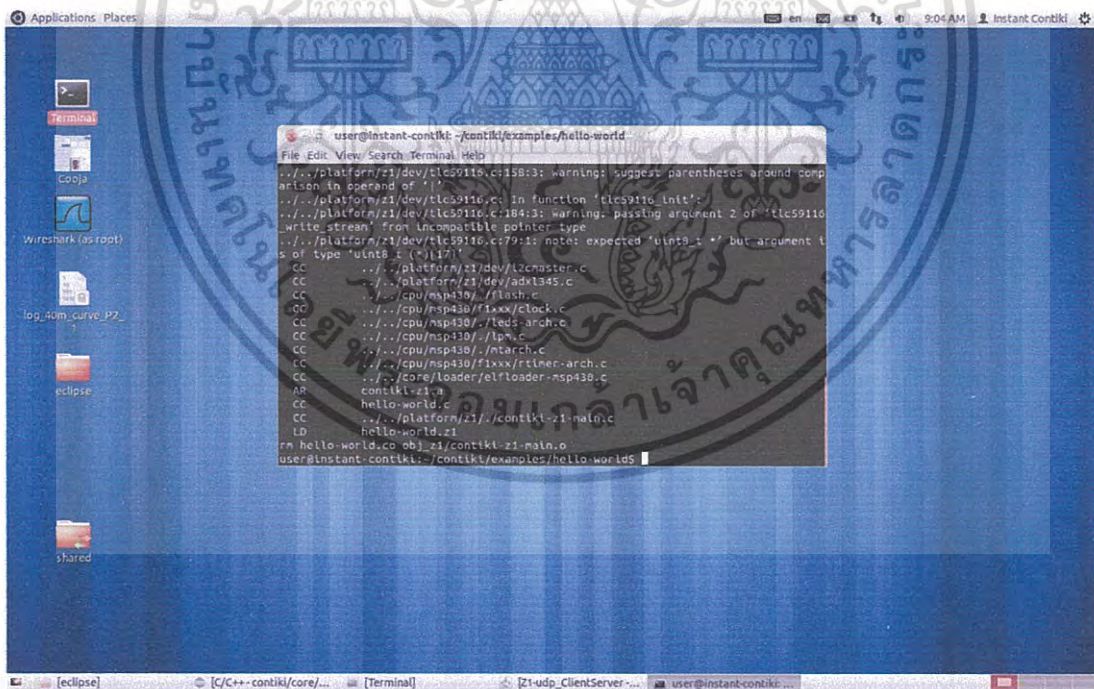
เริ่มทำการ compile ด้วยการพิมพ์คำสั่ง make hello-world ดังรูปที่ 3.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki

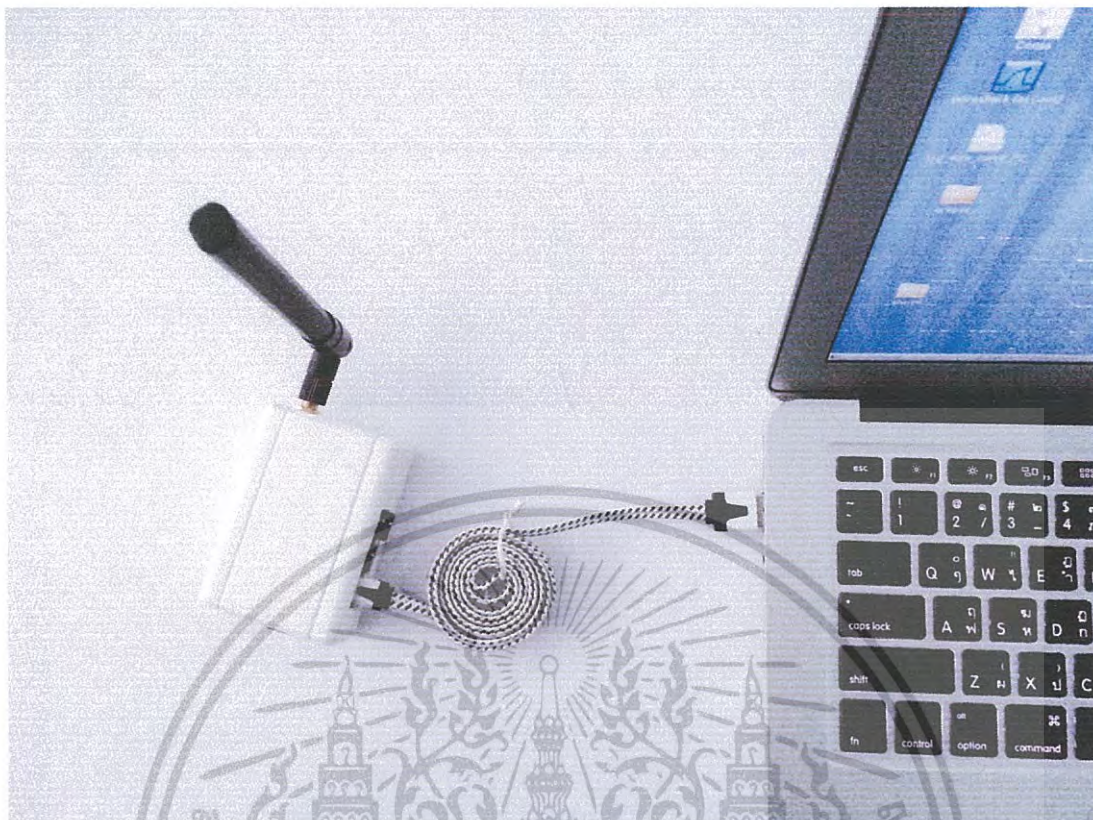
หาก compile คำนี้จะจะมีลักษณะดังรูปที่ 3.17



รูปที่ 3.17 ภาพโครงสร้าง OS ของ Contiki บน Instant Contiki

- ตรวจสอบการเชื่อมต่อระหว่าง Z1 กับคอมพิวเตอร์ที่ทำการติดตั้ง virtual machine และอัปโหลดโปรแกรม ทำการเสียบสาย micro-USB ระหว่างตัวอุปกรณ์ Z1 และคอมพิวเตอร์เพื่อเชื่อมต่อกับ virtual machine ดังรูปที่ 3.18

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเขียนขึ้นเพื่อการวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



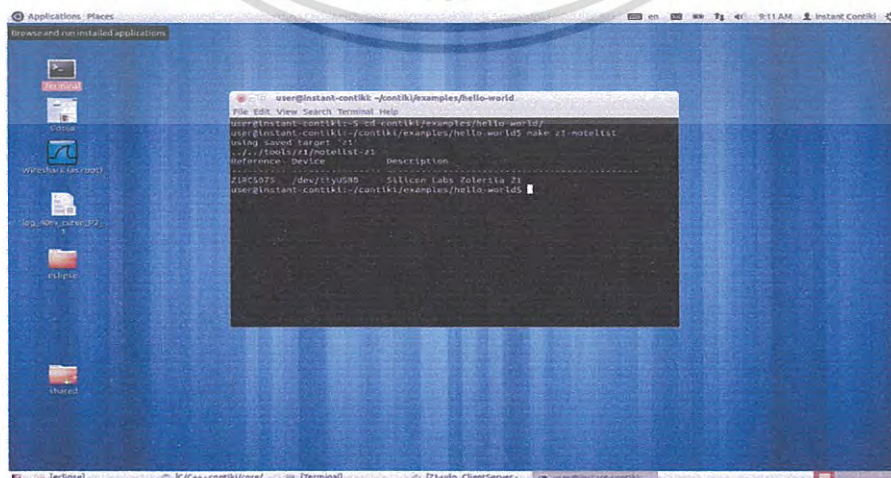
รูปที่ 3.18 การเชื่อมต่อระหว่าง Z1 กับ คอมพิวเตอร์

ใน VMplayer: Player → Removable Devices → Signal Integrated Zolertia Z1 → Connect

ใน VMWare Fusion : Devices → USB Devices → Silicon Labs Zolertia Z1

ใน VirtualBox → Devices → USB → Silicon Labs Zolertia Z1

จากนั้นทดสอบดูว่า อุปกรณ์ Z1 ได้เชื่อมต่อกับคอมพิวเตอร์แล้ว โดยการพิมพ์คำสั่ง
make z1-motelist



รูปที่ 3.19 ตรวจสอบการเชื่อมต่อ Z1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ฟรีโดยไม่คิดค่าใช้จ่าย แต่ขอสงวนสิทธิ์ในนโยบายด้านการใช้งาน
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากเชื่อมต่อสำเร็จจะมีข้อความขึ้นบอกว่า อุปกรณ์ Z1RCxxxx (ID) เชื่อมต่อมายังทาง USB0 จากนั้นลองทำการ อัปโหลดโปรแกรมไปยังอุปกรณ์ Z1 โดยการ

```
make hello-world.upload MOTES=/dev/ttyUSB0 && make z1-reset && make login
```

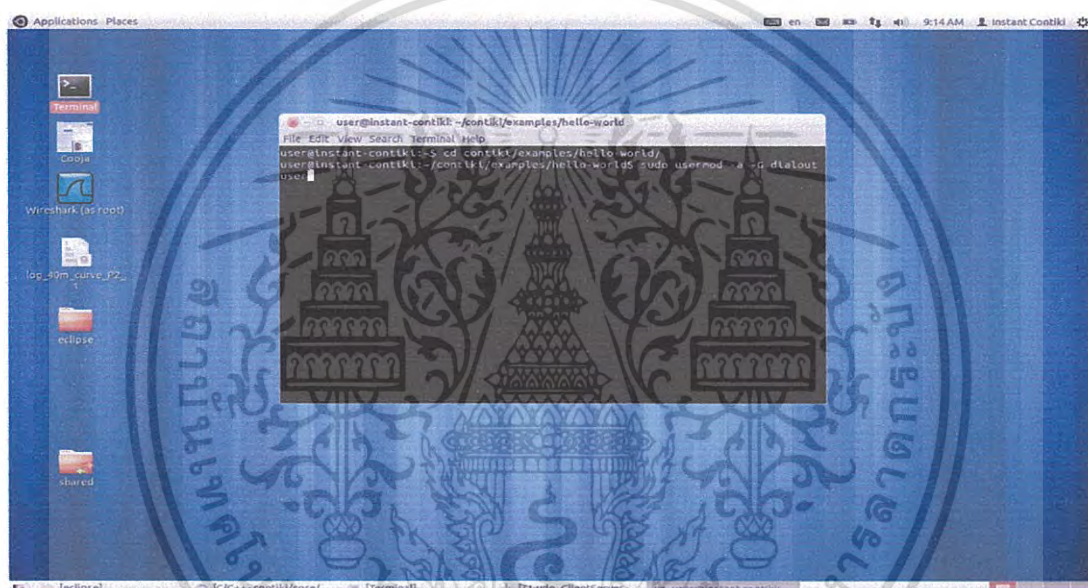
หากมีข้อความ Error ขึ้นมาว่า

```
serial.serialutil.SerialException: could not open port /dev/ttyUSB0: [Errno 13] Permission denied: '/dev/ttyUSB0'
```

ให้ทำการ เพิ่มตัวอุปกรณ์ลงไปยัง dialout group ของ linux โดยการพิมพ์คำสั่ง

```
sudo usermod -a -G dialout user
```

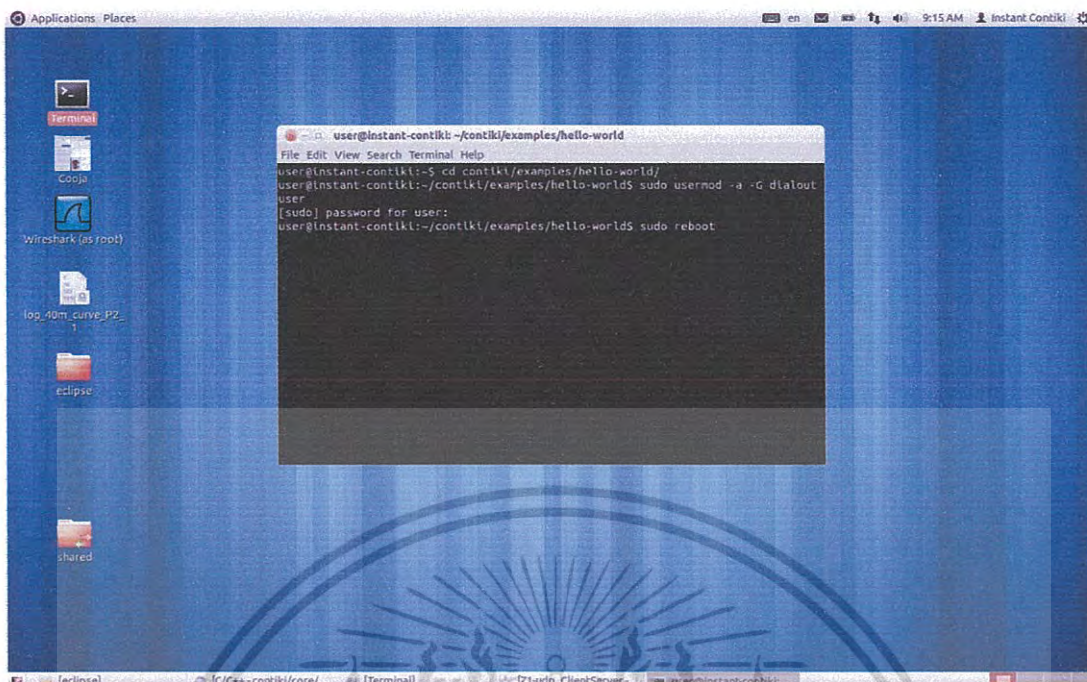
ดังรูปที่ 3.20



รูปที่ 3.20 ตรวจสอบการเชื่อมต่อ Z1

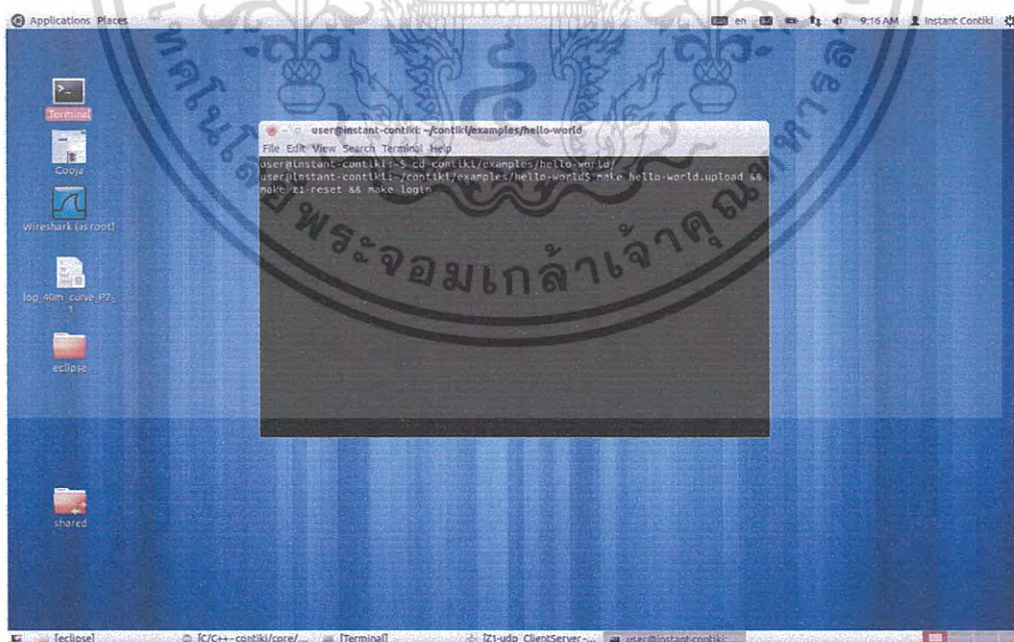
จากนั้นทำการ reboot virtual machine ด้วยการพิมพ์คำสั่ง `sudo reboot` ดังรูปที่ 3.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 การ reboot virtual machine

เมื่อทำการ reboot เสร็จแล้วให้ไปยัง examples/hello-world อีกครั้ง โดยพิมพ์คำสั่ง cd examples/hello-world จากนั้นลองทำการอัปเดตใหม่อีกครั้ง ดังรูปที่ 3.22



รูปที่ 3.22 การอัปเดตข้อมูล หลังจาก reboot virtual machine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลองและสรุป

ในการทดลองได้มีการแบ่งการรูปทดลอง 2 รูปแบบด้วยกัน คือ

- การทดลองใน Cooja Simulator ในส่วนของการทดลองนี้เพื่อเป็นการศึกษาลักษณะการทำงานของ RPL ว่าเมื่อมีการนำ node ที่เป็น parent ออกไป node ที่เป็น child จะมีการทำงานอย่างไร ได้แบ่งการทดลองเป็นอีก 2 แบบ คือ
 - เครือข่ายขนาด 6 node
 - เครือข่ายขนาด 10 node
- การทดลองกับอุปกรณ์ Z1 ในส่วนของการทดลองกับอุปกรณ์ได้มีการแบ่งการทดลองออกเป็น 2 รูปแบบ คือ
 - เครือข่ายขนาด 2 node : ในส่วนของเครือข่ายขนาดสอง โหนดนี้เป็นการศึกษาว่าผลกระทบของสัญญาณรบกวนภายในอาคาร เช่น 802.11 จะมีผลต่อการรับส่งข้อมูลอย่างไร
 - เครือข่ายขนาด 3 node : ในส่วนนี้เป็นการศึกษาว่าหากนำโหนดที่เป็น Parent ออกจากเครือข่าย Child จะมีการทำงานอย่างไรเพื่อรับรู้เส้นทางดังกล่าวไม่สามารถใช้งานได้แล้ว

4.1 การทดลองใน Cooja Simulator

4.1.1 อุปกรณ์

- เครื่องคอมพิวเตอร์ที่ติดตั้ง VMWare Player หรือ VirtualBox
- Instant Contiki Version 2.7 บนระบบปฏิบัติการ Ubuntu Version 12.04 LTS
- Cooja Simulator บน Instant Contiki Version 2.7
- Wireshark Version 1.7.2
- Eclipse IDE for C/C++ Developers Version Mars Release (4.5.0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 รูปแบบการทดลอง

ในการทดลองได้มีการออกแบบไว้ที่ 2 การทดลองคือ

- เครื่องข่ายขนาด 6 node และได้ทำการจำลองสร้างฮาร์ดแวร์แบบ Cooja Mote เป็น 3 ชนิด คือ
 - Cooja Mote Type #1 ทำหน้าที่เป็น Root node
 - Cooja Mote Type #2 ทำหน้าที่เป็น Sender node
 - Cooja Mote Type #3 ทำหน้าที่เป็น Reciever node
- เครื่องข่ายขนาด 10 node และได้ทำการจำลองสร้างฮาร์ดแวร์แบบ Cooja Mote เช่นเดียวกันเป็น 3 ชนิด คือ
 - Cooja Mote Type #1 ทำหน้าที่เป็น Root node
 - Cooja Mote Type #2 ทำหน้าที่เป็น Sender node
 - Cooja Mote Type #3 ทำหน้าที่เป็น Reciever node

สำหรับการทดลองที่ 1 เครื่องข่าย 6 โหนด และการทดลองที่ 2 เครื่องข่าย 10 โหนด โดยโหนดดังกล่าวไม่มีการเคลื่อนที่ และใช้ Simulation script เพื่อเป็นการจัดการนำโหนดออกตามเวลาที่กำหนด และใช้ Eclipse เพื่อเปิด โหมมดการ DEBUG เพื่อแสดง Message การติดต่อกันระหว่างโหนด สุดท้ายได้ทำการใช้ Wireshark เพื่อทำการวิเคราะห์ผลว่า เมื่อทำการนำโหนด ออกไปจะมีการทำงานของ Control Message จะเป็นอย่างไรบ้าง ในเบื้องต้นเพื่อที่จะนำไปใช้ในการคิดค้นการปรับปรุงการทำงานเพื่อปรับปรุงประสิทธิภาพของโปรโตคอล RPL ให้ดียิ่งขึ้น

4.1.3 การทดลองที่ 1 เครื่องข่าย 6 โหนดและผลการทดลอง

การทดลอง

ในการทดลองจะทำการกำหนดคุณลักษณะ ดังตารางที่ 4.1 – 4.3 และ Topology เป็นดังรูป 4.1.3(i) จากนั้นเริ่มทำการทดลองให้เปิดการทำงานของทุกโหนด จากนั้นเมื่อผ่านไปเป็นเวลา 5 นาทีให้ทำการ นำโหนดที่ 3 ออกจากเครือข่ายดังรูป 4.1.3(ii) แล้วบันทึกผล จากนั้นเมื่อเวลาผ่านไปอีก 5 นาทีให้ทำการนำ โหนดที่ 4 ดังรูป 4.1.3(iii) ออกจากเครือข่ายแล้วบันทึกผลการทำงานจนครบ 15 นาที

ตารางที่ 4.1 แสดงคุณลักษณะ Cooja Mote Type #1 (Root node)

No.	Item	Description
1	Identifier	mtype872
2	Description	RPL root
3	Contiki source	/home/user/contiki/regression-tests/12-rpl/code/root-node.c
4	Contiki firmware	/home/user/contiki/regression-tests/12-rpl/code/root-node.cooja
5	Compile commands	make root-node.cooja TARGET=cooja

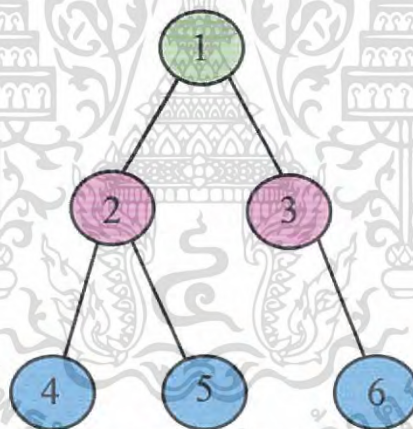
ตารางที่ 4.2 แสดงคุณสมบัติ Cooja Mote Type #2 (Sender node)

No.	Item	Description
1	Identifier	mtype530
2	Description	Sender_node
3	Contiki source	/home/user/contiki/regression-tests/12-rpl/code/sender-node.c
4	Contiki firmware	/home/user/contiki/regression-tests/12-rpl/code/sender-node.cooja
5	Compile commands	make sender-node.cooja TARGET=cooja

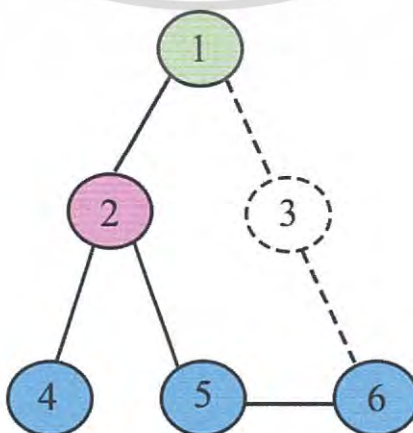
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 แสดงคุณสมบัติ Cooja Mote Type #3 (Receiver node)

No.	Item	Description
1	Identifier	mtype656
2	Description	Receiver_node
3	Contiki source	/home/user/contiki/regression-tests/12-rpl/code/receiver-node.c
4	Contiki firmware	/home/user/contiki/regression-tests/12-rpl/code/receiver-node.cooja
5	Compile commands	make receiver-node.cooja TARGET=cooja

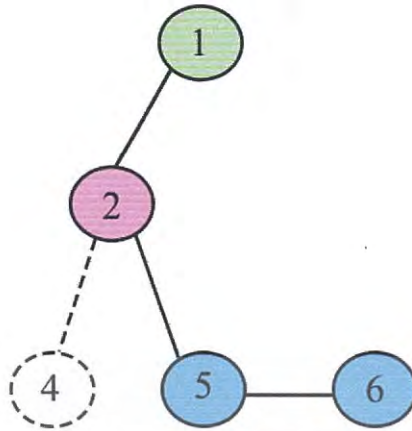


(i)



(ii)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

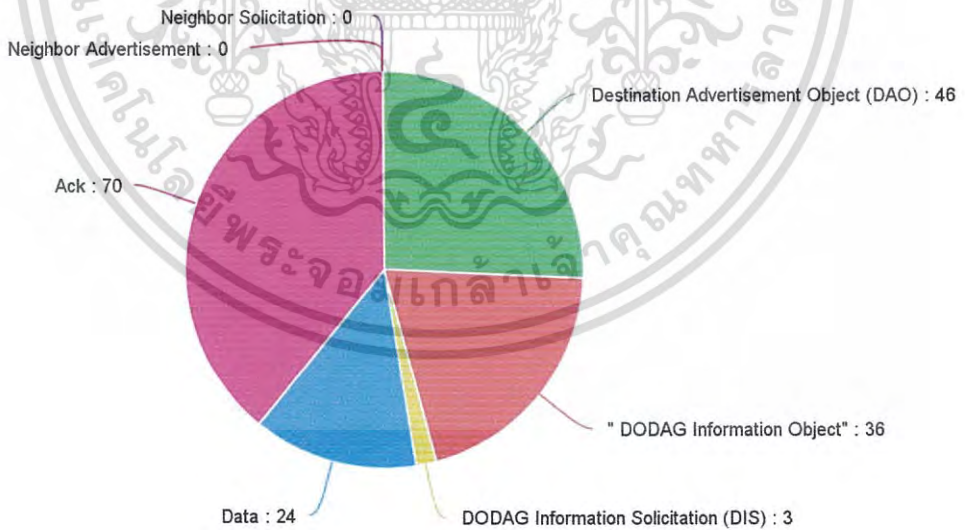


(iii)

รูปที่ 4.1 การทดลองที่ 1 เครือข่าย 6 โหนด

- (i) แสดงลักษณะของ Topology เริ่มต้น
- (ii) ลักษณะ Topology เมื่อนำโหนด 3 ออกเมื่อผ่านไป 5 นาที
- (iii) ลักษณะ Topology เมื่อนำโหนด 4 ออกเมื่อผ่านไปอีก 5 นาที

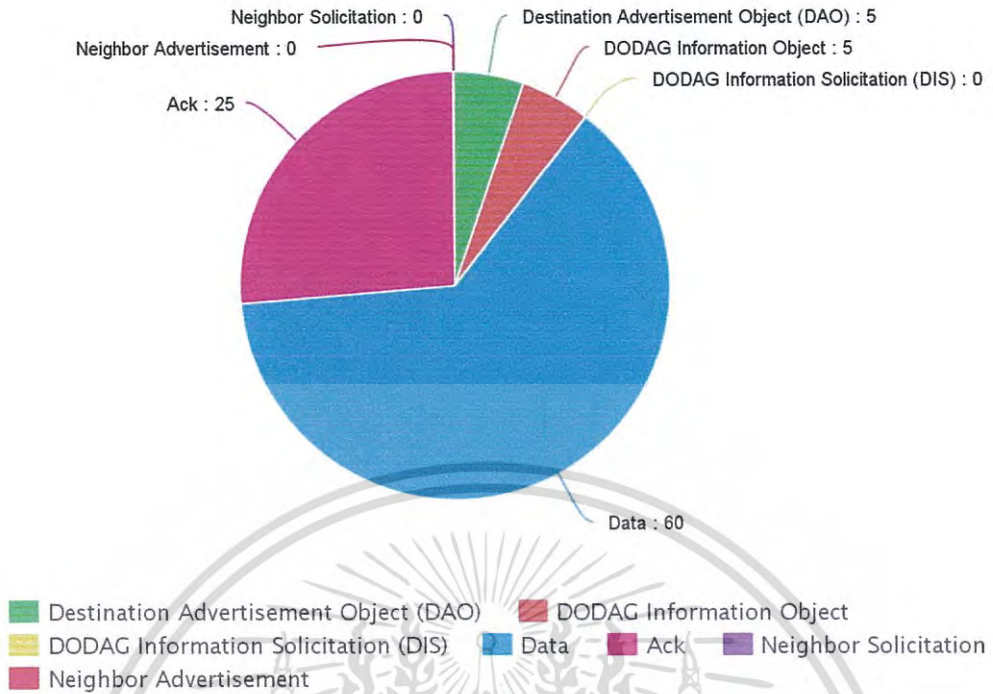
ผลการทดลอง



- Destination Advertisement Object (DAO)
- "DODAG Information Object"
- DODAG Information Solicitation (DIS)
- Data
- Ack
- Neighbor Solicitation
- Neighbor Advertisement

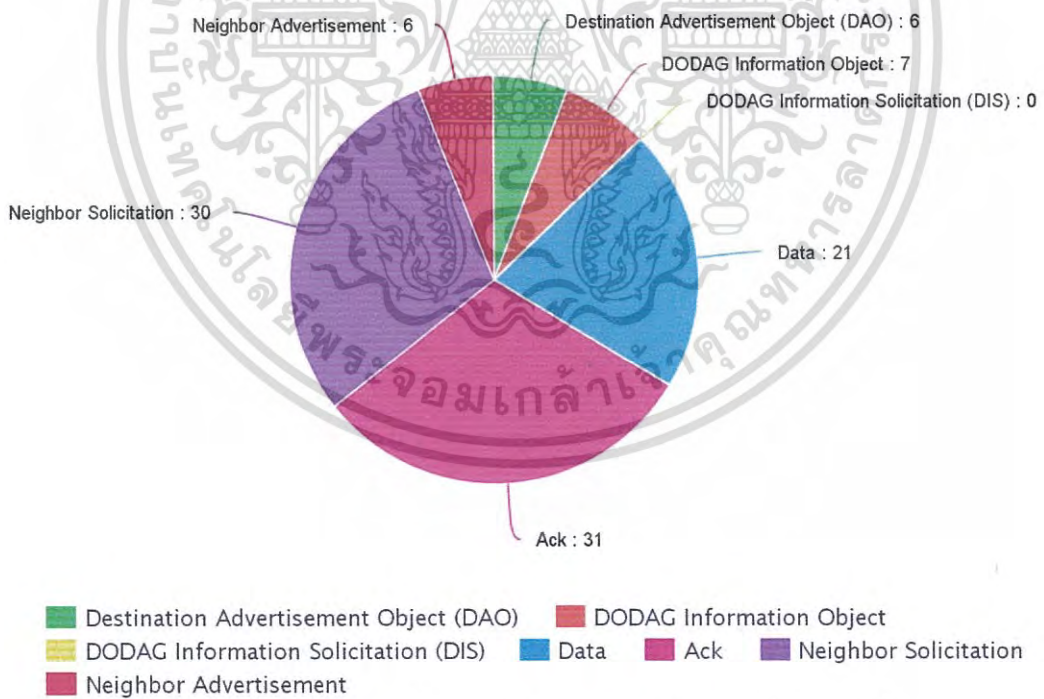
รูปที่ 4.2 ผลการทดลองเรื่องการส่งข้อมูลเพื่อก่ตของการทดลองที่ 1 เมื่อครบ 5 นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 1

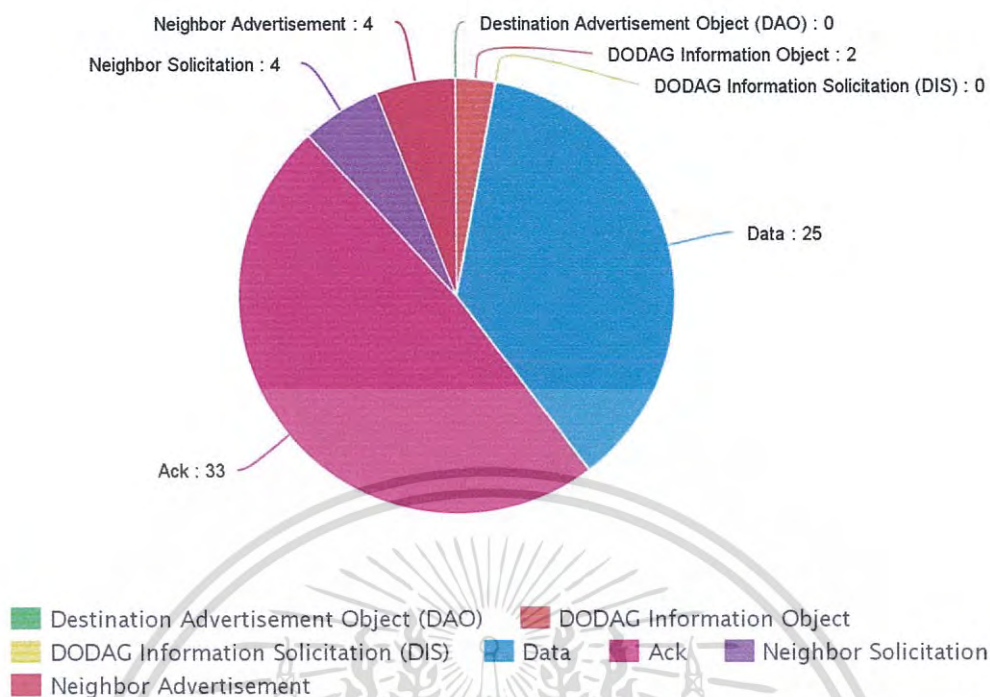
Remove mote 3 ในนาที่ที่ 5



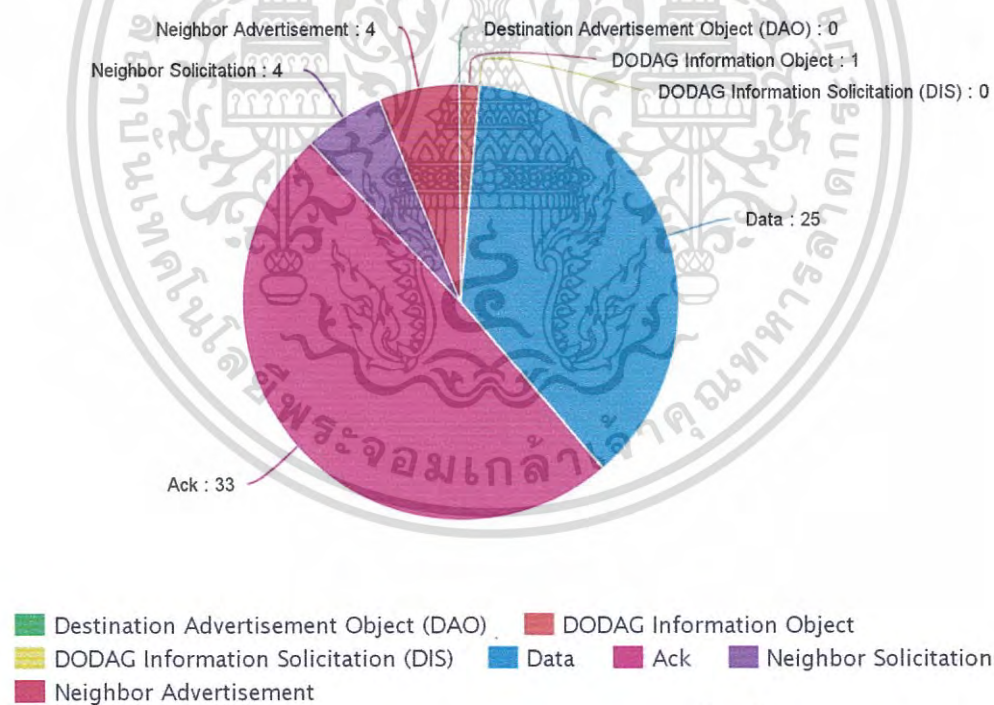
รูปที่ 4.4 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 1

Remove mote 4 ในนาที่ที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 1 เมื่อครบ 20 นาที



รูปที่ 4.6 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 1 เมื่อเวลาผ่านไปอีก 5 นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

จากการทดลองที่ 1 ในช่วงของการทดลองขั้นตอนแรกนั้นจะเห็นว่าการส่งแพ็กเก็ตเกิดประเภท Destination Advertisement Object (DAO), DODAG Information Object (DIO) บนเครือข่ายมีจำนวนมากเนื่องจากในช่วงนี้จะเป็นการสร้าง topology ทำให้มีการส่งแพ็กเก็ตเกิดดังกล่าวเป็นจำนวนมาก เนื่องจากจำเป็นที่จะต้องแลกเปลี่ยนข้อมูลซึ่งกันและกันเพื่อใช้ในการสร้าง topology จากนั้นในขั้นตอนที่ 2 จะเห็นว่า Data แพ็กเก็ตเกิดนั้นมีการส่งบนเครือข่ายเป็นจำนวนมากเป็นเพราะในช่วงนี้ได้มีการ remove mote 3 ออกจากเครือข่ายออกไปทำให้ mote 6 ซึ่งเป็น parent ของ mote 6 นั้นไม่สามารถส่ง Data แพ็กเก็ตเกิดไปยัง root mote ได้ จึงทำให้มีการส่งซ้ำอยู่เรื่อยๆ ทำให้ในช่วงขั้นตอนดังกล่าวมี Data เกิดขึ้นสูงเป็นพิเศษจากนั้นในช่วงขั้นตอนที่ 3 จะเห็นว่าการส่ง Neighbor Solicitation สูงที่สุดเป็นเพราะ mote 6 นั้นยังไม่สามารถกลับเข้าสู่เครือข่ายได้จึงมีการส่ง Neighbor Solicitation หา mote 3 ซ้ำเป็นจำนวนมากและเมื่อ mote 5 ทำการกระจาย DODAG Information Object (DIO) ในขั้นตอนนี้ก็จะทำให้ mote 6 นั้นสามารถกลับเข้ามายังเครือข่ายได้ซึ่งกินเวลาประมาณ 8 นาทีเลยทีเดียวแต่หาก mote ที่ถูก remove นั้นไม่ได้เป็น parent ของ mote ใดก็จะไม่กระทบต่อเครือข่ายดังในขั้นตอนที่ 4 ซึ่งจะเห็นว่าการส่งแพ็กเก็ตเกิดใกล้เคียงกับขั้นตอนสุดท้ายที่มีลักษณะ topology คงที่แล้ว

ตารางที่ 4.4 สรุปผลการทดลองที่ 1

การทดลอง	DAO	DIO	DIS	DATA	ACK	Neighbor Solicitation	Neighbor Advertisement
เมื่อครบ 5 นาที	46	36	3	24	70	0	0
remove mote 3 ในนาทีที่ 5	5	5	0	60	25	0	0
remove mote 4 ในนาทีที่ 10	6	7	0	21	31	30	6
เมื่อครบ 20 นาที	0	2	0	25	33	4	4
เมื่อปล่อยไว้อีก 5 นาที	0	1	0	25	33	4	4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 การทดลองที่ 2 เครื่องข่าย 10 โหนดและผลการทดลอง

การทดลอง

ในการทดลองจะทำการกำหนดคุณลักษณะ ดังตารางที่ 4.4 – 4.6 และ Topology เป็นดังรูป 4.1.4(i) จากนั้นเริ่มทำการทดลองให้เปิดการทำงานของทุกโหนด จากนั้นเมื่อผ่านไปเป็นเวลา 5 นาทีให้ทำการ นำโหนดที่ 3 ออกจากเครือข่ายดังรูป 4.1.4(ii) แล้วบันทึกผล จากนั้นเมื่อเวลาผ่านไปอีก 5 นาทีให้ทำการนำ โหนดที่ 4 ดังรูป 4.1.4(iii) ออกจากเครือข่ายแล้วบันทึกผล สุดท้ายเมื่อผ่านไปอีก 5 นาทีให้ทำการ นำโหนดที่ 9 ออกจากเครือข่ายดังรูป 4.1.4(iv) จนครบ 20 นาที

ตารางที่ 4.5 แสดงคุณลักษณะ Cooja Mote Type #1 (RPL root node)

No.	Item	Description
1	Identifier	mtype872
2	Description	RPL root
3	Contiki source	/home/user/contiki/regression-tests/12-rpl/code/root-node.c
4	Contiki firmware	/home/user/contiki/regression-tests/12-rpl/code/root-node.cooja
5	Compile commands	make root-node.coojaTARGET=cooja

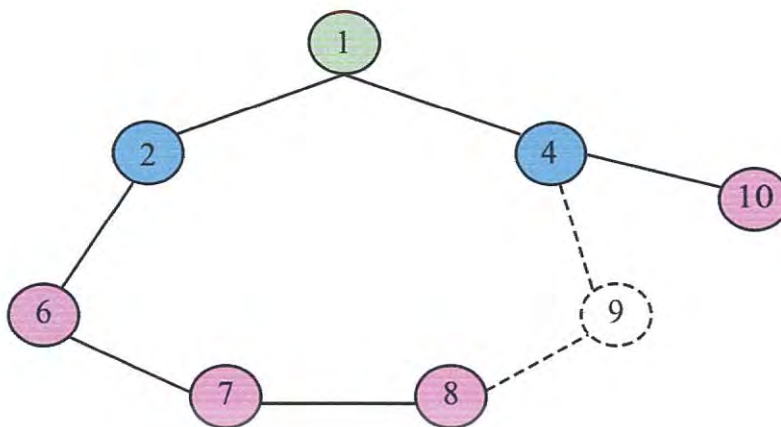
ตารางที่ 4.6 แสดงคุณลักษณะ Cooja Mote Type #2 (Sender node)

No.	Item	Description
1	Identifier	mtype530
2	Description	Sender_node
3	Contiki source	/home/user/contiki/regression-tests/12-rpl/code/sender-node.c
4	Contiki firmware	/home/user/contiki/regression-tests/12-rpl/code/sender-node.cooja
5	Compile commands	make sender-node.cooja TARGET=cooja

ตารางที่ 4.7 แสดงคุณลักษณะ Cooja Mote Type #3 (Receiver node)

No.	Item	Description
1	Identifier	mtype656
2	Description	Receiver_node
3	Contiki source	/home/user/contiki/regression-tests/12-rpl/code/receiver-node.c
4	Contiki firmware	/home/user/contiki/regression-tests/12-rpl/code/receiver-node.cooja
5	Compile commands	make receiver-node.cooja TARGET=cooja

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

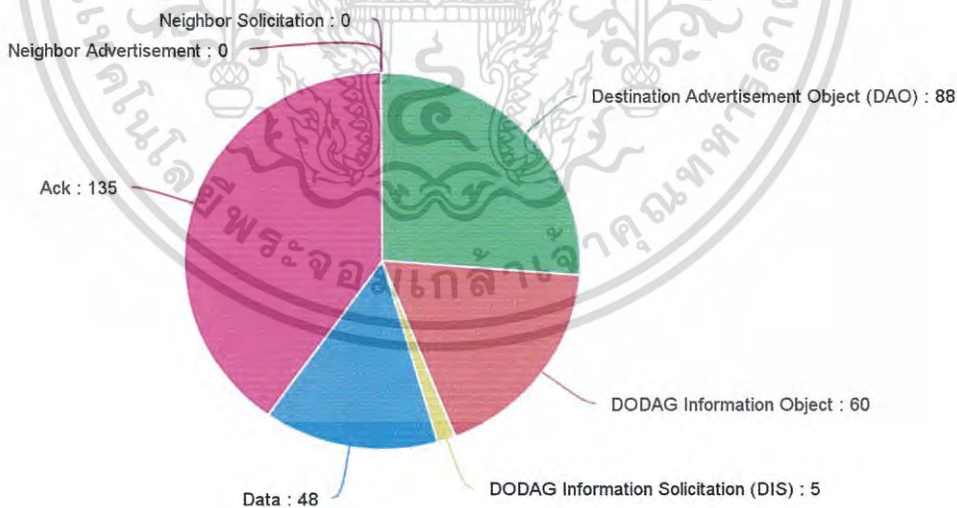


(iv)

รูปที่ 4.7 การทดลองที่ 2 เครื่องข่าย 10 โหนด

- (i) แสดงลักษณะของ Topology เริ่มต้น
- (ii) ลักษณะ Topology เมื่อนำโหนด 3 ออกเมื่อผ่านไป 5 นาที
- (iii) ลักษณะ Topology เมื่อนำโหนด 5 ออกเมื่อผ่านไป 10 นาที
- (iv) ลักษณะ Topology เมื่อนำโหนด 9 ออกเมื่อผ่านไป 15 นาที

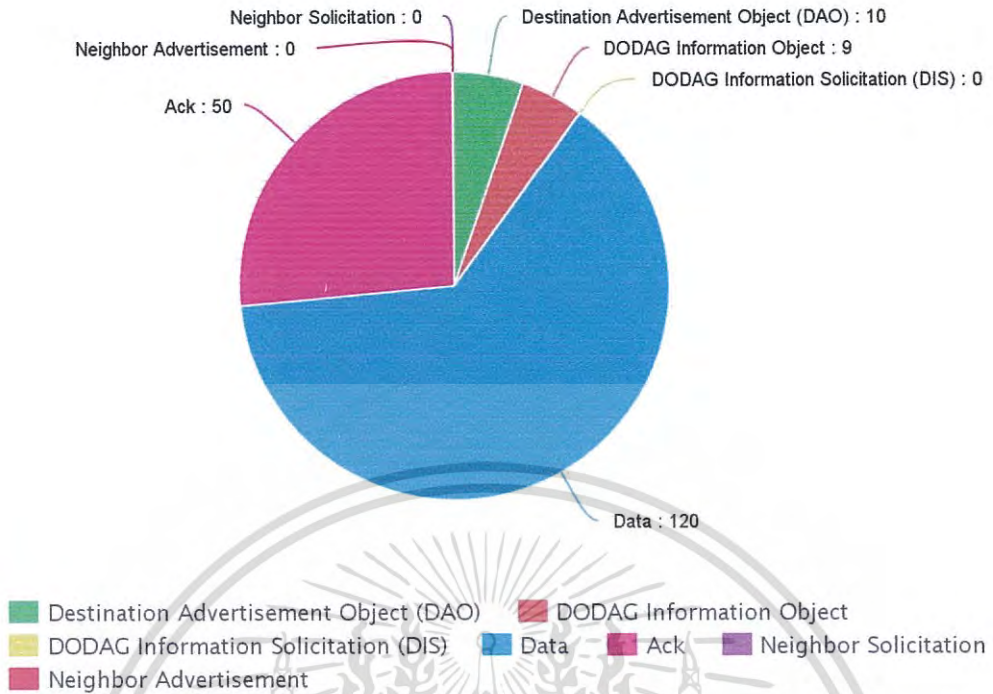
● ผลการทดลอง



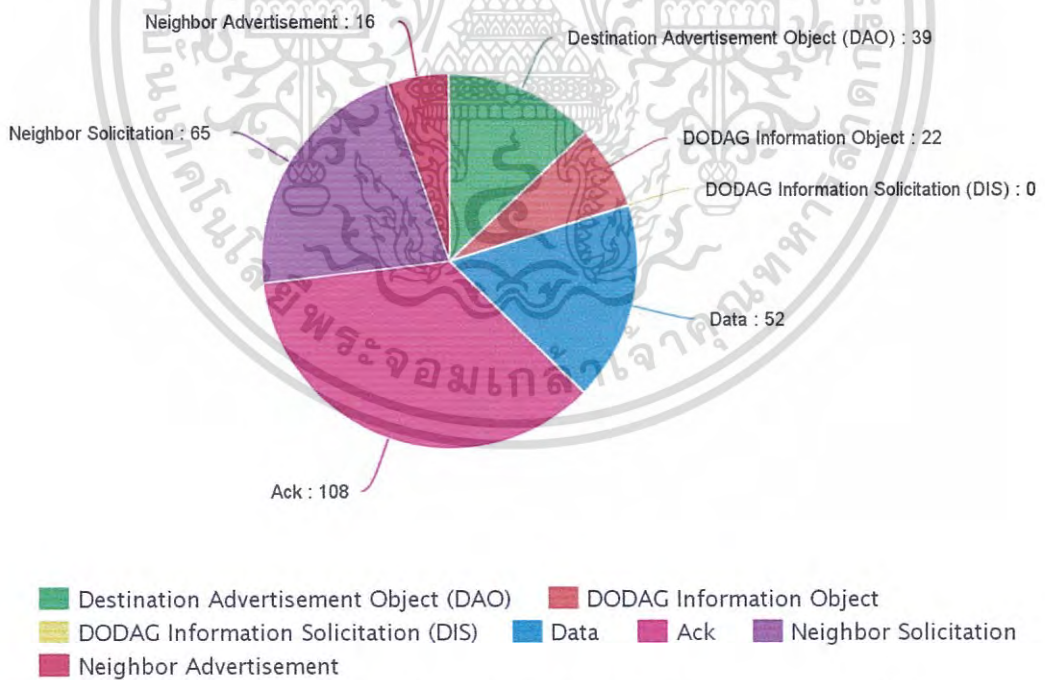
- Destination Advertisement Object (DAO)
- DODAG Information Object
- DODAG Information Solicitation (DIS)
- Data
- Ack
- Neighbor Solicitation
- Neighbor Advertisement

รูปที่ 4.8 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 2 เมื่อครบ 5 นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

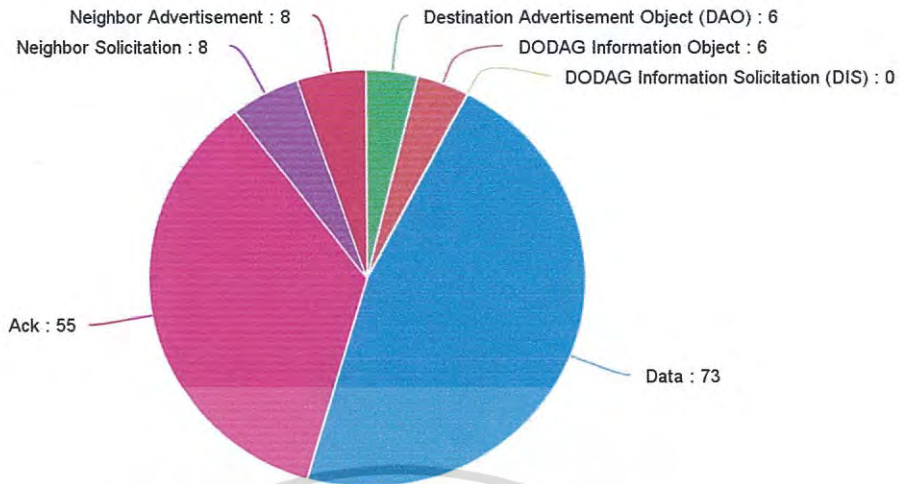


รูปที่ 4.9 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 2
remove mote 3 ในนาที่ที่ 5

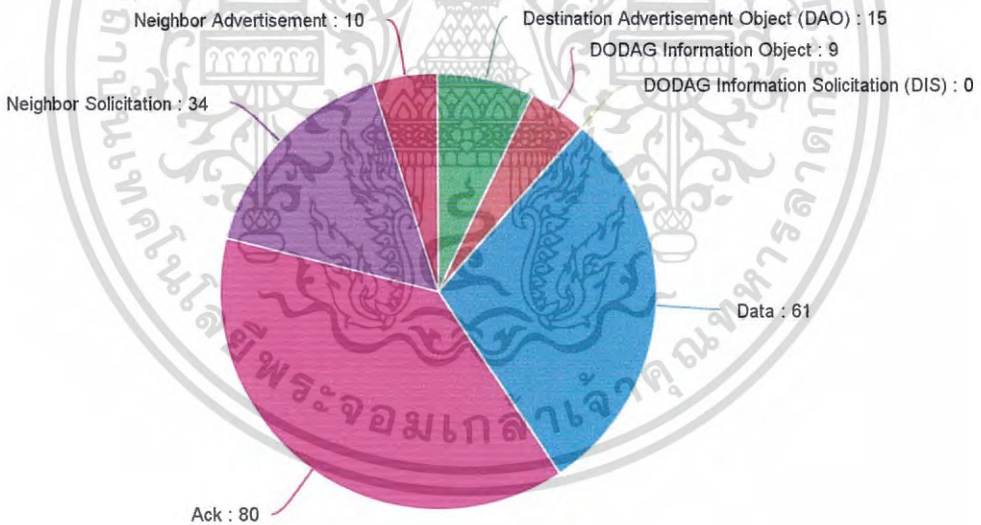


รูปที่ 4.10 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 2
remove mote 5 ในนาที่ที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

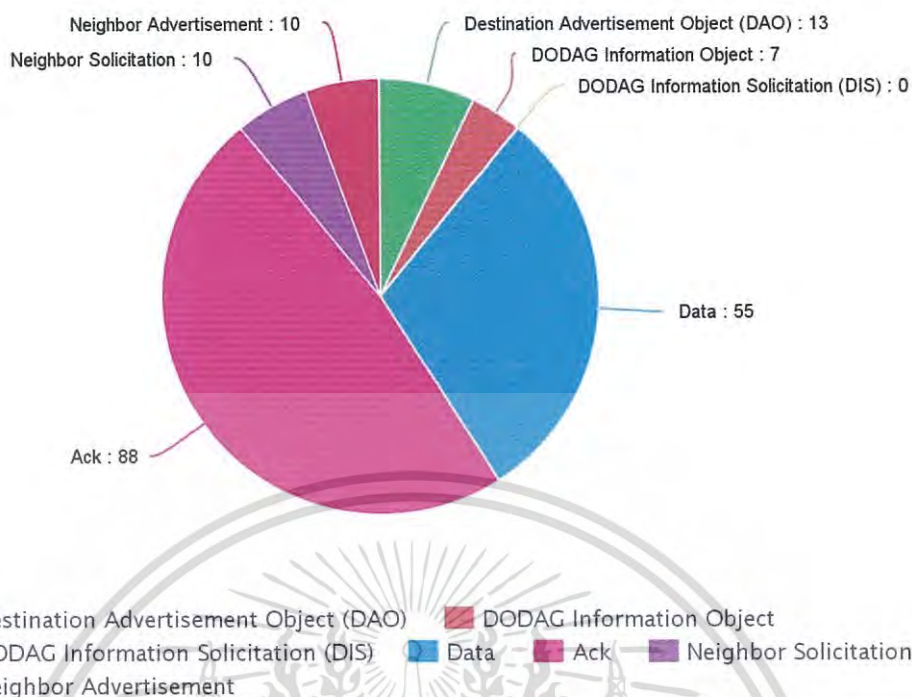


รูปที่ 4.11 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 2 remove mote 9 เมื่อครบ ในนาทีที่ 15



รูปที่ 4.12 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตเกิดของการทดลองที่ 2 เมื่อครบ 25 นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ผลการทดลองเรื่องการส่งข้อมูลแพ็กเก็ตของการทดลองที่ 2 เมื่อปล่อยไว้อีก 5 นาที

สรุปผลการทดลอง

จากการทดลองที่ 2 จะมีความลักษณะการส่งของแพ็กเก็ตคล้ายๆกันคือ ในช่วงของการทดลองขั้นตอนแรกนั้นจะเห็นว่ามี การส่งแพ็กเก็ตประเภท Destination Advertisement Object (DAO), DODAG Information Object (DIO) บนเครือข่ายมีจำนวนมากเนื่องจากในช่วงนี้จะเป็นการ สร้าง topology ทำให้มีการส่งแพ็กเก็ตดังกล่าวเป็นจำนวนมาก เช่นเดียวกันเนื่องจากจำเป็นที่ จะต้องแลกเปลี่ยนข้อมูลซึ่งกันและกันเพื่อใช้ในการสร้าง topology จากนั้นในขั้นตอนที่ 2 จะเห็น ว่า Data แพ็กเก็ตนั้นมีการส่งบนเครือข่ายเป็นจำนวนมากเป็นเพราะในช่วงนี้ได้มีการ remove mote 3 ออกจากเครือข่ายออกไปทำให้ mote 7 และ mote 8 ซึ่ง mote 3 เป็น parent ของ mote 6 นั้น ไม่สามารถส่ง Data แพ็กเก็ตไปยัง root mote ได้ จึงทำให้มีการส่งซ้ำอยู่เรื่อยๆทำให้ในช่วงขั้นตอน ดังกล่าวมี Data เกิดขึ้นสูงเป็นพิเศษ จากนั้นในช่วงขั้นตอนที่ 3 จะเห็นว่ามี การส่ง Neighbor Solicitation สูงที่สุดเป็นเพราะ mote 7 และ mote 8 นั้นยังไม่สามารถกลับเข้าสู่เครือข่ายได้จึงมีการ ส่ง Neighbor Solicitation ห้า mote 3 ซ้ำเป็นจำนวนมากและเมื่อ mote 6 ทำการกระจาย DODAG Information Object(DIO) มายัง mote 7 และ mote 9 ทำการกระจาย DODAG Information Object(DIO) มายัง mote 8 เช่นเดียวกัน จากนั้นทั้งคู่จะตัดสินใจเลือกเส้นทางกัน ในขั้นตอนนี้ก็จะ ทำให้ mote 7 และ mote 8 นั้นสามารถกลับเข้ามายังเครือข่ายได้ซึ่งจะอยู่ในช่วงขั้นตอนที่ 3-4 และ จากนั้นในขั้นตอนที่ 4 มีการ remove mote 9 ออกซึ่งเป็น parent ของ mote 8 ทำให้แพ็กเก็ต Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ Neighbor Solicitation สูงขึ้นมาเช่นเดียวกัน แต่หาก mote ที่ถูก remove นั้นไม่ได้เป็น parent ของ mote ใดก็จะไม่กระทบต่อเครือข่ายด้วยการ remove mote 5

ตารางที่ 4.8 สรุปผลการทดลองที่ 2

การทดลอง	DAO	DIO	DIS	DATA	ACK	Neighbor Solicitation	Neighbor Advertisement
เมื่อครบ 5 นาที	88	60	5	48	135	0	0
remove mote 3 ในนาทีที่ 5	10	9	0	120	50	0	0
remove mote 5 ในนาทีที่ 10	39	22	0	52	108	65	16
remove mote 9 ในนาทีที่ 15	6	6	0	73	55	8	8
เมื่อครบ 25 นาที	15	9	0	61	80	34	10
เมื่อทดสอบอีก 5 นาที	13	7	0	55	88	10	10

4.1.5 สรุปการทดลองใน Cooja Simulator

โดยสรุปแล้วการทดลองบน Cooja Simulator นั้นเป็นการทดลองเบื้องต้นว่าหลักการการทำงานของ RPL นั้นมีการทำงานอย่างไรเมื่อมีมีโหนดหายไปซึ่งก็พบปัญหาของการ Simulation อยู่คือ ปัญหาในเรื่องของเวลาเนื่องจากแต่ละอุปกรณ์นั้นเวลาไม่สอดคล้องกันเช่น อุปกรณ์ Cooja และ อุปกรณ์ Z1 หากเมื่อนำอุปกรณ์มาทำการจำลองแล้วทดสอบจะพบว่าอัตราการรับส่งข้อมูลของ Cooja นั้นจะเร็วกว่าเนื่องจากอุปกรณ์ Cooja นั้น ทางผู้พัฒนา Cooja Simulator ได้ทำการสร้างขึ้นมาซึ่งไม่ได้มีอุปกรณ์นั้นจริงๆ ซึ่งทำให้เราสามารถปรับแต่งรูปแบบของอุปกรณ์นั้นได้ จึงทำให้มีความรวดเร็วกว่าทำให้เกิดปัญหาที่เวลาในของโปรแกรม Simulation นั้นไม่ถูกต้องและเชื่อถือไม่ได้ซึ่งต้องดูจาก Log การส่งของข้อมูล ต่อมาปัญหาเรื่องอัตราการรับส่งข้อมูลจากตารางสรุปผลการทดลองข้างต้นจะเห็นว่าข้อมูลมีในปริมาณที่น้อยมากเนื่องจากการตั้ง Application ของการส่งข้อมูลในทุกๆรอบเวลาที่ 1 นาที กล่าวคือ ทุกๆ 1 นาทีจะมี packet ส่งออก 1

packet บวกกับ ระบบปฏิบัติการ Contiki นั้น ได้มีการนำ ContikiMAC Radio Duty Cycling Protocol มาใช้ด้วยคือ ในช่วงรอบเวลาของการส่ง packet นั้นมีการส่งไปหลาย packet ในช่วงเวลาดังกล่าว ทำให้เมื่อตรวจสอบผลการทดลองจึงมี packet ที่ซ้ำกันซึ่งไม่ตรงกับหลักของ UDP คือจะไม่มีการส่งซ้ำประกอบกับ ContikiMAC Radio Duty Cycling Protocol นั้น ไม่ได้อยู่บน Network Layer ทำให้ผู้วิจัยติดปัญหาดังกล่าว ทำให้ผู้วิจัยได้นำไปสู่การทดลองในอุปกรณ์ Z1 และทดลองในเรื่องของการกวนของสัญญาณและการนำโหนดออกจากเครือข่ายในต่อไป

4.2 การทดลองกับอุปกรณ์ Z1

4.2.1 อุปกรณ์

- เครื่องคอมพิวเตอร์ที่ติดตั้ง VMWare Player หรือ VirtualBox
- Instant Contiki Version 2.7 บนระบบปฏิบัติการ Ubuntu Version 12.04 LTS
- Cooja Simulator ที่มาบน Instant Contiki Version 2.7
- Wireshark Version 1.7.2
- อุปกรณ์ Z1 Zolertia

4.2.2 รูปแบบการทดลอง

การทดลองได้มีการออกแบบไว้ 2 รูปแบบคือ

- แบบ 2 โหนด จะมีการแบ่งรูปแบบการทดลองออกเป็น 2 กรณี คือ
 - แบบที่มีการกวนของสัญญาณภายนอกต่ำ คือใช้ช่องสัญญาณที่ 26 และแบ่งออกเป็นอีก 3 กรณีคือ
 - อัตราการส่งของข้อมูล 10s ต่อ 1 แพ็กเก็ต จำนวน 5 การทดลอง ระยะเวลาแต่ละการทดลอง 15 นาที
 - อัตราการส่งของข้อมูล 5s ต่อ 1 แพ็กเก็ต จำนวน 5 การทดลอง ระยะเวลาแต่ละการทดลอง 15 นาที
 - อัตราการส่งของข้อมูล 1s ต่อ 1 แพ็กเก็ต จำนวน 5 การทดลอง ระยะเวลาแต่ละการทดลอง 15 นาที
 - แบบที่มีการกวนของสัญญาณภายนอกสูง คือใช้ช่องสัญญาณที่ 22 และแบ่งออกเป็นอีก 3 กรณีคือ
 - อัตราการส่งของข้อมูล 10s ต่อ 1 แพ็กเก็ต จำนวน 3 การทดลอง ระยะเวลาแต่ละการทดลอง 15 นาที
 - อัตราการส่งของข้อมูล 5s ต่อ 1 แพ็กเก็ต จำนวน 3 การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ทดลอง ระยะเวลาแต่ละการทดลอง 15 นาที โดยที่ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

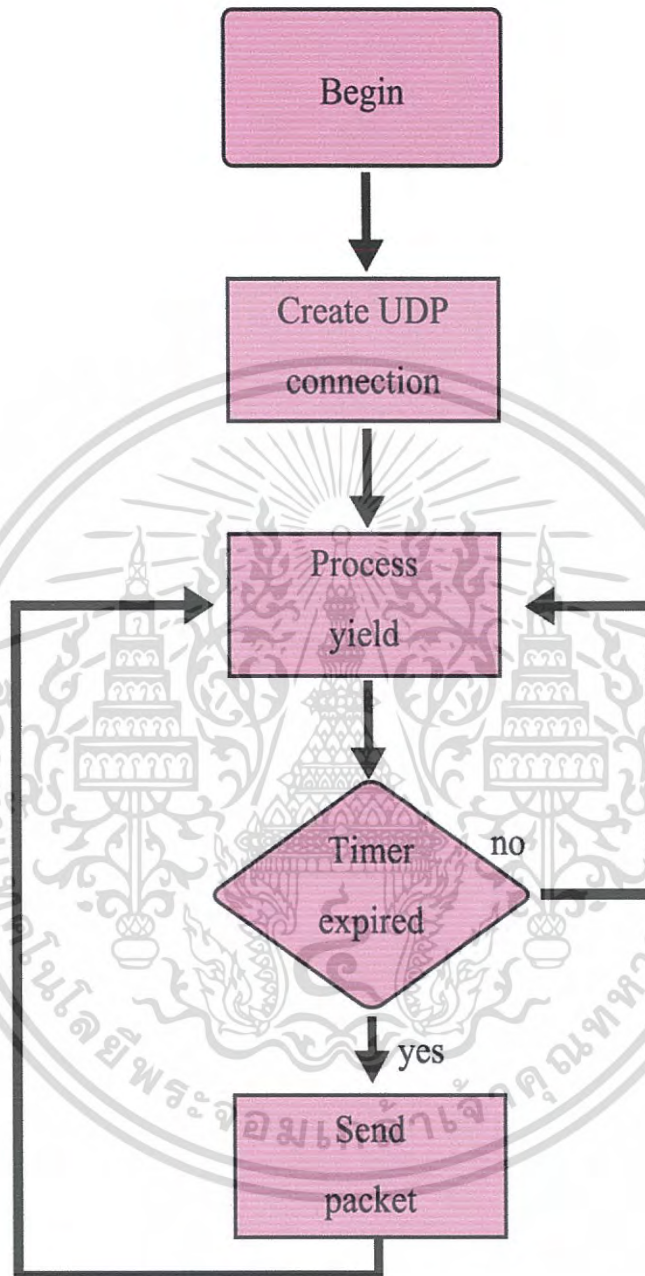
- อัตราการส่งของข้อมูล 1s ต่อ 1 แพ็กเก็ต จำนวน 3 การทดลอง ระยะเวลาแต่ละการทดลอง 15 นาที
- แบบ 3 โหนด จะใช้ช่องสัญญาณที่ 26 ในการรับส่งข้อมูล และแบ่งออกเป็น 3 กรณี คือ
 - อัตราการส่งของข้อมูลสำหรับ โหนดที่ 3 10s ต่อ 1 แพ็กเก็ต จำนวน 5 การทดลอง ระยะเวลาแต่ละการทดลอง 15 นาที
 - อัตราการส่งของข้อมูลสำหรับ โหนดที่ 3 5s ต่อ 1 แพ็กเก็ต จำนวน 5 การทดลอง ระยะเวลาแต่ละการทดลอง 15 นาที
 - อัตราการส่งของข้อมูลสำหรับ โหนดที่ 3 1s ต่อ 1 แพ็กเก็ต จำนวน 5 การทดลอง ระยะเวลาแต่ละการทดลอง 15 นาที

ในการทดลองจะมีการทำงานของ node จะมีการ โปรแกรม โหนด โดยใช้โปรแกรมซึ่งอยู่ที่ `contiki/examples/ipv6/rpl-udp` ดังนี้

- `udp-server.c` เป็น Root โหนด ซึ่งเป็นตัวรับข้อมูลเพียงอย่างเดียว
- `udp-client.c` เป็น Child โหนด ซึ่งเป็นตัวส่งข้อมูลและได้มีการตั้งค่ารอบของการส่งตามรูปแบบการทดลองข้างต้น
- `udp-Recvclient.c` เป็น Parent ซึ่งคัดแปลงจาก Client คือ บังคับให้ไม่มีการส่งข้อมูลแต่ไม่ใช่ Root โหนด

ซึ่งการทำงานของ Server และ Client เป็นไปดังนี้

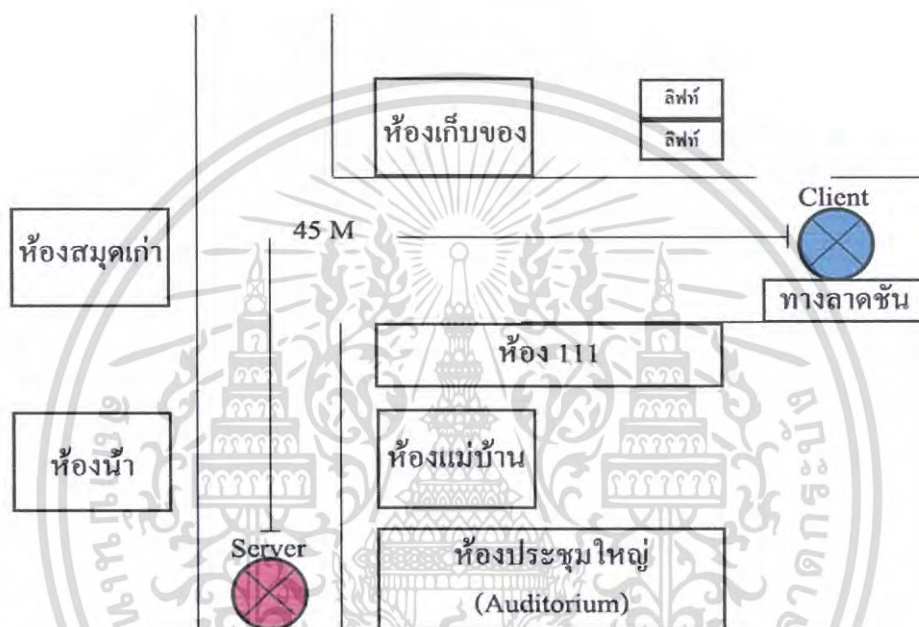
- การทำงานของ Client



รูปที่ 4.15 การทำงานของ Client

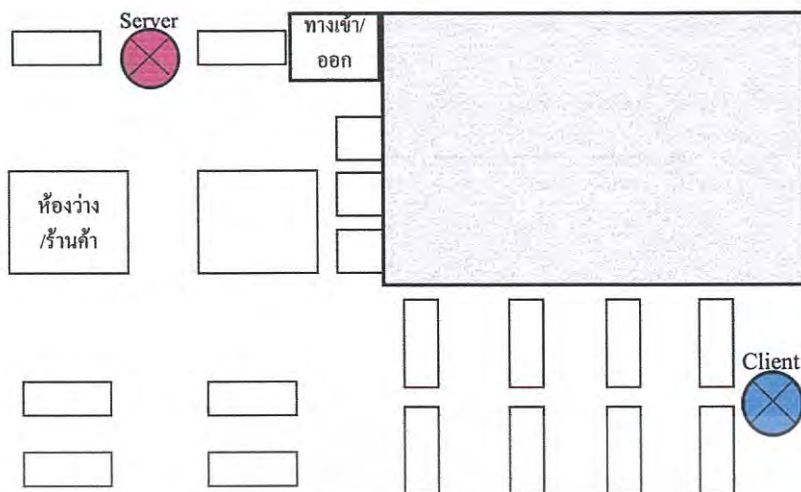
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3.1 ใช้อุปกรณ์ 2 mote แบ่งเป็น Client กับ Server ตามรูปแบบการทำงานข้างต้น โดยทำการทดลองตรงทางโค้งแบบไม่มี Line of sight (LOS) และกำหนดให้ Server อยู่ตรงหน้าห้อง Auditorium และ Client อยู่ตรงข้างหน้าบริเวณทางลาดชันซึ่งมีระยะทาง 45 เมตร ดังรูปที่ 4.18 และทำการทดลองโดยตั้งค่า radio frequency (RF) ไว้ที่ 26 และรอบในการส่งข้อมูล (Period) เท่ากับ 10 , 5 และ 1 วินาทีตามลำดับ โดยทำการทดลองเป็นจำนวน 5 ครั้งในแต่ละรอบเวลา ใช้เวลาในการทดลอง 15 นาที



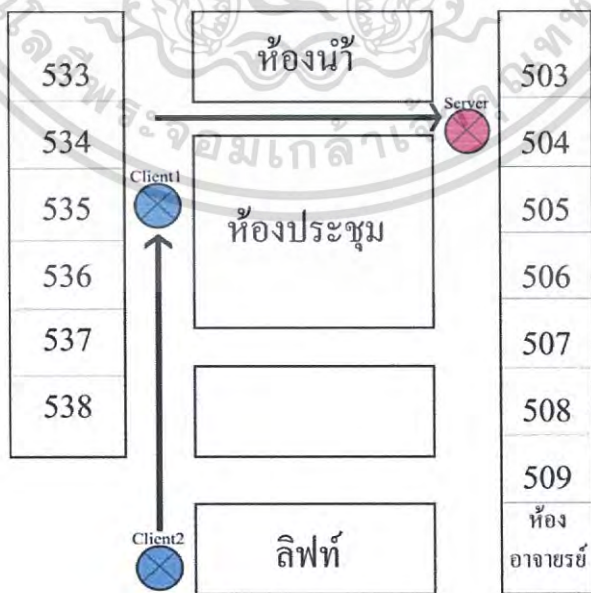
รูปที่ 4.16 การทดลองที่ 3 ระยะ 45 เมตร ทางโค้ง โดยใช้ 2 mote

การทดลองที่ 4 ทำการทดลองที่บริเวณห้อง Common Room ชั้น 4 ซึ่งใช้ 2 mote ดังรูปที่ 4.19 โดยทำการทดลองโดยตั้งค่า radio frequency (RF) ไว้ที่ 22 และรอบในการส่งข้อมูล (Period) เท่ากับ 10 , 5 และ 1 วินาทีตามลำดับ โดยทำการทดลองเป็นจำนวน 3 ครั้งในแต่ละรอบเวลา ใช้เวลาในการทดลอง 15 นาที



รูปที่ 4.17 การทดลองที่ 4 บนห้อง Common Room ใช้ 2 mote

การทดลองที่ 5 นั้นดำเนินการทดลองที่ชั้น 5 บริเวณโถงว่างทางด้านซ้ายมือสุดทางเป็น Client 2 ได้ขึ้นไปด้านบนที่หน้าห้อง 535 เป็นอีกจุดในการวางอุปกรณ์ Client1 และ วางจุดสุดท้ายที่หน้าห้อง 504 ซึ่งเป็นตำแหน่งวาง Server ดังรูปที่ 4.20 โดยให้ Client2 เป็นตัวส่งข้อมูลมายัง Client1 และส่งต่อมายัง Server โดยที่ Client1 ไม่สร้างข้อมูลของตัวเองและส่งกล่าวคือเป็นตัว Receive นั่นเอง ดังรูปที่ 4.20 และรอบในการส่งข้อมูล (Period) เท่ากับ 10 , 5 และ 1 วินาทีตามลำดับ และเมื่อเวลาเริ่มการทดลองและเวลาผ่านไป 7:30 นาที จะทำการนำ Client1 ออกจากเครือข่าย โดยทำการทดลองเป็นจำนวน 5 ครั้งในแต่ละรอบเวลา ใช้เวลาในการทดลอง 15 นาที

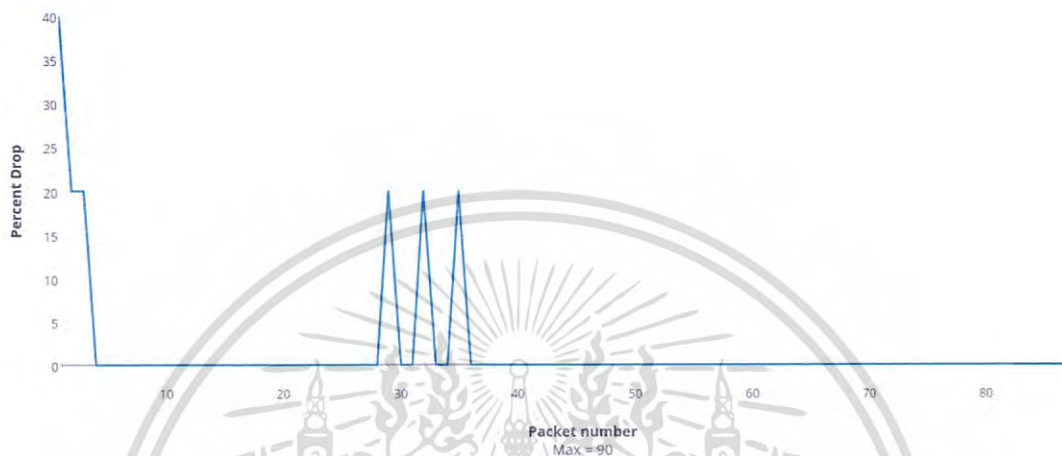


รูปที่ 4.18 การทดลองที่ 4 ใช้ 3 mote โดยให้ Client1 รับและส่งข้อมูล

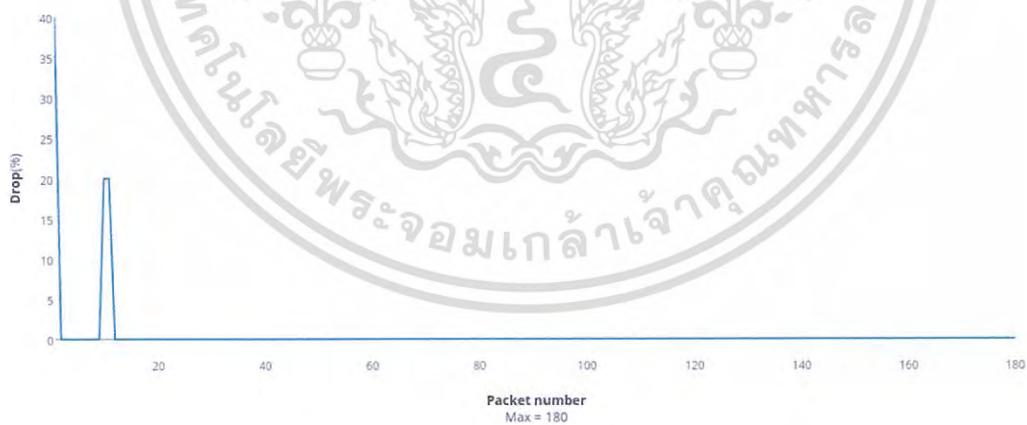
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การทดลองที่ 3 เครื่องข่าย 2 โหนด

- ผลการทดลองแบบที่มีการกวนของสัญญาณภายนอกต่ำ อัตราการส่งข้อมูล 10 วินาที



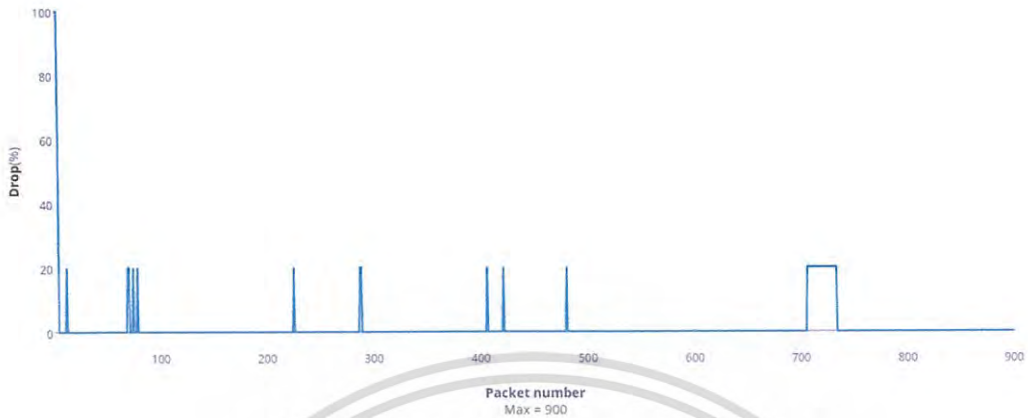
รูปที่ 4.19 การทดลองที่ 3 อัตราการส่งข้อมูล 10 วินาที
อัตราการส่งข้อมูล 5 วินาที



รูปที่ 4.20 การทดลองที่ 3 อัตราการส่งข้อมูล 5 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

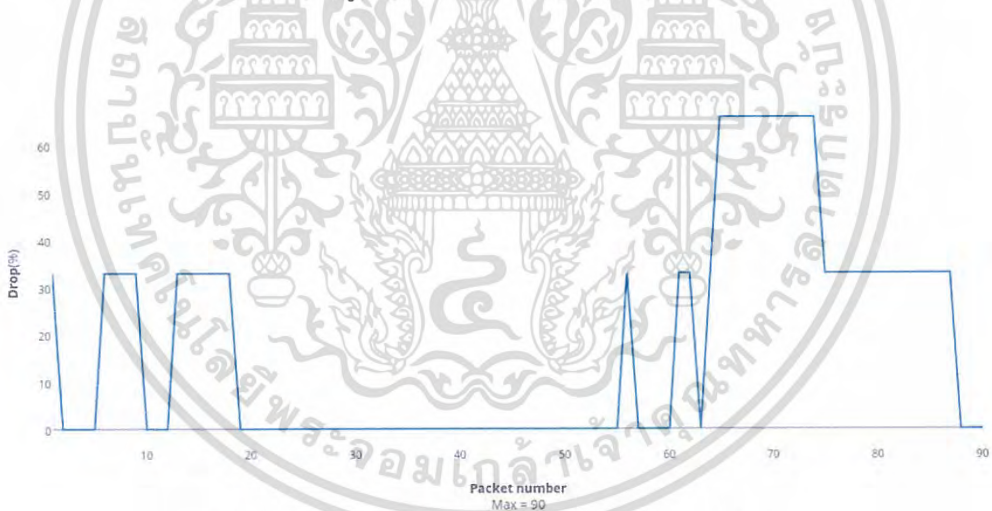
อัตราการส่งข้อมูลที่ 1 วินาที



รูปที่ 4.21 การทดลองที่ 3 อัตราการส่งข้อมูล 1 วินาที

- ผลการทดลองแบบที่มีการกวนของสัญญาณภายนอกสูง

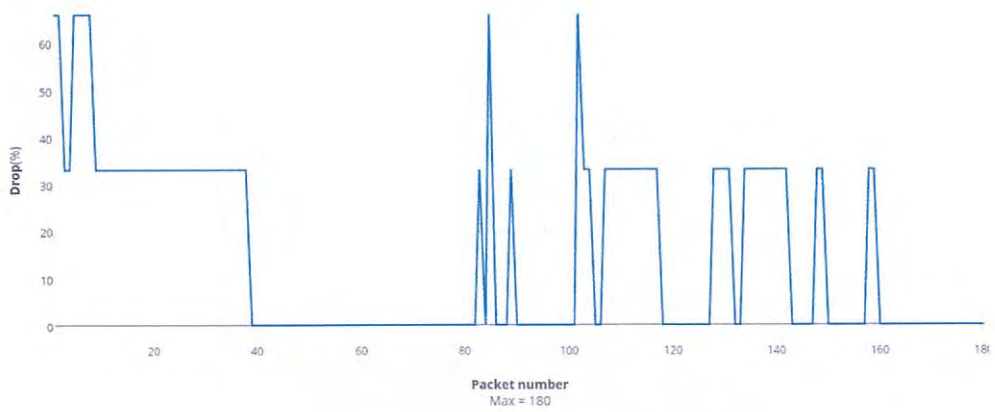
อัตราการส่งข้อมูลที่ 10 วินาที



รูปที่ 4.22 การทดลองที่ 4 อัตราการส่งข้อมูล 10 วินาที

อัตราการส่งข้อมูลที่ 5 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 การทดลองที่ 4 อัตราการส่งข้อมูล 5 วินาที

อัตราการส่งข้อมูล 1 วินาที



รูปที่ 4.24 การทดลองที่ 4 อัตราการส่งข้อมูล 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

จากการทดลองที่ 3 เครือข่าย 2 โหนด เป็นการทดลองที่แสดงถึงให้อัตราความสำเร็จของการรับส่งข้อมูลภายในอาคารเมื่อมีอัตราการกวนของสัญญาณภายนอกที่ต่ำ เนื่องจากใช้ช่วงของสัญญาณข้อมูลที่ช่อง 26

ตารางที่ 4.9 ตารางสรุปผลการทดลองที่ 3

อัตราการส่ง	ค่าเฉลี่ยการส่งสำเร็จ	ค่าเฉลี่ยจำนวน packet drop ต่อๆกัน
00	0.98	1.4
5s	0.99	1.3
1s	0.98	5.1

จากตารางจะเห็นว่าในช่วงของช่องสัญญาณดังกล่าวอัตราการส่งสำเร็จนั้นจะสูงโดยเกือบจะเป็นถึง 100% โดยช่วงที่อัตราการครอบที่สูงคือช่วงข้อมูลแรกของการส่งข้อมูลเนื่องจากฝั่งของ Client นั้นจะส่งทุกๆ รอบเวลาที่กำหนดดังนั้นแพ็กเก็ตแรกก็ออกไปนั้นเป็นไปได้อีกที่ทั้งสองฝั่งนั้นยังไม่ได้ทำการเชื่อมต่อถึงกันทำให้ทางฝั่งของ Server ไม่ได้รับข้อมูลในช่วงเวลาดังกล่าว และจะเห็นได้ว่าในช่วงอัตราการส่ง 1 วินาทีนั้นจะเห็นว่าการครอบที่ติดกันสูงเนื่องจากอัตราการส่งที่เร็วทำให้เมื่อมีการขาดการเชื่อมต่อไปช่วงหนึ่งก็จะทำให้เกิดการสูญหายของข้อมูลนั้นสูงตามไป

จากการทดลองที่ 4 เป็นการทดลองที่แสดงถึงให้อัตราความสำเร็จของการรับส่งข้อมูลภายในอาคารเมื่อมีอัตราการกววนของสัญญาณภายนอกที่สูง เนื่องจากใช้ช่วงของสัญญาณข้อมูลที่ช่อง 22 โดยปริมาณคนใช้งานในช่องนี้ไม่เยอะมาก

ตารางที่ 4.10 ตารางสรุปผลการทดลองที่ 4

อัตราการส่ง	ค่าเฉลี่ยการส่งสำเร็จ	ค่าเฉลี่ยจำนวน packet drop ต่อๆกัน
10s	0.69	29.3
5s	0.73	25.78
1s	0.72	84.6

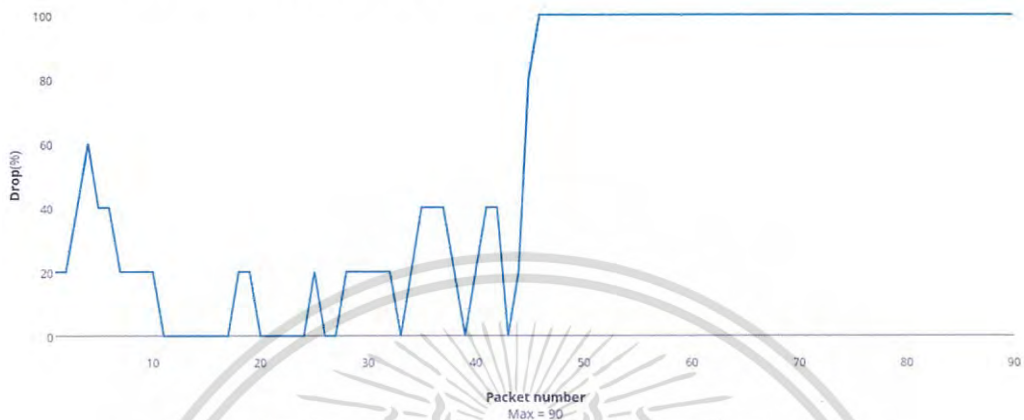
จากตารางสรุปผลการทดลองจะเห็นว่าเมื่อใช้ช่องสัญญาณที่มีการกววนสูงจะเห็นว่าม้ออัตราการส่งสำเร็จน้อยลงมากเพราะมีการรบกวนที่มีการทับซ้อนจากสัญญาณของ 802.11n คือ ช่อง 11 ที่ใช้ช่องสัญญาณเดียวกันทำให้เกิดการสูญหายของข้อมูล และเมื่อเพิ่มอัตราการส่งข้อมูลขึ้นก็ส่งผลให้จำนวนแพ็กเก็ตที่ดรอปต่อๆกันก็จะสูงด้วย แต่ก็ขึ้นกับสภาพแวดล้อมของคนที่ใช้ช่องสัญญาณนั้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 การทดลองที่ 4 เครือข่าย 3 โหนด

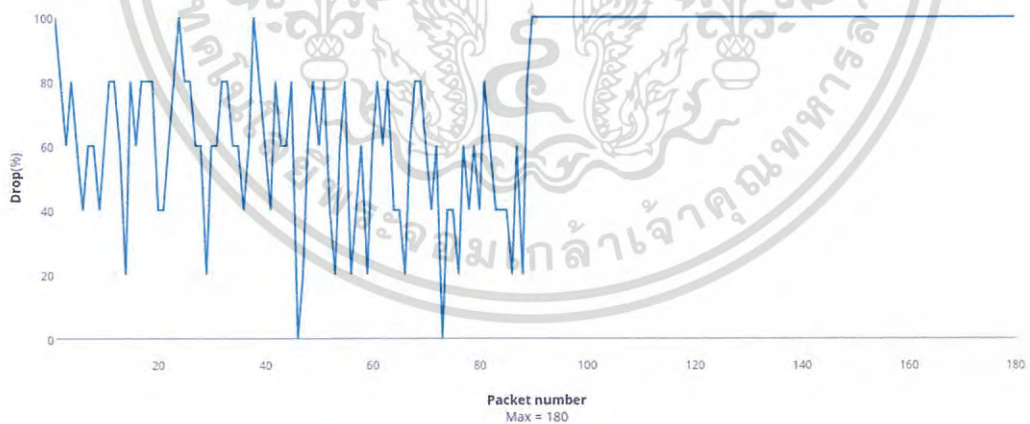
ผลการทดลอง

อัตราการส่งข้อมูลที่ 10 วินาที



รูปที่ 4.25 การทดลองที่ 5 อัตราการส่งข้อมูล 10 วินาที

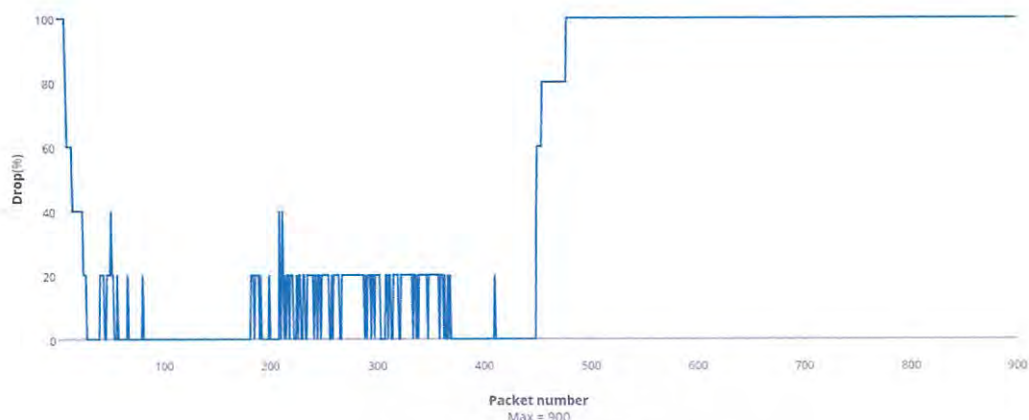
อัตราการส่งข้อมูลที่ 5 วินาที



รูปที่ 4.26 การทดลองที่ 5 อัตราการส่งข้อมูล 5 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราการส่งข้อมูลที่ 1 วินาที



รูปที่ 4.27 การทดลองที่ 5 อัตราการส่งข้อมูล 1 วินาที

สรุปผลการทดลอง

จากการทดลองที่ 5 มีลักษณะเครือข่ายเป็นแบบ 3 mote เพื่อเป็นการศึกษาว่าเมื่อนำโหนดออก RPL มีกระบวนการอย่างไรในการเรียนรู้ว่า โหนดที่เป็นเครือญาตินั้นหายไปแล้วใช้เวลาเท่าไรในกระบวนการนี้

ตารางที่ 4.11 ตารางสรุปผลการทดลองที่ 5

อัตราการส่ง	อัตราการส่งสำเร็จ	เวลาเฉลี่ยที่เรียนรู้ว่าโหนดใช้งานไม่ได้
10 s	0.40	3.4 นาที
5 s	0.21	4.04 นาที
1 s	0.45	3.76 นาที

จากตารางจะเห็นว่าเวลาที่เรียนรู้ว่าโหนดที่เป็นเครือญาตินั้นหายไปใช้เวลาโดยเฉลี่ยถึง 3.03 นาที ซึ่งทำให้อัตราการส่งสำเร็จของข้อมูลนั้นต่ำลงไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

แบบเสนอแนวทางการปรับปรุง

แนวทางการปรับปรุงนี้เหมาะสำหรับ application ที่มีการรับส่งข้อมูลต่อเนื่องอยู่เสมอ เนื่องจากจะมีผลทำให้เกิดการสูญหายของข้อมูลนั้นสูง หากเป็น application ที่มีการรับส่งทุกๆ ครั้ง ชั่วโมงก็จะไม่เปิดประสิทธิภาพเท่าที่ โดยแนวทางที่เป็นไปได้คือ

- ในการทดลองที่ 3 และ 4 นั้น โดย Default ของ RPL ใน Contiki นั้นอาศัยหลักของ MRHOF ในการปรับเปลี่ยนค่า metric ของ DIO message เท่านั้นในการตัดสินใจว่า Parent ยังทำงานได้อยู่ปกติอยู่หรือใช้ในการตัดสินใจในการเลือก parent แต่จากการทดลองพบว่าในกรณี ที่ ณ ช่วงเวลาที่มีการครอบของข้อมูลสูงนั้น (ค่า PRR ต่ำ) ผู้ส่งข้อมูลนั้นยังคิดว่า Parent นั้นทำงานอยู่เป็นปกติเนื่องจากในตลอดเวลาช่วงที่ผ่านมานั้นยังไม่ถึงรอบ Interval (Trickle Timer) ในการส่ง DIO message ซึ่ง DIO message จะถูกส่งตามค่า DIO Minimum Interval และค่า DIO Interval Doublings จึงทำให้ช่วงเวลาดังกล่าว ไม่เกิดกระบวนการตัดสินใจเลือก parent ทำให้เกิดการสูญหายของข้อมูลสูง ณ ช่วงเวลานั้น จึงควรให้มีการใช้ค่า PRR มาเป็นตัวกำหนด เช่น PRR หากน้อยกว่า 90% ให้ทำการส่ง DIO message ไปยัง โหนดที่เป็น parent ดังกล่าว เพื่อไปอัปเดตค่า metric และก่อให้เกิดกระบวนการตัดสินใจเลือก parent ณ เวลานั้นได้
- ในการเรียนรู้ของโหนดว่าเมื่อมี Parent นั้นออกจากเครือข่ายไปนั้นพบว่าจะเกิดขึ้นเมื่อเวลาผ่านไป ทุกๆ 10 นาที โดยจะเป็นค่าตายตัวที่ทาง Contiki ได้มีการกำหนดไว้ในส่วนของ platform ของในแต่ละอุปกรณ์ซึ่งจะพบว่าเมื่อไม่ว่าจะนำโหนดออกก่อนเวลา 10 นาที โหนดก็จะเรียนรู้ในทุกๆ 10 นาทีทำให้มีความล่าช้าและก่อนให้เกิดการสูญหายของข้อมูล

ในช่วงเวลาดังกล่าว จึงควรมีการกำหนดค่าที่มีการที่ไม่เป็นค่าคงที่เพื่อให้เป็นไปตามเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Neighbor Discovery ในส่วนของ Neighbor Unreachability Detection algorithm เช่น หากไม่ได้มี DIO Message มายัง โหนดภายในเวลาที่กำหนดก็จะมีการเลื่อนลำดับเป็นจาก REACHABLE เป็น STALE จนเมื่อมีการส่งข้อมูลก็จะเลื่อนเป็น DELAY และสุ่มค่า delay และส่ง probe และเลื่อนเป็น PROBE NS1 หากไม่ได้รับ acknowledgment ก็จะเปลี่ยนเป็น PROBE NS2 จนครบ 3 ครั้งก็ยังไม่ได้รับ ก็จะเลื่อนเป็น PROBE (END) และลบ โหนดที่เป็น Parent ออกไปในที่สุดซึ่งมีความรวดเร็วกว่าที่จะต้องรอทุกๆ 10 นาทีแบบเช่นเคย

- การปรับปรุงในเรื่องของลด DIO interval ลง ก็เป็นอีกหนึ่งทางเลือกซึ่งต่อจากข้อเสนอ ก่อนหน้า คือ เมื่อมีการปรับให้มีการลดค่า DIO interval ลง ก็จะทำให้โอกาสที่โหนดจะได้รับ DIO ก็จะมีสูงขึ้นทำให้เพิ่มอัตราการรับรู้ของโหนดเมื่อโหนดไม่อยู่ในเครือข่ายและเพื่อเป็นการให้ปรับปรุงค่า Rank ของ Parent ได้รวดเร็วยิ่งขึ้นอีกด้วย แต่ก็แลกมาด้วย Control Traffic Overhead ของข้อมูลในเครือข่ายและพลังงานที่ใช้ก็จะสูงขึ้นตามไปด้วย

บรรณานุกรม

- [1] Nopparat. “รู้จักกับ 6LoWPan เบื้องต้น.”
[Online] Available: <http://tinyurl.com/q95fsux>. 2015.
- [2] “6LoWPAN.”
[Online] Available: <https://en.wikipedia.org/wiki/6LoWPAN>.
- [3] Zach Shelby, Carsten Bormann. **6LoWPAN: The Wireless Embedded Internet.**
A John Wiley and Sons, Ltd, Publication, Inc.2009
- [4] Olfa Gaddour, Anis Koubâa. “RPL in a nutshell: A survey.”
CISTER Research Unit Polytechnic Institute of Porto (ISEP-IPP). 2013
- [5] Siarhei Kuryla, “RPL: IPv6 Routing Protocol for Low power and Lossy Networks”
[Online] Available: <http://tinyurl.com/q6ft153>. 2010.
- [6] “Wireless Sensor Networks (WSN).”
[Online] Available: <http://tinyurl.com/p7pr46b>.
- [7] Hazrat Ali. “A Performance Evaluation of RPL in Contiki.”
Master’s Thesis of Swedish Institute of Computer Science. 2012
- [8] ปนัดดา ชุมกระ โทก. “การปรับปรุงกระบวนการเชื่อมต่อเส้นทางแบบเฉพาะที่ของโปรโตคอล RPL.”
วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย, มหาวิทยาลัยเกษตรศาสตร์. 2557
- [9] “Contiki: The Open Source OS for the Internet of Things.”
[Online] Available: <http://www.contiki-os.org>.
- [10] “ไวร์ชาร์ค.”
[Online] Available: <http://tinyurl.com/pl7kddm>
- [11] “Internet of Things”
[Online] Available: https://en.wikipedia.org/wiki/Internet_of_Things
- [12] “Wireshark.”
[Online] Available : <https://www.wireshark.org>
- [13] “Zolertia Z1 Datasheet.”
[Online] Available
: http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ)

[14] Adam Dunkels. **The ContikiMAC Radio Duty Cycling Protocol.**

SICS Technical Report, T2011:13, ISSN 1100-3154, December 2011



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

ข้อกำหนดสำหรับ absolute maximum ratings ของโมดูล Z1 WSN แสดงดังรูปที่ ก.1

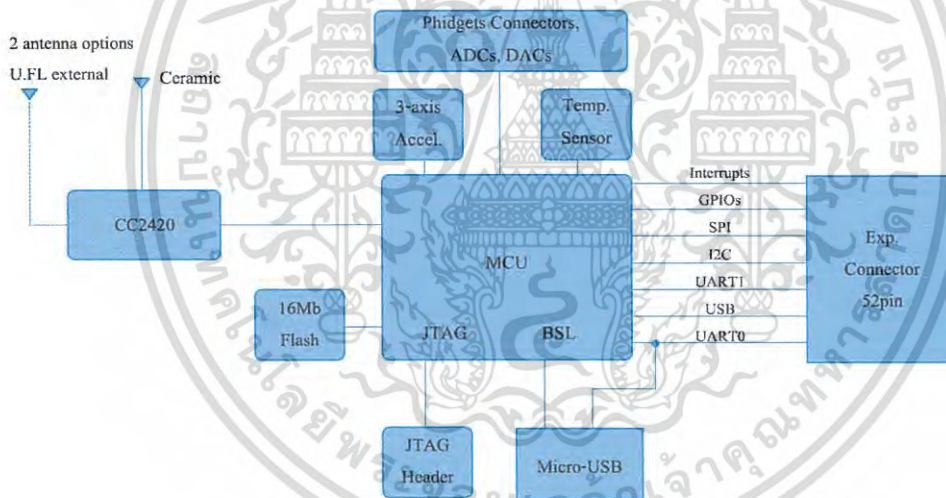
Description	Rating
Power Supply Voltage Vcc –	0.3V to +3.6V
Voltage on any digital Pin –	0.3 to Vcc+0.3V
Max. RF Input Power	10dBm
Storage Temperature Range	-40°C to +105°C
Operating Temperature Range	-40°C to +85°C

รูปที่ ก.1 absolute maximum ratings

โมดูลที่มีการระบุในการทำงานอย่างถูกต้องในช่วงอุณหภูมิอุตสาหกรรม

Functional Block Diagram

ฟังก์ชันแผนภาพบล็อกของ Z1 WSN โมดูลแสดงดังรูปที่ ก.2



รูปที่ ก.2 Functional Block Diagram of Z1 WSN Module

ในรูปที่ ก.2 จะเห็น MCU ตั้งอยู่ในใจกลางของโมดูลการควบคุมอุปกรณ์ต่อพ่วงรอบนอกทั้งหมดสำหรับการเขียนโปรแกรมหนึ่ง บนบอร์ด Z1 มีสองเซ็นเซอร์ : เซ็นเซอร์วัดอุณหภูมิด้วยระบบดิจิทัลและโปรแกรม accelerometer แบบดิจิทัลมีช่องต่อ USB สำหรับการเปิดการเขียนโปรแกรมและการติดต่อสื่อสารในระหว่างพัฒนาและส่วนหัว JTAG ที่มีความสามารถในการตรวจแก้จุดบกพร่อง พร้อม USB ที่มีการสื่อสารแบบดิจิทัลอื่น ๆ ที่ได้รับการสนับสนุนเช่น I2C, SPI และ UART UART แต่ใช้ 0V และ 3V เพื่อให้ระดับแรงดันไฟฟ้าไม่อนุญาตให้ Z1 ต้องเชื่อมต่อกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS232 (ที่แรงดันไฟฟ้าที่มี +12V และ -12V). มี 2 พอร์ตพื้นฐานพร้อมใช้งานสำหรับ 3V phidgets พอร์ตเหล่านี้โดยทั่วไป ADCs และ DACs และสามารถนำมาใช้กับเซนเซอร์อะนาล็อกอื่น ๆ ได้เช่นกัน

Powering the Z1

Z1 สามารถทำงานได้หลายวิธีขึ้นอยู่กับโปรแกรมที่เราต้องการ อย่างไรก็ตามวิธีที่ง่ายที่สุดและสำคัญที่สุดในการให้พลังงานแก่ Z1 ดังต่อไปนี้

- Battery-powered : ใช้ถ่าน AA จำนวน 2 ก้อน หรือ AAA จำนวน 1 ก้อน หรือ CR2032 จำนวน 1 ก้อน
- USB-powered : เพียงแค่เชื่อมต่อ USB ของ Z1 ผ่านทางสาย microUSB

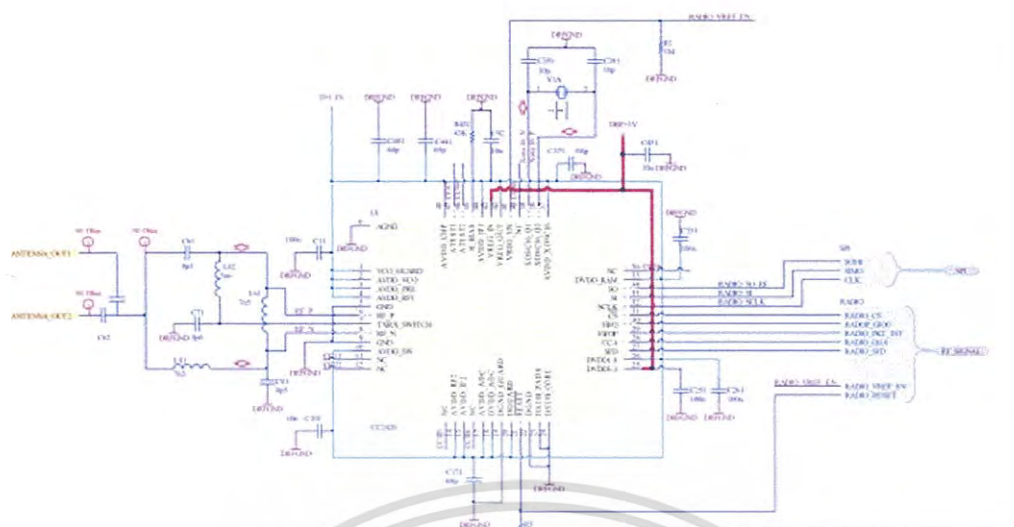
ทั้งสองรูปแบบการให้พลังงานนั้นสามารถทำได้พร้อมกัน หากใช้ Z1 เชื่อมต่อกับสาย USB เพื่อพัฒนาเป็นเวลานานๆ ก็ควรที่จะถอดแบตเตอรี่

Expansion Connector (XPcon)

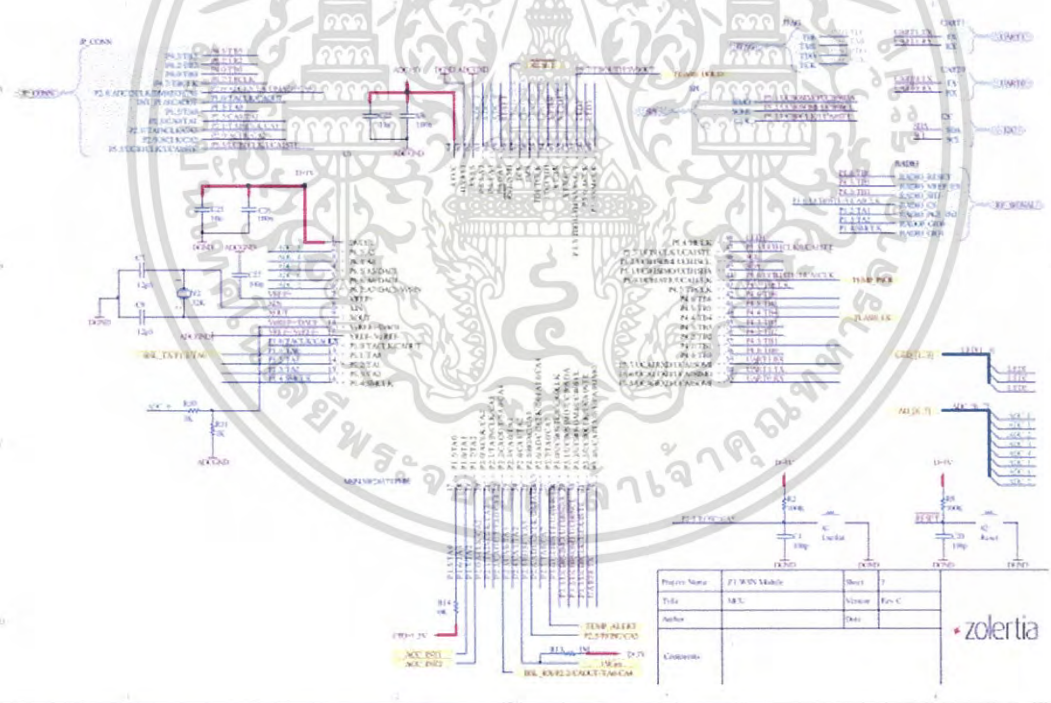
แบ่งออกเป็นหลายพอร์ต การจัดกลุ่มแต่ละประเภทของการเชื่อมต่อใน 4 ทิศทาง ดังต่อไปนี้

- North port - มีไว้สำหรับ Analog I/O สามารถค้นหา ADCs (ไม่เกิน 8) และ DACs (ไม่เกิน 2)
- East port - เหมาะสำหรับการเชื่อมต่อทางสายแบบดิจิทัล (USB, I2C, SPI, 2xUARTs) เช่นเดียวกับบาง GPIOs และ พลังงาน (power) และ สายดิน (ground)
- South port - มีไว้สำหรับ GPIOs เช่นเดียวกับฟังก์ชันการกำหนดค่าอื่น ๆ เช่น ขัดจังหวะการป้อนข้อมูลที่ขา (pin) การเปรียบเทียบข้อมูลที่เข้ามา (input) บางส่วนของขา (pin) ที่มีอยู่แล้วในการใช้งาน โดยคุณสมบัติบางอย่างของ Z1 จึงสามารถตรวจสอบหรือดักสำหรับ โปรแกรมอื่นได้จากที่นี่
- West port - มีไว้สำหรับการสื่อสารไร้สายทั้งโดยเสาอากาศฝังหรือเสาอากาศภายนอกโดยใช้โปรโตคอลเครือข่ายที่สนับสนุนไร้สายเช่น Zigbee, 6LoWPAN ฯลฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

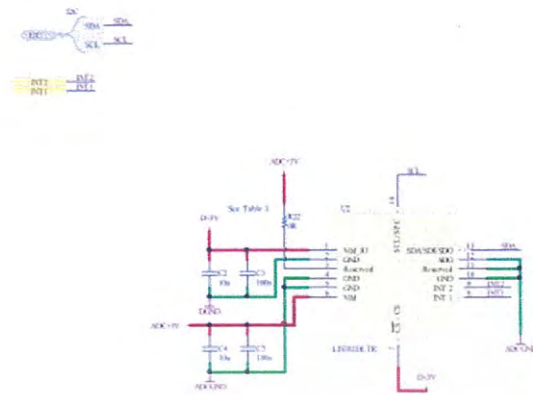


รูปที่ ก.5 Board Schematics



รูปที่ ก.6 Board Schematics

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Project Name	ZUWON Watch	Sheet	01
File	Accelerometer	Version	Rev. C
Author		Date	
Company			

• zolertia

รูปที่ ก.13 Board Schematics

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ นางสาว กัญญ์ณพัชร คุณากรพัฒนาการ
 วัน เดือน ปีเกิด 12 ธันวาคม 2537
 สถานที่เกิด จังหวัด กรุงเทพฯ
 ที่อยู่ บ้านเลขที่ 13/4 หมู่ 5 ถนนเทศบาล 3 ตำบล โสนลอย
 อำเภอ บางบัวทอง นนทบุรี 11110
 ประวัติการศึกษา วิทยาศาสตร์บัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อ นาย วิษณุ ทองประดิษฐ์
 วัน เดือน ปีเกิด 12 มิถุนายน 2536
 สถานที่เกิด จังหวัด กรุงเทพฯ
 ที่อยู่ บ้านเลขที่ 94/276 หมู่บ้านพรอรัวิลล์ ซอย ทิวลิป 4
 ถนนสุวินทวงศ์ 38 แขวงลำผักชี เขตหนองจอก กรุงเทพฯ 10530
 ประวัติการศึกษา วิทยาศาสตร์บัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาประสิทธิภาพการทำงานของโปรโตคอลสำหรับการค้นหาเส้นทางในเครือข่ายที่มีพลังงานต่ำและมีอัตราการสูญหายของข้อมูลสูงในการใช้งานภายในอาคาร

วิชญ์ ทองประดิษฐ์ และ กัญญ์ณพัทธ์ คุณาภรณ์พัฒนาการ

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

Emails: master2430@gmail.com, beobonus@gmail.com

บทคัดย่อ

IPv6 Routing Protocols for Low Power and Lossy Network (RPL) เป็นโปรโตคอลในการค้นหาเส้นทางที่ออกแบบมาเพื่อใช้งานสำหรับอุปกรณ์ในระบบเครือข่ายที่มีพลังงานต่ำและการรับส่งข้อมูลบนเครือข่ายที่มีโอกาสสูญหายข้อมูลสูง จากลักษณะคุณสมบัติที่น่าสนใจดังกล่าวโครงงานนี้ได้นำ RPL มาศึกษาประสิทธิภาพการทำงานภายใต้สภาพแวดล้อมภายในอาคาร ในโครงงานนี้ใช้โปรโตคอล RPL ที่มากับระบบปฏิบัติการ Contiki ทำการจำลองด้วยโปรแกรม cooja simulator และการนำไปใช้กับอุปกรณ์จริง เพื่อศึกษาและวิเคราะห์ประสิทธิภาพ RPL ในการใช้งานภายในอาคาร ในกรณีที่มีการรบกวนจากระบบภายนอกต่ำและสูง เพื่อนำผลการทดลองมาวิเคราะห์และเสนอแนะแนวทางในการปรับปรุงค่าพารามิเตอร์ของ RPL ให้มีประสิทธิภาพดียิ่งขึ้น หวังว่าจะเป็นส่วนช่วยสำหรับผู้สนใจเพื่อนำไปต่อยอดและประยุกต์ใช้งานต่อไป

คำสำคัญ – RPL; Z1; เทคโนโลยีเครือข่ายไร้สาย; Cooja; Contiki

1. บทนำ

ปัจจุบันอุปกรณ์เซนเซอร์แบบไร้สาย เป็นเทคโนโลยีที่มีการใช้กันอย่างกว้างขวาง ไม่ว่าจะเป็นทางด้านอุตสาหกรรม ด้านเกษตรกรรม หรือในชีวิตประจำวันสามารถติดตั้งอุปกรณ์เซนเซอร์และเข้ามาเป็นส่วนหนึ่งของระบบเครือข่ายได้ เพื่อเป็นตัวช่วยอำนวยความสะดวกในด้านต่างๆ เช่น ตู้เย็น, เครื่องใช้ไฟฟ้าต่าง ทำให้อินเทอร์เน็ตโพรโทคอลเวอร์ชัน 4 ไม่เพียงพอต่อการใช้งาน และต้องใช้ อินเทอร์เน็ตโพรโทคอลเวอร์ชัน 6 และเทคโนโลยีนี้ก็มีข้อจำกัดที่อาจหลีกเลี่ยงได้คือ ข้อจำกัดด้านพลังงานที่มีอยู่อย่างจำกัด และข้อจำกัดด้านสภาพแวดล้อมเช่น อากาศ ความร้อน ความชื้น เป็นต้น ซึ่งล้วนแล้วแต่มีผลต่อการเกิดการสูญหายของข้อมูลได้ ซึ่งในเทคโนโลยีเครือข่ายไร้สาย นั้นบนรูปแบบมาตรฐานอินเทอร์เน็ตโพรโทคอลเวอร์ชัน ที่มีการบริหารจัดการโดยใช้พลังงานต่ำสอดคล้องกับซิกซ์โลว์แพน(6LoWPAN)นั้น RPL (IPv6 Routing Protocol for Low Power and Lossy Network) เป็นตัวเลือกหนึ่งใน Routing Protocol ที่สามารถ

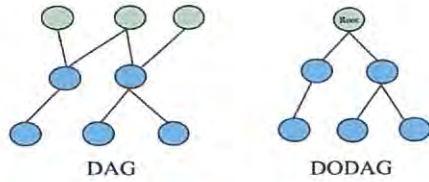
ตอบโจทย์ในข้อจำกัดของเทคโนโลยีเซนเซอร์แบบไร้สายและยังเหมาะกับการประยุกต์ใช้

2. ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

2.1 IPV6 ROUTING PROTOCOL FOR LOW POWER AND LOSSY NETWORKS (RPL)

RPL คือ Routing Protocol เป็นเครือข่ายที่ถูกออกแบบมาเป็นพิเศษสำหรับการใช้ในอุปกรณ์ที่มีพลังงานต่ำ และมีอัตราการสูญเสียข้อมูลสูง (LLNs) ซึ่งสอดคล้องกับ 6LoWPAN RPL ใช้อัลกอริทึมแบบ distance-vector (DV) และอยู่บนพื้นฐานแนวคิดของโทโพโลยีของ Directed Acyclic Graphs (DAGs) Directed Acyclic Graphs (DAGs) คือ กราฟที่สามารถระบุทิศทาง(Directed Graph) ได้และไม่มีวัฏจักร (Cycles) ส่วน Destination Oriented DAG (DODAG) คือ มี Root Node ได้เพียงโหนดเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1. กราฟDAGและกราฟDODAG

RPL ได้กำหนด control messages สำหรับการสร้างและการบำรุงรักษาโครงสร้างของ RPL และตารางเส้นทาง มีสามประเภทดังต่อไปนี้

2.1.1. DAG Information Object (DIO) :

ถูกใช้โดยRPL ในการนำข้อมูลที่ช่วยให้โหนดค้นพบ RPL Instance DIO ประกอบไปด้วยข้อมูลการกำหนดค่าของ DODAG และเรียนรู้การเลือก DODAG Parents ด้วยการเลือกเส้นทางที่มีค่า Cost น้อยที่สุดไปยัง Root ของ DODAG

2.1.2. DAG Information Solicitation (DIS) :

โหนดใดๆที่มีการร้องขอ DIO จาก RPL โหนดเพื่อนบ้าน เนื่องจากตัวเองไม่ได้รับ DIO จึงได้ทำการร้องขอไปยังโหนดเพื่อนบ้าน

2.1.3 Destination Advertisement Object (DAO) :

ถูกใช้โดยRPL ใช้ในการเผยแพร่ข้อมูลปลายทาง ขึ้นไปด้านบนของ DODAG เพื่อสนับสนุนการทำ DOWNWARD TRAFFIC

2.2 Z1 ZOLERTIA

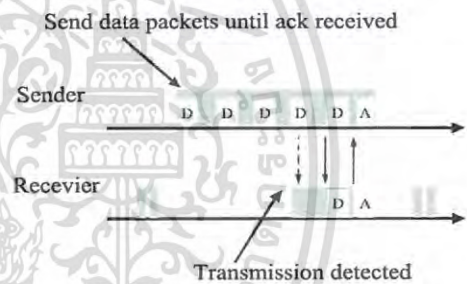
เป็นแพลตฟอร์มในการพัฒนาสำหรับ เครือข่ายเซ็นเซอร์ไร้สายพลังงานต่ำ (WSN) ตามมาตรฐาน IEEE 802.15.4 ออกแบบมาเพื่อทำปฏิต่างานในงานวิจัย และผู้ที่ชื่นชอบในการพัฒนา สามารถทำงานร่วมกันได้ระหว่างตระกูลของ TMOTE™ อีกด้วยนอกจากนี้ สถาปัตยกรรมหลักของ Z1 อยู่บนพื้นฐานของ MSP430 กับ CC2420 ของไมโครคอนโทรลเลอร์และส่งสัญญาณวิทยุโดย TEXAS INSTRUMENTS ทำให้มันเข้ากันได้กับ MOTE ที่ใช้ในสถาปัตยกรรมเดียวกัน อย่างไรก็ตาม MCU ที่มีอยู่ใน Z1 เป็น MSP430F2XXX แทนที่จะเป็น MSP430F1XXX อย่างที่เป็นใน MOTES อื่นๆ เช่น CROSSBOW'S TELOSБ, MOTEIV/S TMOTE



รูปที่ 2. Z1 Zolertia

2.3 Contiki MAC

CONTIKIMAC เป็นโปรโตคอลชนิด RADIO DUTY CYCLING ที่มีช่วงเวลาตื่นขึ้นมารับแพ็กเก็ต จากโหนด เพื่อนบ้านที่อยู่ใกล้กัน หากช่วงเวลาการส่งแพ็กเก็ตเป็นช่วงเวลาที่ไม่โหนดตื่นตัวรับก็ยังคงที่จะได้รับแพ็กเก็ต เมื่อได้รับแพ็กเก็ตเรียบร้อยแล้ว ตัวรับจะทำการส่ง ACKNOWLEDGMENT การส่งแพ็กเก็ตของผู้ส่งจะส่งซ้ำๆจนกว่าจะได้รับ ACKNOWLEDGMENT จากตัวรับ ว่ามันได้รับแพ็กเก็ตแล้ว



รูปที่ 3. ContikiMAC

3. วิธีการดำเนินงานวิจัย

3.1. ศึกษาแอปพลิเคชันที่ใช้ในการจำลอง

3.1.1. COOJA SIMULATOR

เป็นแอปพลิเคชันที่มาพร้อมกับตัว CONTIKI OS ใช้ในการจำลองสภาพแวดล้อมของระบบระบบเครือข่าย และยัง สามารถจำลอง MOTE ไม่ว่าจะเป็น SKY MOTE ,Z1 MOTE ,CC430 MOTE , COOJA MOTE เป็นต้นออกมาได้ และ FIRMWARE นั้นถูกพัฒนาด้วยภาษา C ที่สามารถแก้ไขดัดแปลงต่างๆของตัว MOTE ได้อีกด้วยที่ช่วยให้นักพัฒนาได้เห็นถึงการทำงานของสภาพเครือข่าย และรายละเอียดต่างๆ โดย COOJA SIMULATOR ยังเป็น OPEN SOURCE สำหรับ INTERNET OF THINGS รับรองการสื่อสารที่ใช้พลังงานต่ำ อีกทั้งยังรองรับ IPV4 และ IPV6 รองรับมาตรฐาน 6LOWPAN , RPL , COAP

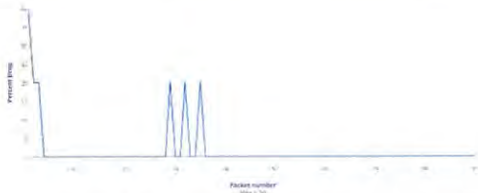
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ผลการทดลองจากอุปกรณ์ Z1

4.1. รูปแบบการทดลอง

4.1.1. แบบที่มีการรบกวนของสัญญาณภายนอกต่ำ

เครือข่ายขนาด 2 mote , radio frequency (RF) = 26 ใช้
เวลาในการทดลอง 15 นาที (ผลการทดลองที่ 3)



รูปที่ 4. อัตราการส่งข้อมูลที่ 10 วินาที

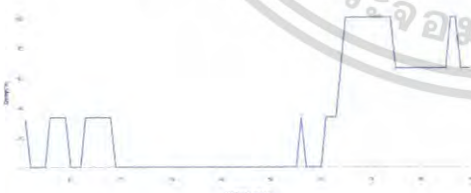


รูปที่ 5. อัตราการส่งข้อมูลที่ 5 วินาที

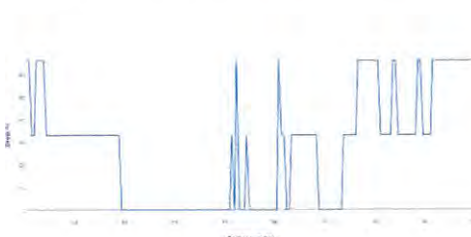


รูปที่ 6. อัตราการส่งข้อมูลที่ 1 วินาที

เครือข่ายขนาด 2 mote , radio frequency (RF) = 22 ใช้
เวลาในการทดลอง 15 นาที (ผลการทดลองที่ 4)

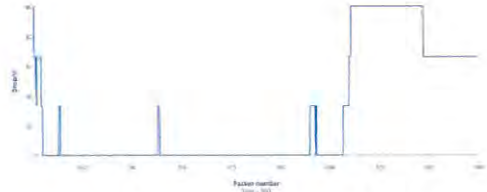


รูปที่ 7. อัตราการส่งข้อมูลที่ 10 วินาที



รูปที่ 8. อัตราการส่งข้อมูลที่ 5 วินาที

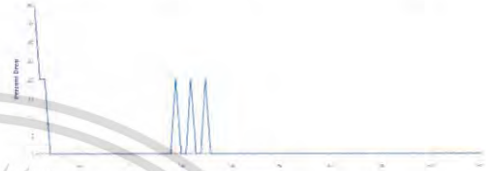
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9. อัตราการส่งข้อมูลที่ 1 วินาที

4.1.2. แบบที่มีการรบกวนของสัญญาณภายนอกสูง

เครือข่ายขนาด 3 mote, radio frequency (RF) = 26 ใช้
เวลาในการทดลอง 15 นาที (ผลการทดลองที่ 5)



รูปที่ 10. อัตราการส่งข้อมูลที่ 10 วินาที ของServer



รูปที่ 11. อัตราการส่งข้อมูลที่ 5 วินาที ของServer



รูปที่ 12. อัตราการส่งข้อมูลที่ 1 วินาที ของServer

5. บทสรุป

5.1 สรุปผลการทดลองที่ 3 (เครือข่าย 2 mote)

จากการทดลองที่ 3 เป็นการทดลองที่แสดงถึงให้อัตราความสำเร็จของการรับส่งข้อมูลภายในอาคารเมื่อมีอัตราการรบกวนของสัญญาณภายนอกที่ต่ำ เนื่องจากใช้ช่วงของสัญญาณข้อมูลที่ช่อง 26

ตารางที่ 1. ตารางสรุปผลการทดลองที่ 3

อัตราการส่ง	ค่าเฉลี่ยการส่งสำเร็จ	ค่าเฉลี่ยจำนวน packet drop ต่อๆกัน
10s	0.98	1.4

จะนำโหนดออกก่อนเวลา 10 นาที โหนดก็จะเรียนรู้ในทุกๆ 10 นาที ทำให้มีความล่าช้าและก่อนให้เกิดการสูญหายของข้อมูลในช่วงเวลาดังกล่าว จึงควรมีการกำหนดค่าที่มีการไม่เป็นค่าคงที่เพื่อให้เป็นไปตาม Neighbor Discovery ในส่วนของ Neighbor Unreachability Detection algorithm เช่น หากไม่ได้รับ DIO Message มาถึงโหนดภายในเวลาที่กำหนดก็จะมีการเลื่อนลำดับเป็นจาก REACHABLE เป็น STALE จนเมื่อมีการส่งข้อมูลก็จะเลื่อนเป็น DELAY และส่งค่า delay และส่ง probe และเลื่อนเป็น PROBE NS1 หากไม่ได้รับ acknowledgment ก็เปลี่ยนเป็น PROBE NS2 จนครบ 3 ครั้งก็ยังไม่ได้รับ ก็จะเลื่อนเป็น PROBE (END) และลบโหนดที่เป็น Parent ออกไปในที่สุดซึ่งมีความรวดเร็วกว่าที่จะต้องรอทุกๆ 10 นาทีแบบเช่นเคย

การปรับปรุงในเรื่องของลด DIO interval ลง ก็เป็นอีกหนึ่งทางเลือกซึ่งต่อจากข้อเสนอก่อนหน้านี้ คือ เมื่อมีการปรับให้มีการลดค่า DIO interval ลง ก็จะทำให้โอกาสที่โหนดจะได้รับ DIO ก็จะมีสูงขึ้นทำให้เพิ่มอัตราการรับรู้ของโหนดเมื่อโหนดไม่อยู่ในเครือข่ายและ เพื่อเป็นการให้ปรับค่า Rank ของ Parent ได้รวดเร็วยิ่งขึ้นอีกด้วย แต่ก็แลกมาด้วย Control Traffic Overhead ของข้อมูลในเครือข่ายและพลังงานที่ใช้ก็จะสูงขึ้นตามไปด้วย

เอกสารอ้างอิง

[1] Nopparat. “รู้จักกับ 6LoWPan เบื้องต้น.”

[Online] Available:

<http://tinyurl.com/q95fsux>. 2015.

[2] “6LoWPAN.”

[Online] Available:

<https://en.wikipedia.org/wiki/6LoWPAN>.

[3] Zach Shelby, Carsten Bormann. 6LoWPAN: The Wireless Embedded Internet.

A John Wiley and Sons, Ltd, Publication, Inc.2009

[4] Olfa Gaddour, Anis Koubâa. “RPL in a nutshell: A survey.”

CISTER Research Unit Polytechnic Institute of Porto (ISEP-IPP). 2013

[5] Sjarhei Kuryla, “RPL: IPv6 Routing Protocol for Low power and Lossy Networks”

[Online] Available:

<http://tinyurl.com/q6ftt53>. 2010.

[6] “Wireless Sensor Networks (WSN).”

[Online] Available:

<http://tinyurl.com/p7pr46b>.

[7] Hazrat Ali. “A Performance Evaluation of RPL in Contiki.”

Master’s Thesis of Swedish Institute of Computer Science. 2012

[8] ปณิตดา ชุมกระโทก. “การปรับปรุงกระบวนการซ่อมเส้นทางแบบเฉพาะที่ของโปรโตคอล RPL.” วิทยานิพนธ์ วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย, มหาวิทยาลัยเกษตรศาสตร์. 2557

[9] “Contiki: The Open Source OS for the Internet of Things.”

[Online] Available:

<http://www.contiki-os.org>.

[10] “ไวร์ชาร์ก.”

[Online] Available:

<http://tinyurl.com/pl7kddm>

[11] “Internet of Things”

[Online] Available:

https://en.wikipedia.org/wiki/Internet_of_Things

[12] “Wireshark”

[Online] Available:

<https://www.wireshark.org>

[13] “Zolertia Z1 Datasheet”

[Online] Available:

http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf.

[14] Adam Dunkels. The ContikiMAC Radio Duty Cycling Protocol.

SICS Technical Report, T2011:13, ISSN 1100 - 3154, December 2011.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้