



**กระบวนการปรับรอบทำงานสำหรับโพรโทคอล MAC ของมาตรฐาน IEEE 802.15.4**

**A Duty-Cycle Adjust Process for MAC Protocol of IEEE 802.15.4**

**จรัสศักดิ์ รักษาชุม**

**Jirasak Raksachum**

**วิทยานิพนธ์นี้สำหรับการศึกษิตตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณเฑิต**

**สาขาวิชาวิศวกรรมคอมพิวเตอร์**

**มหาวิทยาลัยสงขลานครินทร์**

**A Thesis Submitted in Fulfillment of the Requirements for the Degree**

**of Master of Engineering in Computer Engineering**

**Prince of Songkla University**

**2556**

**ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์**



**กระบวนการปรับรอบทำงานสำหรับโพรโทคอล MAC ของมาตรฐาน IEEE 802.15.4**

**A Duty-Cycle Adjust Process for MAC Protocol of IEEE 802.15.4**

**จรัสศักดิ์ รักษาชุม**

**Jirasak Raksachum**

**วิทยานิพนธ์นี้สำหรับการศึกษิตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต**

**สาขาวิชาวิศวกรรมคอมพิวเตอร์**

**มหาวิทยาลัยสงขลานครินทร์**

**A Thesis Submitted in Fulfillment of the Requirements for the Degree**

**of Master of Engineering in Computer Engineering**

**Prince of Songkla University**

**2556**

**ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์**

ชื่อวิทยานิพนธ์ กระบวนการปรับรอบทำงานสำหรับ โพรโทคอล MAC ของมาตรฐาน IEEE  
802.15.4

ผู้เขียน นายจรัสศักดิ์ รักษาชุม

สาขาวิชา วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

คณะกรรมการสอบ

..... ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต) (ผู้ช่วยศาสตราจารย์ ดร.ณัฐฐา จินดาเพ็ชร)

..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.ศุภชัย วรพจน์พิศุทธิ์)

..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้รับวิทยานิพนธ์ฉบับนี้  
เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรม  
คอมพิวเตอร์

.....  
(รองศาสตราจารย์ ดร.ธีระพล ศรีชนะ)

คณบดีบัณฑิตวิทยาลัย

(3)

ขอรับรองว่า ผลงานวิจัยนี้มาจากการศึกษาวิจัยของนักศึกษาเอง และได้แสดงความขอบคุณบุคคลที่มีส่วนช่วยเหลือแล้ว

ลงชื่อ.....

(ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต)

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ลงชื่อ.....

(นายจรัสศักดิ์ รักษาชุม)

นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน และ  
ไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ.....

(นายจรัสศักดิ์ รักษาชุม)

นักศึกษา

ชื่อวิทยานิพนธ์ กระบวนการปรับรอบทำงานสำหรับ โพรโทคอล MAC ของมาตรฐาน IEEE  
802.15.4  
ผู้เขียน นายจรัสศักดิ์ รักษาชม  
สาขาวิชา วิศวกรรมคอมพิวเตอร์  
ปีการศึกษา 2555

### บทคัดย่อ

อุปกรณ์ที่ทำงานด้วยพลังงานจากแบตเตอรี่บนเครือข่ายเซนเซอร์ไร้สาย จำเป็นต้องใช้พลังงานให้น้อยที่สุดเพื่อยืดอายุการใช้งาน ซึ่งในมาตรฐาน IEEE 802.15.4 นั้น End Device ได้ใช้วิธีการเปิด-ปิด ภาครับ-ส่งสัญญาณวิทยุให้ตรงจังหวะกับ Beacon ที่ส่งมาจาก Coordinator เพื่อประหยัดพลังงาน ซึ่ง End Device เหล่านี้สามารถปิดภาครับ-ส่งสัญญาณวิทยุได้แทบตลอดเวลา ขึ้นอยู่กับปริมาณการรับ-ส่งข้อมูล แต่ Coordinator นั้นจะต้องเปิดภาครับ-ส่งสัญญาณวิทยุไว้รอการส่งข้อมูลจาก End Device ซึ่ง Coordinator จะเปิดภาครับ-ส่งสัญญาณวิทยุไว้นานเท่าไรนั้นขึ้นอยู่กับค่า Duty-cycle ใน Coordinator

อย่างไรก็ตามในบางลักษณะงานเช่น งานด้านการเฝ้าระวังสุขภาพ อุปกรณ์ชนิด Coordinator ต้องติดอยู่กับตัวผู้ป่วยตลอดเวลา เพื่อสื่อสารกับเซนเซอร์หลากหลายชนิดที่อยู่บนร่างกาย ดังนั้น Coordinator จึงจำเป็นต้องทำงานด้วยพลังงานจากแบตเตอรี่เช่นกัน การปรับให้ Duty-cycle ต่ำเพื่อให้สามารถปิดภาครับ-ส่งสัญญาณวิทยุได้นานขึ้นนั้นจะต้องแลกมาด้วยความเร็วที่ลดลงและระยะเวลาหน่วงในการส่งข้อมูลที่เพิ่มขึ้น ในขณะเดียวกันหาก Duty-cycle มีค่าสูง จะสามารถลดการใช้พลังงานลงได้แต่ก็ลดความเร็วในการส่งข้อมูลลงด้วยเช่นกัน

วิทยานิพนธ์ฉบับนี้จึงนำเสนอและวิเคราะห์วิธีการใช้พลังงานอย่างมีประสิทธิภาพของเซนเซอร์ไร้สาย โดยกระบวนการปรับ Duty-cycle แบบอัตโนมัติ บนโพรโทคอลชั้น MAC ของมาตรฐาน IEEE 802.15.4 ซึ่งทำการทดลองบนโหนดที่ชื่อ Unode พบว่า MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัติ สามารถยืดอายุของแบตเตอรี่ขนาด AA ความจุ 1600 mAh จำนวน 2 ก้อนให้ทำงานได้มากกว่า MAC โพรโทคอลแบบ Duty-cycle คงที่ สูงสุดกว่า 1000 วัน ที่อัตราการส่งข้อมูล 1 ไบต์ต่อวินาที

คำสำคัญ: เครือข่ายเซนเซอร์ไร้สาย, โพรโทคอลชั้น MAC, IEEE 802.15.4, Duty-Cycle

**Thesis Title**            A Duty-Cycle Adjust Process for MAC Protocol of IEEE 802.15.4  
**Author**                    Mr. Jirasak Raksachum  
**Major Program**        Computer Engineering  
**Academic Year**        2012

### **ABSTRACT**

It is important that the battery-powered devices in a wireless sensor network have to operate for extended period of time. To achieve this in IEEE 802.15.4 personal area network, these devices synchronize with periodic beacons transmitted from coordinators and their transceivers are kept off most of time.

However, there are some applications, e.g. health care, which the coordinator can also run on battery. It communicates with various kinds of body area sensor nodes. Increasing superframe duration comes at the expense of reduced battery life while increasing beacon interval will directly affect data rate and latency.

This study has two major purposes: (1) To analyze protocol performance of MAC sublayer over IEEE 802.15.4. (2) To suggest methods that dynamically adjust beacon interval and superframe duration of the coordinator to reduce its power consumption while preserve optimized communication speed and latency.

**Keywords:** Wireless sensor networks, MAC protocol, IEEE 802.15.4, Duty-Cycle

## กิตติกรรมประกาศ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้เสียสละเวลาในการให้คำปรึกษา พร้อมทั้งแนะนำแนวคิดในการทำวิทยานิพนธ์ รวมถึงการแนะนำวิธีการแก้ไขปัญหาที่เกี่ยวกับการวิทยานิพนธ์ ตลอดจนตรวจสอบและแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างลุล่วงสมบูรณ์

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ณัฐา จินดาเพชร ที่ได้กรุณาเสียสละเวลาเป็นประธานกรรมการสอบวิทยานิพนธ์

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ศุภชัย วรพจน์พิศุทธิ ที่ได้กรุณาเสียสละเวลาเป็นกรรมการสอบวิทยานิพนธ์

ขอขอบพระคุณ ทูนการศึกษา NTC Scholarship สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ (NBTC, TRIDI) ที่ช่วยสนับสนุนทุนการศึกษา

ขอขอบพระคุณ คณาจารย์ บุคลากร นักศึกษาปริญญาเอก และนักศึกษาปริญญาโท ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกคนที่ได้ให้คำปรึกษา และเป็นกำลังใจในการทำงานเป็นอย่างดีเสมอมา

และท้ายสุดนี้ ข้าพเจ้าน้อมรำลึกถึงพระคุณของบิดามารดา และครอบครัวที่ส่งเสริมและสนับสนุนข้าพเจ้าในทุก ๆ เรื่องจนกระทั่งข้าพเจ้าสำเร็จการศึกษา



## สารบัญ

	หน้า
บทคัดย่อ.....	(5)
กิตติกรรมประกาศ.....	(7)
สารบัญ.....	(8)
รายการตาราง.....	(10)
รายการภาพประกอบ.....	(11)
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 ตรวจสอบเอกสาร.....	2
1.3 วัตถุประสงค์.....	9
1.4 ขอบเขต.....	9
1.5 ขั้นตอนและวิธีดำเนินงาน.....	10
1.6 เครื่องมือที่ใช้.....	12
1.7 ประโยชน์ที่คาดว่าจะได้รับ.....	12
บทที่ 2 ทฤษฎีและหลักการ.....	13
2.1 เครื่องข่ายเซนเซอร์ไร้สาย.....	13
2.2 เครื่องข่ายเซนเซอร์ไร้สายกับงานด้านการเฝ้าระวังสุขภาพ.....	14
2.3 MAC โพรโทคอลบนเครือข่ายเซนเซอร์ไร้สาย.....	16
2.4 การทำงานของ MAC โพรโทคอลบนมาตรฐาน IEEE 802.15.4.....	22
2.5 TinyOS.....	26
บทที่ 3 การออกแบบและวิธีทดสอบระบบ.....	28
3.1 วิธีการปรับ Duty-cycle บนมาตรฐาน IEEE 802.15.4.....	28
3.2 การออกแบบกระบวนการปรับ Duty-cycle อัตโนมัติ.....	30

3.3 วิธีการคำนวณหาค่า Duty-cycle ที่เหมาะสม .....	32
บทที่ 4 การทดสอบและวิเคราะห์ประสิทธิภาพ .....	49
4.1 เครื่องมือที่ใช้ในการทดสอบ .....	49
4.2 วิธีการและสภาพแวดล้อมในการทดสอบ .....	52
4.3 ทดสอบอัตราการใช้พลังงานของโพรโทคอลแบบไม่ควบคุมระยะเวลาหน่วง .....	54
4.4 ทดสอบอัตราการใช้พลังงานของโพรโทคอลแบบควบคุมระยะเวลาหน่วง .....	58
4.5 กำหนดระยะเวลาการทำงานบนแบตเตอรี่ .....	64
4.6 ทดสอบการทำงานของเครือข่ายเมื่อมีเซนเซอร์นับก้าวและตรวจจับการหลัดล้ม .....	67
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ .....	69
5.1 สรุปผล .....	69
5.2 ปัญหาและอุปสรรค .....	70
5.3 ข้อเสนอแนะ .....	70
บรรณานุกรม .....	71
ภาคผนวก .....	74
ภาคผนวก ก ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์ .....	75
ประวัติผู้เขียน .....	82

## รายการตาราง

	หน้า
ตารางที่ 2.1 อัตราการส่งข้อมูลของเซนเซอร์ชนิดต่างๆในทางการแพทย์.....	14
ตารางที่ 3.1 Symbol-to-chip mapping [1] .....	36
ตารางที่ 3.2 กลิ่นความถี่และอัตราการส่งข้อมูล [1].....	36
ตารางที่ 4.1 ค่าพารามิเตอร์ที่ใช้ในการทดสอบ .....	53

## รายการภาพประกอบ

	หน้า
รูปที่ 1.1 วิธีการรับ-ส่งข้อมูลของ S-MAC [12].....	3
รูปที่ 1.2 วิธีการรับ-ส่งข้อมูลของ WiseMAC [15] .....	5
รูปที่ 1.3 แผนผังการส่งข้อมูลแบบ Tree ของโปรโตคอล DMAC .....	6
รูปที่ 1.4 การหลับของโหนดภายในระยะเวลา TA ของ T-MAC .....	7
รูปที่ 1.5 การเพิ่ม Duty-cycle เป็นสองเท่าของ DSMAC.....	8
รูปที่ 2.1 โครงสร้างการส่งข้อมูลของ Body Area Network [32].....	15
รูปที่ 2.2 โครงสร้างหลักของ Superframe .....	23
รูปที่ 2.3 การทำงานของ Superframe.....	24
รูปที่ 2.4 ขั้นตอนการส่งข้อมูลจาก Device สู่ Coordinator ในเครือข่ายแบบ Beacon Enable.....	25
รูปที่ 2.5 ขั้นตอนการส่งข้อมูลจาก Coordinator สู่ Device ในเครือข่ายแบบ Beacon Enable.....	26
รูปที่ 3.1 รายละเอียดของ Superframe [1].....	28
รูปที่ 3.2 ลักษณะ Duty-cycle แบบคงที่.....	30
รูปที่ 3.3 Coordinator ขณะที่ยังไม่มี Device เข้ามาในเครือข่าย.....	31
รูปที่ 3.4 เมื่อมี Device Associate เข้ามาในเครือข่าย.....	31
รูปที่ 3.5 รูปแบบของ Superframe [1] .....	33
รูปที่ 3.6 กระบวนการ Spreading และ Modulation สำหรับคลื่น 2.4 GHz [59].....	35
รูปที่ 3.7 การใช้พลังงานรับ-ส่งข้อมูลของ Coordinator.....	37
รูปที่ 3.8 ลักษณะการใช้พลังงานของ Device ขณะส่งข้อมูลแต่ละเฟรม .....	39
รูปที่ 3.9 ลักษณะของเฟรม MAC บนมาตรฐาน IEEE 802.15.4 [1].....	40
รูปที่ 4.1 Unode ที่พัฒนาขึ้นมาโดยห้องวิจัยเครือข่ายเซนเซอร์ไร้สาย .....	49
รูปที่ 4.2 NI USB-6009 สำหรับใช้ในการวัดค่าพลังงาน .....	51
รูปที่ 4.3 การต่อ Unode เข้ากับ NI USB-6009 เพื่อวัดพลังงาน .....	52
รูปที่ 4.4 อัตราการใช้พลังงานเปรียบเทียบกับอัตราการส่งข้อมูล (1-50 Bps).....	54
รูปที่ 4.5 อัตราการใช้พลังงานเปรียบเทียบกับอัตราการส่งข้อมูล (51-1000 Bps).....	55
รูปที่ 4.6 อัตราการใช้พลังงานเปรียบเทียบกับอัตราการส่งข้อมูล (1001-8000 Bps).....	55
รูปที่ 4.7 Latency สูงสุดเปรียบเทียบกับอัตราการส่งข้อมูล (1-50 Bps).....	56
รูปที่ 4.8 Latency สูงสุดเปรียบเทียบกับอัตราการส่งข้อมูล (51-1000 Bps).....	57

รูปที่ 4.9 Latency สูงสุดเปรียบเทียบกับอัตราการส่งข้อมูล (1001-8000 Bps).....	57
รูปที่ 4.10 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที (1-50 Bps).....	58
รูปที่ 4.11 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที (51-1000 Bps) .....	59
รูปที่ 4.12 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที (1001-8000 Bps).....	59
รูปที่ 4.13 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที (1-50 Bps) .....	60
รูปที่ 4.14 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที (51-1000 Bps) .....	60
รูปที่ 4.15 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที(1001-8000 Bps) .....	61
รูปที่ 4.16 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1วินาที (1-50 Bps).....	61
รูปที่ 4.17 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1วินาที (51-1000 Bps) .....	62
รูปที่ 4.18 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1วินาที(1000-8000 Bps) .....	62
รูปที่ 4.19 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1วินาที (1-50 Bps) .....	63
รูปที่ 4.20 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1วินาที (51-1000 Bps) .....	63
รูปที่ 4.21 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1วินาที(1001-8000 Bps) .....	64
รูปที่ 4.22 อายุแบตเตอรี่เปรียบเทียบกับอัตราการส่งข้อมูล (1-100 Bps).....	65
รูปที่ 4.23 อายุแบตเตอรี่เปรียบเทียบกับอัตราการส่งข้อมูล (101-8000 Bps).....	65
รูปที่ 4.24 อายุแบตเตอรี่เทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4 วินาที (1-8000 Bps) ..	66
รูปที่ 4.25 อายุแบตเตอรี่เทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1 วินาที (1-8000 Bps) ..	66

## รายการตัวอย่าง Code

	หน้า
รายการที่ 3.1 ตัวอย่าง Code การหาค่า SO.....	42
รายการที่ 3.2 ตัวอย่างการเรียกใช้งานฟังก์ชัน find_so().....	43
รายการที่ 3.3 ตัวอย่าง Code การวนหาค่า SO และ BO ที่ดีที่สุด .....	45
รายการที่ 3.4 ตัวอย่างการเรียกใช้งานฟังก์ชัน find_so_bo().....	45
รายการที่ 3.5 ตัวอย่าง Code การหาค่า BO จากค่า Latency .....	47
รายการที่ 3.6 ตัวอย่าง Code การเรียกใช้งานฟังก์ชัน get_bo() .....	47
รายการที่ 3.7 การใช้งานฟังก์ชัน find_so_bo() โดยกำหนดค่า Latency .....	48

## บทที่ 1

### บทนำ

#### 1.1 ที่มาและความสำคัญ

โรงพยาบาลในปัจจุบันมีจำนวนผู้ป่วยมาเข้ารับการรักษาเป็นจำนวนมาก การเฝ้าดูแลผู้ป่วยภายในโรงพยาบาลให้ทั่วถึงตลอดเวลานั้นเป็นไปได้ยาก เนื่องจากจำนวนพยาบาลมีน้อยกว่าจำนวนผู้ป่วยที่เข้ามาับการรักษาเป็นจำนวนมาก จึงไม่สามารถที่จะให้พยาบาลเฝ้าคอยติดตามผู้ป่วยในโรงพยาบาลได้ในทุกสถานที่และทุกเวลา แต่ผู้ป่วยทุกคนนั้นมีความจำเป็นที่จะต้องได้รับการเฝ้าระวังอย่างต่อเนื่องจากอาจจะเกิดอุบัติเหตุกับผู้ป่วยได้ตลอดเวลา เช่น การหมดสติหกล้ม หรือ ลื่นหกล้ม โดยเฉพาะผู้ป่วยที่มีร่างกายไม่แข็งแรงหรือผู้ป่วยที่เป็นผู้สูงอายุ ซึ่งในการเฝ้าติดตามบุคคลเหล่านี้จำเป็นจะต้องมีอุปกรณ์ “เฝ้าระวังสุขภาพ (Health Care Monitoring)” เข้ามาช่วย

Low-Rate Wireless Personal Area Network (LR-WPAN)[1] เป็นเทคโนโลยีในการสื่อสารที่เหมาะสมที่สุดในขณะนี้ในการนำมาใช้งานด้านการเฝ้าระวังสุขภาพ เนื่องจาก LR-WPAN ถูกออกแบบมาให้ใช้ในการส่งข้อมูลในระยะสั้น และแตกต่างจากระบบเครือข่าย Wireless Local Area Network (WLAN)[2] ที่ LR-WPAN สามารถเชื่อมต่อเครือข่ายได้โดยไม่ต้องใช้โครงสร้างเครือข่ายพื้นฐาน ด้วยคุณลักษณะนี้ได้ช่วยให้อุปกรณ์ที่ใช้ในเครือข่าย LR-WPAN หรือที่เรียกว่า “โหนด” นั้นสามารถพัฒนามาให้มีขนาดเล็ก ประหยัดพลังงานและราคาไม่แพงได้ โดยอุปกรณ์ที่ได้พัฒนาขึ้นมาใช้งานแล้วในปัจจุบันนั้นก็มียี่ห้อเช่น Tmote-Sky[3], XBee[4] และ EPIC\_mote[5] เป็นต้น ซึ่งอุปกรณ์ทุกตัวก็ได้เลือกใช้และออกแบบวงจรและระบบให้ประหยัดพลังงานมากที่สุดแต่ด้วยข้อจำกัดของการออกแบบด้านฮาร์ดแวร์ ทำให้การประหยัดพลังงานก็สามารถทำได้ในระดับหนึ่งเท่านั้นจึงจำเป็นต้องมีการออกแบบซอฟต์แวร์เข้ามาช่วยในการจัดการเรื่องพลังงานด้วย เพื่อเพิ่มประสิทธิภาพของการใช้พลังงานให้ได้สูงที่สุด

องค์ประกอบของโหนดนั้นจะแบ่งเป็นสองส่วนหลักๆคือ ส่วนของหน่วยประมวลผล (ไมโครคอนโทรลเลอร์) และส่วนของภาครับส่งสัญญาณวิทยุ ซึ่งจากการทดลองและการศึกษา พบว่าภาครับส่งสัญญาณวิทยุของโหนดนั้น เป็นส่วนที่ใช้พลังงานมากที่สุด[6] โดยพลังงานที่สูญเสียไปส่วนใหญ่เกิดจากการรอรับข้อมูลโดยเปล่าประโยชน์ (Idle Listening) จึงจำเป็นต้องใช้โพรโทคอลการสื่อสาร Media access control (MAC)[7] ที่มีประสิทธิภาพเพื่อช่วยให้สามารถใช้งานให้เกิดประโยชน์สูงสุด

โพรโทคอล MAC นั้นถูกออกแบบมาเพื่อให้สามารถเข้าใช้ช่องสัญญาณในการสื่อสารได้โดยหลีกเลี่ยงการชน และตั้งเวลาในการส่งข้อมูล ซึ่งต้องได้อัตราการส่งข้อมูลที่ดีที่สุด มีอัตราการหน่วงของข้อมูลที่ยอมรับได้ และต้องใช้พลังงานอย่างคุ้มค่าที่สุด โดยใช้เทคนิคการเปิดภาครับส่งสัญญาณวิทยุเมื่อจำเป็นจะต้องใช้เท่านั้นซึ่งโพรโทคอล MAC สำหรับ LR-WPAN ในปัจจุบันนั้นก็ได้มีการออกแบบไว้มากมายแต่ที่เป็นที่ยอมรับและได้รับมาตรฐานคือ IEEE 802.15.4[1]

IEEE 802.15.4 เป็นโพรโทคอลที่ถูกออกแบบมาสำหรับ LR-WPAN ควบคุมการใช้พลังงานโดยการ ปิด (หลับ) – เปิด (ตื่น) ของวงจรวิทยุตามคาบเวลา (Duty-cycle) [8] โดยอุปกรณ์ทุกตัวจะต้องตื่นพร้อมกันเพื่อมารับส่งข้อมูล และไม่สามารถเปลี่ยนค่า Duty-cycle ระหว่างการทำงานได้ ทำให้อาจจะเปลืองพลังงานโดยเปล่าประโยชน์

เนื่องจากการเฝ้าระวังสุขภาพนั้น มีความต้องการในการส่งข้อมูลของเซนเซอร์แต่ละชนิดไม่เท่ากัน ทั้งในเรื่องของอัตราการส่งข้อมูลและ Latency ดังนั้นในงานวิจัยนี้จะมุ่งเน้นการวิเคราะห์ และหาวิธีให้ MAC โพรโทคอลบนมาตรฐาน IEEE 802.15.4 นั้นสามารถปรับเปลี่ยน Duty-cycle ของตัวเองได้ตามความต้องการในการใช้ทรัพยากรของเซนเซอร์แต่ละชนิด เพื่อให้การใช้พลังงานของโหนดมีความคุ้มค่าที่สุด

## 1.2 ทรวจเอกสาร

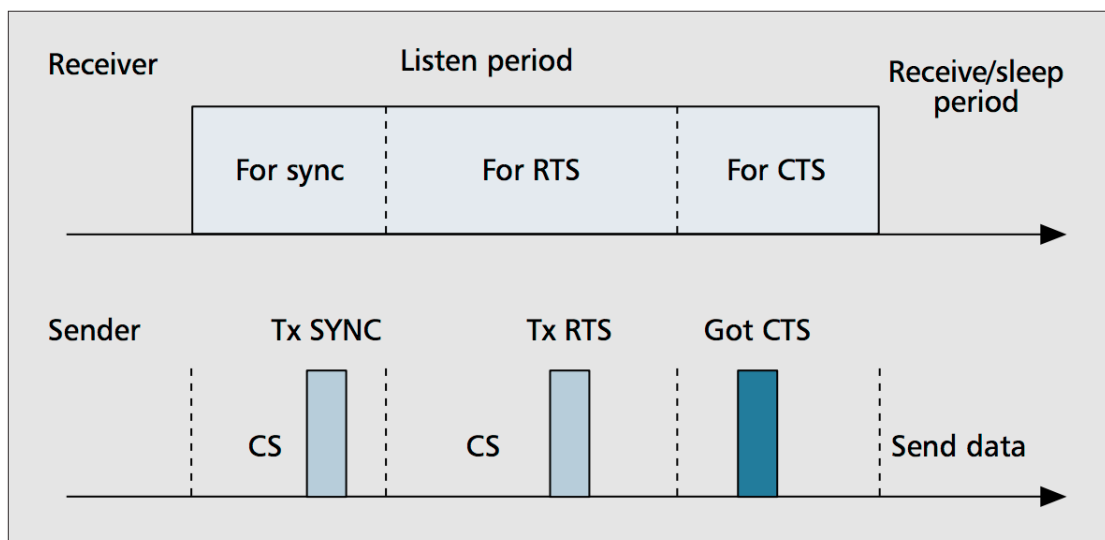
จากการศึกษา MAC โพรโทคอลสำหรับเครือข่ายเซนเซอร์ไร้สาย พบว่าสามารถแบ่งได้ออกได้เป็น 2 ประเภทหลัก[6] คือ ประเภทที่ใช้การแย่งชิงช่องสัญญาณ และประเภทที่ใช้การจัดตาราง การรับ-ส่งข้อมูล ซึ่งประเภทที่ใช้การจัดตาราง การรับ-ส่งข้อมูล สามารถที่จะหลีกเลี่ยงการชนกันของข้อมูล และการรบกวนข้อมูลโดยไม่มีข้อมูลส่งออกมาได้ โดยการจัดการการรับและส่งข้อมูลตามคาบเวลาที่กำหนดไว้ซึ่งกระบวนการเข้าถึงช่องสัญญาณ ในลักษณะนี้เรียกว่า Time Division Multiple Access (TDMA)[9] แต่ MAC โพรโทคอลประเภทนี้จะต้องอาศัยการ Synchronize เวลาที่แม่นยำมาก ส่วนประเภทที่ใช้การแย่งชิงช่องสัญญาณ จะไม่ต้องการการ Synchronize เวลาที่แม่นยำมากนัก และสามารถที่จะปรับเปลี่ยนรูปแบบการเชื่อมต่อของเครือข่ายได้ง่ายกว่า MAC โพรโทคอลประเภทที่ใช้การจัดตาราง การรับ-ส่งข้อมูล โดย MAC โพรโทคอลประเภทที่ใช้การแย่งชิงช่องสัญญาณนี้ จะใช้หลักการเข้าถึงช่องสัญญาณแบบ Carrier Sense Multiple Access (CSMA)[10]



ในปัจจุบันได้มีการพัฒนา MAC โพรโทคอลสำหรับเครือข่ายเซนเซอร์ไร้สาย ไปได้หลายชนิดด้วยกัน ซึ่ง MAC โพรโทคอลแต่ละชนิดก็มีจุดเด่นและข้อดี-ข้อเสีย[11] ที่แตกต่างกันออกไปซึ่งสามารถสรุปได้ดังนี้

### 1.2.1 Sensor-MAC (S-MAC)[12]

S-MAC เป็น MAC โพรโทคอลที่ถูกออกแบบขึ้นมาเพื่อใช้งานกับเครือข่ายเซนเซอร์ไร้สายในยุคแรก ซึ่งได้ปรับปรุงโพรโทคอลมาจาก โหมดประหยัดพลังงานของมาตรฐาน IEEE 802.11 ทำงานโดยการกำหนดช่วงเวลาหลับ และตื่น ของอุปกรณ์เพื่อประหยัดพลังงาน โดยการสื่อสารจะทำได้เฉพาะช่วงเวลาที่อุปกรณ์ตื่นเท่านั้น และการเข้าถึงช่องสัญญาณจะใช้วิธีการแย่งชิงช่องสัญญาณ โดยการส่งเฟรมควบคุมดังรูปที่ 1.1 ซึ่งประกอบด้วย SYNC RTS/CTS[13] และ ACK โดย SYNC จะทำหน้าที่กำหนดจุดเริ่มต้นของการสื่อสาร และหลังจากนั้นหากมีอุปกรณ์ตัวใดต้องการส่งข้อมูลจะส่ง RTS ออกมาเพื่อขอส่งข้อมูล และอุปกรณ์ตัวที่ต้องการรับข้อมูลจะตอบ CTS กลับมาหากพร้อมที่จะรับข้อมูล เมื่อผู้ที่ต้องการจะส่งข้อมูลได้รับสัญญาณ CTS ก็จะส่งข้อมูลออกมาให้ผู้รับ เมื่อผู้รับสามารถรับข้อมูลได้ครบก็จะตอบ ACK ไปยังผู้ส่งเพื่อยืนยันการได้รับข้อมูล



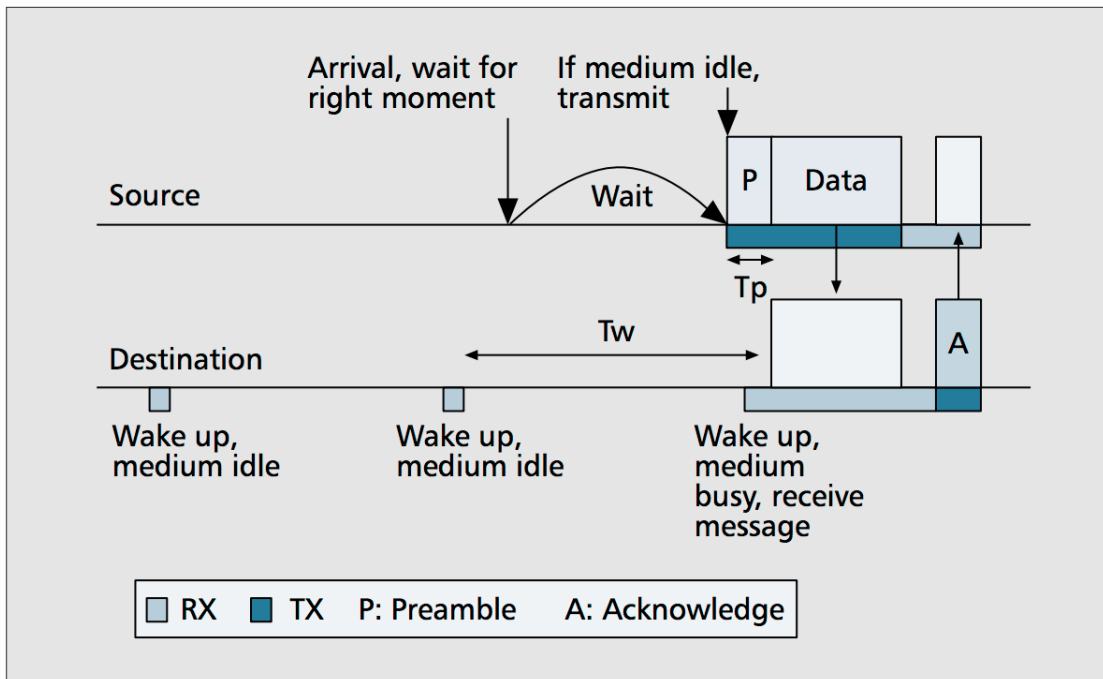
รูปที่ 1.1 วิธีการรับ-ส่งข้อมูลของ S-MAC [12]

ข้อดีของ S-MAC คือ สามารถลดการใช้พลังงานที่เกิดจากการฟังโดยไม่มีข้อมูลส่งออกมาได้ ด้วยการหลับ และตื่น ตามคาบเวลา นอกจากนี้ยังเป็น โพรโทคอลที่สามารถนำไปพัฒนาได้ง่าย เนื่องจากไม่มีความซับซ้อนมากนัก

ข้อเสียของ S-MAC คือ การส่งข้อมูลแบบ Broadcast[14] นั้นจะไม่ใช้เฟรมควบคุม RTS/CTS ซึ่งทำเพิ่มโอกาสการชนกันของข้อมูล และคาบการหลับ และตื่นของอุปกรณ์นั้น ถูกกำหนดค่าไว้ล่วงหน้า ไม่สามารถเปลี่ยนแปลงได้ ซึ่งจะทำให้ประสิทธิภาพของการใช้พลังงานลดลงเมื่อใช้งานในเครือข่ายที่มีอัตราการส่งข้อมูลที่ไม่แน่นอน

### 1.2.2 WiseMAC[15]

WiseMAC ได้พัฒนามาจากแนวคิดของ โพรโทคอลชื่อ “Spatial TDMA and CSMA with Preamble Sampling”[16] ที่คิดค้น โดย A. El-Hoiydi ซึ่งใช้การสื่อสารแบบสองช่องสัญญาณ โดยช่องสัญญาณแรกจะใช้สำหรับการส่งข้อมูล และใช้การเข้าถึงช่องสัญญาณแบบ TDMA ส่วนช่องสัญญาณที่สอง ใช้สำหรับส่งคำสั่งควบคุม และใช้การเข้าถึงช่องสัญญาณแบบ CSMA แต่ WiseMAC จะใช้ช่องสัญญาณเพียงช่องเดียว และใช้วิธีการเข้าถึงช่องสัญญาณแบบ non-persistent CSMA (np-CSMA)[10] โดยใช้กระบวนการ Preamble sampling แบบเดียวกับ โพรโทคอล Spatial TDMA and CSMA with Preamble Sampling เพื่อลดการรบกวนข้อมูล โดยไม่มีข้อมูลส่งออกมา ซึ่งวิธีการทำงานของ Preamble Sampling นั้นได้แสดงไว้ดังรูปที่ 1.2 คือ จะมีการส่งสัญญาณ Preamble ออกมาก่อนที่จะส่งข้อมูลแต่ละเฟรม เพื่อใช้สำหรับบอกโหนดตัวที่จะรับข้อมูล ให้รอรับข้อมูล ซึ่งเมื่อโหนดตัวที่ต้องการจะรับข้อมูลตื่นขึ้นมา แล้วพบว่าช่องสัญญาณไม่ว่าง โหนดตัวนั้นก็รอรับข้อมูลต่อไป จนกว่าจะได้รับข้อมูลหรือจนกว่าช่องสัญญาณจะว่างอีกครั้ง และเมื่อได้รับข้อมูลแล้วก็จะตอบ Acknowledge กลับไปยังตัวที่ส่งข้อมูล เพื่อยืนยันการได้รับข้อมูล



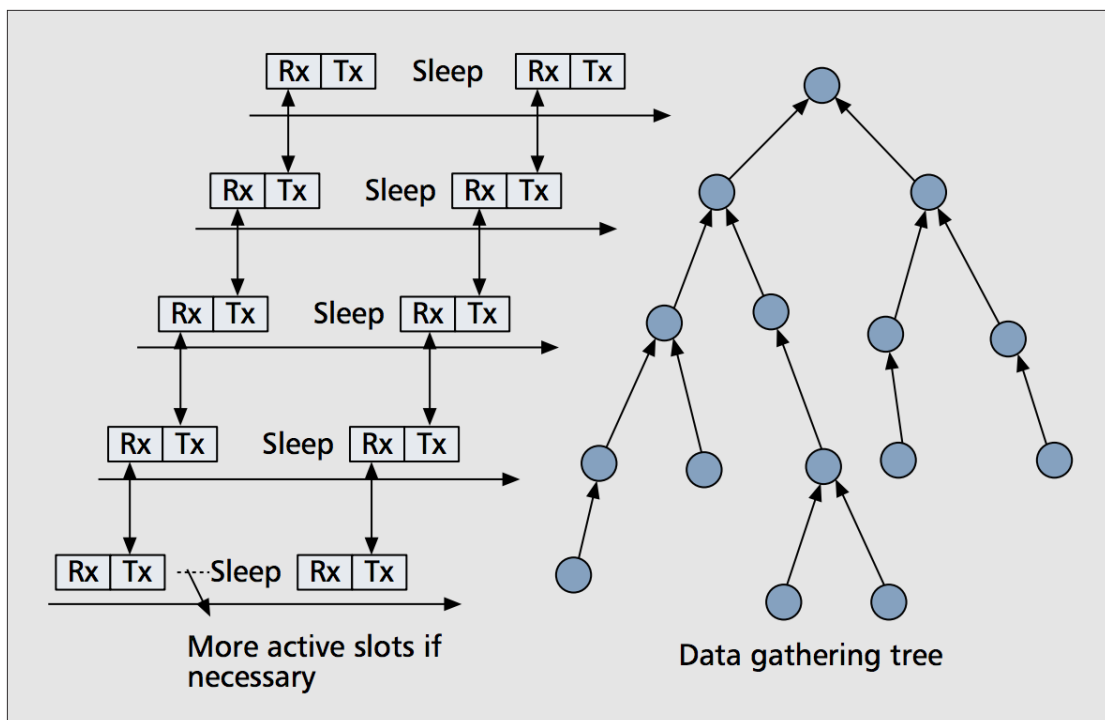
รูปที่ 1.2 วิธีการรับ-ส่งข้อมูลของ WiseMAC [15]

ข้อดีของ WiseMAC คือ ผลจากการ Simulation พบว่าประสิทธิภาพของ WiseMAC นั้นทำได้ดีกว่า S-MAC ภายใต้สภาพเครือข่ายที่มีอัตราการส่งข้อมูลไม่คงที่ และยังสามารถจัดการปัญหา ความเหลื่อมล้ำของเวลาได้ ทำให้ลดการพึ่งพาการ Synchronize เวลาจากภายนอก

ข้อเสียของ WiseMAC คือ การที่ WiseMAC ไม่ได้มีการ Synchronize ตารางการหลับและตื่นให้ตรงกันในแต่ละโหนดที่อยู่ใกล้กัน ทำให้เกิดปัญหาคือ เมื่อมีการส่งข้อมูลแบบ Broadcast ข้อมูลจะต้องถูกเก็บไว้สำหรับโหนดที่อยู่ใกล้กัน หากโหนดนั้นหลับอยู่ และจะส่งออกไปก็ต่อเมื่อโหนดนั้นได้ตื่นขึ้นมา ซึ่งทำให้มี Latency เพิ่มขึ้น และใช้พลังงานมากขึ้น

### 1.2.3 DMAC [17]

รูปแบบการส่งข้อมูลในเครือข่ายเซนเซอร์ไร้สาย ส่วนใหญ่จะส่งข้อมูลไปในทิศทางเดียวคือ เมื่อโหนดแต่ละตัวเก็บข้อมูลจากเซนเซอร์ได้ ก็จะส่งข้อมูลไปยังศูนย์กลางเพียงที่เดียว ไม่มีการส่งข้อมูลกลับไปยังเซนเซอร์โหนด หรืออาจจะเรียกลักษณะการส่งข้อมูลแบบนี้ได้ว่า “การส่งข้อมูลแบบ Trees” ดังนั้น โพรโทคอล DMAC จึงได้ถูกออกแบบมาเพื่อใช้กับการส่งข้อมูลแบบ Trees โดยเน้นในเรื่องของ Latency ที่ต่ำ และยังคงประหยัดพลังงานอยู่ ซึ่งโพรโทคอล DMAC นี้ได้พัฒนามาจากอัลกอริทึม Slotted ALOHA[18][19] โดยการแบ่ง Slots ให้กับโหนดที่อยู่ในเส้นทางการส่งข้อมูลดังรูปที่ 1.3 จากนั้นเมื่อถึงช่วงการรับข้อมูล ทุกตัวที่เป็นโหนดลูกจะส่งข้อมูลเพื่อแข่งชิงการส่งข้อมูล และเส้นทางที่ส่งข้อมูลได้สำเร็จก็จะได้ Slots สำหรับส่งข้อมูล ซึ่งกระบวนการนี้สามารถลด Latency ของเครือข่ายลงได้



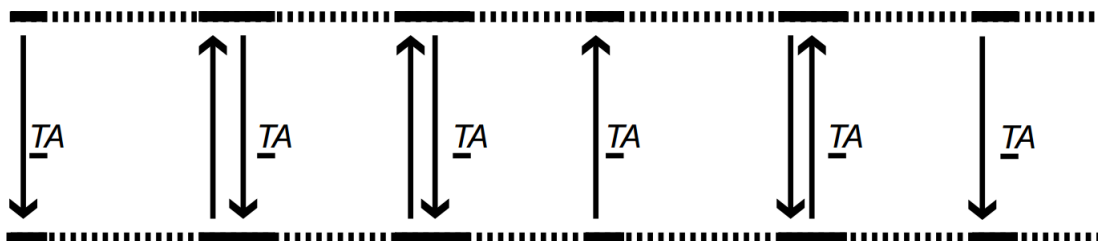
รูปที่ 1.3 แผนผังการส่งข้อมูลแบบ Tree ของโพรโทคอล DMAC

ข้อดีของ DMAC คือ มี Latency ที่ต่ำมากเมื่อเปรียบเทียบกับโพรโทคอล MAC ตัวอื่นๆ ที่ใช้การหลบ และคืน ในการประหยัดพลังงาน

ข้อเสียของ DMAC คือ ไม่มีการนำกระบวนการเพื่อหลีกเลี่ยงการชนกันของข้อมูลมาใช้งาน ดังนั้นเมื่อมีโหนดที่มีตารางการหลับ และตื่นที่พร้อมกัน และอยู่ในลำดับชั้นของ Tree เดียวกัน ก็จะเกิดการชนกันของข้อมูลเกิดขึ้น

#### 1.2.4 Timeout-MAC (T-MAC) [20]

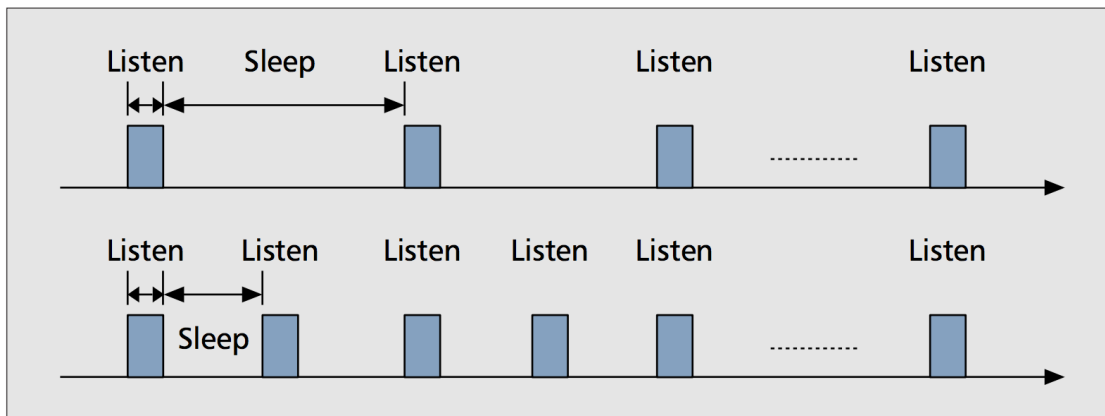
S-MAC เป็นโพรโทคอลที่ใช้คาบเวลาการหลับ และตื่น แบบคงที่ ซึ่งส่งผลให้มี Latency ที่สูง และมีอัตราการส่งข้อมูลต่ำ ดังนั้น จึงได้มีการคิดค้น T-MAC ขึ้นมา เพื่อปรับปรุงปัญหาส่วนนี้ใน S-MAC ที่มีประสิทธิภาพต่ำลง เมื่ออัตราการส่งข้อมูลในเครือข่ายไม่คงที่ โดยใน T-MAC นั้น การตื่นมาฟังข้อมูลจะสิ้นสุดลงและกลับสู่การหลับ เมื่อไม่มีการรับ-ส่ง ข้อมูลภายใน ช่วงเวลาที่กำหนด (TA) ดังรูปที่ 1.4 และนอกจากนี้ T-MAC ยังสามารถแก้ปัญหาการหลับก่อนกำหนดได้อีกด้วย ซึ่งได้อธิบายวิธีการไว้ใน [20] ซึ่งผลการจากทดสอบ[20]โพรโทคอล T-MAC พบว่า มีประสิทธิภาพที่ดีขึ้น ภายได้เครือข่ายที่มีอัตราการส่งข้อมูลไม่คงที่



รูปที่ 1.4 การหลับของโหนดภายในระยะเวลา TA ของ T-MAC

#### 1.2.5 Dynamic Sensor-MAC (DSMAC) [21]

DSMAC เป็นโพรโทคอลที่พัฒนาต่อมาจาก S-MAC ซึ่งเพิ่มความสามารถในการปรับเปลี่ยน Duty-cycle เพื่อลด Latency ของเครือข่าย โดยในช่วงการ SYNC ทุกโหนดจะรายงาน Latency ของตัวเอง (เวลาตั้งแต่รับข้อมูลเข้ามาในคิว จนกระทั่งได้ส่งข้อมูลออกไป) และในช่วงเริ่มต้น ทุกโหนดจะมี Duty-cycle ที่ตรงกัน เมื่อตัวที่ต้องการรับข้อมูลพบว่าค่าเฉลี่ย Latency มีค่าสูงเกินไป มันตัดสินใจลดเวลาการหลับของตัวเองลง และประกาศให้โหนดอื่นรับรู้ผ่าน ช่วงเวลา SYNC เมื่อตัวที่ต้องการจะส่งข้อมูล ได้รับสัญญาณการลดช่วงเวลาการหลับของตัวรับข้อมูล มันจะตรวจสอบคิวสำหรับส่งข้อมูลไปยังปลายทาง หากมีค่าเป็นหนึ่ง มันจะตัดสินใจเพิ่ม Duty-cycle เป็นสองเท่าดังรูปที่ 1.5 หากพลังงานในแบตเตอรี่อยู่ในช่วงที่กำหนด เพื่อลด Latency ลง



รูปที่ 1.5 การเพิ่ม Duty-cycle เป็นสองเท่าของ DSMAC

การเพิ่ม Duty-cycle เป็นสองเท่าของโปรโตคอล DSMAC นั้น เพื่อให้ไม่ส่งผลกระทบกับตารางเวลาของโหนดข้างเคียง และด้วยกระบวนการของ DSMAC นี้ทำให้สามารถลด Latency ลงได้ และยังมีประสิทธิภาพในการใช้พลังงานที่ดีกว่า S-MAC เมื่อเฉลี่ยเป็นค่าพลังงานต่อ Packet

#### 1.2.6 IEEE 802.15.4 [1]

IEEE 802.15.4 เป็นมาตรฐานที่ออกแบบมาสำหรับการสื่อสารแบบไร้สายที่ความเร็วต่ำสำหรับอุปกรณ์ราคาถูกลง ซึ่งได้รับการดูแลโดยคณะกรรมการ IEEE 802.15 โดย IEEE 802.15.4 นี้เป็นที่นิยมอย่างมากและถูกนำไปใช้เป็นโปรโตคอลระดับล่างของโปรโตคอลระดับบนหลายชนิด เช่น ZigBee[21] ISA100.11a[22] WirelessHART[23] และ MiWi[24] เป็นต้น

IEEE 802.15.4 ใช้กระบวนการเข้าถึงช่องสัญญาณแบบ CSMA/CA และสามารถส่งข้อมูลด้วยอัตราส่งข้อมูลสูงสุดที่ 250 kbps บนคลื่นความถี่ 2.4 GHz โดยแบ่งชนิดของโหนดในเครือข่ายได้เป็น 2 ชนิด คือ Full-function device และ Reduced-function device ซึ่งรายละเอียดของโปรโตคอลจะอธิบายเพิ่มเติมในหัวข้อ 2.4 เรื่องการทำงานของ MAC โปรโตคอลบนมาตรฐาน IEEE 802.15.4

MAC โพรโทคอลสำหรับเครือข่ายเซนเซอร์ไร้สายนั้นมีหลายชนิด ซึ่งแต่ละชนิดก็ได้พัฒนาขึ้นมาเพื่อจุดประสงค์การใช้งานที่แตกต่างกันออกไป แต่ MAC โพรโทคอลที่ใช้กันอย่างแพร่หลายมากที่สุดในปัจจุบันก็คือ มาตรฐาน IEEE 802.15.4 ซึ่งนอกจากจะได้รับความนิยมในการใช้งานแล้ว ยังมีฮาร์ดแวร์ และซอฟต์แวร์ ให้เลือกใช้ในการใช้งานจริงอีกด้วย เช่น ฮาร์ดแวร์โมดูลวิทยุ CC2420[25] ที่สามารถทำงานร่วมกับซอฟต์แวร์ TinyOS[26] ได้ เป็นต้น ซึ่งต่างจาก MAC โพรโทคอลชนิดอื่นๆ ที่ส่วนใหญ่จะเป็นแค่แนวคิด ที่จำลองขึ้นมา ไม่ได้มีการนำไปใช้ในอุปกรณ์จริงอย่างแพร่หลาย ดังนั้น ผู้วิจัยจึงเลือก MAC โพรโทคอลบนมาตรฐาน IEEE 802.15.4 มาวิจัยและปรับปรุง ในเรื่องของ การปรับเปลี่ยน Duty-cycle แบบอัตโนมัติ เพื่อให้เหมาะสมกับการนำมาใช้งานในการเฝ้าระวังสุขภาพ

### 1.3 วัตถุประสงค์

เพื่อพัฒนาและปรับปรุง MAC โพรโทคอลบนมาตรฐาน IEEE 802.15.4 ให้ใช้พลังงานอย่างมีประสิทธิภาพ และสามารถปรับตัวเข้ากับสถานการณ์ที่แตกต่างกัน ในงานด้านการเฝ้าระวังสุขภาพ

### 1.4 ขอบเขต

วิเคราะห์เฉพาะ โพรโทคอลชั้น MAC เท่านั้น และใช้โพรโทคอลชั้น PHY ตามมาตรฐานของ IEEE 802.15.4-2006 โดยทำการทดลองด้วยอุปกรณ์เซนเซอร์โหนดไร้สายจริงของ Platform TelosB (Unode) บนโหนดที่ห้องวิจัยได้พัฒนาขึ้น

### 1.5 ขั้นตอนและวิธีดำเนินงาน

การดำเนินงานในการทำวิทยานิพนธ์ชุดนี้ มีขั้นตอนการทำงานดังต่อไปนี้

1. ศึกษาค้นคว้าข้อมูลเอกสารที่เกี่ยวข้องกับวิทยานิพนธ์ดังนี้
  - เครื่องข่ายเซนเซอร์ไร้สาย
  - เครื่องข่ายเซนเซอร์ไร้สายกับงานด้านการเฝ้าระวังสุขภาพ
  - MAC โพรโทคอลบนเครื่องข่ายเซนเซอร์ไร้สาย
  - การทำงานของ MAC โพรโทคอลบนมาตรฐาน IEEE 802.15.4
  - TinyOS
2. วิเคราะห์และปรับปรุง MAC โพรโทคอล
  - วิธีการปรับ Duty-cycle บนมาตรฐาน IEEE 802.15.4
  - การออกแบบกระบวนการปรับ Duty-cycle อัตโนมัติ
  - วิธีการคำนวณหาค่า Duty-cycle
3. ทดสอบและแก้ไขโปรโทคอล
  - เครื่องมือที่ใช้ในการทดสอบ
  - วิธีการและสภาพแวดล้อมในการทดสอบ



- ทดสอบอัตราการใช้พลังงานของโพรโทคอลแบบไม่ควบคุมระยะเวลาหนึ่ง
- ทดสอบอัตราการใช้พลังงานของโพรโทคอลแบบควบคุมระยะเวลาหนึ่ง
- กำหนดระยะเวลาการทำงานบนแบตเตอรี่
- ทดสอบการทำงานของเครือข่ายเมื่อมีเซนเซอร์นับก้าวและตรวจจับการหก  
ดื่ม

#### 4. จัดทำเอกสารรายงานผลการทำวิจัยฉบับสมบูรณ์

## 1.6 เครื่องมือที่ใช้

### 1. ด้านฮาร์ดแวร์

- เครื่องคอมพิวเตอร์ ความเร็วหน่วยประมวลผลที่ 2.66 GHz และมีหน่วยความจำ 4 GB ฮาร์ดดิสก์ความจุ 320 กิกะไบต์ จำนวน 1 ชุด
- Unode ที่ทางห้องวิจัยพัฒนาขึ้นมา
- NI USB-6009 Multifunction Data Acquisition (DAQ)

### 2. ด้านซอฟต์แวร์

- ระบบปฏิบัติการ Microsoft Window XP
- Java 1.6 JDK
- Cygwin
- gcc native compiler
- nesC compiler
- TinyOS source tree
- TI MSP430 Tools

## 1.7 ประโยชน์ที่คาดว่าจะได้รับ

ระบบเครือข่ายเซนเซอร์ไร้สายมีการใช้พลังงานอย่างมีประสิทธิภาพ เพื่อนำไปใช้ในงานด้านการเฝ้าระวังสุขภาพ

## บทที่ 2

### ทฤษฎีและหลักการ

ในบทนี้จะอธิบายถึงทฤษฎีและหลักการที่เกี่ยวข้องกับวิทยานิพนธ์ฉบับนี้ ซึ่งประกอบด้วย ระบบเครือข่ายเซนเซอร์ไร้สาย ระบบการเฝ้าระวังสุขภาพ และ MAC โพรโทคอลที่ใช้ในระบบเครือข่ายเซนเซอร์ไร้สาย

#### 2.1 เครือข่ายเซนเซอร์ไร้สาย

เครือข่ายเซนเซอร์ไร้สาย[27] คือระบบเครือข่ายไร้สายที่ใช้อุปกรณ์ที่มีเซนเซอร์ในการตรวจจับสถานะแวดล้อมต่างๆ เช่น อุณหภูมิ เสียง การสั่นสะเทือน ความกดอากาศ มลพิษ เป็นต้น โดยอุปกรณ์แต่ละตัวสามารถติดต่อกับอุปกรณ์ที่อยู่ใกล้ๆ กันได้ในรัศมีที่ครอบคลุมถึงผ่านทางคลื่นวิทยุ ซึ่งรูปแบบเครือข่ายของเซนเซอร์ไร้สาย อาจจะเป็นเครือข่ายอย่างง่ายแบบ Star[28] หรือเครือข่ายขั้นสูงแบบ Mesh[29] ที่สามารถส่งข้อมูลแบบ multi-hop ได้ โดยการกระจายข้อมูลผ่านแต่ละโหนดในเครือข่ายนั้นมีสองวิธีการคือ ใช้การหาเส้นทาง (Routing) หรือแบบส่งแบบรอบทิศทาง (Flooding)

การพัฒนาเครือข่ายเซนเซอร์ไร้สายได้รับแรงบันดาลใจจากการใช้งานทางทหาร เช่น การเฝ้าระวังสนามรบ เป็นต้น แต่ในปัจจุบัน เครือข่ายดังกล่าว ได้ถูกนำไปใช้อย่างแพร่หลายทั้งในภาคอุตสาหกรรม และภาคครัวเรือน เช่น การนำมาใช้ตรวจสอบกระบวนการทำงานในอุตสาหกรรม และควบคุมเครื่องจักร เป็นต้น

เครือข่ายเซนเซอร์ไร้สายถูกสร้างขึ้นจาก “โหนด” ที่ทำงานร่วมกัน ตั้งแต่จำนวนไม่กี่ตัวจนถึงหลักร้อย หรือพันตัว ซึ่งแต่ละโหนดจะเชื่อมต่อกับเซนเซอร์จำนวนหนึ่งตัว หรืออาจจะมากกว่านั้นก็ได้ และเซนเซอร์โหนดแต่ละตัว จะมีส่วนประกอบหลักหลายส่วนที่เหมือนกัน โดยส่วนแรกคือ ส่วนของวิทยุรับ-ส่งข้อมูล พร้อมเสาอากาศที่อาจจะอยู่ในตัว หรืออาจจะต่อแยกออกมาก็ได้ ส่วนที่สองคือ ไมโครคอนโทรลเลอร์ที่ใช้ในการประมวลผลข้อมูล ส่วนที่สามคือ วงจรอิเล็กทรอนิกส์ที่ใช้สำหรับการเชื่อมต่อกับเซนเซอร์ และแหล่งพลังงาน ซึ่งปกติจะทำงานอยู่บนแบตเตอรี่ หรืออาจจะมีอุปกรณ์ที่ใช้ในการเก็บเกี่ยวพลังงานจากสภาพแวดล้อมก็ได้

เซนเซอร์โหนดอาจจะมีขนาดที่แตกต่างกันออกไปตั้งแต่ขนาดใหญ่กว่ากล่องรองเท้า จนกระทั่งเล็กกว่าเมล็ดข้าว และราคาของเซนเซอร์โหนดก็เช่นเดียวกันคือ มีตั้งแต่ราคาไม่กี่บาท จนถึงหลายพันบาท ขึ้นอยู่กับความซับซ้อนของแต่ละเซนเซอร์โหนด นอกจากนี้ข้อจำกัดในด้านขนาด และราคาแล้ว เซนเซอร์โหนดยังมีข้อจำกัดในด้าน ทรัพยากรอื่นๆอีก เช่น พลังงาน หน่วยความจำ ความเร็วในการประมวลผล และ อัตราการส่งข้อมูล เป็นต้น [30][31]

## 2.2 เครื่องข่ายเซนเซอร์ไร้สายกับงานด้านการเฝ้าระวังสุขภาพ

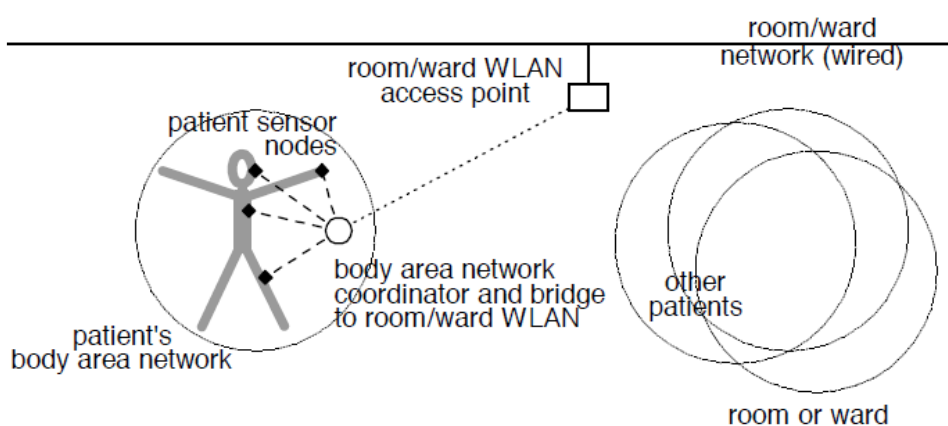
การดูแลสุขภาพในบางรายจำเป็นที่จะต้องมีการเฝ้าระวังอาการอยู่ตลอดเวลา ยกตัวอย่างเช่น ผู้ป่วยโรคหัวใจ ซึ่งโรคนี้เป็นสาเหตุหลักอย่างหนึ่งของการเสียชีวิต จากการบันทึกพบว่า ในประเทศอังกฤษมีคนตายด้วยโรคนี้ถึง 39 % [32] ของการตายทั้งหมดในแต่ละปี แม้ว่าโรคหัวใจจะเกิดขึ้นโดยทันทีไม่มีข้อบ่งชี้ชัดเจน แต่ก็สามารถสังเกตอาการจากการเต้นของหัวใจที่ผิดปกติได้ ซึ่งการที่จะตรวจสอบจังหวะการเต้นของหัวใจตลอดเวลาได้นั้น จำเป็นจะต้องมีอุปกรณ์ที่สามารถส่งข้อมูลการเต้นของหัวใจไปยังแพทย์ และเพื่อความสะดวกในการใช้งานอุปกรณ์ จึงจำเป็นต้องส่งข้อมูลแบบไร้สายได้ด้วย

เซนเซอร์ทางการแพทย์ที่ใช้ในการเฝ้าระวังผู้ป่วยนั้นมีหลายชนิด เช่น Electroencephalography (EEG) [33], Electrocardiography (EKG) [34], Blood Pressure (BP) [35], Pulse Oximeter [36], Cardiac Output [37] และ Body Temperature[38] เป็นต้น ซึ่งอุปกรณ์แต่ละชนิด ใช้อัตราการส่งข้อมูลที่แตกต่างกัน ดังตารางที่ 2.1 [39]

Parameter	Sampling Frequency	Bits per Sample	Number of Channels	Data Rate
EEG	256 Hz	16	24	98,304 bps
EKG	200 Hz	16	1-3	3,200-9,600 bps
Blood Pressure	120 Hz	16	1	1,920 bps
Pulse Oximeter	60 Hz	16	2	1,920 bps
Cardiac Output	40 Hz	16	1	640 bps
Body Temperature	0.2 Hz	12	1	2.4 bps

ตารางที่ 2.1 อัตราการส่งข้อมูลของเซนเซอร์ชนิดต่างๆในทางการแพทย์

จากตารางจะเห็นได้ว่า หากต้องการเฝ้าระวังผู้ป่วยด้วยเซนเซอร์ทั้งหมดนี้จะต้องใช้อัตราการส่งข้อมูลอย่างน้อย 112386.4 บิตต่อวินาที หรือ 14048.3 ไบต์ต่อวินาที ซึ่งในปัจจุบันการเก็บข้อมูลเหล่านี้ของโรงพยาบาลทั่วไป จะทำผ่านเครื่องที่ใช้สาย ซึ่งจะทำให้เกิดความยากลำบากในการเคลื่อนที่ของผู้ป่วย และการปฏิบัติหน้าที่ของพยาบาล นอกจากนี้ ในบางครั้งการเฝ้าระวังผู้ป่วยที่เป็นโรคติดต่อ อาจจะทำให้ผู้ดูแลได้รับเชื้อโรคได้โดยไม่รู้ตัว ดังนั้นจึงเป็นที่มาของการนำระบบการเฝ้าระวังแบบไร้สายมาประยุกต์ใช้งาน



รูปที่ 2.1 โครงสร้างการส่งข้อมูลของ Body Area Network [32]

เครือข่ายเซนเซอร์ไร้สายสำหรับงานด้านสุขภาพ ต้องการอุปกรณ์ที่มีขนาดเล็ก มีน้ำหนักเบา โดยมีเซนเซอร์ หน่วยประมวลผล และ โมดูลในการสื่อสารในตัว ซึ่งอุปกรณ์จะถูกติดตั้งไปกับตัวผู้ป่วย ซึ่งต้องการส่งข้อมูลที่ได้จากเซนเซอร์ในระยะใกล้ๆ ไปยังอุปกรณ์หลักที่อาจจะติดอยู่กับผู้ป่วยหรือเตียงของผู้ป่วย ซึ่งเทคโนโลยีนี้เรียกว่า Body Area Network (BAN) [32] โดยลักษณะของเครือข่ายแบบ BAN นั้นได้แสดงดังรูปที่ 2.1 ซึ่งจะเห็นได้ว่า ในส่วนของ BAN นั้นประกอบไปด้วยเซนเซอร์โหนด ที่ติดตามจุดต่างๆของร่างกาย เพื่อเก็บข้อมูล และส่งข้อมูลเหล่านั้นไปยัง BAN Coordinator ที่ทำหน้าที่ รวบรวมข้อมูลจากเซนเซอร์โหนดในร่างกาย ก่อนที่จะส่งต่อออกจากเครือข่าย BAN ไปยังเครือข่าย WLAN ที่อยู่ในห้องนั้นๆ เพื่อให้แพทย์ได้นำข้อมูลไปใช้ต่อไป

ก่อนหน้านี้ได้มีการสร้างระบบส่งข้อมูลทางการแพทย์แบบไร้สาย โดยเริ่มต้นจากการทำต้นแบบและผลิตภัณฑ์ เช่น อุปกรณ์ EKG และระบบในการเฝ้าระวัง [40] และ [41] โดยส่งข้อมูลผ่านเทคโนโลยีของ Bluetooth และ เครือข่ายไร้สายมาตรฐาน IEEE 802.11b แต่อย่างไรก็ตาม IEEE 802.15.4 ก็ดูเหมือนจะเหมาะสมสำหรับงานทางการแพทย์มากที่สุด เนื่องจากต้องการใช้พลังงานต่ำ และต้องการอัตราการส่งข้อมูลที่ต่ำด้วย แต่ก็ยังต้องการความน่าเชื่อถือสูง อัตราหน่วงต่ำ และสามารถส่งข้อมูลแบบทันทีได้

### 2.3 MAC โพรโทคอลบนเครือข่ายเซนเซอร์ไร้สาย

การสื่อสารของอุปกรณ์ในเครือข่ายเซนเซอร์ไร้สายนั้น จะอยู่บนพื้นฐานของ OSI Model[42] โดยใน OSI Model นี้จะแบ่งการสื่อสารออกเป็น 7 ระดับชั้น แต่ในมาตรฐาน IEEE 802.15.4 ได้กำหนดไว้เพียง 2 ระดับชั้นล่างสุดคือ Physical Layer และ Medium Access Control

Physical Layer (PHY) เป็นลำดับชั้นการสื่อสารลำดับแรกใน OIS Model ซึ่งทำหน้าที่จัดการ การรับ-ส่งคลื่นวิทยุทางกายภาพ ดำเนินการในการเลือกช่องสัญญาณ และจัดการเกี่ยวกับระดับพลังงานของสัญญาณ โดยสามารถเลือกทำงานได้บน 3 ช่วงคลื่นความถี่ที่เปิดให้ใช้งานได้โดยไม่ต้องขออนุญาต คือ 1.ช่วง 868.0-868.6MHz โดยแบ่งออกเป็น 1 ช่องสัญญาณและอนุญาตให้ใช้งานได้ในเขตยุโรป 2.ช่วง 902-928 MHz โดยแบ่งออกเป็น 10 ช่องสัญญาณและอนุญาตให้ใช้งานได้ในเขตอเมริกาเหนือ และ 3.ช่วง 2400-2483.5 MHz โดยแบ่งออกเป็น 16 ช่องสัญญาณและอนุญาตให้ใช้งานได้ทั่วโลก

Medium Access Control (MAC) เป็นลำดับชั้นที่เชื่อม PHY Layer เข้ากับลำดับชั้นที่สูงขึ้นใน OSI Model โดย MAC Layer จะทำหน้าที่จัดการการเข้าใช้ "สื่อ" ที่ใช้ในการสื่อสาร ซึ่งในที่นี้ก็คือช่องสัญญาณในอากาศ เนื่องจากช่องสัญญาณเป็นทรัพยากรที่จะต้องใช้ร่วมกันโดยโหนดที่อยู่ในบริเวณเดียวกัน ดังนั้นจึงจำเป็นต้องมีการ การเข้าใช้ช่องสัญญาณไม่ให้เกิดการชนกันของข้อมูล ซึ่ง MAC โพรโทคอลบนเครือข่ายเซนเซอร์ไร้สาย นอกจากจะทำหน้าที่จัดการไม่ให้เกิดการชนกันของข้อมูลแล้ว ยังต้องจัดการ การใช้พลังงานของภาครับ-ส่งสัญญาณวิทยุอีกด้วย โดยการเปิดใช้เมื่อจำเป็นเท่านั้น และปิดภาครับส่งสัญญาณวิทยุให้ได้มากที่สุดเพื่อประหยัดพลังงาน

เนื่องจากภาครับ-ส่งสัญญาณวิทยุ ของโหนดบนเครือข่ายเซนเซอร์ไร้สาย เป็นส่วนที่ใช้พลังงานมากที่สุด[6] ดังนั้นการออกแบบ MAC โพรโทคอลที่ดี จะช่วยเพิ่มประสิทธิภาพในการใช้พลังงานของโหนดได้ และจะช่วยยืดอายุการใช้งานของเครือข่ายออกไปได้อีกด้วย

### 2.3.1 ลักษณะ MAC โพรโทคอลที่ดี

การออกแบบ MAC โพรโทคอลที่ดี ควรจะต้องมีคุณลักษณะดังนี้ [43]

- การใช้พลังงานให้คุ้มค่า: เนื่องจากเซนเซอร์โหนดใช้พลังงานจากแบตเตอรี่ และบ่อยครั้งที่ไม่สามารถเปลี่ยนหรือชาร์ตแบตเตอรี่ได้ ซึ่งในบางครั้ง อาจจะเป็นการดีกว่าที่จะเปลี่ยนโหนด แทนที่จะชาร์ตแบตเตอรี่ใหม่ ดังนั้น การใช้พลังงานให้คุ้มค่า จึงเป็นสิ่งที่สำคัญอย่างหนึ่ง
- ความหน่วง: เซนเซอร์โหนด เมื่ออ่านค่าจากเซนเซอร์ได้แล้ว ก็ควรจะ รายงานผลได้ในระยะเวลาที่เหมาะสม เนื่องจากข้อมูลจากเซนเซอร์บางชนิด จำเป็นต้องรายงานได้ในทันที
- อัตราการส่งข้อมูล: MAC โพรโทคอลที่ดี ต้องรองรับอัตราการส่งข้อมูลที่แตกต่างกันออกไปได้ ซึ่งความต้องการนั้น จะปรับเปลี่ยนตามชนิดของ เซนเซอร์ที่ใช้งาน
- ความเท่าเทียมกัน: เมื่ออัตราการส่งข้อมูลถูกจำกัด เราต้องมั่นใจได้ว่า จะสามารถรับข้อมูลจากเซนเซอร์โหนดทุกตัวได้อย่างเท่าเทียมกัน

อย่างไรก็ตาม จากคุณลักษณะทั้งหมดนี้ การใช้พลังงานให้คุ้มค่า และอัตราการส่งข้อมูลที่เหมาะสม เป็นหัวข้อหลัก ที่สำคัญในการออกแบบ MAC โพรโทคอล ซึ่งการใช้พลังงานให้คุ้มค่า นั้น สามารถทำได้โดยการลดการใช้พลังงานที่ไม่ก่อให้เกิดประโยชน์

### 2.3.2 ที่มาของการใช้พลังงานที่ไม่ก่อให้เกิดประโยชน์

พื้นฐานของการใช้พลังงานที่ไม่ก่อประโยชน์ในเครือข่ายแบบไร้สายมี 4 อย่าง คือ [43] และ [20]

- การชนกันของข้อมูล: เมื่อข้อมูลถูกส่งออกไป และมีการชนกันของข้อมูลเกิดขึ้น ข้อมูลนั้นจะเสียหายและเซนเซอร์โหนดจะต้องส่งข้อมูลนั้นใหม่อีกครั้ง ทำให้ต้องสิ้นเปลืองพลังงานในการส่งโดยไม่เกิดประโยชน์ นอกจากนี้การชนยังทำให้เกิดการหน่วงเพิ่มขึ้นอีกด้วย
- Overhearing: หมายถึงการรับฟังข้อมูลที่เป้าหมายการส่งคือโหนดอื่น ซึ่งทำให้เสียพลังงานโดยไม่เกิดประโยชน์
- Packet Overhead: ในการรับและส่งข้อมูล จำเป็นต้องมีข้อมูลส่วนหัว ซึ่งข้อมูลในส่วนนี้ ไม่ได้เป็นข้อมูลของเซนเซอร์ แต่จำเป็นต้องมีเพื่อที่จะส่งข้อมูลให้ถึงเป้าหมายได้อย่างถูกต้อง ดังนั้นหากข้อมูลส่วนนี้มีขนาดใหญ่เกินไป ก็ส่งผลให้เกิดการใช้พลังงานเพิ่มขึ้นโดยไม่เกิดประโยชน์
- การรอรับข้อมูลโดยไม่มีข้อมูลส่งออกมา: หากโหนดต้องเปิดวงจรรับวิทยุไว้ตลอด เพื่อรอข้อมูลที่จะส่งมา ก็จะทำให้สิ้นเปลืองพลังงานโดยไม่เกิดประโยชน์ และในหัวข้อนี้ ก็เป็นส่วนที่ทำให้โหนดสิ้นเปลืองพลังงานมากที่สุด

เป้าหมายหลักของการใช้งาน MAC โพรโทคอลบนเครือข่ายเซนเซอร์ไร้สายคือ ลดการใช้พลังงานที่ไม่ก่อให้เกิดประโยชน์ ที่เกิดจากการรอรับข้อมูลโดยไม่มีข้อมูลส่งออกมา, Overhearing และ การชนกันของข้อมูล โดยโพรโทคอลจะต้องไม่มีความซับซ้อนมากเกินไป เพื่อลด Packet Overhead



### 2.3.3 การวัดประสิทธิภาพของ MAC โพรโทคอล

ตัวชี้วัดสำหรับการเปรียบเทียบประสิทธิภาพการใช้พลังงานของ MAC โพรโทคอล นักวิจัยส่วนใหญ่จะเปรียบเทียบตามหัวข้อต่อไปนี้ [6]

- การบริโภคพลังงานต่อบิต: คือพลังงานที่ใช้ไป ต่อจำนวนบิตที่สามารถส่งได้ หน่วยเป็น “จูลต่อบิต” ซึ่งค่ายิ่งน้อยยิ่งดี โดยในส่วนนี้จะได้รับผลกระทบมาจาก สิ่งหลักที่เป็นต้นกำเนิดของการใช้พลังงานที่ไม่ก่อประโยชน์ทั้งหมด เช่น การชนกันของข้อมูล, Packet Overhead, Overhearing และการรอรับข้อมูลโดยไม่มีข้อมูลส่งออกมา
- สัดส่วนค่าเฉลี่ยการส่ง: คือจำนวนข้อมูลที่ได้รับหารด้วยจำนวนข้อมูลที่ส่งเฉลี่ยกันทุกเซนเซอร์โหนด
- ค่าเฉลี่ยการหน่วงของข้อมูล: คือเวลาเฉลี่ยของการส่งข้อมูลตั้งแต่เริ่มส่งจนถึงปลายทาง
- อัตราการส่งข้อมูลของเครือข่าย: คือจำนวนรวมของข้อมูลที่ถูกลงส่งไป และได้รับต่อหน่วยเวลา

โดยในงานวิจัยชิ้นนี้ จะวัดประสิทธิภาพของ MAC โพรโทคอลในหัวข้อ การบริโภคพลังงานต่อบิต และค่าเฉลี่ยการหน่วงของข้อมูล

### 2.3.4 MAC โพรโทคอลสำหรับเครือข่ายเซนเซอร์ไร้สายในปัจจุบัน

MAC โพรโทคอลสำหรับเครือข่ายเซนเซอร์ไร้สายส่วนใหญ่ที่ถูกเผยแพร่ออกมาจะไปเน้นการออกแบบให้ใช้พลังงานอย่างคุ้มค่า อย่างไรก็ตาม ก็มีงานอีกไม่น้อย ที่ทำได้สำเร็จในการออกแบบ MAC โพรโทคอลด้านอื่น[6] เช่น

- ความปลอดภัยของเครือข่าย: เช่น TinySec [44] ที่สร้างระบบรักษาความปลอดภัยของข้อมูลในระดับ Link Layer
- ความสามารถในการเคลื่อนย้ายโหนด: โดยปกติของเครือข่ายเซนเซอร์ไร้สายจะสมมุติให้เซนเซอร์โหนดอยู่กับที่ แต่ก็มียานบางชนิด ที่ต้องการการเคลื่อนที่ของเซนเซอร์โหนด เช่น งานด้านการดูแลสุขภาพ เป็นต้น ซึ่ง MAC โพรโทคอลที่เน้นในเรื่องของการเคลื่อนที่ของโหนด ก็มีออกมาในชื่อ Mobility-adaptive, collision-free MAC protocol for wireless sensor networks (MMAC) [45] ซึ่งในหัวข้อนี้ยังมีหัวข้องานอีกมากให้วิจัยในอนาคต
- ความสามารถในการใช้งานกับโหนดจริง: MAC โพรโทคอลสำหรับเครือข่ายเซนเซอร์ไร้สายหลายชนิด ถูกพัฒนาขึ้นมาและทดลองโดยการจำลอง อย่างไรก็ตาม MAC โพรโทคอลที่ถูกพัฒนาขึ้นมา ควรจะทดลองและสามารถนำไปใช้งานได้กับโหนดจริง เช่น MAC โพรโทคอลบนมาตรฐาน IEEE 802.15.4 [46, p. 802]
- ความสามารถในการทำงานบนระบบเวลาจริง: การใช้พลังงานให้คุ้มค่าเป็นการออกแบบหลักสำหรับ MAC โพรโทคอลบนเครือข่ายเซนเซอร์ไร้สาย แต่ความเสถียรในการส่งข้อมูลและการส่งข้อมูลตามเวลาจริง ก็เป็นสิ่งที่สำคัญสำหรับงานที่เคร่งครัดเรื่องเวลา เช่น A real-time and energy-efficient MAC protocol for wireless sensor networks (AREA-MAC) [47]

### 2.3.5 MAC โพรโทคอลแบบต่างๆกับการนำมาใช้งานบนเครือข่ายไร้สาย

MAC โพรโทคอลบนการสื่อสารแบบไร้สายในปัจจุบันนี้มีอยู่ 4 รูปแบบด้วยกัน ซึ่งแต่ละแบบก็มีข้อดีและข้อเสียที่จะนำมาใช้ในเครือข่ายเซนเซอร์ไร้สายดังนี้

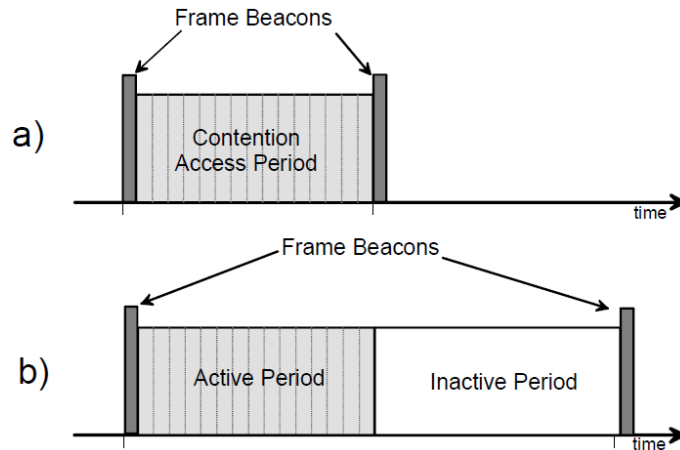
- Schedule Based (TDMA)[9]: สามารถจัดการกับการชนได้ แต่ต้องการเวลาของโหนดที่แม่นยำ นอกจากนั้น ยังยากที่จะปรับเปลี่ยนรูปแบบของเครือข่าย เช่น การเพิ่มหรือลดโหนด
- Contention Based (CSMA)[10]: มีการหน่วงต่ำ และอัตราการส่งข้อมูลสูง ไม่ต้องการเวลาของโหนดที่แม่นยำมากนัก แต่มีปัญหาเรื่องการชนกันของข้อมูล
- Frequency Division Multiple Access (FDMA)[48] ไม่มีการชนเกิดขึ้น แต่เซนเซอร์โหนดจะต้องมีวงจรที่ซับซ้อน เพื่อรองรับการทำงานหลายช่องสัญญาณ ทำให้ราคาของเซนเซอร์โหนดเพิ่มขึ้น ซึ่งเป็นปัจจัยหลักของระบบเครือข่ายเซนเซอร์ไร้สาย
- sCode Division Multiple Access (CDMA)[49]: สามารถจัดการการชนได้ แต่มีความซับซ้อนสูง ซึ่งเป็นข้อจำกัดของการประมวลผลบนอุปกรณ์เครือข่ายเซนเซอร์ไร้สาย ที่ถูกจำกัดด้วยพลังงาน

## 2.4 การทำงานของ MAC โพรโทคอลบนมาตรฐาน IEEE 802.15.4

ในเครือข่ายแบบ IEEE 802.15.4 นั้นได้แบ่งอุปกรณ์ออกเป็น 2 ชนิด คือ Full Function Device (FFD) และ Reduce Function Device (RDF) โดย FFD สามารถที่จะทำงานได้สามรูปแบบ คือ (PAN) Coordinator, Coordinator และ Device ซึ่ง FFD สามารถติดต่อกับ RDF หรือ FFD ตัวอื่นๆ ได้ แต่ RDF จะสามารถติดต่อกับ FFD ได้เท่านั้น ซึ่ง RDF นี้มีไว้สำหรับการทำงานที่ง่ายไม่ซับซ้อนมากนัก เช่น สวิตช์ไฟ หรือ อ่านข้อมูลจากเซนเซอร์ เป็นต้น และ RDF เหล่านี้ จะไม่ต้องการส่งข้อมูลที่มีจำนวนมากนัก ซึ่งอาจจะทำเพียงแค่เชื่อมต่อกับ FFD เท่านั้น ดังนั้น RDF จึงสามารถทำงานได้โดยใช้ทรัพยากรน้อยที่สุด

MAC โพรโทคอลบน IEEE 802.15.4 นั้นได้กำหนดรูปแบบการทำงานของเครือข่ายไว้เป็นสองรูปแบบคือ แบบ Beacon-enable และแบบ Non-Beacon-enable โดยแบบ Beacon-enable จะมีการ Synchronize เวลาระหว่างอุปกรณ์ ด้วยการส่ง Beacon ออกมาตามคาบเวลาที่กำหนด ซึ่งอุปกรณ์ทุกตัวที่เป็น RDF จะต้องตื่นขึ้นมารับ Beacon จาก FFD เพื่อรับ-ส่งข้อมูล และหลังจากนั้นอุปกรณ์ทุกตัวก็จะหลับไปอีกครึ่ง รอการส่ง Beacon ครั้งถัดไป แต่ในแบบ Non-Beacon-enable นั้นจะไม่มี การ Synchronize เวลาเพื่อหลับและตื่น โดยอุปกรณ์ที่เป็น FFD จะต้องตื่นตลอดเวลาเพื่อรอฟังข้อมูลที่อาจถูกส่งมาจาก RDF หรือ FFD ทำให้อุปกรณ์ที่เป็น FFD นั้นมีการใช้พลังงานมากอยู่ตลอดเวลา ทั้งช่วงที่มีการส่งข้อมูลและไม่มีการส่งข้อมูล ดังนั้นรูปแบบการสื่อสารแบบ Beacon-enable จึงเหมาะกับลักษณะงานที่อุปกรณ์ทุกตัวทำงานโดยใช้แหล่งพลังงานจากแบตเตอรี่ มากกว่าแบบ Non-Beacon-enable ที่ FFD ต้องเปิดวงจรวิทยุไว้ตลอดเวลา

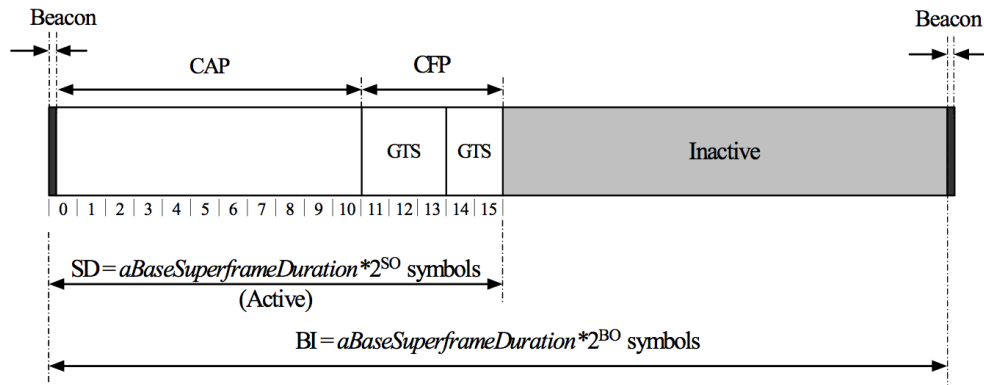
### 2.4.1 Superframe Structure



รูปที่ 2.2 โครงสร้างหลักของ Superframe

ในรูปแบบการส่งข้อมูลแบบ Beacon-enable บนมาตรฐาน IEEE 802.15.4 นั้นจำเป็นต้องมีการใช้โครงสร้างข้อมูลแบบ Superframe structure โดยรูปแบบของ Superframe structure จะถูกกำหนดโดย Coordinator ซึ่งขอบของตัว Superframe จะมี Beacons ที่ถูกส่งออกมาโดย Coordinator คั่นอยู่ ดัง

รูปที่ 2.2 ซึ่งตัว Superframe ถูกแบ่งออกเป็น 16 ช่องเท่าๆกันและ Superframe สามารถมีส่วนที่เป็น Active และ Inactive โดยในช่วง Inactive นั้น ตัว Coordinator จะเข้าสู่รูปแบบประหยัดพลังงาน และ Beacon frame จะถูกส่งในช่องแรกในแต่ละ Superframe ซึ่งถ้า Coordinator ไม่ต้องการที่จะใช้ Superframe structure จะปิดการส่ง Beacon โดย Beacon นี้จะถูกใช้ในการ synchronize กับอุปกรณ์ Device ใช้ในการระบุ PAN ของเครือข่าย และ ใช้ในการอธิบายโครงสร้างของ Superframes และหากอุปกรณ์ใดๆต้องการที่จะสื่อสารกับอุปกรณ์ตัวอื่นจะต้องสื่อสารกันในช่วง Contention Access Period (CAP) ซึ่งเป็นช่วงระหว่างการส่ง Beacon สอง Beacon โดยการส่งข้อมูล จะใช้วิธีการแบบ Slotted CSMA-CA ซึ่งทุกการสื่อสาร จะต้องเสร็จก่อนที่จะมี Network Beacon ตัวถัดไปถูกส่งมา



รูปที่ 2.3 การทำงานของ Superframe

เวลาที่แตกต่างกันระหว่างช่วงของ Superframe สามารถอธิบายได้ด้วยตัวแปร  $macBeaconOrder$  และ  $macSuperFrameOrder$  โดย  $macBeaconOrder$  จะอธิบายถึงช่วงเวลาที่ Coordinator ต้องส่ง Beacon Frame ซึ่ง Beacon Interval, BI นี้จะสัมพันธ์กับ  $macBeaconOrder$ , BO ตามสมการ

$$BI = aBaseSuperFrameDuration \times 2^{BO} \text{ เมื่อ } 0 \leq BO \leq 14 \quad [1]$$

และ Superframe จะถูกเพิกเฉยเมื่อ  $BO = 15$

ค่าของ  $macSuperFrameOrder$  จะอธิบายถึงขนาดของช่วง Active ของ Superframe โดย Superframe Duration, SD นี้จะสัมพันธ์กับ  $macSuperFrameOrder$ , SO ตามสมการ

ss

$$SD = aBaseSuperFrameDuration \times 2^{SO} \text{ เมื่อ } 0 \leq SO \leq 14 \quad [1]$$

และถ้า  $SO = 15$  แล้ว Superframe จะไม่มีส่วน Active หลังจาก Beacon โดยภาพแสดงการทำงานของ Superframe ได้แสดงดัง

รูปที่ 2.3

ในช่วง SD ของ Superframe นั้นได้แบ่งออกเป็น 2 ส่วนคือ Contention Access Period (CAP) และ Contention-Free Period (CFP) โดยในส่วน CAP เช่นเซอร์โหนดจะต้องทำกระบวนการ CSMA-CA เพื่อเข้ามาส่งข้อมูล แต่ส่วน CFP นั้นจะแบ่งเป็นช่องของ Guaranteed Time Slots (GTSs) ซึ่งในแต่ละช่องจะไม่มีการใช้ CSMA-CA เพื่อเข้ามาส่งข้อมูล แต่การส่งข้อมูลในแต่ละช่องของ GTS นั้นจะถูกกำหนดไว้ล่วงหน้าแล้วใน Beacon

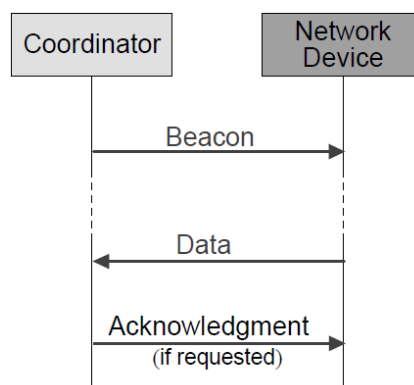
#### 2.4.2 รูปแบบการรับ-ส่งข้อมูล

ชนิดของการส่งข้อมูลบนมาตรฐาน IEEE 802.15.4 นั้นแบ่งออกได้เป็นสองชนิดคือ ส่งจาก Coordinator ถึง Device และ ส่งจาก Device ถึง Coordinator โดยกระบวนการของการส่งแต่ละรูปแบบนั้น จะขึ้นอยู่กับเครือข่ายที่สร้างขึ้นมานั้นเป็นแบบ Beacon Enable หรือ Non-Beacon-enable ซึ่งในที่นี้จะเน้นเฉพาะเครือข่ายแบบ Beacon-enable เท่านั้น

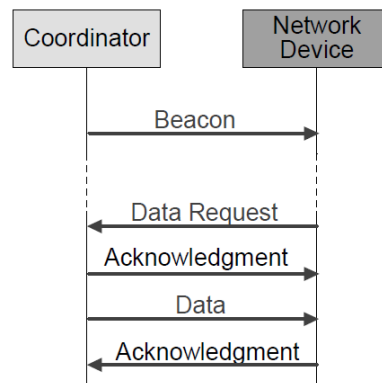
เมื่อ Device ต้องการที่จะส่งข้อมูลในเครือข่ายแบบ Beacon-enable เริ่มต้นจะรอฟัง Beacon เมื่อพบ Beacon แล้วก็จะทำ.

รูปแบบของ Superframe ให้ตรงกัน ซึ่งข้อมูลเหล่านี้จะถูกส่งมากับ Beacon หลังจากทำรูปแบบของ Superframe ให้ตรงกันเสร็จแล้ว ก็จะเริ่มส่งข้อมูลถึง Coordinator โดยการใช้ Slotted CSMA-CA ซึ่งในที่นี้เราอาจจะให้ Coordinator ตอบ Acknowledgment กลับมา เมื่อได้รับข้อมูลเรียบร้อยแล้วก็ได้ โดยกระบวนการที่กล่าวมานั้นได้แสดงไว้ใน

รูปที่ 2.4



รูปที่ 2.4 ขั้นตอนการส่งข้อมูลจาก Device สู่ Coordinator ในเครือข่ายแบบ Beacon Enable



รูปที่ 2.5 ขั้นตอนการส่งข้อมูลจาก Coordinator สู่ Device ในเครือข่ายแบบ Beacon Enable

ในงานที่ต้องการส่งข้อมูลเพื่อไปควบคุมตัว Device เมื่อ Coordinator ต้องการที่จะส่งข้อมูลไปยัง Device ในเครือข่ายแบบ Beacon-enable ตัว Coordinator จะบอกให้รู้โดยส่ง Data Message มาใน Beacon และตัว Device เองก็จะรอฟัง Beacon เป็นระยะๆ เมื่อ Device ได้รับ Beacon ที่มี Data Message ตัว Device จะส่ง MAC Command Requesting เพื่อไปขอ Data จาก Coordinator โดยใช้ Slotted CSMA-CA ซึ่งในที่นี้อาจจะมีการ Acknowledges กลับมาจาก Coordinator เมื่อมีการส่งข้อมูล MAC Command Requesting สำเร็จก็ได้หากต้องการ ซึ่งหลังจาก Coordinator ได้รับ Message ในการร้องขอข้อมูลจาก Device แล้ว ข้อมูลที่รอการส่งอยู่ก็จะถูกส่งจาก Coordinator มายัง Device โดยใช้ Slotted CSMA-CA ซึ่งตัว Device อาจจะมีการตอบ Acknowledged กลับมา เมื่อสามารถรับข้อมูลเรียบร้อยแล้วก็ได้ หากต้องการ หลังจากที่ Device ได้รับข้อมูลเรียบร้อยแล้ว Message ก็จะถูกลบออกจากรายการข้อมูลที่รอส่งใน Beacon ซึ่งกระบวนการที่กล่าวมานั้นได้แสดงไว้ใน

รูปที่ 2.5

## 2.5 TinyOS

TinyOS เป็นระบบปฏิบัติการแบบ Open source ที่ออกแบบมาสำหรับเครือข่ายเซนเซอร์ไร้สายโดยเฉพาะ พัฒนาโดยภาษา NesC ซึ่งเป็นภาษาที่มีพื้นฐานมาจาก C และถูกปรับปรุงให้เหมาะกับอุปกรณ์ที่มีหน่วยความจำจำกัด เช่น อุปกรณ์ในเครือข่ายเซนเซอร์ไร้สาย โดย TinyOS จะมีเครื่องมือที่ใช้ในการติดต่อกับอุปกรณ์โหนด ที่ส่วนใหญ่จะพัฒนาโดยภาษา Java[50] และ Shell script[51] และมี Libraries พร้อมเครื่องมือในการ Compile ภาษา NesC ที่



พัฒนามาจากภาษา C โดยการทำงานของระบบปฏิบัติการ TinyOS นั้นจะเป็นแบบ Non-blocking[52] และมี Stack[53] เพียง 1 Stack ดังนั้นการเรียกใช้งาน I/O[54] ที่ต้องรอนานกว่า 100 ไมโครวินาที จะถูกส่งไปทำงานแบบ Asynchronous และเมื่อ I/O ทำงานเสร็จ ก็จะมี Callback[55] ส่งกลับมา และเพื่อให้ Compiler มีประสิทธิภาพในการเรียก Callback ระบบปฏิบัติการ TinyOS จึงใช้คุณสมบัติของภาษา NesC เพื่อเชื่อมเหล่า Callback ต่างๆเข้าด้วยกันแบบคงที่ และ TinyOS จะจัดการสิ่งต่างๆที่ต้องทำงานพร้อมๆกันด้วย Stack เพียง 1 Stack ซึ่งด้วยวิธีการนี้ จะทำให้ผู้เขียนโปรแกรมจะต้องเขียน Code ที่ซับซ้อนมาก เพื่อที่จะจัดการเหล่า Event ที่จะเกิดขึ้นจาก Callback ต่างๆ ดังนั้น TinyOS จึงได้มีการสร้าง Tasks ขึ้นมาเพื่อจัดการเหตุการณ์ที่อาจจะเกิดขึ้นพร้อมๆกัน โดยเมื่อมีการสร้าง Task ขึ้นมาหนึ่งตัว ตัวระบบปฏิบัติการจะสร้างตารางขึ้นมาเพื่อเก็บงานไว้ประมวลผลงานที่ค้าง และ Task ที่สร้างขึ้นมาจะเป็นแบบ Non-preemptive[56] และเข้าลำดับการประมวลผลแบบ First In, First Out (FIFO)[57] ด้วยกระบวนการจัดการงานแบบพร้อมกันอย่างง่ายนี้ ก็เพียงพอที่จะนำไปใช้ในงานที่ต้องใช้ I/O ทั่วไปได้โดยไม่ต้องหยุดรอ I/O

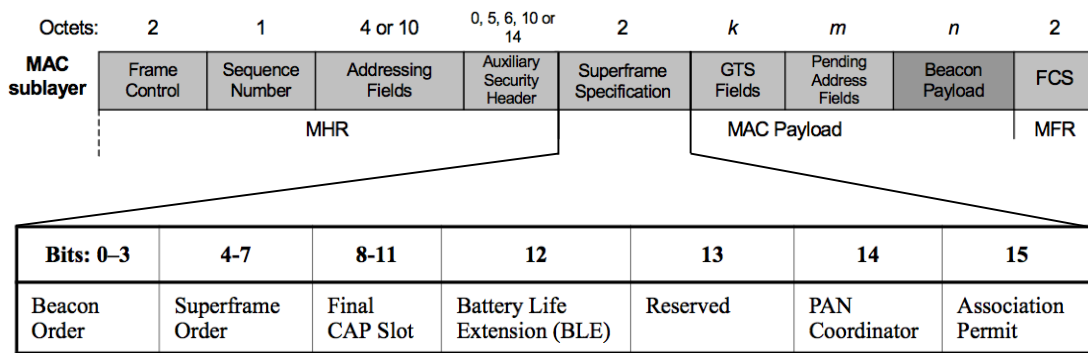
TinyOS ถูกพัฒนาขึ้นจากมหาวิทยาลัย Berkeley ซึ่งเป็นส่วนหนึ่งของโครงการ Defense Advanced Research Projects Agency (DARPA)[58] ที่มีนักวิชาการ นักพัฒนาในเชิงพาณิชย์ และผู้ใช้ทั่วโลกกว่าพันคน มาร่วมกันพัฒนา ทำให้ TinyOS ยังถูกใช้งานและมีการพัฒนาอย่างต่อเนื่อง

### บทที่ 3

#### การออกแบบและวิธีทดสอบระบบ

ในบทนี้จะกล่าวถึงการออกแบบ MAC โพรโทคอล โดยอ้างอิงจากมาตรฐาน IEEE 802.15.4 ให้สามารถปรับเปลี่ยน Duty-cycle ได้ตามความต้องการของเซนเซอร์แต่ละชนิด

#### 3.1 วิธีการปรับ Duty-cycle บนมาตรฐาน IEEE 802.15.4



รูปที่ 3.1 รายละเอียดของ Superframe [1]

Duty-cycle คืออัตราส่วนของเวลาที่แบ่งระหว่างช่วงที่ภาครับ-ส่งวิทยุเปิด ต่อคาบการของการส่งข้อมูล ตามสมการ  $Duty\ cycle = \frac{Active}{Active+Inactive}$

สำหรับ IEEE 802.15.4 แบบ Beacon-enable นั้นสามารถปรับเปลี่ยน Duty-cycle ได้จากการตัดสินใจของ Coordinator ผ่านพารามิเตอร์ 2 ตัวคือ BeaconOrder และ SuperframeOrder โดยค่าพารามิเตอร์นี้จะอยู่ส่วนของ Superframe Specification ที่ส่งมาพร้อมกับ Beacon Frame ดังรูปที่ 3.1 ซึ่งใน IEEE 802.15.4 แบบเดิมนั้น จะถูกกำหนดไว้ล่วงหน้า ไม่สามารถปรับเปลี่ยนได้ระหว่างที่เครือข่ายกำลังทำงานอยู่ และหากกำหนดค่าไม่เหมาะสมต่อลักษณะข้อมูลที่ต้องการส่งเช่น ปรับให้มีค่า Duty-cycle สูงเกินไป ก็จะทำให้ใช้พลังงานไปโดยไม่เกิดประโยชน์และหากปรับ Duty-cycle ให้ต่ำเกินไปก็จะทำให้อัตราในการส่งข้อมูลต่ำลงและระยะเวลาหน่วงสูงขึ้น ซึ่งส่งผลให้ประสิทธิภาพของเครือข่ายต่ำลงด้วย

ด้วย Duty-cycle ที่เท่ากัน ใน IEEE 802.15.4 นั้น สามารถเกิดได้จากการปรับเปลี่ยนค่าพารามิเตอร์ BeaconOrder และ SuperframeOrder หลากหลายรูปแบบมาผสมกัน ซึ่งแต่ละรูปแบบก็จะส่งผลกระทบต่อระยะเวลาหน่วงที่จะเกิดขึ้นแตกต่างกันออกไป ดังนั้นการเลือกค่าพารามิเตอร์ที่เหมาะสมจะช่วยให้ลดการใช้พลังงานที่ไม่เกิดประโยชน์ และยังช่วยเพิ่มประสิทธิภาพของระบบเครือข่ายอีกด้วย แต่การที่ Coordinator จะสามารถตัดสินใจเลือก BeaconOrder และ SuperframeOrder ที่เหมาะสมได้นั้น จะต้องทราบถึงลักษณะของเครือข่ายที่กำลังทำงานอยู่ ซึ่งข้อมูลนี้จะได้มาจาก Device แต่ละตัวที่เข้ามาอยู่ในเครือข่าย โดย Device ในเครือข่ายก็คืออุปกรณ์ที่จะส่งข้อมูลเซนเซอร์แต่ละชนิดนั่นเอง

เนื่องจากบุคคลแต่ละคนต้องการติดตามเซนเซอร์ในการเฝ้าระวังที่แตกต่างกันออกไปและเซนเซอร์แต่ละชนิดก็มีความต้องการในการส่งข้อมูลที่แตกต่างกันออกไป ดังนั้น Coordinator จะต้องรองรับความต้องการในการส่งข้อมูลของ Device เหล่านี้ให้ได้ และยังคงใช้พลังงานได้อย่างคุ้มค่าที่สุด โดยการปรับ Duty-cycle ให้เหมาะสมกับข้อมูลที่จะส่งมา และการที่จะปรับ Duty-cycle ให้เหมาะสมได้นั้นก็ต้องอาศัยข้อมูลจาก Device ดังนี้

#### 1. ระยะเวลาหน่วงสูงสุดที่ Device ขอมอบได้

เนื่องจากเซนเซอร์แต่ละชนิดมีความรีบเร่งในการส่งข้อมูลที่ไม่เท่ากัน บางชนิดเมื่ออ่านข้อมูลมาแล้วสามารถเก็บไว้แล้วค่อยส่งออกไปภายหลังได้ เช่น ข้อมูลอุณหภูมิร่างกาย แต่บางชนิดเมื่ออ่านข้อมูลออกมาแล้วต้องการส่งข้อมูลให้ได้ภายในเวลาที่กำหนด เช่น ข้อมูลจากเซนเซอร์ตรวจจับการหกล้ม ดังนั้น Device จึงต้องส่งข้อมูลระยะเวลาหน่วงสูงสุดที่ยอมรับได้ เพื่อให้ Coordinator กำหนดค่า Duty-cycle ที่เหมาะสมให้กับแต่ละ Device

#### 2. อัตราการส่งข้อมูลของ Device

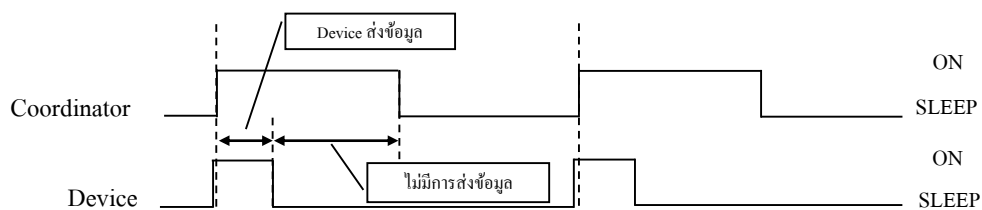
เซนเซอร์แต่ละชนิดมีจำนวนข้อมูลที่อ่านออกมาต่อหนึ่งหน่วยเวลาแตกต่างกัน เช่น Pulse oximeter มีข้อมูลที่อ่านออกมาได้ 1920 บิตต่อวินาที และ อุณหภูมิร่างกาย มีข้อมูลที่อ่านออกมาได้ 2.4 บิตต่อวินาที เป็นต้น จะเห็นได้ว่าเซนเซอร์ต่างชนิดกันมีความต้องการในการส่งข้อมูลแตกต่างกันออกไป ดังนั้น Coordinator จะต้องทราบค่าอัตราการส่งข้อมูลของแต่ละ Device ที่จะเข้ามาอยู่ในเครือข่าย

### 3. จำนวนข้อมูลต่อเฟรม

อุปกรณ์แต่ละชนิดมีลักษณะข้อมูลที่ต้องการส่งแตกต่างกันออกไป โดยข้อมูลแต่ละชนิดก็มีขนาดไม่เท่ากันดังนั้นจำนวนข้อมูลต่อเฟรมจึงแตกต่างกันออกไปด้วย ซึ่งส่งผลต่อระยะเวลาในการส่งข้อมูล ดังนั้น จำนวนข้อมูลต่อเฟรม จึงเป็นข้อมูลที่จำเป็นต่อการตัดสินใจของ Coordinator เพื่อเลือกค่า Duty-cycle ที่เหมาะสม

### 3.2 การออกแบบกระบวนการปรับ Duty-cycle อัตโนมัติ

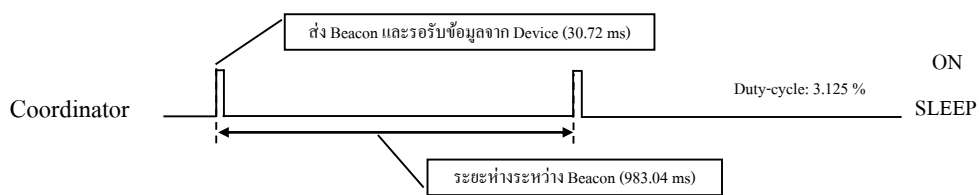
ในหัวข้อนี้จะกล่าวถึงวิธีการออกแบบและปรับปรุง MAC โพรโทคอลบนมาตรฐาน IEEE 802.15.4 ให้สามารถปรับ Duty-cycle ได้เองโดยอัตโนมัติ ตามการใช้งานของเซนเซอร์โหนดแต่ละชนิด ที่ผู้วิจัยได้ทดลองและพัฒนาขึ้นมา โดยเน้นการออกแบบไปในการใช้พลังงานอย่างมีประสิทธิภาพ



รูปที่ 3.2 ลักษณะ Duty-cycle แบบคงที่

MAC โพรโทคอลบนมาตรฐาน IEEE 802.15.4 นั้นออกแบบมาให้สามารถปรับ Duty-cycle ได้ตามความต้องการของแต่ละงาน แต่จะต้องกำหนดค่าที่แน่นอนลงไปก่อน ไม่สามารถเปลี่ยนแปลงขณะใช้งานได้ ดังนั้นหากเป็นงานที่เราไม่ทราบถึงลักษณะข้อมูลที่ต้องการส่ง เราต้องกำหนด Duty-cycle เพื่อไว้ก่อน เพื่อให้สามารถรองรับข้อมูลที่จะส่งได้ แต่การกำหนดค่า Duty-cycle ให้สูงเกินไป ก็จะทำให้เปลืองพลังงานโดยไม่เกิดประโยชน์ ดังรูปที่ 3.2 ได้แสดงถึง Coordinator ที่เปิดรับข้อมูลด้วย Duty-cycle คงที่ตลอดเวลาซึ่งไม่สัมพันธ์กับปริมาณข้อมูลที่ใช้งานจริง

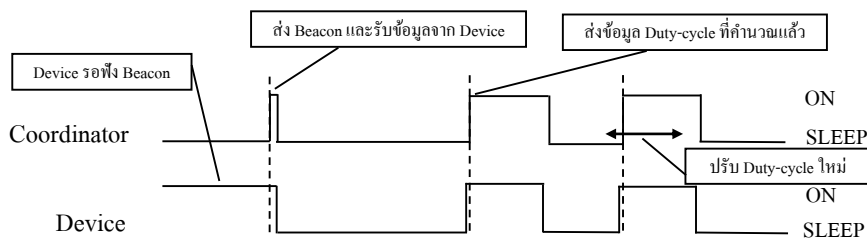
ในการใช้งานอุปกรณ์จริงนั้น บุคคลแต่ละคนจะติดเซนเซอร์แตกต่างกันออกไป ทั้งชนิดและจำนวน การตั้งค่าพารามิเตอร์ไว้ล่วงหน้าให้เหมาะสำหรับแต่ละบุคคลนั้น ย่อมเป็นเรื่องที่ทำได้ยาก และแต่ละบุคคลอาจจะมีการเปลี่ยนแปลงเซนเซอร์ที่ติดอยู่เรื่อยๆ ดังนั้น Coordinator จึงต้องสามารถรับข้อมูลจาก Device และปรับเปลี่ยน Duty-cycle ให้เหมาะสมสำหรับเซนเซอร์แต่ละชนิดได้ โดยจะเสนอวิธีการปรับเปลี่ยน Duty-cycle แบบอัตโนมัติตามขั้นตอนดังต่อไปนี้



รูปที่ 3.3 Coordinator ขณะที่ยังไม่มี Device เข้ามาในเครือข่าย

1. ปรับ Duty-cycle ให้ต่ำที่สุดขณะเริ่มทำงาน

เมื่อ Coordinator เริ่มทำงานและยังไม่มี Device เข้ามาในเครือข่าย ตัว Coordinator จะต้องใช้พลังงานให้น้อยที่สุดโดยการตื่นเพื่อส่ง Beacon และรับข้อมูลให้สั้นที่สุดเท่าที่จะเป็นไปได้ ซึ่งสามารถทำได้จากการตั้งค่าพารามิเตอร์ Superframe Order ให้เท่ากับ 1 และ Coordinator จะต้องส่ง Beacon เท่าที่จำเป็นเท่านั้น โดยกำหนดให้ระยะเวลาในการส่ง Beacon คือทุกๆ 1 วินาที ซึ่งสามารถทำได้จากการตั้งพารามิเตอร์ Beacon Order ให้เท่ากับ 6 ดังรูปที่ 3.3 ที่แสดงถึงการเปิด-ปิดวงจรวิทยุของ Coordinator ขณะที่ยังไม่มี Device เข้ามาในเครือข่าย โดยจะใช้เวลาในช่วง Active 30.72 ms และระยะห่างระหว่าง Beacon เป็น 983.04 ms ซึ่งวิธีการคำนวณเวลาจะกล่าวถึงในหัวข้อถัดไป



รูปที่ 3.4 เมื่อมี Device Associate เข้ามาในเครือข่าย

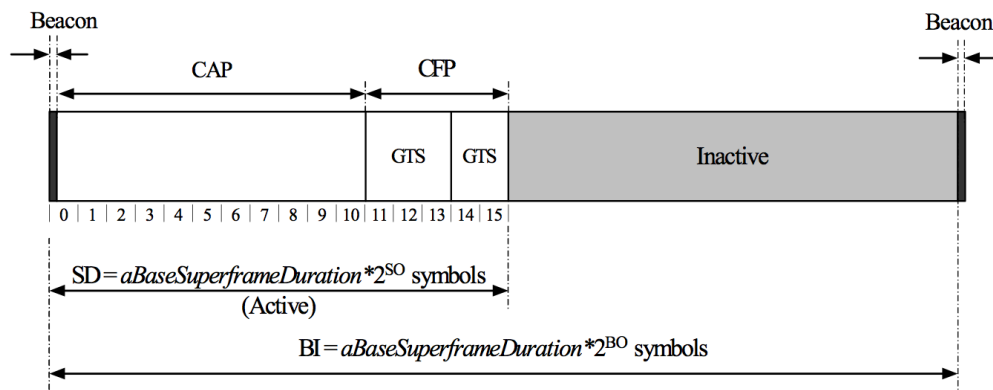
## 2. ปรับ Duty-cycle ตามความต้องการของ Device

เมื่อ Device ต้องการเข้ามาในเครือข่าย Device จะต้องเปิดวงจรวิทยุขึ้นมาเพื่อหา Beacon จาก Coordinator และขอเข้าไปในเครือข่าย ดังรูปที่ 3.4 หลังจากสามารถเข้าไปในเครือข่ายได้ Device จะส่งข้อมูล ซึ่งประกอบด้วย ระยะเวลาหน่วงสูงสุดที่ยอมรับได้ อัตราการส่งข้อมูล และจำนวนข้อมูลต่อเฟรม ไปยัง Coordinator แล้วจึงกลับไป เพื่อรอให้ Coordinator กำหนดค่า Duty-cycle ที่เหมาะสม แล้วส่งมาใน Beacon ถัดไป หลังจาก Device ได้รับ Beacon พร้อมพารามิเตอร์ใหม่เพื่อปรับ Duty-cycle จาก Coordinator แล้ว ก็จะปรับ Duty-cycle ให้เหมาะสมกับการส่งข้อมูลและพร้อมที่จะส่งข้อมูล

จากกระบวนการปรับ Duty-cycle แบบอัตโนมัติตามวิธีที่กล่าวมานั้นจะเห็นว่า Coordinator มีบทบาทสำคัญในการปรับ Duty-cycle โดยอาศัยข้อมูลจาก Device ที่อยู่ในเครือข่ายดังนั้นกระบวนการที่ Coordinator นำข้อมูลจาก Device มาคำนวณนั้นจึงเป็นขั้นตอนสำคัญที่จะทำให้กระบวนการปรับ Duty-cycle แบบอัตโนมัติสมบูรณ์โดยวิธีการคำนวณนั้นจะกล่าวในหัวข้อถัดไป

### 3.3 วิธีการคำนวณค่า Duty-cycle ที่เหมาะสม

ในหัวข้อนี้จะกล่าวถึงวิธีที่ Coordinator ใช้ในการคำนวณค่า Duty-cycle ที่เหมาะสมต่อการส่งข้อมูลในอัตราการส่งข้อมูลที่ต่างกันออกไปตามความต้องการของ Device โดยในมาตรฐาน IEEE 802.15.4 นั้น Duty-cycle ของการส่งข้อมูลแบบ Beacon-enabled นั้นขึ้นอยู่กับพารามิเตอร์ 2 ตัวหลักดังรูปที่ 3.5 คือ 1. Beacon interval (BI) ซึ่งหมายถึงช่วงเวลาที่ Coordinator จะต้องส่ง Beacon frame หรือช่วงเวลา 1 คาบของการหลับและตื่นของ Coordinator และ 2. Superframe Duration (SD) ซึ่งหมายถึงช่วงเวลาที่ใช้ในการรับ-ส่งข้อมูลหรือช่วงที่ Coordinator ตื่น



รูปที่ 3.5 รูปแบบของ Superframe [1]

ในการใช้งานจริงนั้นเราไม่สามารถปรับค่า Beacon interval (BI) และ Superframe Duration (SD) ได้ตรงๆ เนื่องจากใน Superframe Specification ที่ส่งมาพร้อม Beacon นั้นไม่ได้มีพารามิเตอร์ Beacon interval (BI) และ Superframe Duration (SD) ส่งมาด้วย เราจึงต้องปรับค่าพารามิเตอร์เหล่านี้ด้วยตัวแปร macBeaconOrder (BO) และ macSuperFrameOrder (SO) แทน ซึ่ง Beacon interval (BI) นี้จะสัมพันธ์กับ macBeaconOrder (BO) ตามสมการ (3.1)

$$BI = aBaseSuperFrameDuration \times 2^{BO} \text{ symbols} \quad (3.1) [1]$$

และ Superframe Duration (SD) จะสัมพันธ์กับ macSuperFrameOrder (SO) ตามสมการ (3.2)

$$SD = aBaseSuperFrameDuration \times 2^{SO} \text{ symbols} \quad (3.2) [1]$$

จากสมการ (3.1) และ (3.2) จะเห็นได้ว่าค่าจะมีตัวแปรที่ยังไม่ทราบค่า นั่นคือ  $aBaseSuperFrameDuration$  ซึ่งเป็นค่าคงที่ได้ถูกกำหนดไว้ดังสมการที่ (3.3)

$$aBaseSuperFrameDuration = aBaseSlotDuration \times aNumberSuperframe! \quad (3.3) [1]$$

โดย

$$aBaseSlotDuration = 60 \text{ symbols} \quad (3.4) [1]$$

และ

$$aNumberSuperframeSlots = 16 \quad (3.5) [1]$$

ดังนั้นจากสมการที่ (3.3), (3.4) และ (3.5) จึงได้ว่า

$$aBaseSuperFrameDuration = 60 \times 16 = 960 \text{ symbols} \quad (3.6)$$

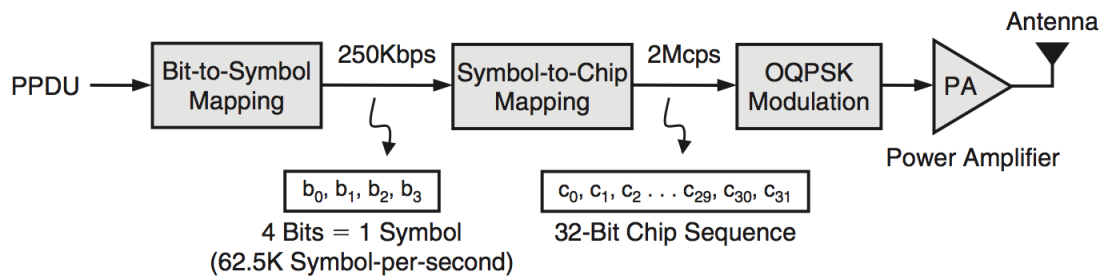
เมื่อทราบค่า  $aBaseSuperFrameDuration$  แล้วก็สามารถนำมาแทนกลับในสมการที่ (3.1) และ (3.2) ได้เป็นสมการที่ (3.7) และ (3.8)

$$BI = 960 \times 2^{B_0} \text{ symbols} \quad (3.7)$$

$$SD = 960 \times 2^{B_0} \text{ symbols} \quad (3.8)$$

จากสมการ (3.7) และ (3.8) ทำให้เราสามารถคำนวณเวลาหาช่วง Active จากค่า SD ได้ และสามารถคำนวณหาช่วงเวลาระหว่างคาบได้จาก BI แต่หน่วยที่ได้ออกมาในขณะจะเป็นหน่วย Symbols ซึ่งยากต่อความเข้าใจของผู้ใช้งาน เนื่องจากผู้ใช้งานจะต้องกำหนดระยะเวลาของระยะเวลาหน่วยสูงสุดที่ยอมรับได้ให้แก่ Device เป็นหน่วยวินาที ดังนั้นจึงมีความจำเป็นที่จะต้องแปลงหน่วย Symbols ให้เป็นหน่วยวินาที โดยก่อนที่จะแปลงได้ จะต้องทราบความถี่ที่ใช้งานเสียก่อน ซึ่งในวิทยานิพนธ์นี้จะกล่าวถึงเฉพาะความถี่ 2.4 GHz เท่านั้น และก่อนที่จะทราบค่าของหน่วย symbols ได้นั้น จะต้องทราบถึงกระบวนการส่งข้อมูลของมาตรฐาน IEEE 802.15.4 ด้วย





รูปที่ 3.6 กระบวนการ Spreading และ Modulation สำหรับคลื่น 2.4 GHz [59]

กระบวนการการส่งข้อมูลผ่านอากาศของคลื่นความถี่ 2.4 GHz นั้นสามารถอธิบายได้ตามรูปที่ 3.6 คือ เมื่อมีข้อมูลต้องการส่งออกทางอากาศ จะเริ่มต้นด้วยกระบวนการ Bit-to-Symbol Mapping เพื่อแปลงข้อมูล Bit ให้เป็น Symbol โดย 1 Symbol สามารถแทนข้อมูลได้ 4 Bit และตามมาตรฐาน IEEE 802.15.4 จะต้องสามารถส่งข้อมูลได้ด้วยความเร็ว 250 Kb/s หรือ 62.5 KSymbol/s หลังจากนั้นจึงทำกระบวนการ Symbol-to-Chip Mapping โดย 1 Symbol จะถูกแทนที่ด้วย 32 Bit Chip ตามตารางที่ 3.1 เพื่อให้ข้อมูลที่ส่งออกไป สามารถทนต่อสัญญาณรบกวนได้สูงมากยิ่งขึ้น แล้วจึงส่งออกไปด้วยความเร็ว 2Mcps ผ่านกระบวนการ Modulation แบบ O-QPSK ออกทางอากาศ

Data symbol (Decimal)	Data symbol (binary) ( $b_0 b_1 b_2 b_3$ )	Chip values ( $c_0 c_1 \dots c_{30} c_{31}$ )
0	0 0 0 0	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1	1 0 0 0	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
2	0 1 0 0	0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
3	1 1 0 0	0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
4	0 0 1 0	0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1
5	1 0 1 0	0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0
6	0 1 1 0	1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1
7	1 1 1 0	1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1
8	0 0 0 1	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
9	1 0 0 1	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1
10	0 1 0 1	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1
11	1 1 0 1	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
12	0 0 1 1	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0
13	1 0 1 1	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1
14	0 1 1 1	1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0
15	1 1 1 1	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0

ตารางที่ 3.1 Symbol-to-chip mapping [1]

PHY (MHz)	Frequency band (MHz)	Spreading parameters		Data parameters		
		Chip rate (kchip/s)	Modulation	Bit rate (kb/s)	Symbol rate (ksymbol/s)	Symbols
868/915	868-868.6	300	BPSK	20	20	Binary
	902-928	600	BPSK	40	40	Binary
868/915 (optional)	868-868.6	400	ASK	250	12.5	20-bit PSSS
	902-928	1600	ASK	250	50	5-bit PSSS
868/915 (optional)	868-868.6	400	O-QPSK	100	25	16-ary Orthogonal
	902-928	1000	O-QPSK	250	62.5	16-ary Orthogonal
<b>2450</b>	<b>2400-2483.5</b>	<b>2000</b>	<b>O-QPSK</b>	<b>250</b>	<b>62.5</b>	<b>16-ary Orthogonal</b>

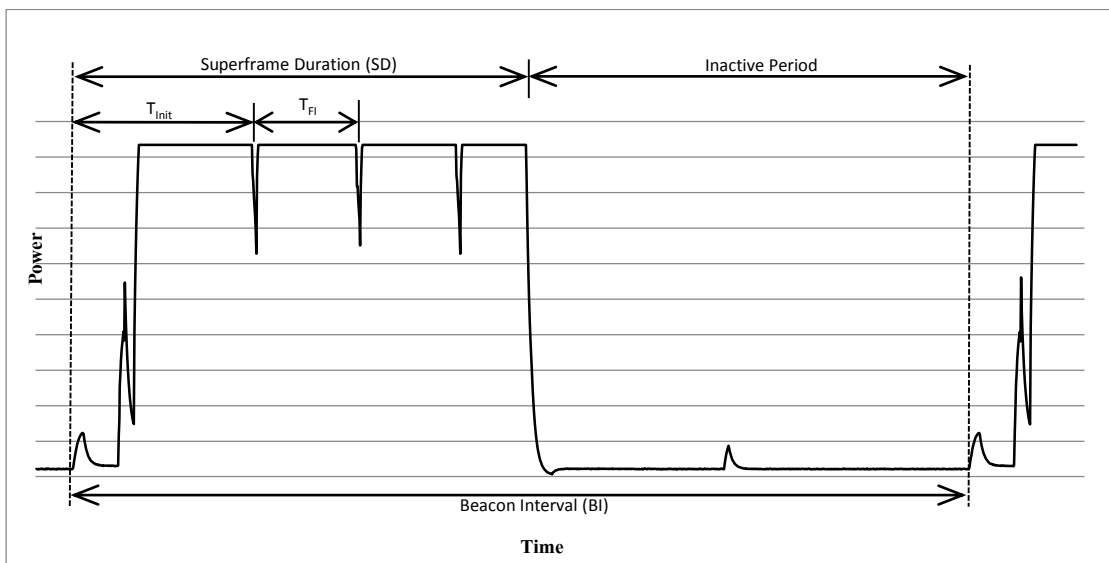
ตารางที่ 3.2 คลื่นความถี่และอัตราการส่งข้อมูล [1]

จากกระบวนการส่งข้อมูลข้างต้นที่กล่าวมา สามารถสรุปได้เป็นข้อมูลในตารางที่ 3.2 ซึ่งแสดงความสัมพันธ์ระหว่างคลื่นความถี่และอัตราการส่งข้อมูลที่เป็นหน่วย kbps และ ksymbol/s บนมาตรฐาน IEEE 802.15.4 จะเห็นได้ว่าที่คลื่นความถี่ 2.4 GHz นั้นสามารถส่งข้อมูลได้ด้วยอัตรา 62.5 ksymbol/s ซึ่งคิดเป็น  $1.6 \times 10^{-5}$  วินาทีต่อ symbol เมื่อนำมาแทนค่าในสมการที่ (3.7) และ (3.8) จะได้ดังสมการที่ (3.9) และ (3.10)

$$BI = 960 \times 2^{BO} \times 1.6 \times 10^{-5} = 15.36 \times 2^{BO} \text{ms} \quad (3.9)$$

$$SD = 960 \times 2^{SO} \times 1.6 \times 10^{-5} = 15.36 \times 2^{SO} \text{ms} \quad (3.10)$$

จากสมการที่ (3.9) และ (3.10) ทำให้เราสามารถทราบถึงระยะเวลาตื่น (Active) และหลับ (Inactive) ของอุปกรณ์ได้ เมื่อเปลี่ยนค่า BO และ SO เป็นค่าต่างๆ หลังจากนั้นเราจะต้องคำนวณหาจำนวน Packet ที่สามารถส่งได้ในช่วง Active Period เพื่อที่จะคำนวณออกมาเป็นปริมาณข้อมูลที่จะสามารถส่งได้ในช่วง Active Period ที่แตกต่างกันออกไป



รูปที่ 3.7 การใช้พลังงานรับ-ส่งข้อมูลของ Coordinator

จำนวน Packet ที่ส่งได้ในแต่ละช่วง Beacon Interval นั้นขึ้นอยู่กับระยะเวลาของช่วง Active period โดยจากรูปที่ 3.7 แสดงการใช้พลังงานของ Coordinator ทำให้เราสามารถทราบถึงลักษณะการรับส่งข้อมูลได้จากระดับพลังงานที่แตกต่างกันออกไปในแต่ละช่วงเวลาซึ่งสามารถสร้างเป็นสมการได้ดังสมการที่ (3.11)

$$SD \geq T_{Init} + (T_{FI} \times (F_{Total} - 1)) \quad (3.11)$$

เมื่อ

SD คือ ช่วง Active Period หรือช่วงเวลาที่ Coordinator จะตื่นขึ้นมาเพื่อให้ Device ส่งข้อมูลได้

$T_{Init}$  คือ ช่วงเวลาในการส่งข้อมูล Frame แรกโดยนับตั้งแต่เริ่มต้น Beacon จนกระทั่งมีการ Acknowledge Frame แรกซึ่งจากการทดลองพบว่ามีค่าเฉลี่ยเท่ากับ 26.1 ms

$F_{Total}$  คือ จำนวน Frame ทั้งหมดที่ส่งได้ภายในเวลาช่วง Active Period

$T_{FI}$  คือ ระยะเวลาที่ใช้ในการส่งข้อมูลแต่ละเฟรม โดยระยะเวลานี้จะขึ้นอยู่กับจำนวนข้อมูลที่ส่งออกไปในเฟรมนั้นๆ เมื่อมีจำนวนข้อมูลต่อเฟรมมากขึ้น ก็ต้องใช้ระยะเวลาในการส่งข้อมูลเพิ่มขึ้นด้วย

จากสมการที่ (3.11) สิ่งที่เราสนใจคือ จำนวนเฟรมที่ส่งได้ในช่วง Active Period หรือตัวแปร  $F_{Total}$  ดังนั้นจึงต้องปรับสมการที่ (3.11) ให้เป็นสมการที่ (3.12)

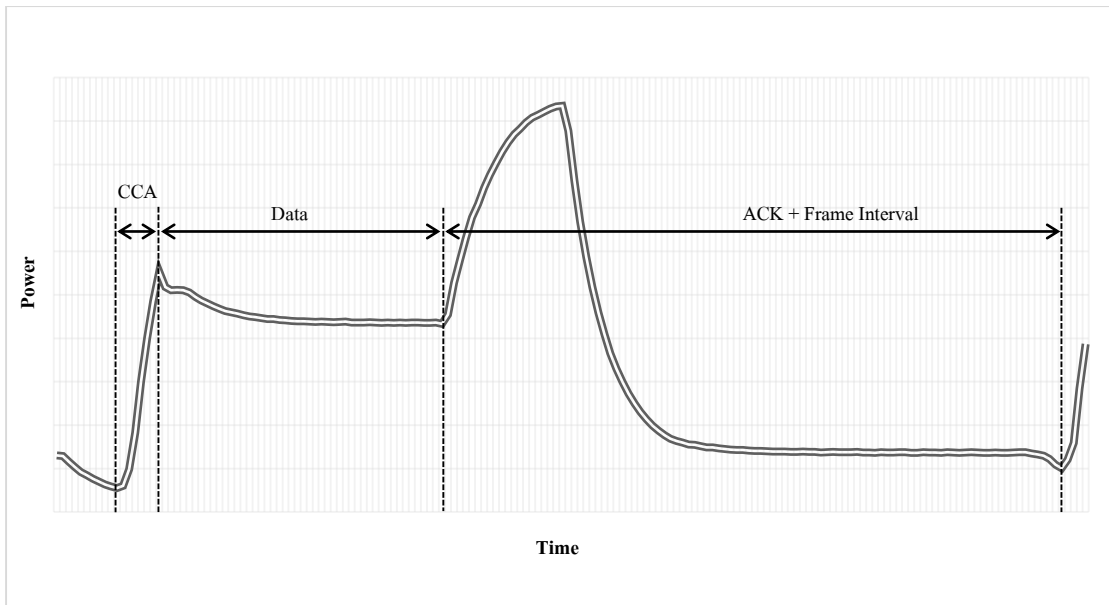
$$F_{Total} \leq \left( \frac{SD - T_{Init}}{T_{FI}} \right) + 1 \quad (3.12)$$

แทนค่า สมการที่ (3.12) ด้วยสมการที่ (3.10) จะได้สมการที่ (3.13)

$$F_{Total} \leq \left( \frac{(0.01536 \times 2^{S_0}) - T_{Init}}{T_{FI}} \right) + 1 \quad (3.13)$$

แทนค่าคงที่  $T_{Init}$  ลงในสมการที่ (3.13) จะได้สมการที่ (3.14)

$$F_{Total} \leq \left( \frac{(0.001536 \times 2^{S_0}) - 0.0261}{T_{FI}} \right) + 1 \quad (3.14)$$



รูปที่ 3.8 ลักษณะการใช้พลังงานของ Device ขณะส่งข้อมูลแต่ละเฟรม

เนื่องจากค่า  $T_{FI}$  นั้นค่าแตกต่างกันตามขนาดข้อมูลที่ส่งในแต่ละเฟรม ดังนั้นจึงต้องคำนวณหาค่า  $T_{FI}$  ของการส่งข้อมูลขนาดต่างๆกัน โดยอาศัยข้อมูลจากการวัดพลังงานของ Device ดังรูปที่ 3.8 ซึ่งจะเห็นได้ว่า ในการส่งข้อมูลแต่ละเฟรม จะแบ่งเป็นช่วงการป้องกันการชนของข้อมูล (CCA) ช่วงการส่งข้อมูล และช่วงการ Acknowledgment (ACK) ดังนั้นจึงสามารถสร้างเป็นสมการหาค่า  $T_{FI}$  ได้ดังสมการที่ (3.15)

$$T_{FI} = T_{CCA} + T_{Data} + T_{ACK} \quad (3.15)$$

เมื่อ

$T_{CCA}$  คือ ช่วงเวลาที่ Device ตรวจสอบว่าช่องสื่อสารว่างหรือไม่ ก่อนที่จะส่งข้อมูล เพื่อป้องกันการชนกันของข้อมูลที่ส่งออกไป ซึ่งจากการทดลองพบว่ามีค่าเฉลี่ยอยู่ที่ 0.058 ms

$T_{ACK}$  คือ ช่วงเวลาหลังจากที่ Device ส่งข้อมูลออกไปแล้วรอ Acknowledgment จาก Coordinator บวกกับระยะเวลาระหว่างเฟรม ซึ่งเป็นเวลาที่ Device รอก่อนจะส่งข้อมูลเฟรมถัดไป โดยจากการทดลองพบว่ามีค่าเฉลี่ยอยู่ที่ 1 ms

$T_{Data}$  คือ ช่วงเวลาที่ Device ส่งข้อมูลออกไป โดยช่วงเวลานี้จะขึ้นอยู่กับขนาดข้อมูลที่ส่งออกไปจากชั้น MAC Layer ซึ่งจากรูปที่ 3.9 จะเห็นได้ว่าข้อมูลที่ส่งออกจาก MAC Layer นั้นประกอบด้วย MHR, MAC Payload และ MFR โดยค่า MHR ในการทดลองนี้จะใช้ขนาดที่เล็กที่สุดนั่นคือ 3 Byte

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10/14	variable	2
Frame Control	Sequence Number	Destination PAN Identifier	Destination Address	Source PAN Identifier	Source Address	Auxiliary Security Header	Frame Payload	FCS
Addressing fields								
MHR							MAC Payload	MFR

รูปที่ 3.9 ลักษณะของเฟรม MAC บนมาตรฐาน IEEE 802.15.4 [1]

จากหัวข้อก่อนหน้าทำให้เราทราบแล้วว่าในการส่งข้อมูล 1 symbol จะต้องใช้เวลา  $1.6 \times 10^{-5}$  วินาที และแต่ละ Symbol จะสามารถแทนข้อมูลในชั้น MAC Layer ได้ 4 bit ดังนั้นในการส่งข้อมูล 1 Byte จะต้องใช้เวลาตามสมการที่ (3.16)

$$T_{Byte} = 1.6 \times 10^{-5} \times 2 = 3.2 \times 10^{-5} \text{ Second} \quad (3.16)$$

จากสมการที่ (3.16) เราสามารถสร้างสมการหา  $T_{Data}$  ได้ดังสมการที่ (3.17)

$$T_{Data} = DPF_{MAC} \times 3.2 \times 10^{-5} \text{ Second} \quad (3.17)$$

เมื่อ  $DPF_{MAC}$  คือ จำนวนข้อมูลที่ต้องการส่งออกจากชั้น MAC Layer โดยมีหน่วยเป็น Byte

เมื่อได้ข้อมูลในการหาค่า  $T_{FI}$  ครบแล้วก็สามารถแทนค่ากลับไปในสมการที่ (3.15) ได้ ดังนั้นแทนค่าคงที่และสมการ (3.17) ลงไปในสมการ (3.15) จะได้สมการ (3.18)

$$T_{FI} = 0.01058 + (DPF_{MAC} \times 3.2 \times 10^{-5}) \quad (3.18)$$

แทนค่าจากสมการที่ (3.18) กลับไปในสมการที่ (3.14) จะได้สมการที่ (3.19)

$$F_{\text{Total}} \leq \left( \frac{(0.01536 \times 2^{\text{SO}}) - 0.0261}{0.01058 + (\text{DPF}_{\text{MAC}} \times 3.2 \times 10^{-5})} \right) + 1 \quad (3.19)$$

ในขณะนี้ เราได้สมการความสัมพันธ์ระหว่างค่า SO,  $\text{DPF}_{\text{MAC}}$  และ  $F_{\text{Total}}$  ดังสมการที่ (3.19) แต่ในการใช้งานจริงจะต้องเปลี่ยน  $F_{\text{Total}}$  ให้เป็นอัตราการส่งข้อมูลต่อหน่วยเวลาเสียก่อน ดังนั้นจึงเปลี่ยนหน่วยการส่งข้อมูลจากจำนวนเฟรมต่อคาบ ให้เป็นจำนวนเฟรมต่อวินาที โดยการนำจำนวนเฟรมที่สามารถส่งได้ มาหารด้วยเวลาในการส่งข้อมูลต่อคาบ (BI) ซึ่งจะได้ดังสมการที่ (3.20)

$$\text{FPS}_{\text{MAC}} = \frac{F_{\text{Total}}}{\text{BI}} \quad (3.20)$$

อัตราการส่งข้อมูลในหน่วยเฟรมต่อวินาทีซึ่งนำไปใช้งานจริงไม่ได้ จะต้องแปลงหน่วยอีกครั้งให้เป็น ไบต์ต่อวินาที จึงต้องหาความสัมพันธ์ระหว่างเฟรมต่อวินาที กับ ไบต์ต่อวินาที ซึ่งสามารถคำนวณได้จาก อัตราการส่งข้อมูล (ไบต์ต่อวินาที) หารด้วยจำนวนข้อมูลต่อเฟรมดังสมการที่ (3.21)

$$\text{FPS}_{\text{MAC}} = \frac{\text{BPS}}{\text{DPF}_{\text{MAC}}} \quad (3.21)$$

จากสมการที่ (3.20) และ สมการที่ (3.21) สามารถรวมกันได้เป็นสมการที่ (3.22)

$$\frac{\text{BPS}}{\text{DPF}_{\text{MAC}}} = \frac{F_{\text{Total}}}{15.36 \times 10^{-3} \times 2^{\text{BO}}} \quad (3.22)$$

จากสมการที่ (3.22) สามารถจัดรูปใหม่ให้อยู่ในรูปของ  $F_{\text{Total}} = f(\text{BPS})$  ได้ดังสมการที่ (3.23)

$$F_{\text{Total}} = \frac{\text{BPS}}{\text{DPF}_{\text{MAC}}} \times 15.36 \times 10^{-3} \times 2^{\text{BO}} \quad (3.23)$$

นำสมการที่ (3.23) กลับไปแทนค่าในสมการที่ (3.19) ได้ดังสมการที่ (3.24)

$$\frac{\text{BPS}}{\text{DPF}_{\text{MAC}}} \times 15.36 \times 10^{-3} \times 2^{\text{BO}} = \left( \frac{(0.01536 \times 2^{\text{SO}}) - 0.0261}{0.01058 + (\text{DPF}_{\text{MAC}} \times 3.2 \times 10^{-5})} \right) + 1 \quad (3.24)$$

จากสมการที่ (3.24) สามารถจัดรูปใหม่ได้ดังสมการที่ (3.25)

$$\text{BPS} = \frac{3125 \times \text{DPF}_{\text{MAC}} \times 2^{8-\text{BO}} (15 \times 2^{\text{SO}+5} + \text{DPF}_{\text{MAC}} - 485)}{1563 \times (8 \times \text{DPF}_{\text{MAC}} + 2645)} \quad (3.25)$$

สมการที่ (3.25) เป็นสมการที่แสดงความสัมพันธ์ระหว่าง อัตราการส่งข้อมูล (BPS), จำนวนข้อมูลที่ต้องการส่งต่อเฟรม ( $\text{DPF}_{\text{MAC}}$ ), ระยะเวลาการตื่นของ Coordinator (SO) และ ระยะเวลาต่อคาบ (BO) ซึ่งในที่นี้เราต้องการหาระยะเวลาการตื่นของ Coordinator (SO) ที่จะทำให้ Device สามารถส่งข้อมูลหา Coordinator ได้ทันก่อนที่เข้าสู่ช่วงการหลับ แต่การแก้สมการที่ (3.25) เพื่อหาค่า SO โดยตรงนั้นทำได้ยาก จึงต้องใช้วิธีการวนลูปหาค่า SO โดยการแทนค่า SO ลงในสมการที่ (3.25) แล้วหาค่า SO ที่น้อยที่สุดที่ทำให้สมการมีค่ามากกว่าหรือเท่ากับอัตราการส่งข้อมูลที่ต้องการ (BPS) ซึ่งสามารถเขียนเป็น Code ตัวอย่างได้ดังรายการที่ 3.1

```
#define SO_ERROR 15
uint8_t find_so(uint16_t bps, uint8_t dpf, uint8_t bo)
{
    uint8_t so;
    for(so = 1; so <= bo; so++)
    {
        if(bps <= (3125 * dpf * 2 ^ (8 - bo) * ((15 * 2 ^ (so + 5)) + dpf -
        485))/(1563 * (8 * dpf + 2645)))
        {
            return so;
        }
    }
    return SO_ERROR;
}
```

รายการที่ 3.1 ตัวอย่าง Code การหาค่า SO



ในการใช้งานตัวอย่าง Code ในรายการที่ 3.1 นั้น สามารถเรียกฟังก์ชัน `find_so()` พร้อมกับส่งพารามิเตอร์ต่างๆที่ฟังก์ชันต้องการ จะได้ค่า SO ที่เหมาะสมกลับมา แต่ในบางกรณี เราอาจจะไม่สามารถระบุพารามิเตอร์ให้แก่ฟังก์ชันทั้งหมดได้ หรือเราต้องการที่จะระบุแค่อัตราการส่งข้อมูลเพียงอย่างเดียว แล้วต้องการส่งข้อมูลในอัตราการส่งข้อมูลที่ระบุ โดยใช้พลังงานน้อยที่สุด ก็สามารถทำได้โดยการกำหนดจำนวนข้อมูลที่ต้องการส่งต่อเฟรม (dpf) ให้เป็นจำนวนสูงสุดที่จะส่งออกจากชั้น MAC Layer ได้ ซึ่งบนระบบปฏิบัติการ TinyOS ระบุไว้ว่า สามารถส่ง MAC Payload ได้สูงสุด 115 Byte และกำหนดระยะเวลาระหว่าง Beacon (BO) ให้เป็นค่าสูงสุดเช่นกันโดยมาตรฐาน IEEE 802.15.4 กำหนดให้ค่าสูงสุดที่สามารถจะกำหนดได้คือ 14 แต่ในการใช้งานบน Unode สามารถทำได้สูงสุดที่ 12 หาก BO มากกว่า 12 ก็จะทำให้ไม่สามารถ Synchronization Beacon ได้สำเร็จเนื่องจากปัญหาความแม่นยำของเวลาที่อยู่บน Unode

จากวิธีการที่กล่าวมา หากมี Device ต้องการส่งข้อมูล Blood Pressure ไปยัง Coordinator ด้วยอัตราการส่งข้อมูล 240 ไบต์ต่อวินาที โดยไม่ได้สนใจว่าข้อมูลจะส่งด้วยระยะเวลาหนึ่งวงเท่าไร เราสามารถคำนวณหาค่า SO ที่ทำให้ส่งข้อมูลได้โดยใช้พลังงานน้อยที่สุดจากการเรียกฟังก์ชัน `find_so()` ตามกระบวนการที่กล่าวมาซึ่งสามารถเขียนเป็น Code การเรียกใช้งานฟังก์ชันได้ดังรายการที่ 3.2

```
uint8_t so, bo = 12;
uint8_t mac_payload_len = 115;
uint8_t mac_header_and_checksum_len = 5;
uint16_t data_rate = 240;
so = find_so( data_rate, mac_payload_len + mac_header_and_checksum_len,
bo)
.....
```

รายการที่ 3.2 ตัวอย่างการเรียกใช้งานฟังก์ชัน `find_so()`

จากตัวอย่างเราสามารถคำนวณค่า SO ออกมาได้เท่ากับ 7 ซึ่งเป็นค่าที่ทำให้ส่งข้อมูล 240 ไบต์ต่อวินาทีได้ โดยใช้พลังงานน้อยที่สุดบนอุปกรณ์ U-node แต่ในขณะนี้ เราได้กำหนดค่า BO ให้เป็นค่าที่มากที่สุด ซึ่งส่งผลให้ระยะเวลาหน่วงของเครือข่าย อาจจะสูงโดยไม่จำเป็น เราจึงต้องการค่า BO ที่จะทำให้อุปกรณ์ใช้พลังงานเท่า เดิมแต่มีระยะเวลาหน่วงที่ลดลงจากเดิม ซึ่งสามารถทำได้โดยการหาค่า BO ที่น้อยที่สุด ที่ยังคงทำให้ Duty-cycle เท่าเดิม เนื่องจาก Duty-cycle ส่งผลต่อการใช้พลังงานของอุปกรณ์โดยตรง ถ้าหาก Duty-cycle สูงขึ้นก็จะทำให้ใช้พลังงานมากขึ้นแต่ก็ได้อัตราการส่งข้อมูลที่สูงขึ้นเช่นกัน โดย Duty-cycle นี้จะสามารถคำนวณได้จากพารามิเตอร์ BO และ SO โดยมีความสัมพันธ์ดังสมการที่ (3.26)

$$\text{Duty cycle} = \frac{2^{SO}}{2^{BO}} \quad (3.26)$$

ในการคำนวณหา Duty-cycle ด้วยสมการที่ (3.26) อาจเกิดความจำเป็นเนื่องจากเราต้องการเพียงเปรียบเทียบ Duty-cycle ว่ามีค่าเท่ากันหรือไม่ซึ่งสามารถทำได้โดยการหาผลต่างของ BO และ SO ดังสมการที่ (3.27)

$$\text{Order}_{\text{diff}} = BO - SO \quad (3.27)$$

การหาค่า BO และ SO ที่น้อยที่สุดที่ยังคงใช้พลังงานในการส่งข้อมูลเท่าเดิม เพื่อลดระยะเวลาหน่วงของเครือข่าย สามารถเขียนเป็น Code ตัวอย่างได้ดังรายการที่ 3.3

```

#define SO_ERROR 15
void find_bo_so(uint16_t bps, uint8_t dpf, uint8_t bo_max, uint8_t *bo,
uint8_t *so)
{
    uint8_t order_diff;
    *bo = 0;
    *so = find_so( bps, dpf, bo_max);
    if(*so == SO_ERROR)
        return;
    order_diff = bo_max - *so;
    for(*bo = order_diff + 1, *bo <= bo_max, *bo++)
    {
        *so = find_so( bps, dpf, *bo);
        if(order_diff <= (*bo - *so))
            return;
    }
}

```

### รายการที่ 3.3 ตัวอย่าง Code การวนหาค่า SO และ BO ที่ดีที่สุด

การใช้งานฟังก์ชัน `find_bo_so()` นั้นคล้ายกับการใช้งานฟังก์ชัน `find_so()` คือ นำข้อมูลที่ได้จาก Device ส่งไปให้ฟังก์ชัน แต่จะมีข้อแตกต่างที่จะต้องส่งตัวแปรเพื่อใช้ในการรับค่า BO และ SO กลับมาซึ่งสามารถเขียนเป็น Code ตัวอย่างการเรียกใช้งานฟังก์ชันได้ดังรายการที่ 3.4

```

uint8_t so, bo, bo_max = 12;
uint8_t mac_payload_len = 115;
uint8_t mac_header_and_checksum_len = 5;
uint16_t data_rate = 240;
find_so_bo( data_rate, mac_payload_len + mac_header_and_checksum_len,
bo_max, &bo, &so);
.....

```

### รายการที่ 3.4 ตัวอย่างการเรียกใช้งานฟังก์ชัน `find_so_bo()`

จากตัวอย่างการเรียกใช้งานฟังก์ชัน `find_so_bo()` ในรายการที่ 3.4 จะได้ค่า BO เท่ากับ 9 และ SO เท่ากับ 4 จะเห็นได้ว่าเมื่อเปรียบเทียบกับ การเรียกฟังก์ชัน `find_so()` ตามตัวอย่าง Code ในรายการที่ 3.2 เราได้ค่า BO ที่ต่ำลงแต่ยังคงมีค่า Duty-cycle เท่าเดิม ซึ่งส่งผลให้ระยะเวลาหน่วงของเครือข่ายต่ำลงและยังคงใช้พลังงานเท่าเดิม

ฟังก์ชัน `find_so_bo()` ในรายการที่ 3.4 นั้นสามารถคำนวณหาค่า Duty-cycle ที่เหมาะสม ที่จะทำให้ส่งข้อมูลด้วยอัตราการส่งข้อมูลที่กำหนด โดยใช้พลังงานน้อยที่สุด แต่ระยะเวลาหน่วงของเครือข่ายนั้นจะไม่แน่นอน เนื่องจากฟังก์ชันจะหาค่า BO ที่ทำให้ระยะเวลาหน่วงต่ำที่สุด และยังคงใช้พลังงานน้อยที่สุดด้วย ซึ่งในบางครั้งค่า BO ที่คำนวณออกมาได้อาจจะมากเกินไป ส่งผลให้ระยะเวลาหน่วงของเครือข่ายสูงขึ้นตามไปด้วย ดังนั้นเราจึงต้องกำหนดค่า BO ที่เหมาะสมให้กับฟังก์ชัน `find_so_bo()` ตามระยะเวลาหน่วงสูงสุดที่ Device ยอมรับได้ โดยจากสมการ(3.9) ทำให้เราทราบว่าระยะเวลาระหว่าง Beacon นั้นแปรผันตามค่า BO และค่าระยะเวลาหน่วงสูงสุดของเครือข่ายนั้น เท่ากับระยะเวลาระหว่าง Beacon ดังนั้นจากสมการที่ (3.9) จึงได้ว่าระยะเวลาหน่วงสูงสุดของเครือข่ายแปรผันตาม BO ดังสมการที่ (3.28)

$$\text{Latency}_{\max} = 15.36 \times 2^{\text{BO}} \text{ms} \quad (3.28)$$

จากตัวอย่างก่อนหน้านี้เราได้สมมติการส่งข้อมูล Blood Pressure ด้วยอัตราการส่งข้อมูล 240 ไบต์ต่อวินาที โดยไม่สนใจระยะเวลาหน่วงที่จะเกิดขึ้น เมื่อคำนวณด้วยฟังก์ชัน `find_so_bo()` แล้วพบว่าได้ค่า BO เท่ากับ 9 แล้วนำมาคำนวณด้วยสมการที่ (3.28) พบว่าระยะเวลาหน่วงสูงสุดที่จะเกิดขึ้นได้คือ  $15.36 \times 2^9 = 7864.32 \text{ ms}$  ในทางกลับกันหากต้องการเปลี่ยนค่า ระยะเวลาหน่วงเป็นค่า BO จะต้องเปลี่ยนสมการที่ (3.28)ให้อยู่ในรูปของ  $\text{BO} = f(\text{Latency}_{\max})$  ดังสมการที่ (3.29)

$$\text{BO} \approx 1.4427 \log(0.0651042 \times \text{Latency}_{\max}) \quad (3.29)$$

เนื่องจากเซนเซอร์โหนดเป็นอุปกรณ์ที่มีทรัพยากรจำกัด ดังนั้นการแปลงค่า Latency เป็น BO โดยใช้สมการที่ (3.29) ที่อยู่ในรูปของ log ต้องใช้ทรัพยากรในการประมวลผลค่อนข้างสูง อีกทั้งค่า BO นั้นเป็นจำนวนเต็มที่อยู่ในช่วง 0-14 เท่านั้น ดังนั้นจึงใช้วิธีการวนลูบหาค่า BO โดยอ้างอิงจากสมการที่ (3.28) ซึ่งสามารถเขียนเป็น Code ตัวอย่างได้ตามรายการที่ 3.5

```

uint8_t get_bo(uint16_t latency_max_ms)
{
    uint8_t bo;

    for(bo = 14; bo >= 1; bo--)
    {
        if(15.36*(2^bo) <= latency_max_ms)
        {
            return bo;
        }
    }
    return 0;
}

```

### รายการที่ 3.5 ตัวอย่าง Code การหาค่า BO จากค่า Latency

ตัวอย่าง Code ในรายการที่ 3.5 นั้นสามารถใช้คำนวณหาค่า BO จากค่าระยะเวลาหน่วง โดยการแทนค่า ระยะเวลาหน่วงสูงสุดที่ยอมรับได้ ซึ่งมีหน่วยเป็น ms ลงในฟังก์ชัน `get_bo()` ก็จะได้ค่า BO ที่ทำให้มีระยะเวลาหน่วงน้อยกว่าหรือเท่ากับที่กำหนดได้

ในที่นี้จะยกตัวอย่างการใช้งานฟังก์ชัน `find_so_bo()` ที่มีการจำกัดระยะเวลาหน่วงสูงสุด โดยสมมติให้ Device ส่งข้อมูล Cardiac Output ที่มีอัตราการการส่งข้อมูล 80 ไบต์ต่อวินาที และกำหนดให้ระยะเวลาหน่วงสูงสุดของเครือข่ายมีค่าไม่เกิน 1 วินาที ชั้นแรกจะต้องแปลงค่าระยะเวลาหน่วงให้เป็นค่า BO เสียก่อน โดยการเรียกใช้ฟังก์ชัน `get_so()` ซึ่งสามารถเขียนเป็น Code ตัวอย่างได้ดังรายการที่ 3.6

```

uint16_t latency_max_ms = 1000;
uint8_t bo_max;
bo_max = get_bo(latency_max_ms);
.....

```

### รายการที่ 3.6 ตัวอย่าง Code การเรียกใช้งานฟังก์ชัน `get_bo()`

เมื่อได้ค่า BO ที่ต้องการแล้วก็เรียกใช้ฟังก์ชัน `get_bo_so()` แล้วส่งค่า BO และ อัตราการส่งข้อมูลที่ต้องการไปเป็นพารามิเตอร์ดังตัวอย่าง Code ในรายการที่ 3.7

```

uint16_t latency_max_ms = 1000;
uint8_t  mac_payload_len = 115;
uint8_t  mac_header_and_checksum_len = 5;
uint16_t data_rate = 80;
uint8_t  bo_max, bo, so;
bo_max = get_bo(latency_max_ms);
find_so_bo( data_rate, mac_payload_len + mac_header_and_checksum_len,
bo_max, &bo, &so);
.....

```

### รายการที่ 3.7 การใช้งานฟังก์ชัน find\_so\_bo() โดยกำหนดค่า Latency

จากตัวอย่าง Code ในรายการที่ 3.7 จะได้ผลลัพธ์ค่า BO เท่ากับ 6 และค่า SO เท่ากับ 1 ซึ่งเมื่อนำค่า BO ไปแทนในสมการที่ (3.28) พบว่าระยะเวลาหน่วงสูงสุดจะมีค่าเท่ากับ 983.04 ms ซึ่งมีค่าน้อยกว่าระยะเวลาหน่วงสูงสุดที่เรากำหนดไว้ แต่การใช้พลังงานก็จะสูงขึ้นตามไปด้วยเนื่องจาก Duty-cycle มีค่าสูงขึ้น

จากตัวอย่างที่ผ่านมาเราได้กำหนดอัตราการส่งข้อมูล และระยะเวลาหน่วงสูงสุดที่ยอมรับได้ แล้วใช้ฟังก์ชัน find\_so\_bo() ในการคำนวณหาค่า BO และ SO ที่เหมาะสมออกมา โดยกำหนดขนาดข้อมูลต่อเฟรมเป็นค่าคงที่ นั่นหมายถึงโปรแกรมของเราจะต้องสามารถนำข้อมูลที่อ่านมาได้จากเซนเซอร์ เก็บรวมไว้จนได้ขนาดที่ต้องการแล้วจึงค่อยส่งข้อมูลออกไป ซึ่งในบางครั้งขนาดของข้อมูลต่อเฟรม ก็อาจจะไม่แน่นอน ดังนั้นจึงจำเป็นที่จะต้องให้ Device สามารถระบุขนาดข้อมูลต่อเฟรมที่ต้องการจะส่งได้ โดยวิธีการแบบเดียวกันกับตัวอย่าง Code ในรายการที่ 3.7 แต่เปลี่ยนจากการระบุขนาดของข้อมูลต่อเฟรมแบบคงที่ เป็นระบุขนาดข้อมูลต่อเฟรมที่ Device ต้องการลงไปในการามิเตอร์ dpf ของฟังก์ชัน find\_bo\_so() ในตัวอย่าง Code รายการที่ 3.3

จากกระบวนการข้างต้นที่กล่าวมา ทำให้สามารถสร้างเครือข่ายที่ปรับขนาดของ Duty-cycle ได้ตามความต้องการของเซนเซอร์แบบทันที ส่งผลให้สามารถลดการใช้พลังงานที่ไม่จำเป็นของอุปกรณ์ และยังสามารเพิ่มประสิทธิภาพของเครือข่ายได้ตามความต้องการอีกด้วย

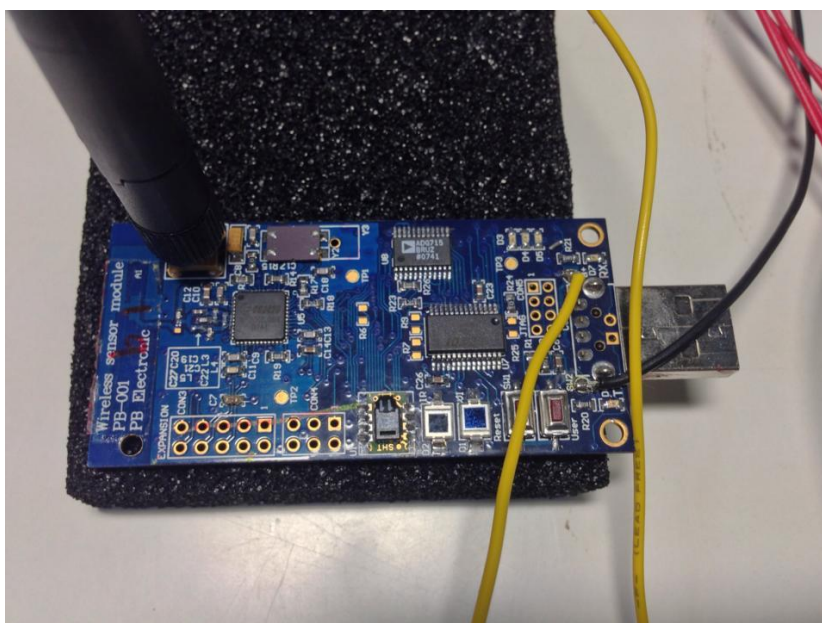
## บทที่ 4

### การทดสอบและวิเคราะห์ประสิทธิภาพ

ในบทนี้จะกล่าวถึงวิธีการทดสอบและวัดประสิทธิภาพของกระบวนการปรับ Duty-cycle แบบอัตโนมัติที่ได้พัฒนาขึ้นมาจากงานวิจัยนี้ โดยจะทดสอบอัตราการใช้พลังงานของอุปกรณ์เมื่อมีอัตราการส่งข้อมูลที่แตกต่างกันออกไป และ ทดสอบระยะเวลาหน่วงสูงสุดที่จะเกิดขึ้นได้ในเครือข่าย เปรียบเทียบกันระหว่างแบบ Duty-cycle คงที่ กับ แบบปรับ Duty-cycle ได้ อัตโนมัติที่ได้พัฒนาขึ้นมาในงานวิจัยนี้

#### 4.1 เครื่องมือที่ใช้ในการทดสอบ

##### 4.1.1 โหนดที่ใช้ในการทดสอบ



รูปที่ 4.1 Unode ที่พัฒนาขึ้นมาโดยห้องวิจัยเครือข่ายเซนเซอร์ไร้สาย

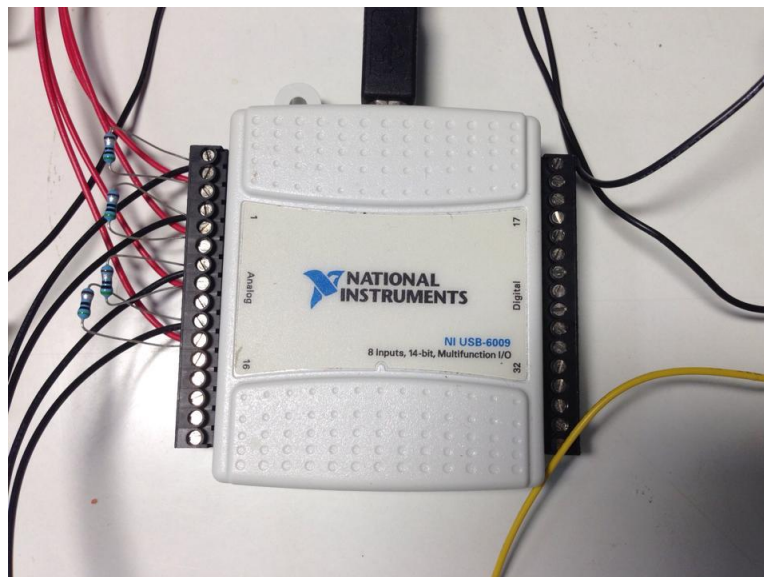
ในการทดสอบประสิทธิภาพของกระบวนการปรับ Duty-cycle แบบอัตโนมัติ นั้น เราได้เลือกใช้ “Unode” ในรูปที่ 4.1 ซึ่งเป็นโหนดที่พัฒนาขึ้นมาโดยห้องวิจัยเครือข่าย เซนเซอร์ไร้สาย มหาวิทยาลัยสงขลานครินทร์ อ้างอิงการออกแบบจาก Tmote Sky[3] ประมวลผล ด้วยไมโครคอนโทรลเลอร์ MSP430f1611[60] ความเร็วสัญญาณนาฬิกา 8 MHz และ รับ-ส่ง สัญญาณวิทยุด้วยโมดูล CC2420[25, p. 24] ที่รองรับมาตรฐาน IEEE 802.15.4 ทำงานอยู่บน ระบบปฏิบัติการ TinyOS[26] และภาษาที่ใช้ในการเขียนโปรแกรมคือ ภาษา NesC

#### 4.1.2 เซนเซอร์ที่ใช้ในการทดสอบ

การเลือกเซนเซอร์ที่จะนำมาใช้นั้น จะต้องเลือกเซนเซอร์ที่ใช้พลังงานต่ำ และสามารถประมวลผลข้อมูลได้เองบ้างส่วน เพื่อลดภาระของไมโครคอนโทรลเลอร์ ซึ่งในที่นี้เราได้เลือกเซนเซอร์ Digital Accelerometer 3 Axis เพื่อนำมาใช้ในการตรวจจับการล้มและนับก้าวการเดิน โดยเลือกเซนเซอร์รุ่น ADXL345[27] ที่สามารถตรวจสอบการตกอย่างอิสระได้ (Free-fall detection) และใช้พลังงานต่ำ

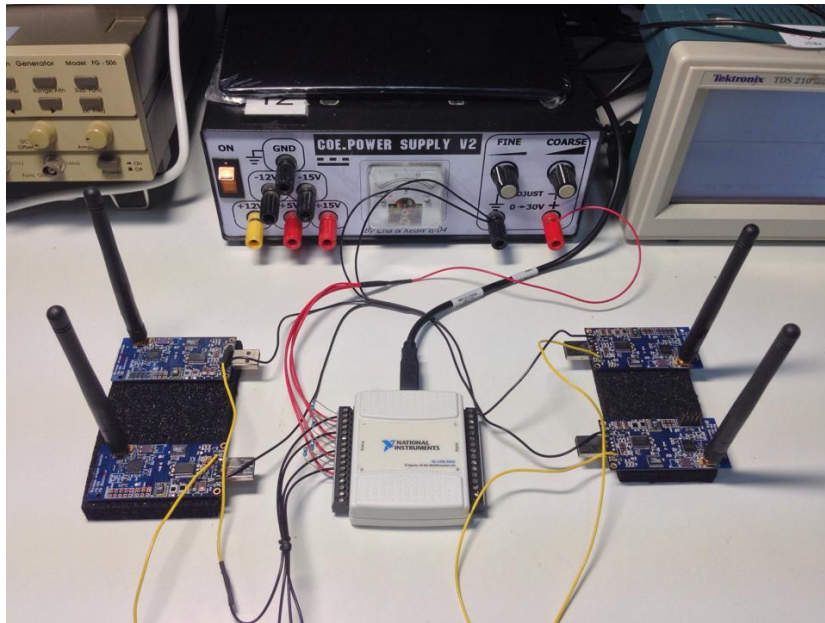


### 4.1.3 อุปกรณ์ที่ใช้ในการวัดค่าพลังงาน



รูปที่ 4.2 NI USB-6009 สำหรับใช้ในการวัดค่าพลังงาน

การวัดค่าพลังงานเป็นกระบวนการที่สำคัญต่อการวิจัยนี้ เพราะนอกจากจะใช้ในการวัดประสิทธิภาพของโปรโตคอลที่ออกแบบแล้ว ยังต้องใช้ค่าพลังงานที่วัดได้ มาวิเคราะห์หาพฤติกรรมการหลับ และตื่นของ Unode ดังนั้นอุปกรณ์ที่เลือกใช้จะต้องมีความเร็วในการเก็บข้อมูลที่สูง และต้องสามารถส่งข้อมูลออกมาเก็บไว้ เพื่อนำไปวิเคราะห์ย้อนหลังได้ ซึ่งอุปกรณ์ที่เลือกใช้คือ “NI USB-6009” ในรูปที่ 4.2 ที่สามารถแปลงค่าอนาล็อกเป็นดิจิทัลได้ด้วยความละเอียด 12 บิตและอัตราการเก็บข้อมูลสูงสุดที่ 48 KS/s และวัดกระแส โดยการวัดแรงดันที่ไหลผ่านตัวต้านทานขนาด 5 โอห์ม ความคลาดเคลื่อน 1 % ดังรูปที่ 4.3



รูปที่ 4.3 การต่อ Unode เข้ากับ NI USB-6009 เพื่อวัดพลังงาน

#### 4.2 วิธีการและสภาพแวดล้อมในการทดสอบ

การทดสอบโพรโตคอลที่ได้วิจัยขึ้นมานั้น จะทดสอบโดยแบ่งออกเป็นสองช่วง คือ 1) ช่วงการทดสอบโดยไม่ได้ใช้เซนเซอร์ เพื่อควบคุมปัจจัยการใช้พลังงานจากภายนอก ซึ่งในช่วงนี้ จะทดสอบโดยการจำลองข้อมูลที่มีขนาดแตกต่างกันออกไป โดยข้อมูลนี้จะถูกสุ่มขึ้นมา 2) ช่วงการทดสอบโดยใช้ข้อมูลจากเซนเซอร์จริง โดยจะนำเซนเซอร์นับก้าวและตรวจจับการหกล้มมาต่อใช้งานจริงแล้ววัดผลการใช้พลังงาน

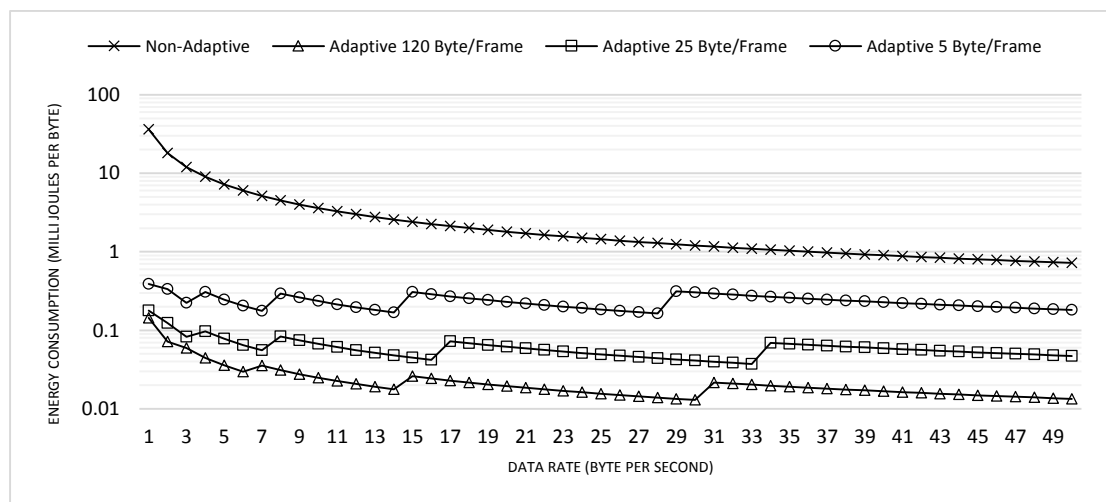
Parameter	Value	Unit
Supply voltage	2.4	V
Free air temperature	25	°C
Current Consumption: Active Period [MCU on, Radio RX]	30	mA
Current Consumption: Sleep Period [MCU standby, Radio standby]	45	uA
Current Consumption: Sleep Period [MCU standby (with interrupt sensor), Radio standby]	450	uA
Current Consumption: Sleep Period [MCU on (Read analog sensor), Radio standby]	2400	uA
Current Consumption: Fall Sensor [ADXL345]	140	uA
Battery Capacity	1600	mAh
BO Value for Fixed Duty-cycle	7	-
SO Value for Fixed Duty-cycle	6	-

#### ตารางที่ 4.1 ค่าพารามิเตอร์ที่ใช้ในการทดสอบ

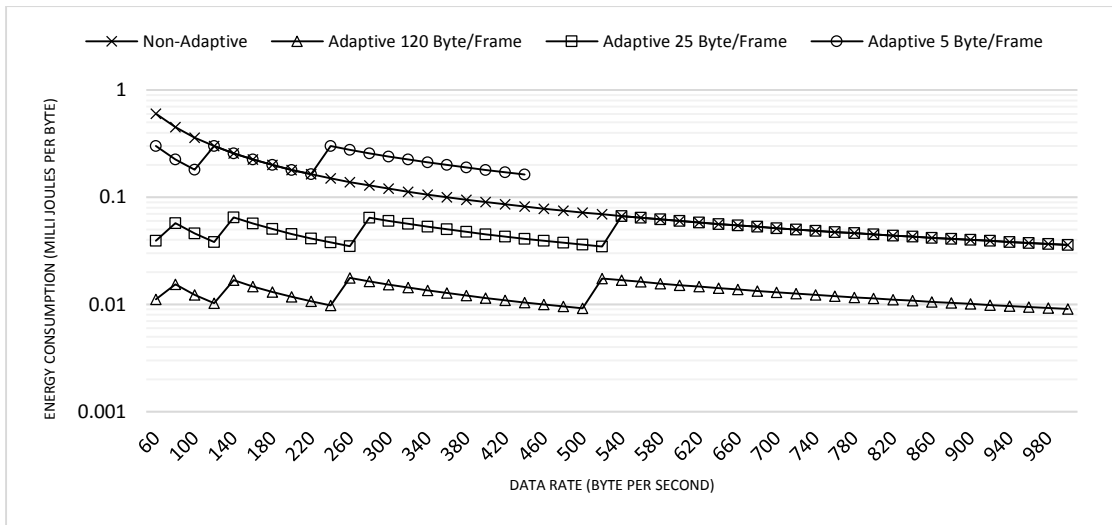
ในการทดสอบโปรโตคอลนั้น เราจะต้องกำหนดสภาพแวดล้อมเพื่อควบคุมการทดสอบโดยจะมีค่าพารามิเตอร์ต่างๆที่จำเป็นสำหรับการทดสอบดังตารางที่ 4.1 ซึ่งประกอบด้วย แรงดันที่ใช้มีค่า 2.4 โวลต์โดยอ้างอิงแรงดันจากแบตเตอรี่ชนิด Nickel-Metal Hydride ขนาด AA จำนวน 2 ก้อน อุณหภูมิอากาศโดยรอบที่ใช้ทดสอบ 25 °C และความจุแบตเตอรี่ที่ใช้ในการคำนวณกำหนดให้มีค่า 1600 mAh

### 4.3 ทดสอบอัตราการใช้พลังงานของโพรโทคอลแบบไม่ควบคุมระยะเวลาหน่วง

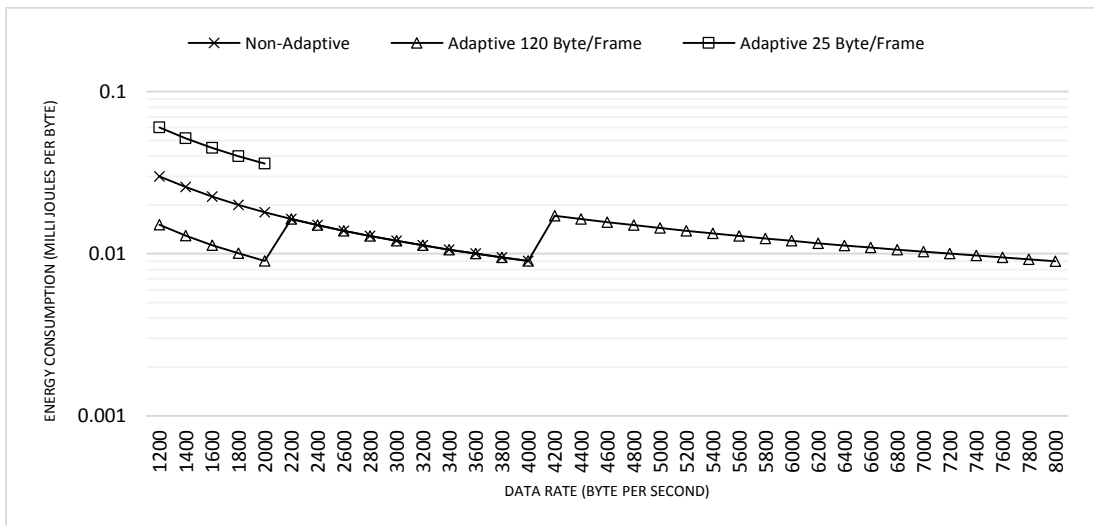
ในหัวข้อนี้จะทดสอบพร้อมเปรียบเทียบอัตราการใช้พลังงานและระยะเวลาหน่วง ระหว่าง MAC โพรโทคอลแบบ Duty-cycle คงที่ (Non-Adaptive) กับ MAC โพรโทคอลแบบปรับเปลี่ยน Duty-cycle อัตโนมัติที่ได้วิจัยขึ้นมา (Adaptive) โดยในส่วนของพลังงานนั้นจะเปรียบเทียบระหว่าง อัตราการใช้พลังงาน (จุดต่อไบต์) กับ อัตราการส่งข้อมูล (ไบต์ต่อวินาที) โดยอัตราการส่งข้อมูลจะแปรผันตั้งแต่ 1 ไบต์ต่อวินาที จนกระทั่งถึง 8000 ไบต์ต่อวินาที ซึ่งเป็นอัตราการส่งข้อมูลสูงสุดที่สามารถทำได้ และเพื่อให้เห็นการเปลี่ยนแปลงของข้อมูลได้ชัดเจน จึงได้แบ่งช่วงกราฟออกเป็น 3 ช่วงคือ 1-50, 51-1000 และ 1001-8000 ไบต์ต่อวินาทีตามลำดับ และส่วนของระยะเวลาหน่วงจะเปรียบเทียบระหว่างระยะเวลาหน่วงสูงสุดที่จะเกิดขึ้นหน่วยเป็นวินาที กับอัตราการส่งข้อมูลหน่วยเป็นไบต์ต่อวินาที โดยกำหนดจำนวนข้อมูลต่อเฟรมของ MAC โพรโทคอลแบบ Duty-cycle ไว้ที่ค่าสูงสุดคือ 120 ไบต์ต่อเฟรมเนื่องจากการแปรผันจำนวนข้อมูลต่อเฟรมใน MAC โพรโทคอลแบบ Duty-cycle ไม่ส่งผลกระทบต่อการใช้พลังงานของโหนด แต่ใน MAC โพรโทคอลแบบปรับเปลี่ยน Duty-cycle อัตโนมัตินั้น จะเปรียบเทียบกับขนาดข้อมูลต่อเฟรม 3 รูปแบบด้วยกันคือ 5 ไบต์ต่อเฟรม 25 ไบต์ต่อเฟรมและ 120 ไบต์ต่อเฟรม เนื่องจากกราฟที่แสดงผลเป็นกราฟในรูปแบบ Log ดังนั้นการปรับค่าขนาดข้อมูลต่อเฟรมจึงต้องปรับเป็นลักษณะของค่ายกกำลัง



รูปที่ 4.4 อัตราการใช้พลังงานเปรียบเทียบกับอัตราการส่งข้อมูล (1-50 Bps)



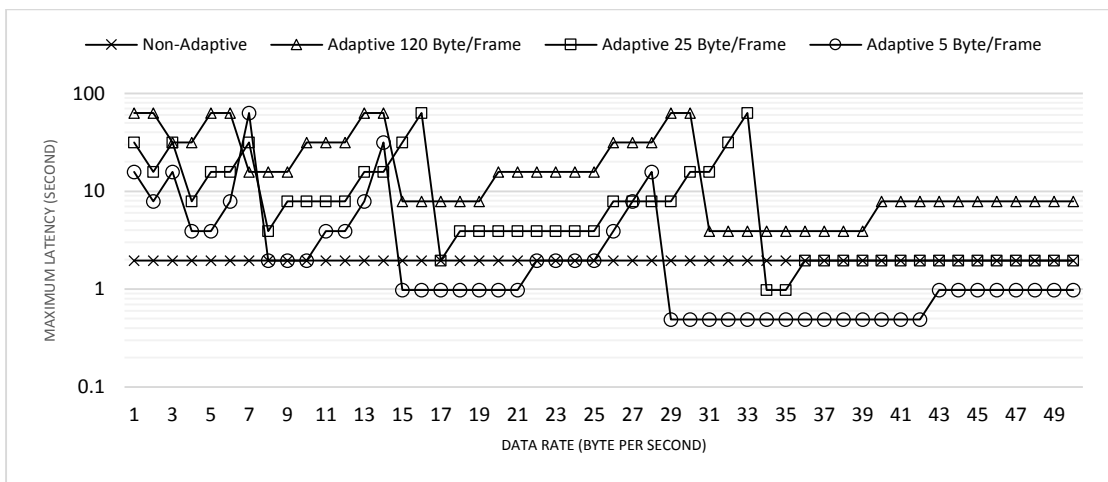
รูปที่ 4.5 อัตราการใช้พลังงานเปรียบเทียบกับอัตราการส่งข้อมูล (51-1000 Bps)



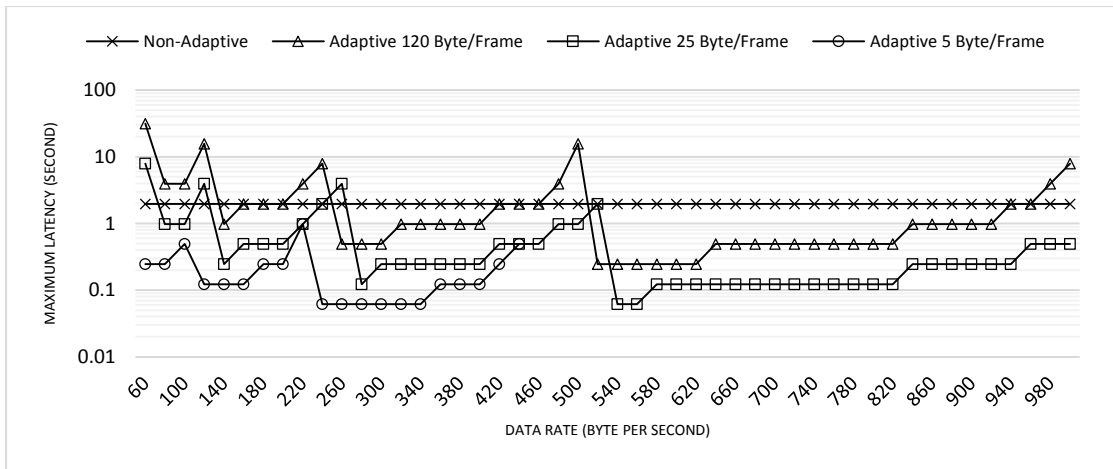
รูปที่ 4.6 อัตราการใช้พลังงานเปรียบเทียบกับอัตราการส่งข้อมูล (1001-8000 Bps)

จากรูปที่ 4.4 ถึง รูปที่ 4.6 แสดงถึงอัตราการใช้พลังงานต่ออัตราการส่งข้อมูล โดยแบ่งอัตราการส่งข้อมูลเป็น 3 ช่วงคือ ช่วง 1-50 ไบต์ต่อวินาที, 51-1000 ไบต์ต่อวินาที และ 1001-8000 ไบต์ต่อวินาที เพื่อให้สามารถมองเห็นการเปลี่ยนแปลงได้อย่างชัดเจน ซึ่งในแต่ละช่วงนั้นจะเปรียบเทียบกันระหว่าง MAC โพรโทคอลแบบ Duty-cycle คงที่ และ MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัติ โดย MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมนั้นแบ่งออกเป็น 3 ชนิดตามขนาดข้อมูลต่อเฟรมที่ต้องการจะส่งคือ 5 ไบต์, 25 ไบต์ และ 120 ไบต์

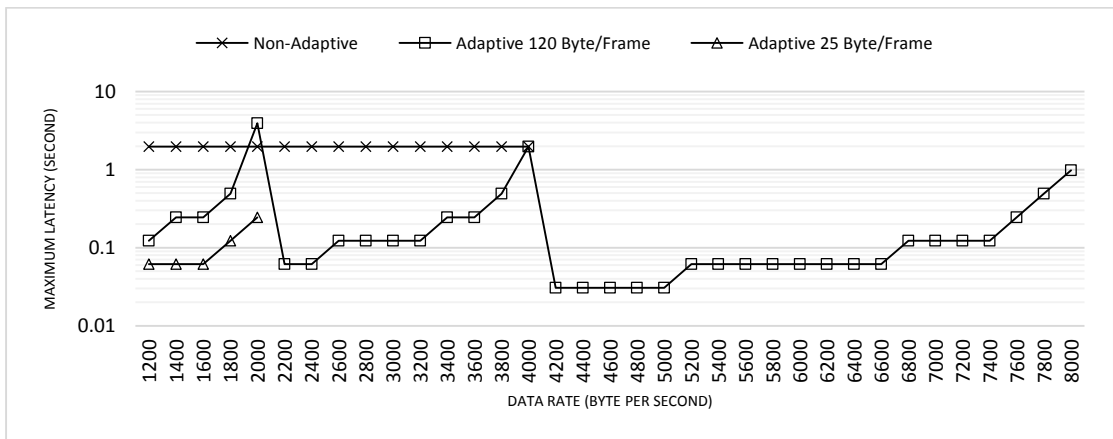
ผลการทดสอบประสิทธิภาพในด้านการใช้พลังงานพบว่าหากส่งข้อมูลด้วยอัตราการส่งข้อมูลที่ต่ำ อัตราการใช้พลังงานของ MAC โพรโทคอลแบบ Duty-cycle คงที่นั้น จะสูงกว่าแบบปรับ Duty-cycle อัตโนมัติกว่า 100 เท่า แต่เมื่อเพิ่มอัตราการส่งข้อมูลให้สูงขึ้นก็จะส่งผลให้อัตราการใช้พลังงานใกล้เคียงกัน แต่โดยภาพรวมแล้ว MAC โพรโทคอลแบบ Duty-cycle คงที่ก็ยังใช้พลังงานสูงกว่า และหากเปรียบเทียบ MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัติด้วยกัน ก็พบว่า ยังมีขนาดข้อมูลที่ต้องการส่งต่อเฟรมต่ำก็จะส่งผลให้อัตราการใช้พลังงานสูงขึ้นตามไปด้วย เนื่องจาก Overhead ที่เกิดขึ้นในการส่งแต่ละเฟรมและยังทำให้อัตราการส่งข้อมูลสูงสุดลดลงดังรูปที่ 4.6 จะเห็นได้ว่าอัตราการส่งข้อมูลสูงสุดของ 120 ไบต์ต่อเฟรม จะได้ประมาณ 8000 ไบต์ต่อวินาที แต่ในรูปที่ 4.5 อัตราการส่งข้อมูลสูงสุดของ 5 ไบต์ต่อเฟรม จะหยุดอยู่ที่ประมาณ 400 ไบต์ต่อวินาที เท่านั้น



รูปที่ 4.7 Latency สูงสุดเปรียบเทียบกับอัตราการส่งข้อมูล (1-50 Bps)



รูปที่ 4.8 Latency สูงสุดเปรียบเทียบกับอัตราการส่งข้อมูล (51-1000 Bps)



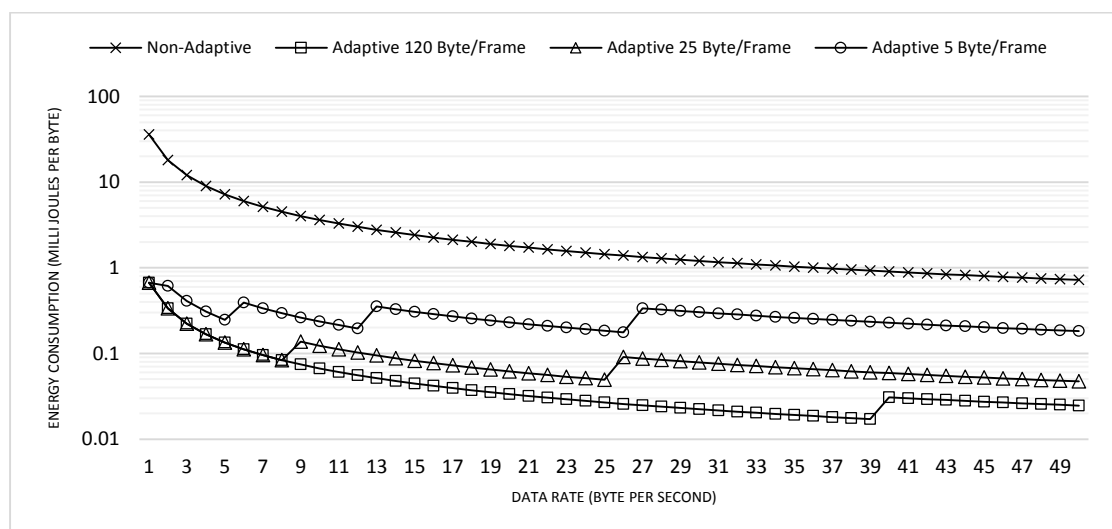
รูปที่ 4.9 Latency สูงสุดเปรียบเทียบกับอัตราการส่งข้อมูล (1001-8000 Bps)

จากรูปที่ 4.7 ถึง รูปที่ 4.9 แสดงถึงระยะเวลาช่วงเวลาหน่วงสูงสุดที่จะเกิดขึ้นได้ในเครือข่าย เมื่อมีอัตราการส่งข้อมูลที่แตกต่างกันออกไป โดยเปรียบเทียบกันระหว่าง MAC โพรโทคอล แบบ Duty-cycle คงที่ กับ MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัติ จะเห็นได้ว่าระยะเวลาหน่วงของ MAC โพรโทคอล แบบ Duty-cycle คงที่นั้น จะมีค่าคงที่ตลอดเวลา และในช่วงที่อัตราการส่งข้อมูลต่ำระยะเวลาหน่วงของ MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัติ จะมีค่าสูงกว่า MAC โพรโทคอล แบบ Duty-cycle คงที่ที่กว่า 10 เท่า แต่เมื่อมีอัตราการส่งข้อมูลที่สูงขึ้นระยะเวลาหน่วงของ MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัติก็จะลดลงมาต่ำกว่าแบบ Duty-cycle คงที่ เป็นบางช่วง

จากผลการทดลองที่ผ่านมาซึ่งเปรียบเทียบอัตราการใช้พลังงาน และระยะเวลา  
 หน่วงสูงสุดระหว่าง MAC โพรโทคอลแบบ Duty-cycle คงที่ กับ MAC โพรโทคอลแบบปรับ  
 Duty-cycle อัตโนมัติ สามารถสรุปได้ว่า MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัติ  
 สามารถปรับตัวเองให้เหมาะกับอัตราการส่งข้อมูลที่จะเกิดขึ้นได้ ทำให้ลดการใช้พลังงานในช่วง  
 ที่ต้องการส่งข้อมูลด้วยอัตราส่งข้อมูลต่ำได้อย่างมาก แต่สิ่งที่ส่งผลกระทบต่อตามมาคือ เราไม่  
 สามารถควบคุมระยะเวลาหน่วงสูงสุดที่จะเกิดขึ้นได้ ซึ่งอาจจะมีค่าสูงสุดกว่า 1 นาฬิกา ทำให้ส่งผล  
 ต่อการอ่านข้อมูลจากเซนเซอร์บางชนิดที่ต้องการส่งข้อมูลด้วยระยะเวลาหน่วงที่ต่ำ ไม่สามารถ  
 ใช้งานได้ เช่น เซนเซอร์ตรวจจับการหกล้ม ซึ่งหากตรวจจับการหกล้มแล้วก็ต้องส่งข้อมูลออกไป  
 ให้ได้ในเวลาที่กำหนด

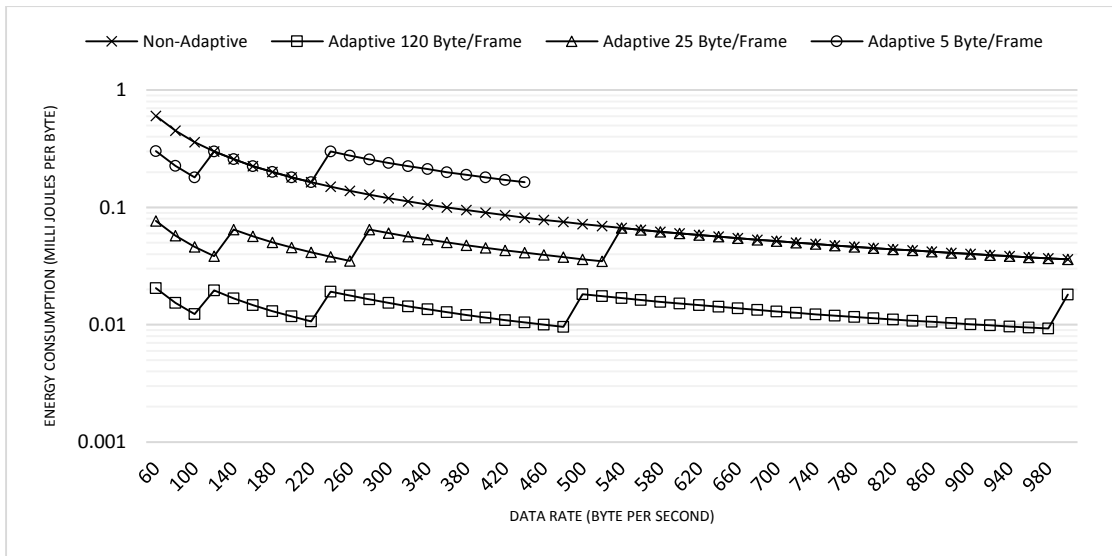
#### 4.4 ทดสอบอัตราการใช้พลังงานของโพรโทคอลแบบควบคุมระยะเวลาหน่วง

จากการทดสอบในหัวข้อที่ 0 จะเห็นได้ว่า MAC โพรโทคอลแบบปรับ Duty-cycle  
 อัตโนมัติสามารถลดการใช้พลังงานลงให้เหมาะสมกับอัตราการส่งข้อมูลได้ แต่หากจะ  
 นำไปใช้กับการส่งข้อมูลจากเซนเซอร์ที่ต้องการระยะเวลาหน่วงต่ำ เราจำเป็นต้องกำหนดขอบเขต  
 ของระยะเวลาหน่วงที่จะเกิดขึ้นไม่ให้เกินที่กำหนดตามสมการที่ (3.8) โดยในการควบคุม  
 ระยะเวลาหน่วงนั้น จะแปรผันด้วยค่า 4 วินาที และ 1 วินาที เนื่องจาก การแทนค่า BO ด้วย 8 และ  
 6 ตามลำดับ

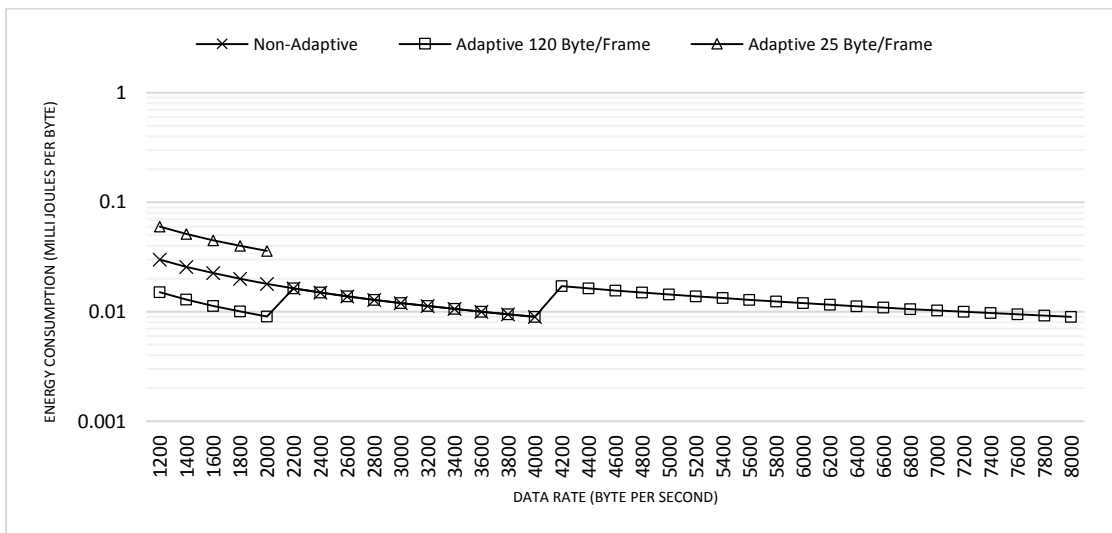


รูปที่ 4.10 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที (1-50 Bps)



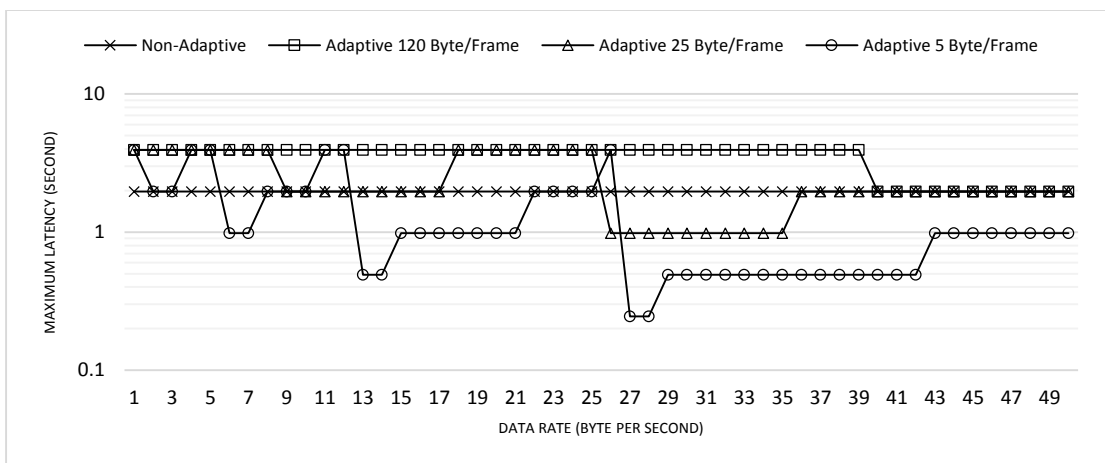


รูปที่ 4.11 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที (51-1000 Bps)

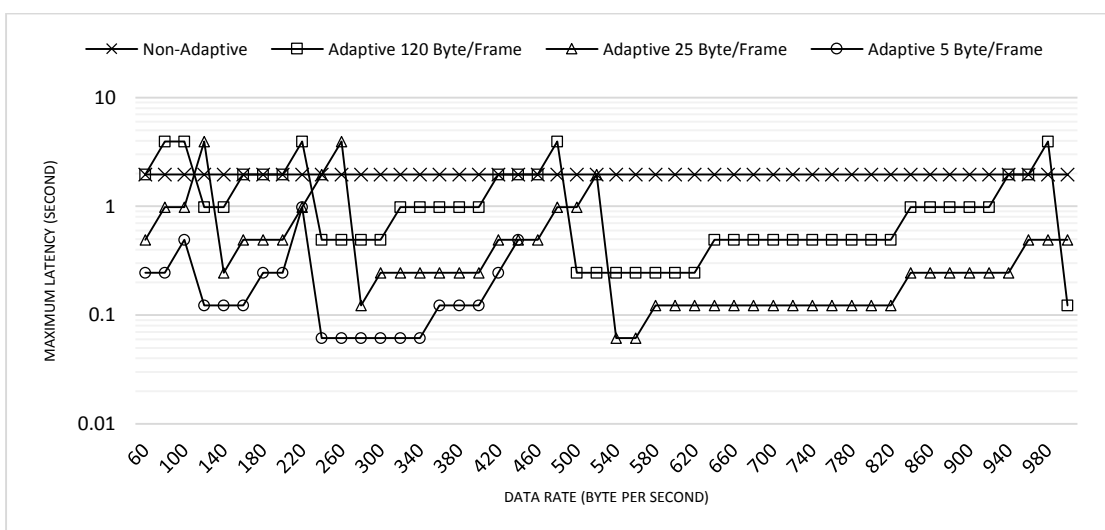


รูปที่ 4.12 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที (1001-8000 Bps)

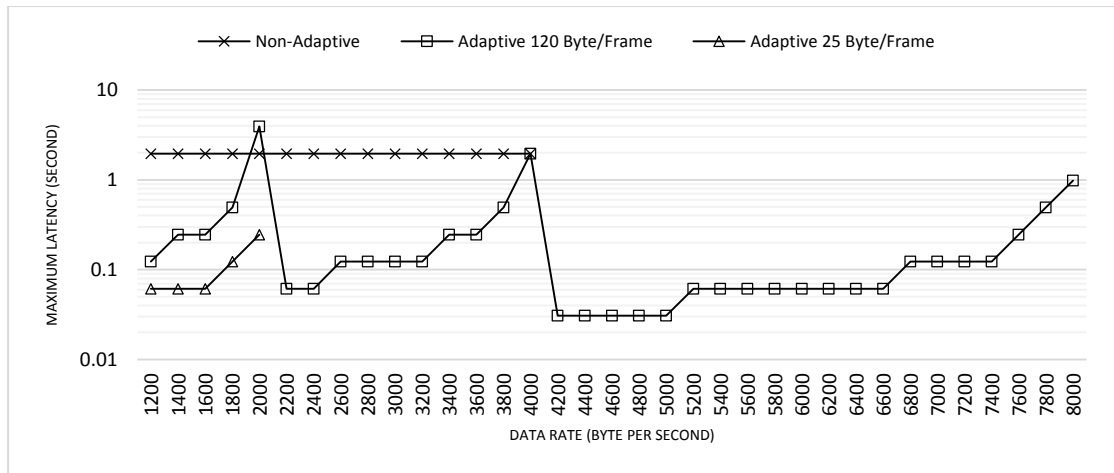
จากรูปที่ 4.10 ถึง รูปที่ 4.12 แสดงถึงอัตราการใช้พลังงานเปรียบเทียบกับอัตราการส่งข้อมูลโดยมีการควบคุมระยะเวลาหน่วงสูงสุดไม่ให้มีค่าเกิน 4 วินาที การทดสอบได้แบ่งอัตราการส่งข้อมูลออกเป็น 3 ช่วง เช่นเดียวกับการทดลองในหัวข้อที่ 0 ซึ่งจะเห็นได้ว่าการควบคุมระยะเวลาหน่วงไม่ให้มีค่าเกิน 4 วินาที การใช้พลังงานของ MAC โพรโทคอลแบบ Duty-cycle คงที่ ก็ยังคงใช้พลังงานเท่าเดิม แต่ MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัตินี้จะใช้พลังงานเพิ่มมากขึ้นกว่าตอนไม่ควบคุมระยะเวลาหน่วงอยู่ในช่วง 1-2 % ขึ้นอยู่กับอัตราการส่งข้อมูล ซึ่งโดยภาพรวมแล้ว การใช้พลังงานของ MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัตินี้ก็ยังคงใช้พลังงานน้อยกว่า MAC โพรโทคอลแบบ Duty-cycle คงที่



รูปที่ 4.13 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที (1-50 Bps)

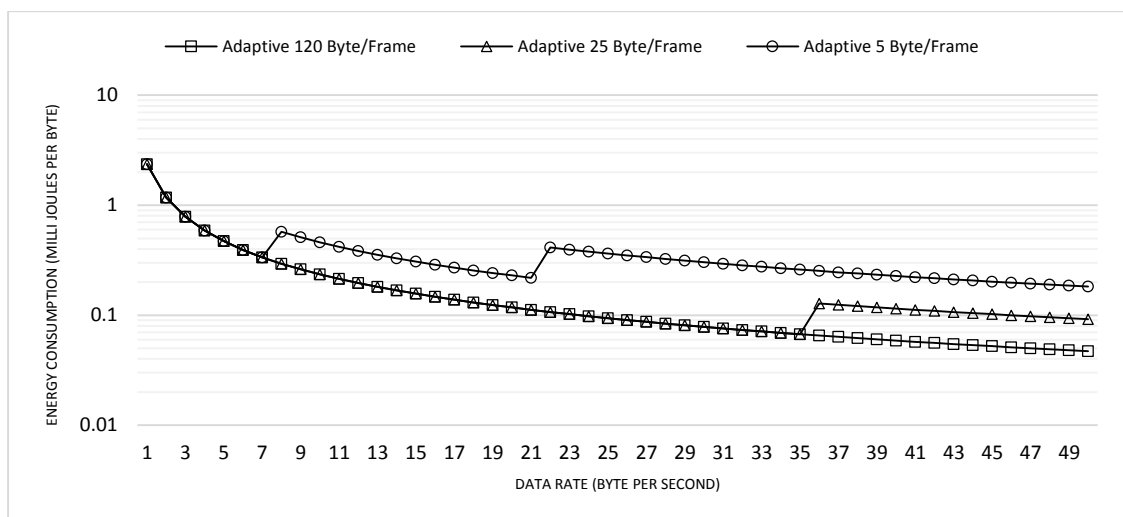


รูปที่ 4.14 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที (51-1000 Bps)

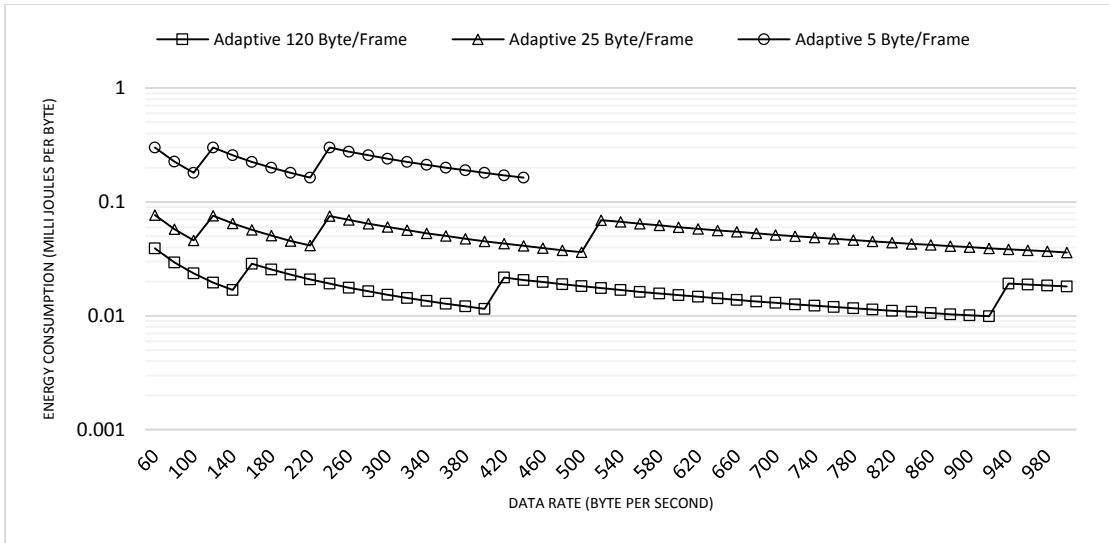


รูปที่ 4.15 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4วินาที(1001-8000 Bps)

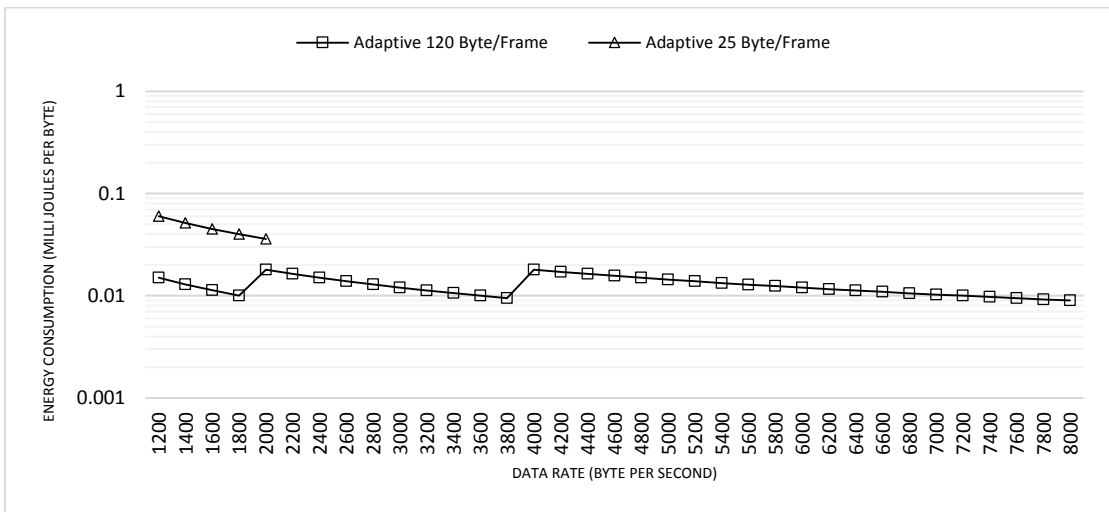
จากรูปที่ 4.13 ถึง รูปที่ 4.15 แสดงถึงระยะเวลาหน่วงสูงสุดที่มีโอกาสเกิดขึ้นเมื่อเปลี่ยนแปลงอัตราการส่งข้อมูลเป็นค่าต่างๆ จะเห็นได้ว่าเมื่อมีการควบคุมระยะเวลาหน่วงไม่ให้มีค่าเกิน 4 วินาที ระยะเวลาหน่วงของ MAC โพรโทคอลแบบ Duty-cycle คงที่ก็ยังคงมีค่าคงที่ เช่นเดิมที่ 2 วินาที แต่ระยะเวลาหน่วงของ MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัตินั้นปรับลดลงจากเดิม โดยตลอดช่วงการส่งข้อมูลจะมีระยะเวลาหน่วงสูงสุดไม่เกิน 4 วินาที



รูปที่ 4.16 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1วินาที (1-50 Bps)

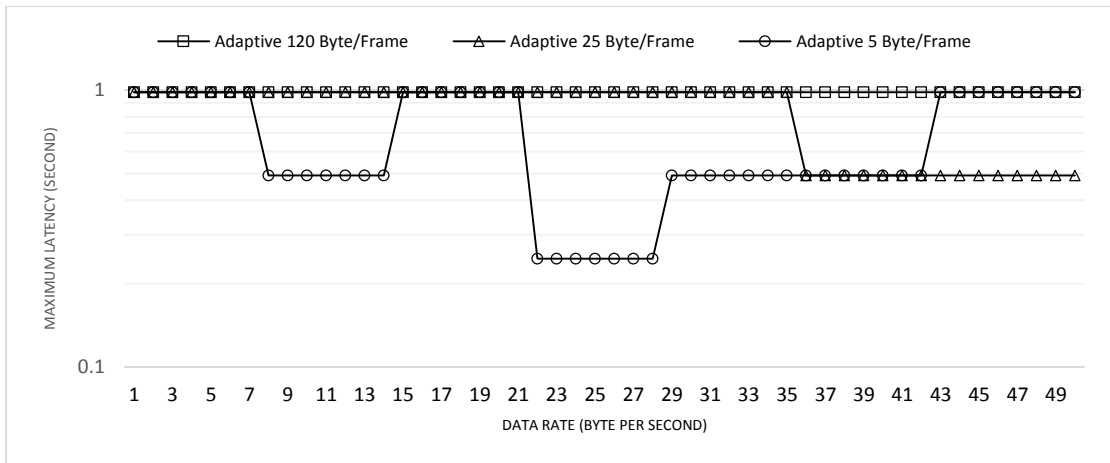


รูปที่ 4.17 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1 วินาที (51-1000 Bps)

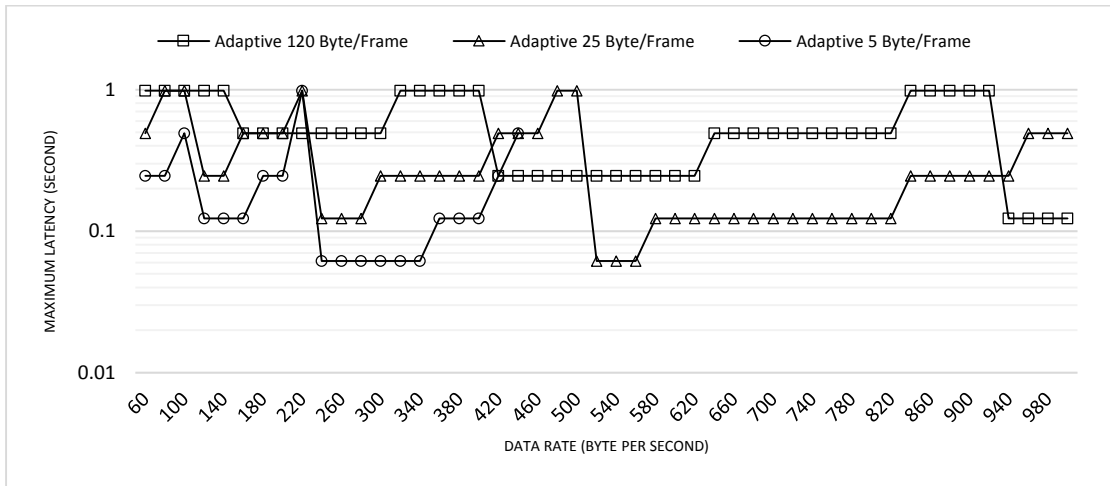


รูปที่ 4.18 การใช้พลังงานเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1 วินาที(1000-8000 Bps)

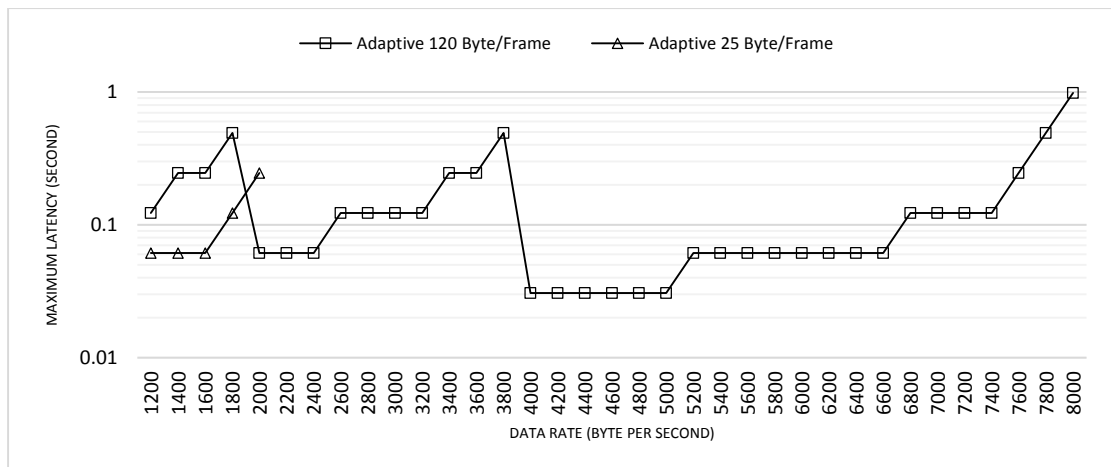
เมื่อลดค่าระยะเวลาหน่วงที่ควบคุมจาก 4 วินาที ลงมาเป็น 1 วินาที จะเห็นได้ว่า อัตราการใช้พลังงานของ MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัติจะมีค่าสูงกว่าเดิมกว่า 15 เท่า ดังแสดงในรูปที่ 4.16 ถึง รูปที่ 4.18 และในส่วนของ MAC โพรโทคอลแบบ Duty-cycle คงที่นั้น ไม่สามารถที่จะส่งข้อมูลได้หากควบคุมระยะเวลาหน่วงให้ต่ำกว่า 2 วินาที



รูปที่ 4.19 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1 วินาที (1-50 Bps)



รูปที่ 4.20 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1 วินาที (51-1000 Bps)



รูปที่ 4.21 Latency สูงสุดเทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1 วินาที(1001-8000 Bps)

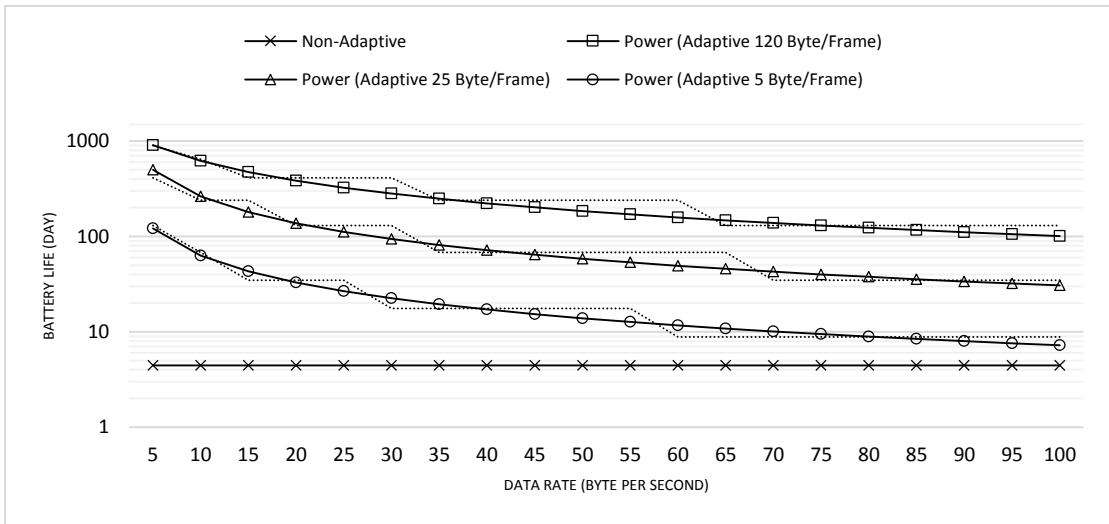
จากรูปที่ 4.19 ถึง รูปที่ 4.21 แสดงถึงระยะเวลาหน่วงสูงสุดที่มีโอกาสจะเกิดขึ้นเมื่อควบคุมระยะเวลาหน่วงไว้ที่ 1 วินาที ซึ่งจะเห็นได้ว่า MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัติก็ยังสามารถรักษาระดับระยะเวลาหน่วงไว้ไม่ให้เกิน 1 วินาทีได้ตลอดช่วงอัตราการส่งข้อมูล

จากผลการทดสอบในหัวข้อนี้สามารถสรุปได้ว่า MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมนิตินั้น สามารถปรับตัวเองให้ใช้พลังงานได้เหมาะสมกับอัตราการส่งข้อมูลและระยะเวลาหน่วงที่เปลี่ยนแปลงไป โดยพลังงานที่ใช้ขึ้นจะแปรผกผันกับระยะเวลาหน่วง กล่าวคือ ยิ่งต้องการลดระยะเวลาหน่วงให้น้อยลง ก็ต้องแลกมาด้วยการใช้พลังงานที่สูงขึ้น และ MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมนิตินั้น สามารถปรับตัวเองให้ใช้พลังงานมากขึ้นเพื่อลดระยะเวลาหน่วงลงได้ ในขณะที่ MAC โพรโทคอลแบบ Duty-cycle คงที่ จะถูกจำกัดระยะเวลาหน่วงไว้แล้ว ไม่สามารถปรับเปลี่ยนได้อีก และในการจำกัด Latency ที่ 4 วินาที MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมนิตินั้น สามารถประหยัดพลังงานได้สูงสุดได้มากกว่า MAC โพรโทคอลแบบ Duty-cycle คงที่ประมาณ 100 เท่า

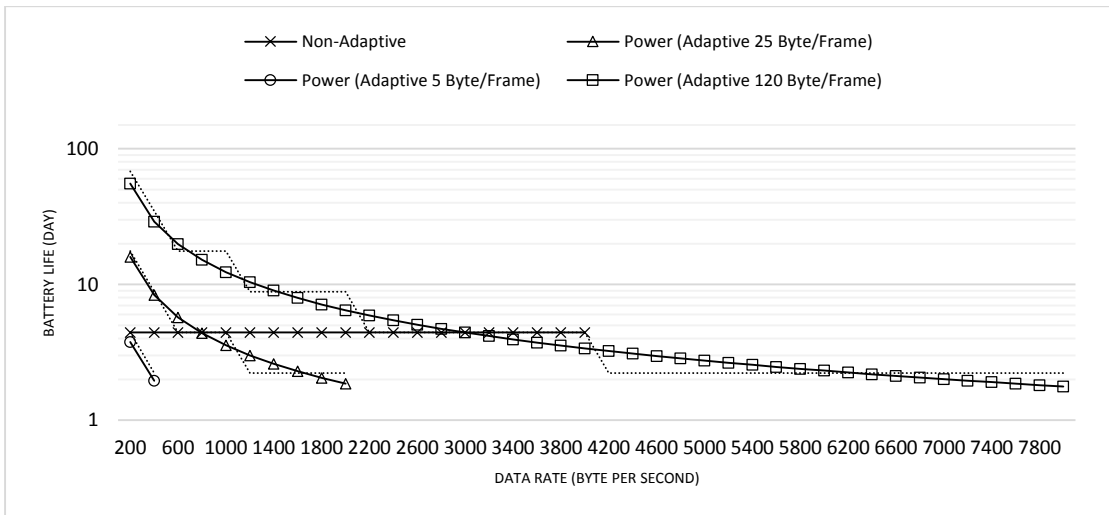
#### 4.5 กำหนดระยะเวลาการทำงานบนแบตเตอรี่

อุปกรณ์เก็บข้อมูลแบบ ไร้สายที่ใช้เก็บข้อมูลทางการแพทย์จากตัวผู้ที่ต้องการเฝ้าระวังนั้น จำเป็นต้องได้รับพลังงานจากแบตเตอรี่ ดังนั้นการทราบถึงระยะเวลาที่อุปกรณ์จะทำงานได้จนกว่าแบตเตอรี่จะหมด จึงเป็นสิ่งที่จำเป็นต้องทราบ โดยการคำนวณระยะเวลา จะอ้างอิงจากแบตเตอรี่ชนิด Nickel-Metal Hydride จำนวน 2 ก้อน ที่มีแรงดันก้อนละ 1.2 โวลต์ ต่ออนุกรมกัน

ซึ่งจะได้แรงดัน 2.4 โวลต์ และกำหนดให้แบตเตอรี่มีความจุ 2000 mAh แต่มีประสิทธิภาพที่จะจ่ายไฟได้ 60 % ดังนั้นความจุที่จะนำมาคำนวณคือ 1600 mAh และในการคำนวณนี้จะกำหนดให้อัตรา Self-discharge ของแบตเตอรี่มีค่าต่ำมากจนไม่มีนัยสำคัญ



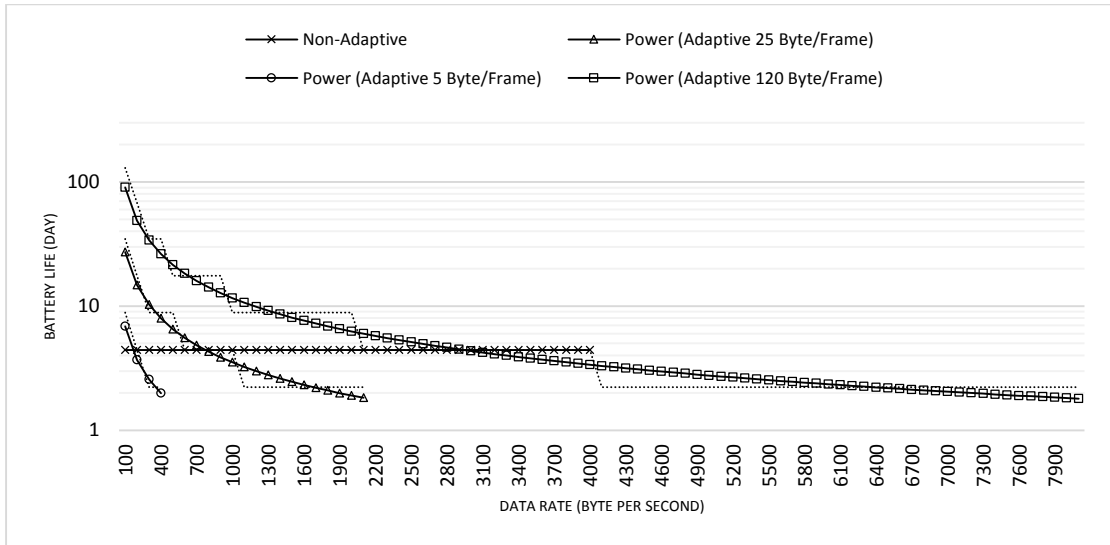
รูปที่ 4.22 อายุแบตเตอรี่เปรียบเทียบกับอัตราการส่งข้อมูล (1-100 Bps)



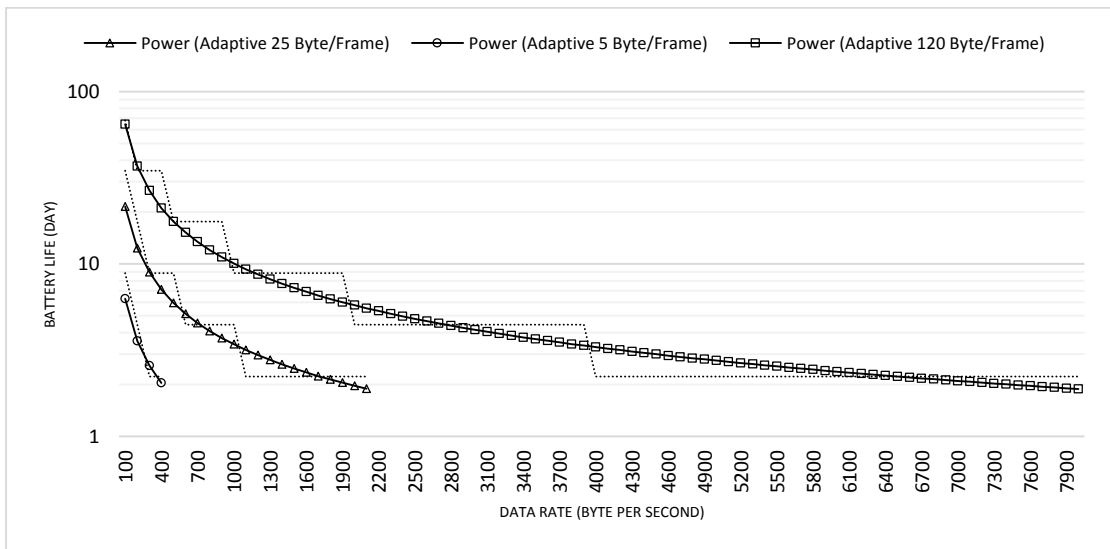
รูปที่ 4.23 อายุแบตเตอรี่เปรียบเทียบกับอัตราการส่งข้อมูล (101-8000 Bps)

รูปที่ 4.22 และ รูปที่ 4.23 แสดงถึงอายุการใช้งานแบตเตอรี่เมื่อมีการส่งข้อมูลที่อัตราการส่งข้อมูลที่แตกต่างกันออกไปโดยไม่มีการควบคุมระยะเวลาหน่วง จะเห็นได้ว่าหากส่งข้อมูลด้วยขนาดข้อมูลต่อเฟรมที่สูง และใช้อัตราการส่งข้อมูลที่ต่ำ จะสามารถยืดอายุแบตเตอรี่ได้

กว่า 1000 วัน หรือประมาณ 4 ปี และหากส่งข้อมูลด้วยอัตราส่งข้อมูลสูงสุด จะลดอายุของแบตเตอรี่ลงเหลือเพียง 2 วัน



รูปที่ 4.24 อายุแบตเตอรี่เทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 4 วินาที (1-8000 Bps)



รูปที่ 4.25 อายุแบตเตอรี่เทียบกับอัตราการส่งข้อมูลแบบควบคุม Latency 1 วินาที (1-8000 Bps)

หากลดระยะเวลาระยะเวลาหนึ่งให้เหลือ 4 วินาที และ 1 วินาที จะส่งผลให้อายุแบตเตอรี่สูงสุดลดลงเหลือ 150 และ 70 วัน ตามลำดับ และหากส่งข้อมูลด้วยอัตราส่งข้อมูลสูงสุด อายุของแบตเตอรี่ก็จะลดลงเหลือ 2 วัน ซึ่งเท่ากับการควบคุม Latency ค่าอื่นๆ ดังรูปที่ 4.24 และรูปที่ 4.25



จากการคำนวณอายุการใช้งานของแบตเตอรี่ในสภาวะแวดล้อมที่แตกต่างกันออกไปสามารถสรุปได้ว่า ปัจจัยที่ส่งผลต่ออายุการใช้งานของแบตเตอรี่นั้นมีทั้งเรื่องของ อัตราการส่งข้อมูล จำนวนข้อมูลต่อเฟรมและ การกำหนดระยะเวลาหน่วง โดยการเปลี่ยนแปลงค่า บางอย่างเพียงเล็กน้อย ก็อาจจะส่งผลกระทบต่ออายุของแบตเตอรี่ลดลงได้อย่างมาก เช่น ในรูปที่ 4.22 หากเปลี่ยนอัตราการส่งข้อมูลที่ขนาดข้อมูล 120 ไบต์ต่อเฟรม จาก 1 ไบต์ต่อวินาที เป็น 100 ไบต์ต่อ วินาที จะทำให้อายุของแบตเตอรี่ลดลงจาก 4 ปี เหลือแค่ 100 วัน ดังนั้นหากต้องการยืดอายุของแบตเตอรี่ให้นานมากที่สุด ก็จะต้องส่งข้อมูล โดยใช้ MAC แบบปรับ Duty-cycle อัตโนมัติที่ขนาดข้อมูล 120 ไบต์ต่อเฟรม และไม่จำกัดระยะเวลาหน่วงของข้อมูล ซึ่งสามารถยืดอายุของแบตเตอรี่ได้มากกว่าแบบ Duty-cycle คงที่มากที่สุดมากกว่า 1000 วัน

#### 4.6 ทดสอบการทำงานของเครือข่ายเมื่อมีเซนเซอร์นับก้าวและตรวจจับการหลัด

ในหัวข้อก่อนหน้านี้ เราได้ทดสอบอายุแบตเตอรี่ด้วยการส่งข้อมูลแบบสุ่ม ด้วยอัตราการส่งข้อมูลที่แตกต่างกันออกไปเพื่อให้เห็นถึงพลังงานที่ใช้ในการส่งข้อมูลจริง โดยตัดปัจจัยอื่นๆที่ส่งผลต่อการใช้พลังงานออกไป เช่น ไม่มีการใช้พลังงานจากเซนเซอร์ และ ในช่วงที่อุปกรณ์หลับ ไมโครคอนโทรลเลอร์ก็ไม่มีประมวลผลข้อมูลใดๆ เป็นต้น แต่ในการใช้งานจริงจะต้องมีการใช้พลังงานเพิ่มขึ้นจากตัวเซนเซอร์ และ จากไมโครคอนโทรลเลอร์ที่ใช้ในการประมวลผลข้อมูลจากเซนเซอร์ด้วย โดยพลังงานเหล่านี้ จะส่งผลกระทบต่ออายุของเครือข่ายที่ทำงานบนแบตเตอรี่ได้

การเลือกใช้เซนเซอร์ที่สามารถประมวลผลข้อมูลด้วยตัวเองได้ และใช้พลังงานต่ำ เป็นปัจจัยสำคัญ ที่จะช่วยยืดอายุของเครือข่าย โดยในการทดลองนี้ จะเปรียบเทียบการใช้พลังงาน โดยการนำเซนเซอร์วัดความเร่ง 3 แกน 2 ชนิดคือ เซนเซอร์ที่ไม่สามารถประมวลผลข้อมูลได้ด้วยตัวเอง กับเซนเซอร์วัดความเร่ง 3 แกน ที่สามารถประมวลผลการตกแบบอิสระ และตรวจจับการเคลื่อนไหวได้ พบว่าเซนเซอร์ทั้งสองชนิด ใช้พลังงานขณะอ่านข้อมูลไม่แตกต่างกัน คือประมาณ 140  $\mu\text{A}$  แต่พลังงานที่ไมโครคอนโทรลเลอร์ ต้องใช้ในการประมวลผลแตกต่างกันมาก โดยชนิดที่ไม่สามารถประมวลผลข้อมูลได้ ไมโครคอนโทรลเลอร์จะต้องคอยอ่านข้อมูลและประมวลผลข้อมูลตลอดเวลา จึงทำให้ใช้พลังงานในการประมวลผลเฉลี่ย 2400  $\mu\text{A}$  และเซนเซอร์ที่สามารถประมวลผลข้อมูลได้ สามารถลดการใช้พลังงานของไมโครคอนโทรลเลอร์ลงไปได้ เนื่องจากระหว่างที่เซนเซอร์ประมวลผลข้อมูล ไมโครคอนโทรลเลอร์สามารถเข้าสู่โหมดประหยัดพลังงาน และเมื่อเซนเซอร์ประมวลผลข้อมูลเสร็จ ก็จะส่งสัญญาณไปยังไมโครคอนโทรลเลอร์ ให้นำข้อมูลไปประมวลผลต่อไป ซึ่งด้วยวิธีนี้ จะทำให้ไมโครคอนโทรลเลอร์ใช้พลังงานเฉลี่ยเพียง 450  $\mu\text{A}$  ซึ่งน้อยกว่าแบบแรกกว่า 5 เท่า

## บทที่ 5

### สรุปผลการวิจัยและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงการสรุปผลและข้อเสนอแนะที่ได้จากการดำเนินการทำวิทยานิพนธ์ ตลอดจนปัญหาและอุปสรรคที่เกิดขึ้นขณะทำวิทยานิพนธ์ และท้ายที่สุดจะกล่าวถึงข้อเสนอแนะแก่ผู้สนใจที่จะนำวิทยานิพนธ์ชิ้นนี้เพื่อพัฒนาต่อไป

#### 5.1 สรุปผล

งานวิจัยนี้ได้นำเสนอกระบวนการปรับปรุง MAC โพรโทคอลบนมาตรฐาน IEEE 802.15.4 ให้ทำงานได้เหมาะสมกับเซนเซอร์ทางการแพทย์ที่มีความหลากหลาย โดยการออกแบบกระบวนการปรับ Duty-cycle อัตโนมัติ และวิธีการคำนวณหาค่า Duty-cycle ที่เหมาะสมต่ออัตราการส่งข้อมูล ซึ่งการออกแบบกระบวนการปรับ Duty-cycle แบบอัตโนมัตินี้ทำให้ Coordinator สามารถปรับเปลี่ยนลักษณะของเครือข่ายได้ขณะที่กำลังทำงานอยู่ ทำให้การใช้พลังงานของ Coordinator สามารถปรับเปลี่ยนตามชนิดของเซนเซอร์ที่ต้องการส่งข้อมูลได้ ส่งผลให้ลดการใช้พลังงานที่ไม่จำเป็นลง และยืดอายุของแบตเตอรี่

เมื่อนำ MAC โพรโทคอลที่ได้ปรับปรุง มาเปรียบเทียบกับ MAC โพรโทคอลแบบ Duty-cycle คงที่เพื่อวัดประสิทธิภาพพบว่า MAC โพรโทคอลที่ได้ปรับปรุงในงานวิจัยนี้สามารถลดการใช้พลังงานลงได้ตามอัตราการส่งข้อมูล โดยเฉพาะในเซนเซอร์ที่ต้องการอัตราการส่งข้อมูลต่ำ MAC โพรโทคอลที่ได้รับการปรับปรุง สามารถยืดอายุแบตเตอรี่ได้มากกว่า MAC โพรโทคอลแบบ Duty-cycle คงที่กว่า 100 เท่า และนอกจากจะช่วยลดในเรื่องของการใช้พลังงานแล้ว ยังสามารถเพิ่มอัตราการส่งข้อมูลสูงสุด เมื่อเทียบกับ MAC โพรโทคอลแบบ Duty-cycle คงที่ โดยการเพิ่มการใช้พลังงานให้มากขึ้นเพื่อแลกกับอัตราการส่งข้อมูลที่สูงขึ้นได้อีกด้วย และเมื่อพิจารณาถึงรูปแบบของ MAC โพรโทคอลที่สามารถใช้พลังงานได้อย่างมีประสิทธิภาพมากที่สุดก็พบว่า MAC โพรโทคอลแบบปรับ Duty-cycle อัตโนมัติ ที่การส่งข้อมูล 120 ไบต์ต่อเฟรมสามารถใช้พลังงานได้อย่างมีประสิทธิภาพมากที่สุด โดยสามารถประหยัดได้มากกว่า MAC โพรโทคอลแบบ Duty-cycle คงที่ สูงสุดกว่า 150 %

## 5.2 ปัญหาและอุปสรรค

ปัญหาและอุปสรรคของงานวิจัยในครั้งนี้คือ ไม่มีเครื่องมือที่ใช้ในการเก็บและวิเคราะห์ข้อมูลทางอากาศ (Spectrum analyzer) จึงต้องใช้ข้อมูลที่ได้มาจากการวัดค่าพลังงานเพียงอย่างเดียวในการคำนวณหาระยะเวลาการส่งข้อมูล และปัจจัยที่จะส่งผลกระทบต่อระยะเวลาที่ซึ่งทำได้ค่อนข้างยาก เนื่องจากข้อมูลพลังงานที่ได้ออกมานั้น ไม่ได้เป็นพลังงานในส่วนของการรับ-ส่งวิทยุเพียงอย่างเดียว แต่มีปัจจัยจากอุปกรณ์อื่นๆบนตัวอุปกรณ์เข้ามาเกี่ยวข้องด้วย ทำให้ตอนวิเคราะห์ข้อมูลจะต้องลดการใช้งานของอุปกรณ์ที่ไม่เกี่ยวข้องให้มากที่สุด เพื่อที่จะได้ข้อมูลที่ถูกต้องเพื่อนำการวิเคราะห์

## 5.3 ข้อเสนอแนะ

ในงานวิจัยชิ้นนี้ได้ทดสอบโพรโทคอลบนอุปกรณ์ Unode ซึ่งเป็นอุปกรณ์ที่พัฒนาขึ้นมาโดยห้องวิจัยเท่านั้น ยังไม่ได้ทำการทดสอบกับอุปกรณ์ชนิดอื่นๆ ดังนั้นผลการทดสอบด้วยอุปกรณ์ชนิดอื่นๆ อาจจะได้ผลที่แตกต่างจากผลในการวิจัยครั้งนี้ได้ ขึ้นอยู่กับชนิดของอุปกรณ์ที่จะนำมาใช้ และการคำนวณอายุของแบตเตอรี่ในงานวิจัยนี้ ได้กำหนดสภาพแวดล้อมและลักษณะของแบตเตอรี่ไว้ ดังนั้นหากนำไปใช้งานในสภาพแวดล้อมที่แตกต่างออกไป หรือ ใช้แบตเตอรี่ที่มีคุณสมบัติแตกต่างกันจากที่ระบุไว้ จะส่งผลให้อายุของแบตเตอรี่เปลี่ยนแปลงไปด้วย

### บรรณานุกรม

- [1] IEEE Computer Society. LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for information technology telecommunications and information exchange between systems--local and metropolitan area networks--specific requirements. Part 15.4, Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. New York, NY: Institute of Electrical and Electronics Engineers, 2006.
- [2] IEEE Computer Society. LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE standard for information technology telecommunications and information exchange between systems: local and metropolitan area networks--specific requirements. Part 11, Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*. New York: Institute of Electrical and Electronics Engineers, 2012.
- [3] "Tmote-sky-datasheet-101.doc - tmote-sky-datasheet.pdf." [Online]. Available: <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>. [Accessed: 16-Apr-2013].
- [4] "Digi XBee® Wireless RF Modules - Digi International." [Online]. Available: <http://www.digi.com/xbee/>. [Accessed: 29-Apr-2013].
- [5] "Epic: An Open Mote Platform for Application-Driven Design." [Online]. Available: <http://www.cs.berkeley.edu/~prabal/projects/epic/>. [Accessed: 29-Apr-2013].
- [6] R. Yadav, S. Varma, and N. Malaviya, "A survey of MAC protocols for wireless sensor networks," *UbiCC journal*, vol. 4, no. 3, pp. 827–833, 2009.
- [7] "Media access control," *Wikipedia, the free encyclopedia*. 28-Apr-2013.
- [8] "Duty cycle," *Wikipedia, the free encyclopedia*. 25-Apr-2013.
- [9] "Time division multiple access," *Wikipedia, the free encyclopedia*. 19-Apr-2013.
- [10] "Carrier sense multiple access," *Wikipedia, the free encyclopedia*. 20-Mar-2013.
- [11] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115–121, 2006.
- [12] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.
- [13] "IEEE 802.11 RTS/CTS," *Wikipedia, the free encyclopedia*. 27-Apr-2013.
- [14] "Broadcasting (networking)," *Wikipedia, the free encyclopedia*. 29-Mar-2013.
- [15] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks," in *Algorithmic Aspects of Wireless Sensor Networks*, S. E. Nikolettseas and J. D. P. Rolim, Eds. Springer Berlin Heidelberg, 2004, pp. 18–31.
- [16] A. El-Hoiydi, "Spatial TDMA and CSMA with preamble sampling for low power ad hoc wireless sensor networks," in *Seventh International Symposium*

- on Computers and Communications, 2002. Proceedings. ISCC 2002, 2002, pp. 685–692.*
- [17] G. Lu, B. Krishnamachari, and C. S. Raghavendra, “An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks,” in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International, 2004*, p. 224.
  - [18] N. Abramson, “THE ALOHA SYSTEM: another alternative for computer communications,” in *Proceedings of the November 17-19, 1970, fall joint computer conference*, New York, NY, USA, 1970, pp. 281–285.
  - [19] “ALOHAnet,” *Wikipedia, the free encyclopedia*. 02-Apr-2013.
  - [20] T. Van Dam and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks,” in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 171–180.
  - [21] “ZigBee Alliance.” [Online]. Available: <http://www.zigbee.org/>. [Accessed: 04-May-2013].
  - [22] “ISA100.11a,” *Wikipedia, the free encyclopedia*. 19-Apr-2013.
  - [23] “WirelessHART,” *Wikipedia, the free encyclopedia*. 19-Apr-2013.
  - [24] “MiWi,” *Wikipedia, the free encyclopedia*. 06-Apr-2013.
  - [25] “Proprietary RF - 2.4 GHz - CC2420 - TI.com.” [Online]. Available: <http://www.ti.com/product/cc2420>. [Accessed: 21-Mar-2013].
  - [26] “TinyOS,” *Wikipedia, the free encyclopedia*. 12-Apr-2013.
  - [27] “Wireless sensor network,” *Wikipedia, the free encyclopedia*. 28-Apr-2013.
  - [28] “Star network,” *Wikipedia, the free encyclopedia*. 04-May-2013.
  - [29] “Mesh networking,” *Wikipedia, the free encyclopedia*. 26-Apr-2013.
  - [30] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*. John Wiley & Sons, 2010.
  - [31] K. Sohrawy, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*. John Wiley & Sons, 2007.
  - [32] B. Lo and G.-Z. Yang, “BODY SENSOR NETWORKS - RESEARCH CHALLENGES AND OPPORTUNITIES,” in *2007 IET Seminar on Antennas and Propagation for Body-Centric Wireless Communications*, 2007, pp. 26–32.
  - [33] “Electroencephalography,” *Wikipedia, the free encyclopedia*. 19-Apr-2013.
  - [34] “Electrocardiography,” *Wikipedia, the free encyclopedia*. 25-Apr-2013.
  - [35] “Blood pressure,” *Wikipedia, the free encyclopedia*. 25-Apr-2013.
  - [36] “Pulse oximetry,” *Wikipedia, the free encyclopedia*. 23-Apr-2013.
  - [37] “Cardiac output,” *Wikipedia, the free encyclopedia*. 19-Apr-2013.
  - [38] “Human body temperature,” *Wikipedia, the free encyclopedia*. 24-Apr-2013.
  - [39] J. Misic and V. B. Misic, “Bridging between iee 802.15.4 and IEEE 802.11b networks for multiparameter healthcare sensing,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 435–449, May 2009.
  - [40] R. Matthews, N. McDonald, H. Anumula, J. Woodward, P. Turner, M. Steindorf, K. Chang, and J. Pendleton, “Novel hybrid bioelectrodes for ambulatory zero-prep EEG measurements using multi-channel wireless EEG system,” *Foundations of Augmented Cognition*, pp. 137–146, 2007.
  - [41] H. Chen and J. Lee, “The implementation of a wireless electroencephalogram information system using WLAN,” *International Journal of Computer Science and Network Security*, vol. 8, no. 2, pp. 166–172, 2008.

- [42] “OSI model,” *Wikipedia, the free encyclopedia*. 07-May-2013.
- [43] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002, vol. 3, pp. 1567–1576.
- [44] C. Karlof, N. Sastry, and D. Wagner, “TinySec: a link layer security architecture for wireless sensor networks,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, New York, NY, USA, 2004, pp. 162–175.
- [45] M. Ali, T. Suleman, and Z. A. Uzmi, “MMAC: A mobility-adaptive, collision-free mac protocol for wireless sensor networks,” in *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*, 2005, pp. 401–407.
- [47] P. Kumar, M. Gunes, Q. Mushtaq, and B. Blywis, “A real-time and energy-efficient MAC protocol for wireless sensor networks,” in *IFIP International Conference on Wireless and Optical Communications Networks, 2009. WOCN '09*, 2009, pp. 1–5.
- [48] “Frequency-division multiple access,” *Wikipedia, the free encyclopedia*. 18-Apr-2013.
- [49] “Code division multiple access,” *Wikipedia, the free encyclopedia*. 19-Apr-2013.
- [50] “Java (programming language),” *Wikipedia, the free encyclopedia*. 07-May-2013.
- [51] “Shell script,” *Wikipedia, the free encyclopedia*. 25-Apr-2013.
- [52] “Asynchronous I/O,” *Wikipedia, the free encyclopedia*. 21-Apr-2013.
- [53] “Call stack,” *Wikipedia, the free encyclopedia*. 06-May-2013.
- [54] “Input/output,” *Wikipedia, the free encyclopedia*. 07-May-2013.
- [55] “Callback (computer programming),” *Wikipedia, the free encyclopedia*. 19-Apr-2013.
- [56] “Preemption (computing),” *Wikipedia, the free encyclopedia*. 01-May-2013.
- [57] “FIFO,” *Wikipedia, the free encyclopedia*. 03-May-2013.
- [58] “DARPA,” *Wikipedia, the free encyclopedia*. 01-May-2013.
- [59] S. F. PhD, *ZigBee Wireless Networks and Transceivers*. Newnes, 2008.
- [60] “MSP430™16-bit Ultra-Low Power MCUs - 1 Series - MSP430F1611 - TI.com.” [Online]. Available: <http://www.ti.com/product/msp430f1611>. [Accessed: 21-Mar-2013].

**ภาคผนวก**



ภาคผนวก ก  
ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์

ICESIT 2011: International Conference on Embedded Systems and Intelligent Technology  
February 9-11 2011, at Phuket Thailand



# ICESIT 2011

*International Conference  
on Embedded Systems  
and Intelligent Technology*

# Performance Evaluation of IEEE 802.15.4: Experiment on Compatible Telos Platform

J. Raksachum, W. Suntiamorntut  
Collaborative Research Unit on Wireless Sensor Network,  
Department of Computer Engineering, Faculty of engineering,  
Prince of Songkla University, Hatyai, Songkhla 90112 Thailand  
Email: baby\_step@hotmail.com, wannarat@coe.psu.ac.th

**Abstract**— IEEE 802.15.4 is a standard specifically designed for low-rate wireless personal area networks (LR-WPAN) with a focus on enabling the wireless sensor networks. It attempts to provide a low data rate, low power, and low cost wireless networking on the device-level communication. In this paper, we have established a realistic environment for the preliminary performance evaluation of the IEEE 802.15.4 wireless networks. Several sets of practical experiments are conducted to study its various features, including the effects of data payload size and the data throughput in beacon-enabled mode.

**Keyword:** *Wireless Sensor Network, IEEE 802.15.4, TinyOs, Performance Evaluation*

## I. INTRODUCTION

WIRELESS Sensor Networks (WSNs) have emerged as one of the dominant technologies such as military defense and healthcare applications. These WSNs can be deployed for different purposes such as target tracking, intrusion detection, wildlife habitat monitoring, climate control and disaster warning management [1]. A small device called *node* in WSN consists of sensors, embedded processor, moderate amount of memory and transmitter/receiver circuitry. These sensor nodes are normally battery powered and they coordinate among themselves to perform a common task.

Wireless Sensor Networks have to concern about resource constraints and energy conservation. The sensor node's radio in the WSNs consumes a significant amount of energy. Substantial research has been done on the design of low power electronic devices in order to reduce energy consumption of these sensor nodes. Because of hardware limitations further energy efficiency can be achieved through the design of energy efficient communication protocols.

Medium access control (MAC) is an important technique to ensure the successful operation of the network. One of the main functions in the MAC protocol is to avoid collisions from interfering nodes. Even MAC protocol in IEEE 802.11 is in the idle listening state most of the time, the energy is still dissipated quite a lot. These bring us to explore the behavior of IEEE802.15.4 in wireless sensor network using Telos platform. The experimental results of the performance

evaluation will be considered to develop power efficient MAC protocol in the future. The goal of designing MAC protocol is to prolong the life time of the network. In this work we also carried the study of the energy efficient MAC protocols for the wireless sensor network.

The rest of the paper is organized as follows. Section 2 explains an overview of the MAC protocol based on IEEE 802.15.4[3]. Section 3 describes an overview of the TinyOS[4][5] and our in-house node architecture. Section 4 reports the performance evaluation and analysis results. Finally, Section 5 draws a conclusion of this work.

## II. OVERVIEW OF THE MAC PROTOCOL IN IEEE 802.15.4

There are two different devices that can participate in an IEEE 802.15.4 network; a full-function device (FFD) and a reduced-function device (RFD). The FFD can operate in three modes serving as a personal area network (PAN) coordinator, a coordinator, or a normal device. An FFD can communicate with RFDs or other FFDs while an RFD can talk only to an FFD. IEEE 802.15 standard normally performs on both beacon enabled and non-beacon enabled mode. In this section, a brief overview of FFD and RFD are presented including the information on the superframe structure, CSMA-CA algorithm and the data transfer model.

### A. Superframe structure

IEEE 802.15.4 standard allows user to select a superframe structure option to beacon-enabled mode. The format of the superframe is defined by the coordinator. The superframe is bounded by network beacon and is divided into sixteen slots equally. The first slot of each superframe is sent without the use of CSMA. The beacons are used to synchronize the attached devices and to describe the structure of superframes. An example beacon frame is shown in Figure 1. This frame has been captured by program z-monitor [6].

The superframe has both an active portion and an inactive portion. The coordinator may enter a low-power (sleep) mode during the inactive portion. The structure of this superframe is expressed by the values of *macBeaconOrder* and *macSuperframeOrder*. The *macBeaconOrder* is the interval at which the coordinator shall transmit its beacon frames. The



TABLE I POWER PARAMETERS

Parameters	Value (mW)
Idle (Radio off)	0.1
Process full load	7.4
Transmit with Power 0 db	39.5
Transmit with Power -20 db	21.2
Receive	42.9
LED Red On	8.3
LED-Green On	21.6
LED-Blue On	3.3

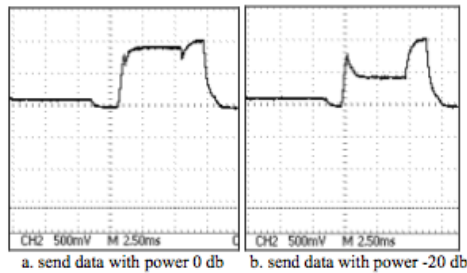


Figure 5. Transmit Power

Applications are component based, code is encapsulated in components that are defined by the interfaces they provide and by those they can consume. Interfaces are bidirectional, they do not only define the commands that have to be implemented by the lower level that implements the interface, but also events that have to be implemented by the higher level that uses the component implementing the interface. The component based model decouples API and implementation and hides the specific properties of an implementation. At compile time, implementations can be substituted by different implementation, either in hardware or in software, only the wiring has to be changed. This makes TinyOS very flexible.

Our in-house node named *Unode* is an ultra low power wireless module for using in sensor network. Unode leverages industry standards like USB and IEEE 802.15.4 to interoperate seamlessly with other devices. By using industry standards, providing flexible interconnection with peripherals, Unode enables a wide range of mesh network applications. Unode is a drop-in replacement for Telos platform by Moteiv company.

Unode module has implemented using ultra low power TI MSP430F1611[7] microcontroller. This 16-bit RISC processor features extremely low current consumption in both active and sleep modes. In order to minimize power consumption, it is in sleep mode most of the time. However, it can wake up as fast as possible to process the task and then return to sleep mode again. Unode uses a USB controller from FTDI to communicate with the host computer and features the Chipcon CC2420 [8] radio, which is an IEEE 802.15.4 compliant radio providing reliable wireless communication for wireless communications. The radio provides fast data rate and robust signal. It is controlled by the microcontroller through the SPI port and can be turn off for low power duty cycled operation.

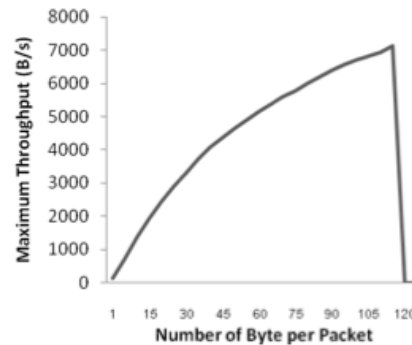


Figure 6. Maximum Throughput VS Number of Byte per Packet

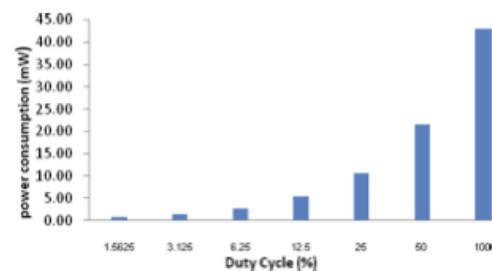


Figure 7. Power consumption VS Duty cycle

The internal Inverted-F microstrip antenna is a pseudo omni directional antenna that may attain 50-meter range indoors and up to 125-meter range outdoors. The key feature of the Unode is listed below:

- 250kbps 2.4GHz IEEE 802.15.4 Chipcon Wireless Transceiver, Interoperability with other IEEE 802.15.4 devices,
- 8MHz Texas Instruments MSP430 microcontroller (10k RAM, 48k Flash),
- Integrated ADC, DAC, Supply Voltage Supervisor, and DMA Controller, Ultra low current consumption, Fast wakeup from sleep (<math><6\mu\text{s}</math>), Programming and data collection via USB, 34-pin expansion support
- TinyOS support : mesh networking and communication implementation

#### IV. PERFORMANCE EVALUATION AND RESULTS ANALYSIS

To evaluate the performance of IEEE 802.15.4 standard on sensor platforms, we use TinyOS version 2.1 and Unode device in our experiments. We started to measure the energy consumption on each Unode operation e.g. idle, processing full load, transmitting data, receiving data and toggle LED on Unode. The results are shown in Table 1. We found that the radio part in sensor node consumes a significant amount of energy. Received mode dissipated energy about 42.9 mW,

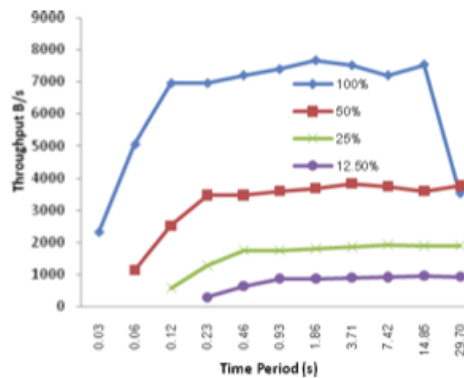


Figure 8. Time Period VS Throughput

Transmitting uses energy vary from 21-35 mW as shown in Figure 5. Our performance evaluation study reveals some of the key throughput, time period, duty cycle and byte per packet.

#### A. Throughput VS Number of Packet

Figure 6 shows the effective data rate with varied data payload ratio between the coordinator and device. The results show that with the increase of payload size, the data rate also increased. This is because the effect of overhead was reduced leading to a raise of data coding efficiency.

#### B. Throughput VS Time Period

In IEEE 802.15.4, the coordinator decides the duty cycle and beacon interval that the PAN operates on. This is done by setting the SO and BO parameter explained earlier. The duty cycle is give by the expression  $2^{SO}/2^{BO}$ . Low duty cycle conserves energy by putting devices to sleep, in figure 7 shown power consumes vs. duty cycle. However, a low duty cycle reduces the bandwidth and increase latency. With the same duty cycle, we can have different combination of SO and BO which means the superframe structure has different active period and inactive period.

In this experiment, the maximum throughput at time period 1.856 s and duty cycle 100% this throughput is 7625.5 B/s as shown in fig 8. Duty cycle 50% can have maximum throughput 3826.105 B/s at time period 3.712 s, Duty cycle 25% can have maximum throughput 3826.105 B/s at time period 7.424 s, Duty cycle 12.5% can have maximum throughput 956.52 B/s at time period 14.848 s

### V. CONCLUSION

We have described the IEEE 802.15.4 standard for low-rate and low-power wireless network using the experiment on real sensor platform. We found that the sensor node's radio in the Unode consumes a significant amount of energy. Moreover, the extremely low duty cycle operation can enable a significant energy saving. However, these saving will trade-off with the cost of significantly higher latency and lower

bandwidth.

#### ACKNOWLEDGEMENT

Authors would like to think TRIDI for a postgraduate scholarship, no.008/2553 .

#### REFERENCES

- [1] N. Malaviya, Shirshu Varma, and Rajesh Yadav, "A SURVEY OF MAC PROTOCOLS FOR WIRELESS SENSOR NETWORKS," UbiCC Journal, vol. 4, no. 3, pp. 827-833, 2009.
- [2] IEEE Standard 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (1999)."
- [3] "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003), 2006. [Online]. Available: 10.1109/IEEESTD.2006.232110. [Accessed: 02-Oct-2010].
- [4] P. Levis, "TinyOS: An Open Operating System for Wireless Sensor Networks (Invited Seminar)," in 7th International Conference on Mobile Data Management (MDM'06), pp. 63-63.
- [5] "TinyOS Home Page." [Online]. Available: <http://www.tinyos.net/>. [Accessed: 05-Oct-2010].
- [6] "Main Page - Z-Monitor." [Online]. Available: [http://www.z-monitor.org/index.php?title=Main\\_Page](http://www.z-monitor.org/index.php?title=Main_Page). [Accessed: 05-Oct-2010].
- [7] "MSP430™16-bit Ultra-Low Power MCUs - 1xx 8MHz Series - MSP430F1611 - TI.com." [Online]. Available: <http://focus.ti.com/docs/prod/folders/print/msp430f1611.html>. [Accessed: 05-Oct-2010].
- [8] "Low Power RF Protocols - ZigBee - CC2420 - TI.com." [Online]. Available: <http://focus.ti.com/docs/prod/folders/print/cc2420.html>. [Accessed: 05-Oct-2010].

## ประวัติผู้เขียน

ชื่อ สกุล	นายจรัสศักดิ์ รักษาชุม	
รหัสประจำตัวนักศึกษา	5310120050	
วุฒิการศึกษา		
	วุฒิ	ชื่อสถาบัน
	วิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์
		ปีที่สำเร็จการศึกษา
		2552

## ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

- ทุนการศึกษา NTC Scholarship สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ (NBTC, TRIDI) ประจำปีการศึกษา 2553

## การตีพิมพ์เผยแพร่ผลงาน

- J. Raksachum, W. Suntiamorntut, “Performance Evaluation of IEEE 802.15.4 Experiment on Compatible Telos Platform,” In Proceedings of 4<sup>th</sup> International Conference on Embedded System and Intelligent Technology 2011, (ICESIT2011), Phuket, Thailand, 9<sup>th</sup> – 11<sup>th</sup> Feb 2010, pages 154-158.