

อัลกอริทึมการไม่ประสานเวลาแบบจัดฟันสำหรับเครือข่ายตัวรับรู้ไร้สาย

นายพงษ์ภักดิ์ เตชะเลิศไพศาล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย  
ปีการศึกษา 2554  
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)  
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

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

AN ORTHODONTIC DESYNCHRONIZATION  
ALGORITHM FOR WIRELESS SENSOR NETWORKS

Mr.Pongpakdi Taechalerpaisarn

A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2011

Copyright of Chulalongkorn University



พงษ์ภักดิ์ เตชะเลิศไพศาล : อัลกอริทึมการไม่ประสานเวลาแบบจัดฟันสำหรับ  
เครือข่ายตัวรับรู้ไร้สาย (AN OROTHODONTIC DESYNCHRONIZATION  
ALGORITHM FOR WIRELESS SENSOR NETWORKS)  
อ.ที่ปรึกษา : ผศ.ดร.เฉลิมเอก อินทนากรวิวัฒน์, 45 หน้า

การไม่ประสานเวลาเป็นปัญหาตรงข้ามกันกับการประสานเวลา เพื่อให้โหนดทำงาน  
ไม่ตรงกัน โดยมีจุดมุ่งหมายเพื่อจัดสรรทรัพยากรต่างๆ เพื่อให้สมาชิกเข้าใช้งานไม่พร้อมกัน  
ตัวอย่างการใช้งาน เช่น ระบบทีดีเอ็มเอ เป็นต้น โดยอัลกอริทึมการไม่ประสานเวลาเดิมที่เสนอ  
ไว้นั้นได้ทำการจัดสรรเวลาโดยปรับตำแหน่งเวลาของตนเองไปยังจุดกึ่งกลางของเพื่อนบ้าน  
เพื่อให้เกิดการทำงานไม่พร้อมกัน แต่อัลกอริทึมการไม่ประสานเวลาที่นำเสนอมาแล้วนั้นยังมี  
ความผิดพลาดสูงอยู่ โดยมีความแปรปรวนของความผิดพลาดอยู่ในระดับสูงเนื่องจากไม่ได้มี  
กลไกใดๆมาจัดการการปรับที่ไม่เหมาะสม

วิทยานิพนธ์นี้ได้นำเสนออัลกอริทึมที่จะช่วยปรับปรุงการไม่ประสานเวลาแบบเดิม  
โดยแนวคิดมาจากกระบวนการจัดฟันของทันตแพทย์ที่ได้ใช้ยางรัดฟันเข้าด้วยกันเพื่อรักษา  
ระยะห่างที่ถูกต้องเอาไว้ ซึ่งวิทยานิพนธ์ฉบับนี้ได้นำแนวคิดแบบเดียวกันนี้ไปใช้เพื่อรักษาระยะ  
ช่วงเวลาระหว่างของโหนดเพื่อนบ้าน เพื่อให้ระบบเข้าสู่สถานะการไม่ประสานเวลาโดยมีความ  
แปรปรวนของความผิดพลาดต่ำลงและทำให้ระบบเข้าสู่สถานะการไม่ประสานเวลาได้เร็วขึ้น

ภาควิชา ..... วิศวกรรมคอมพิวเตอร์ ..... ลายมือชื่อนิสิต .....

สาขาวิชา ..... วิศวกรรมคอมพิวเตอร์ ..... ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา ..... 2554 .....

# # 5270782121 : MAJOR COMPUTER ENGINEERING

KEYWORDS : WIRELESS SENSOR NETWORKS/ DESYNCHRONIZATION/  
ORTHODONTICS/ GROUPING/ NODE SCHEDULING

PONGPAKDI TAECHALERTPAISARN : AN OROTHODONTIC  
DESYNCHRONIZATION ALGORITHM FOR WIRELESS SENSOR  
NETWORKS. ADVISOR : ASST. PROF. CHALERMEK  
INTANAGONWIWAT, Ph.D., 45 pp.

Desynchronization is the inverse problem of synchronization to ensure that each node will not gain control over the resource at the same time. The potential application is TDMA. The original desynchronization algorithm achieve the desynchronization state by adjusting each node time phase to the middle of its neighbor distributing time space and ensuring that each node will not active at the same time. Despite from the simplicity of the algorithm, the system error still fluctuate quite high due to lack of any mechanism to deal with improper adjustment.

This thesis purpose an improvement algorithm inspired by Orthodontics method which the dentist will use power-chain to tie up each teeth to maintain correct gap. This thesis is using the same principle to maintain the correct time phase between each node. Our experiment result indicate the our purposed algorithm help original algorithm converge to desynchronizaiton state faster with lower in error fluctuation.

Department ... Computer Engineering ..... Student's Signature.....  
Field of Study ... Computer Engineering ..... Advisor's Signature .....  
Academic Year ... 2011 .....

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้จะสำเร็จเรียบร้อยสมบูรณ์มิได้ หากปราศจากความช่วยเหลือจากอาจารย์ที่ปรึกษา วิชา ผศ.ดร. เฉลิมเอก อินทนากรวิวัฒน์ ที่คอยแนะนำแนวทางการวิจัย ความรู้และความคิดเห็นต่าง ๆ อันเป็นประโยชน์ต่องานวิจัยมาโดยตลอด อีกทั้งรวมไปถึงความรู้ในด้านอื่นๆ ระเบียบวินัยในการทำงานต่างๆ ที่ช่วยขัดเกลาให้ผู้วิจัยประสบความสำเร็จ ทุกอย่างที่อาจารย์ถ่ายทอดให้ล้วนเป็นสมบัติล้ำค่ายิ่งที่ช่วยผลักดันให้ผู้วิจัยสามารถมาจนถึงจุดนี้ได้และนำไปใช้ประโยชน์ได้อย่างไม่มีวันหมด ขอกราบขอบพระคุณเป็นอย่างสูง กราบขอบพระคุณ พี่ปิง ศุภเสฏฐ์ ชูชัยศรี รุ่นพี่บัณฑิตปริญญาเอก ผู้เปรียบเสมือนพี่เลี้ยง ที่คอยช่วยเหลือผู้วิจัย ในด้านการฝึกทักษะที่จำเป็นสำหรับงานวิจัย และคอยตรวจสอบข้อบกพร่องของงานให้อย่างต่อเนื่องจนประสบความสำเร็จในการศึกษา กราบขอบพระคุณ ผศ.ดร.เกริก ภิรมย์โสภา อ.ดร. กุลธิดา โรจน์วิบูลย์ชัย และ รศ.ดร. อนันต์ ผลเพิ่ม คณะกรรมการสอบวิทยานิพนธ์ที่สละเวลามาให้ข้อเสนอแนะและเติมเต็มข้อบกพร่องของวิทยานิพนธ์ฉบับนี้ให้ผ่านมาด้วยดี

ขอขอบคุณอาจารย์สองศรี รัชนิกรและคณาจารย์ทุกท่านจากโรงเรียนเตรียมอุดมศึกษาที่คอยปลูกฝังระเบียบวินัยและแนวทางในการดำเนินชีวิตที่ช่วยให้ผู้วิจัยประสบความสำเร็จ

ขอขอบคุณบิดา มารดา และน้องชายและน้องสาว รวมไปถึงทุกคนในครอบครัวของผู้วิจัยที่คอยสนับสนุนและให้กำลังใจมาตลอดระยะเวลา 2 ปีที่ผ่านมา ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย สถานศึกษาที่ได้ประสิทธิประสาทวิชาความรู้อันเป็นประโยชน์ต่อผู้วิจัย โดยเฉพาะเหล่าคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ รวมไปถึงเพื่อนๆ พี่ๆ และน้องๆ ร่วมภาควิชาทุกคนที่คอยให้คำปรึกษาและช่วยเหลือยามที่ผู้วิจัยประสบปัญหาใดๆ อยู่เสมอ นอกจากนี้ยังช่วยสร้างบรรยากาศให้ผู้วิจัยรู้สึกไม่เจียบเหงาและสนุกสนานไปกับการเรียน นับเป็นประสบการณ์ชีวิตและความทรงจำที่มีค่ายิ่ง ขอขอบคุณและขอขมาในสิ่งที่ล่วงเกินไว้ ณ โอกาสนี้

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	ง
บทคัดย่อภาษาอังกฤษ .....	จ
กิตติกรรมประกาศ .....	ฉ
สารบัญ .....	ช
สารบัญภาพ .....	ฌ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา .....	1
1.2 วัตถุประสงค์ของการวิจัย .....	2
1.3 ขอบเขตของการวิจัย.....	3
1.4 ขั้นตอนและวิธีดำเนินการวิจัย .....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย .....	4
1.6 ผลงานตีพิมพ์จากวิทยานิพนธ์.....	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....	5
2.1 ทฤษฎีที่เกี่ยวข้อง .....	5
2.1.1 ทฤษฎีการไม่ประสานเวลา (Desynchronization theory).....	5
2.1.2 การจัดฟัน (Orthodontics Braces).....	6
2.2 งานวิจัยที่เกี่ยวข้อง .....	7
2.2.1 Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks.....	7
2.2.2 DWARF: Desynchronization with artificial force field .....	10
2.2.3 Towards Desynchronization of Multihop Topologies .....	10
บทที่ 3 การออกแบบอัลกอริทึมการไม่ประสานเวลาแบบจัดฟัน .....	11
3.1 ขั้นตอนการตรวจสอบความถูกต้องของโหนด .....	12
3.1.1 การคำนวณหาหน้าหนักฝั่งซ้าย .....	12
3.1.2 การคำนวณหาหน้าหนักฝั่งขวา .....	12
3.2 หาค่าการปรับช่วงเวลาที่เหมาะสม .....	13
3.3 การทำงานของอัลกอริทึม .....	14
3.4 รหัสเทียมของอัลกอริทึม .....	15
บทที่ 4 ผลการทดลองและวิเคราะห์ผล .....	18
4.1 เครื่องมือที่ใช้ในการทดลอง .....	18
4.1.1 ระบบปฏิบัติการ TinyOS .....	18

หน้า	
4.1.2 โปรแกรมจำลองทอสมิม (TOSSIM).....	19
4.1.3 ภาษาโปรแกรมเนสซี (nesC Language) .....	19
4.2 วิธีการประเมินประสิทธิภาพของอัลกอริทึมการไม่ประสานเวลา .....	20
4.3 การทำลองทดสอบอัลกอริทึมการไม่ประสานเวลา .....	21
4.4 ผลการทดลองเปรียบเทียบอัลกอริทึมการไม่ประสานเวลา.....	21
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ .....	41
5.1 สรุปผลการวิจัย .....	41
5.2 ข้อจำกัดและข้อเสนอแนะ .....	41
5.3 ปัญหาและอุปสรรค .....	42
รายการอ้างอิง .....	42
ภาคผนวก .....	44
ประวัติผู้เขียนวิทยานิพนธ์ .....	45



## สารบัญภาพ

หน้า	
ภาพที่ 1.1	โครงสร้างระบบเครือข่ายตัวรับรู้แบบไร้สาย ..... 1
ภาพที่ 1.2	ตัวอย่างโหมดที่ใช้พลังงานจากถ่านไฟฉาย ..... 2
ภาพที่ 2.1	โครงสร้างของการประสานเวลา-ไม่ประสานเวลา ..... 5
ภาพที่ 2.2	การจัดฟันโดยอุปกรณ์ของทันตแพทย์ ..... 7
ภาพที่ 2.3	สมการและรูปการปรับตำแหน่งเวลาของ DESYNC..... 8
ภาพที่ 2.4	ภาพแสดงการกระจายตัวของความผิดพลาด ..... 9
ภาพที่ 2.5	ภาพจำลองแรงของ DWARF ..... 10
ภาพที่ 3.1	แนวคิดของอัลกอริทึม..... 11
ภาพที่ 3.2	การหาน้ำหนักของโหนดทั้งสองฝั่ง ..... 12
ภาพที่ 3.3	การปรับค่า $\alpha$ ตามน้ำหนัก..... 13
ภาพที่ 3.5	รหัสเทียมในส่วนของ Initialization..... 15
ภาพที่ 3.6	รหัสเทียมในส่วนของ Weight Determination ..... 16
ภาพที่ 3.7	รหัสเทียมในส่วนของ Phase adjustment ..... 17
ภาพที่ 4.1	สมการค่าความผิดพลาดและรากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง ... 20
ภาพที่ 4.3	กราฟแสดงค่าความผิดพลาดในความหนาแน่นต่ำ ..... 22
ภาพที่ 4.4	กราฟแสดงค่าความผิดพลาดในความหนาแน่นต่ำที่ทำให้เป็นมาตรฐาน ..... 22
ภาพที่ 4.5	กราฟแสดงค่าความผิดพลาดในความหนาแน่นกลาง ..... 23
ภาพที่ 4.6	กราฟแสดงค่าความผิดพลาดในความหนาแน่นกลางที่ทำให้เป็นมาตรฐาน ..... 23
ภาพที่ 4.7	กราฟแสดงค่าความผิดพลาดในความหนาแน่นสูง ..... 24
ภาพที่ 4.8	กราฟแสดงค่าความผิดพลาดในความหนาแน่นสูงที่ทำให้เป็นมาตรฐาน ..... 24
ภาพที่ 4.9	กราฟสรุปค่าความผิดพลาดหลังจากทำการปรับไป 300 รอบ ..... 26
ภาพที่ 4.10	กราฟสรุปค่าความผิดพลาดหลังจากทำการปรับไป 300 รอบ ที่ทำให้เป็น มาตรฐาน ..... 26
ภาพที่ 4.11	กราฟความผิดพลาดเทียบกับคาบเวลาในความหนาแน่นต่ำ ..... 28
ภาพที่ 4.12	กราฟความผิดพลาดเทียบกับคาบเวลาในความหนาแน่นต่ำที่ทำให้เป็น มาตรฐาน ..... 28
ภาพที่ 4.13	กราฟความผิดพลาดเทียบกับคาบเวลาในความหนาปานกลาง ..... 29
ภาพที่ 4.14	กราฟความผิดพลาดเทียบกับคาบเวลาในความหนาแน่นปานกลางที่ทำให้เป็น มาตรฐาน ..... 29
ภาพที่ 4.15	กราฟความผิดพลาดเทียบกับคาบเวลาในความหนาสูง ..... 30
ภาพที่ 4.16	กราฟความผิดพลาดเทียบกับคาบเวลาในความหนาแน่นสูงที่ทำให้เป็น มาตรฐาน ..... 30

ภาพที่ 4.17 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นต่ำ ..... 32

ภาพที่ 4.18 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นต่ำที่ทำให้เป็น  
มาตรฐาน ..... 32

ภาพที่ 4.19 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นกลาง ..... 33

ภาพที่ 4.20 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นกลางที่ทำให้เป็น  
มาตรฐาน ..... 33

ภาพที่ 4.21 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นสูง ..... 34

ภาพที่ 4.22 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นต่ำที่ทำให้เป็น  
มาตรฐาน ..... 34

ภาพที่ 4.23 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นต่ำ ..... 36

ภาพที่ 4.24 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นต่ำที่ทำให้เป็นมาตรฐาน ..... 36

ภาพที่ 4.25 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นกลาง ..... 37

ภาพที่ 4.26 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นกลางที่ทำให้เป็นมาตรฐาน ..... 37

ภาพที่ 4.27 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นสูง ..... 38

ภาพที่ 4.28 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นสูงที่ทำให้เป็นมาตรฐาน ..... 38

ภาพที่ 4.29 กราฟเปรียบเทียบค่าน้ำหนักและความผิดพลาดในความหนาแน่นต่ำ ..... 39

ภาพที่ 4.30 กราฟเปรียบเทียบค่าน้ำหนักและความผิดพลาดในความหนาแน่นกลาง ..... 40

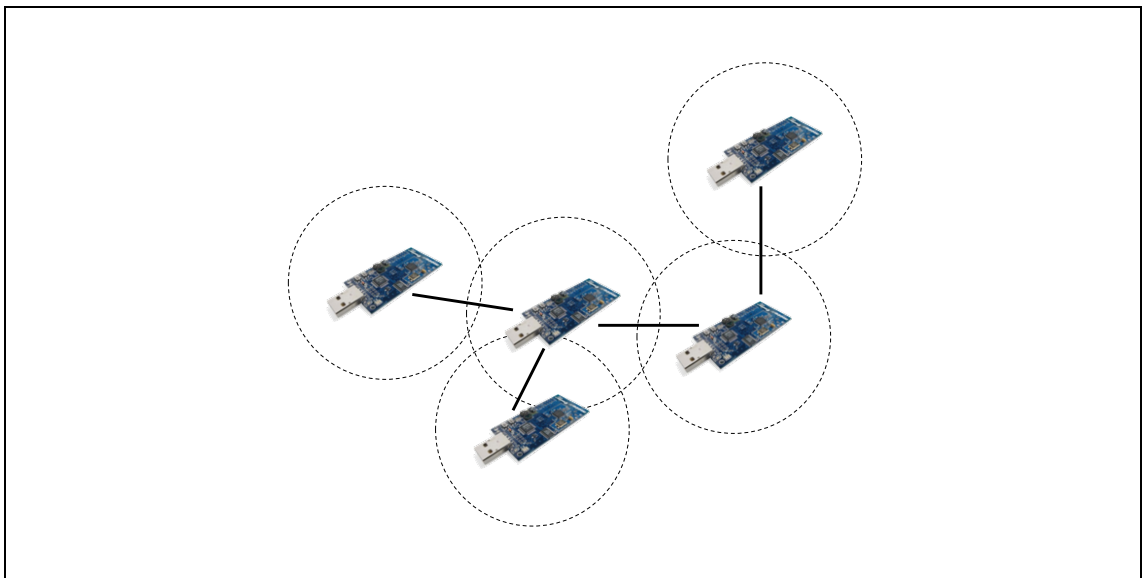
ภาพที่ 4.31 กราฟเปรียบเทียบค่าน้ำหนักและความผิดพลาดในความหนาแน่นสูง ..... 40

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญของปัญหา

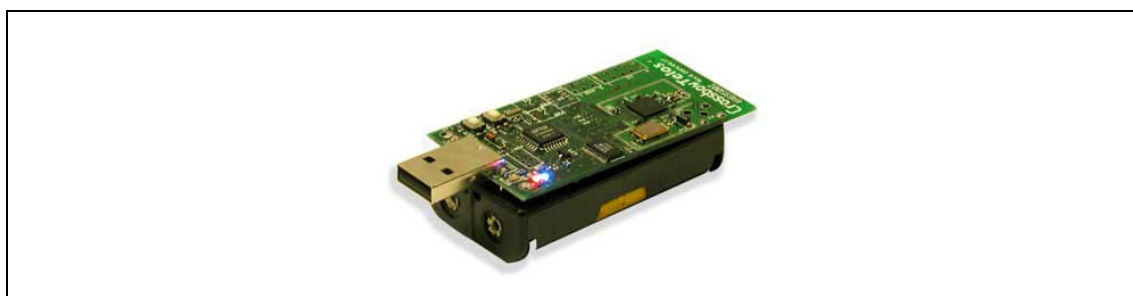
ระบบเครือข่ายเซ็นเซอร์ไร้สาย (Wireless Sensor Networks) เป็นระบบเครือข่ายชนิดหนึ่งประกอบไปด้วยอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็กจำนวนมาก มาประกอบกันเป็นเครือข่ายที่ทำงานร่วมกัน แต่เป็นอิสระต่อกัน ทำงานตรวจจับสภาพแวดล้อมร่วมกันในพื้นที่บริเวณใกล้เคียงกันโดยมีการติดต่อสื่อสารระหว่างกันแบบไร้สายด้วยมาตรฐาน IEEE802.15.4 ซึ่งเป็นมาตรฐานการสื่อสารไร้สายที่เน้นการประหยัดพลังงานดังในรูป 1.1



ภาพที่ 1.1 โครงสร้างระบบเครือข่ายตัวรับรู้แบบไร้สาย

อุปกรณ์ของระบบเครือข่ายเซ็นเซอร์ไร้สายแต่ละตัวจะถูกเรียกว่า โหนด ( Node) ประกอบไปด้วยวงจรมicrocontroller ( Micro Controller) ที่ทำหน้าที่ควบคุมการทำงาน โดยจะรับข้อมูลมาจากเซ็นเซอร์ต่างๆที่ตรวจวัดสภาพแวดล้อม เช่น ความเข้มแสง ,ความดังเสียง และอุณหภูมิ เพื่อนำค่าที่วัดได้มาประมวลผลข้อมูลต่างๆตามที่ผู้ใช้กำหนด โดยข้อมูลจะถูกเก็บลงไว้ในหน่วยความจำและส่งไปประมวลผลที่หน่วยประมวลผล ( Microprocessor) ภายในตัวโหนด และข้อมูลที่ได้นี้จะถูกประมวลผลและส่งผ่านไปสถานีรับข้อมูล ( Base Station) ผ่านทางการติดต่อสื่อสารแบบไร้สายผ่านโหนดที่ใกล้เคียงต่อกันไปเรื่อยๆ จนถึงสถานีรับข้อมูล

โหนดในปัจจุบันจะมีขนาดเล็กและถูกลง โดยจะใช้พลังงานจากแบตเตอรี่หรือ ถ่านไฟฉายขนาดเล็ก ดังในรูป 1.2 ทำให้ผู้ใช้งานต้องพยายามออกแบบให้โหนดพยายาม ประหยัดพลังงานให้มากที่สุด แต่เนื่องจากการสื่อสารแบบไร้สายภายใต้มาตรฐาน IEEE802.15.4 นี้ ไม่ได้มีการจัดการเกี่ยวกับการชนกันของการส่งข้อมูลเช่น CSMA-CA ใน IEEE802.11 อีกทั้งการส่งข้อมูลภายใต้มาตรฐาน IEEE802.15.4 นั้นจะทำการส่งข้อมูลระยะ สั้นหลายๆครั้งต่อกันไป ทำให้มีโอกาสเกิดการชนของข้อมูลค่อนข้างสูง และเมื่อเกิดการชนกัน ของข้อมูลแล้ว โหนดแต่ละตัวจะต้องทำการส่งข้อมูลใหม่ทำให้สิ้นเปลืองทรัพยากรพลังงานเป็น อย่างมาก ดังนั้นหากทำให้โหนดแต่ละตัวทำการส่งข้อมูลในช่วงเวลาที่ไม่ตรงกัน และห่างกัน อย่างสม่ำเสมอแล้ว จะช่วยลดโอกาสของการชนกันของข้อมูลได้ เช่นเดียวกับงานวิจัย Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks



ภาพที่ 1.2 ตัวอย่างโหนดที่ใช้พลังงานจากถ่านไฟฉาย

แต่อัลกอริทึมที่ได้นำเสนอมาแล้วก็ยังประสบกับปัญหาเมื่อนำมาใช้งานจริง เนื่องจากไม่ได้มีกลไกควบคุมการปรับที่เหมาะสมทำให้โหนดกระจายความผิดพลาดในการ ปรับไปยังเพื่อนบ้านทุกตัวได้ วิทยานิพนธ์ฉบับนี้จึงได้เสนอวิธีแก้ไขปัญหาดังกล่าวเพื่อให้ สามารถนำไปใช้งานจริงได้

## 1.2 วัตถุประสงค์ของการวิจัย

นำเสนออัลกอริทึมการไม่ประสานเวลาแบบจัดฟันเพื่อปรับปรุงอัลกอริทึมเดิม โดย มีการทำงานแบบ แบบกระจาย และเฉพาะที่ (Distributed and localized algorithm) สำหรับ ระบบเครือข่ายตัวรับรู้แบบไร้สาย เพื่อให้ระบบเข้าสู่สถานะการไม่ประสานเวลาได้อย่างรวดเร็ว ถูกต้องแม่นยำโดยมีค่าความผิดพลาดอยู่ในระดับต่ำเมื่อเทียบกับอัลกอริทึมเดิม

### 1.3 ขอบเขตของการวิจัย

- 1) อัลกอริทึมนี้ใช้ได้กับการไม่ประสานเวลาที่โหนดมีรัศมีการติดต่อสื่อสารระหว่างโหนด 1 ช่วงเท่านั้น
- 2) การวัดผลจะทำภายใต้โปรแกรมจำลอง TOSSIM [1] เท่านั้น
- 3) เมื่อเทียบระหว่างอัลกอริทึมไม่ประสานเวลาเดิม และอัลกอริทึมไม่ประสานเวลาแบบจัดพื้นที่วัดผลภายใต้ระบบจำลองทอสมิมแบบเดียวกันนี้แล้ว ถือได้ว่าปริมาณโอเวอร์เฮดทางด้านการรับส่งข้อมูลระหว่างโหนดและการคำนวณมีปริมาณที่ไม่ต่างกันมาก จึงจะไม่นำมาพิจารณาเพื่อวัดประสิทธิภาพของอัลกอริทึม (โอเวอร์เฮดด้านรับส่งข้อมูลอาจสูงกว่าปกติได้)
- 4) อัลกอริทึมที่นำเสนอในโครงร่างวิทยานิพนธ์นี้เป็นต้นแบบแนวคิดในการแก้ปัญหา ซึ่งอาจมีการเปลี่ยนแปลงกระบวนการและวิธีการใหม่ได้ในอนาคต หากผลลัพธ์ที่ได้ไม่เป็นตรงตามที่คาดหวังไว้
- 5) อัลกอริทึมนี้จะทดสอบการทำงานบนระบบจำลองที่ไม่มีการสูญเสียข้อมูล (Packet loss) ในรับส่งข้อมูลกันระหว่างโหนด

### 1.4 ขั้นตอนและวิธีดำเนินการวิจัย

- 1) ศึกษาข้อจำกัดและมาตรฐานต่างของเซ็นเซอร์ไร้สาย
- 2) ศึกษาปัญหาการไม่ประสานเวลาของระบบเครือข่ายตัวรับรู้แบบไร้สาย
- 3) ศึกษานิยามและทฤษฎีของการไม่ประสานเวลา
- 4) ศึกษาวิธีการและอัลกอริทึมการไม่ประสานเวลของโหนดในระบบเครือข่าย
- 5) วิเคราะห์และออกแบบอัลกอริทึมของการไม่ประสานเวล โดยปรับปรุงแก้ไขจากแนวทางเดิม
- 6) ทดสอบอัลกอริทึมที่นำเสนอ
- 7) บันทึกและวิเคราะห์ผลการทดลอง
- 8) ปรับปรุงแก้ไขหากพบข้อผิดพลาด

## 9) สรุปผลการทดลองและเรียบเรียงเป็นรายงานวิทยานิพนธ์

### 1.5 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

อัลกอริทึมการไม่ประสานเวลาแบบจัดฟันสำหรับ ระบบเครือข่ายตัวรับรู้แบบไร้สาย ที่นำเสนอสามารถช่วยปรับปรุงอัลกอริทึมการไม่ประสานเวลาเดิมให้สามารถเข้าสู่สภาวะการไม่ประสานเวลาได้อย่างรวดเร็วโดยมีความถูกต้องมากขึ้นกว่าเดิม การทำงานของอัลกอริทึม จะเป็นแบบกระจาย และถูกทดสอบใช้งานบนโปรแกรมจำลองซึ่งจำลองการทำงานของระบบปฏิบัติการที่ใช้งานบนตัวอุปกรณ์โหนดจริง ดังนั้นอัลกอริทึมนี้จึง สามารถนำไปประยุกต์ใช้งานได้ในระบบและสภาพแวดล้อมจริงได้

### 1.6 ผลงานตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของงานวิทยานิพนธ์ได้รับการตีพิมพ์เป็นบทความวิชาการในหัวเรื่อง “An Orthodontic Desynchronization Algorithm for Wireless Sensor Network” โดย พงษ์ภักดิ์ เตชะเลิศไพศาล , ศุภเสฏฐ์ ชูชัยศรี และ เฉลิมเอก อินทนากรวิวัฒน์ ในงานประชุมวิชาการ “The 13 IEEE International Conference on Communication and Technology (ICCT2011)” ซึ่งจัดขึ้น ณ เมืองจีหนาน (Jinan) มณฑลชางตง (Shandong) ประเทศจีน ระหว่างวันที่ 25-28 กันยายน 2554 ดังภาคผนวก ก

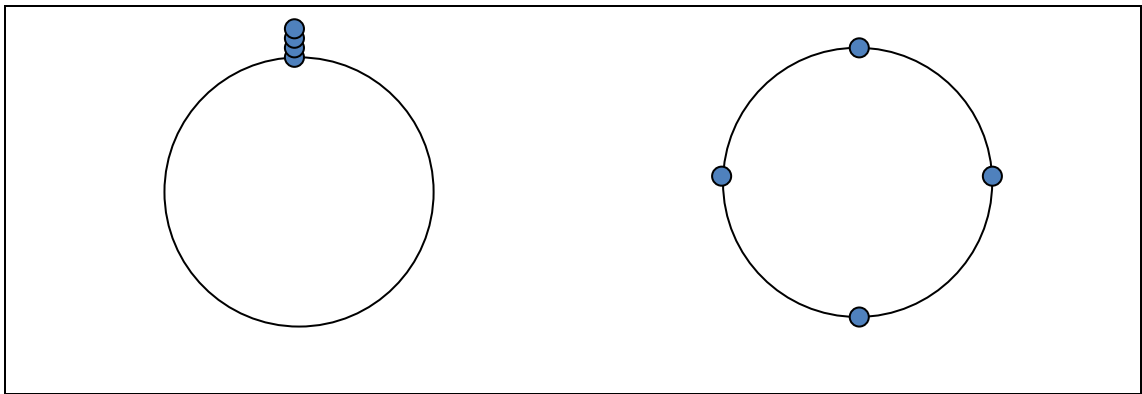
## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

#### 2.1 ทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 ทฤษฎีการไม่ประสานเวลา (Desynchronization theory) [2]

การไม่ประสานเวลา (Desynchronization) เป็นสถานะที่ตรงข้ามกับการประสานเวลา (Synchronization) โดยการประสานเวลานั้นสมาชิกทุกตัวจะต้องตื่นขึ้นมาทำงานพร้อมกันในช่วงเวลาใด ๆ ซึ่งตรงข้ามกับการไม่ประสานเวลาโดยสมาชิกทุกตัวนั้นจะพยายามปรับเวลาดตื่นขึ้นมาให้เกิดความไม่พร้อมเพรียงกัน ให้มากที่สุด ดังในภาพที่ 2.1



ภาพที่ 2.1 โครงสร้างของการประสานเวลา-ไม่ประสานเวลา

จากรูปเราจะแทนช่วงเวลาใด ๆ ด้วยเส้นรอบวงกลมและแทนเวลาที่สมาชิกตื่นขึ้นมาทำงานด้วยจุด หากเป็นการประสานเวลาในรูปซ้าย ทุกโหนดก็จะพยายามตื่นขึ้นมาที่เวลาเดียวกัน ทำงานพร้อม ๆ กัน ซึ่งตรงข้ามกับการไม่ประสานเวลาในรูปทางขวาที่ทุกโหนดจะตื่นขึ้นมาทำงานไม่พร้อมกัน โดยจะพยายามเว้นระยะห่างระหว่างกันให้มากที่สุด เพื่อให้เกิดความไม่พร้อมเพรียงกันมากที่สุด โดยการไม่ประสานเวลานั้นสามารถแบ่งได้เป็น 4 ประเภท ดังนี้

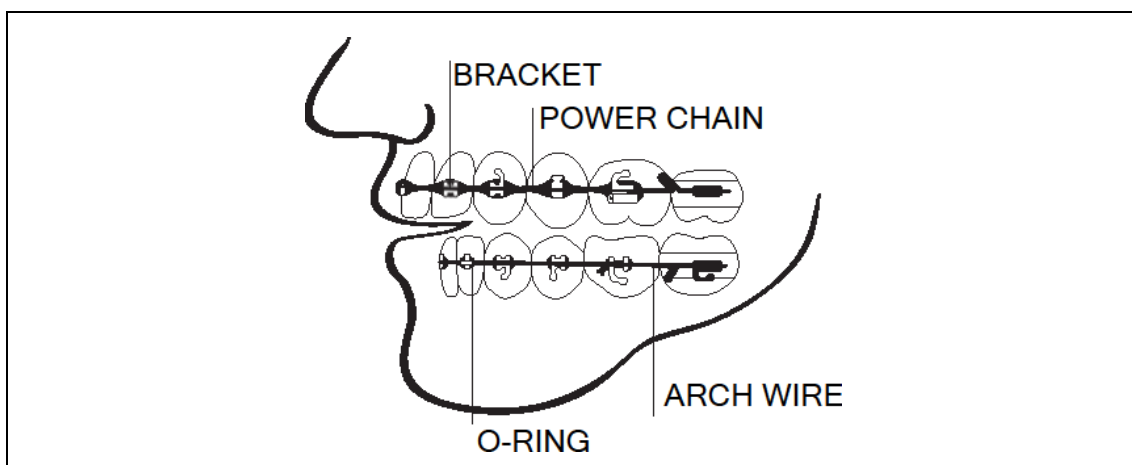
1. **Weak desynchronization** คือการไม่ประสานเวลาที่ช่วงเวลาของ oscillator สามารถเปลี่ยนแปลงได้ โดยเมื่อนำมาใช้ใน TDMA แล้ว โหนดจะไม่สามารถหาช่วงเวลาของตนเองที่แน่นอนได้ เพราะช่วงเวลาที่ใช้แบ่งไม่แน่นอน

2. **Strong desynchronization** คือการไม่ประสานเวลาที่ช่วงเวลาของ oscillator คงที่ตายตัว โดยจะทำให้โหนดแต่ละโหนดใน TDMA ทราบช่วงเวลาของตนเองแน่นอน และจะทำให้ระบบใช้ทรัพยากรได้อย่างเต็มที่
3. **Discrete desynchronization** คือการไม่ประสานเวลาที่กำหนดจำนวนช่องของการแบ่งมาอย่างตั้งแต่เริ่ม โดยจะใช้การแก้ปัญหาเช่นเดียวกับ Graph Coloring problem ซึ่งเป็นปัญหาแบบ NP-Hard ซึ่งต้องทราบจำนวนสีหรือช่องเวลาที่เป็นไปได้ตั้งแต่เริ่มแล้วนำไปแบ่งให้กับโหนดที่ต้องการใช้ช่องเวลาต่อไป ข้อเสียก็คือไม่สามารถใช้ทรัพยากรได้อย่างเต็มที่ ตัวอย่างเช่นหากแบ่งช่องเวลาไว้ 5 ช่องแต่มีผู้ต้องการใช้ 3 โหนด ก็จะทำให้โหนดแต่ละโหนดไม่สามารถจัดสรรช่องได้ลงตัว
4. **Continuous desynchronization** คือการไม่ประสานเวลาที่ไม่ได้กำหนดจำนวนช่องของการแบ่ง โดยโหนดแต่ละโหนดจะใช้ช่วงเวลาที่แบ่งถัดจากโหนดก่อนหน้าเป็นช่วงเวลาที่เท่ากัน จะทำให้จัดสรรทรัพยากรได้ใช้งานอย่างเต็มที่

โดยในอัลกอริทึมที่เสนอในงานวิจัยนี้จะมุ่งเน้นแก้ปัญหาในส่วน Strong and Continuous desynchronization เท่านั้น โดยอาจจะนำไปประยุกต์กับแบบอื่นในภายหลัง

### 2.1.2 การจัดฟัน (Orthodontics Braces)

การจัดฟันเป็นกระบวนการแก้ไขความไม่สมดุลที่เกิดขึ้นจากการเรียงตัวของฟัน และการสบฟัน ที่ผิดปกติให้กลับไปสู่สภาพที่ปกติ ซึ่งโครงสร้างที่ถูกปรับเปลี่ยนรวมทั้งฟันที่ถูกเลื่อนไปจะเกิดจากการใช้แรงที่ละเอียดอ่อนทั้งจากเครื่องมือภายนอกและภายในช่องปากเป็นตัวกระตุ้นให้เกิดการปรับแต่งใหม่ดังในภาพที่ 2.2





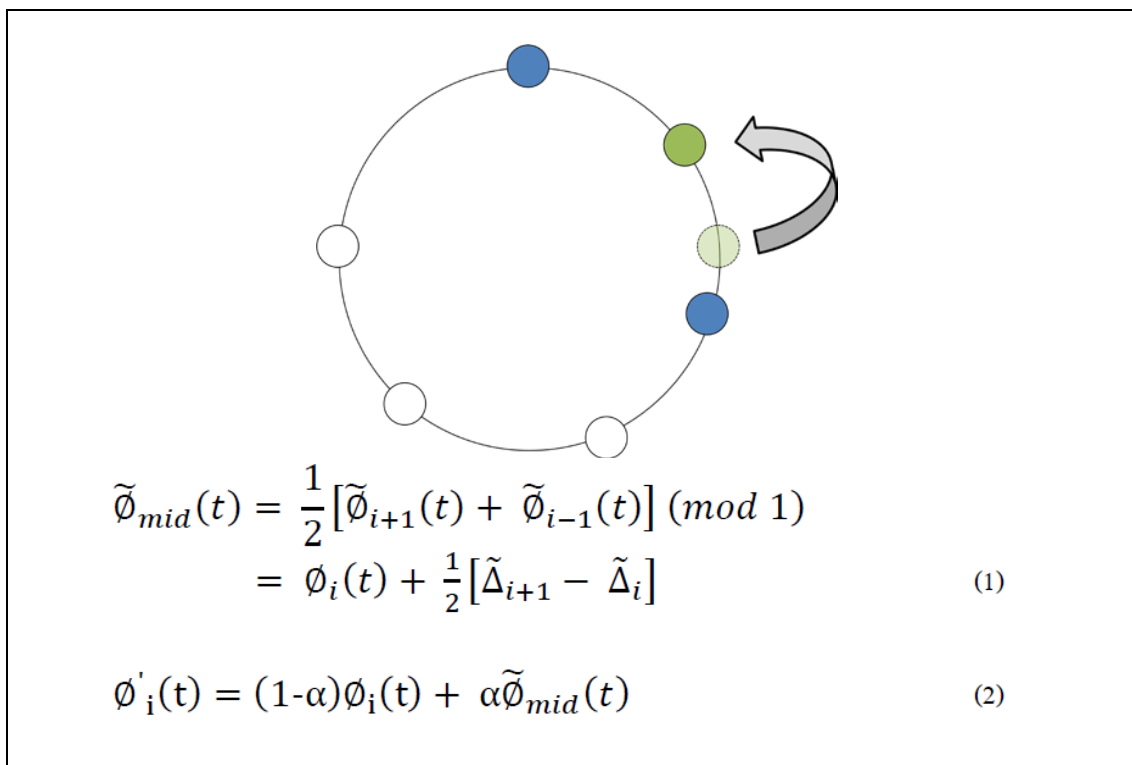
## ภาพที่ 2.2 การจัดฟันโดยอุปกรณ์ของทันตแพทย์

โดยทันตแพทย์จะยึดติดอุปกรณ์ที่ใช้ในการจัดฟันเข้ากับตัวฟันเพื่อจัดระยะระหว่างฟันแต่ละซี่ เมื่อระยะห่างระหว่างฟันได้ระยะพอดีแล้วทันตแพทย์ก็จะมัดรวมฟันเข้าด้วยกันด้วยยางรัด(power-chain) เพื่อให้ฟันกลุ่มนั้นไม่เคลื่อนห่างออกจากกันอีก วิทยานิพนธ์ฉบับนี้ได้นำกระบวนการเดียวกันมาใช้เพื่อรักษาระยะเวลาระหว่างโหนดเพื่อให้ระบบทำการไม่ประสานเวลาได้ถูกต้องแม่นยำยิ่งขึ้น

### 2.2 งานวิจัยที่เกี่ยวข้อง

#### 2.2.1 Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks [3]

เป็นงานวิจัยแรกๆที่กล่าวถึงการไม่ประสานเวลา โดยงานวิจัยนี้จะทำการไม่ประสานเวลาในเครือข่ายเซ็นเซอร์ไร้สายโดยโหนดแต่ละตัวจะทำการ broadcast ความออกไปเมื่อเวลาของตนเองอยู่ที่ตำแหน่ง 12 นาฬิกา และโหนดจะนำข้อมูลตำแหน่งเวลาของเพื่อนบ้านข้างเคียงมาปรับเวลาของตนเองให้มีช่วงเวลาที่ทำงานไปอยู่ในกึ่งกลางระหว่างช่วงเวลาการทำงานของโหนดเพื่อนบ้านสองตัวที่ติดกับตัวมันเอง ดังสมการดังต่อไปนี้

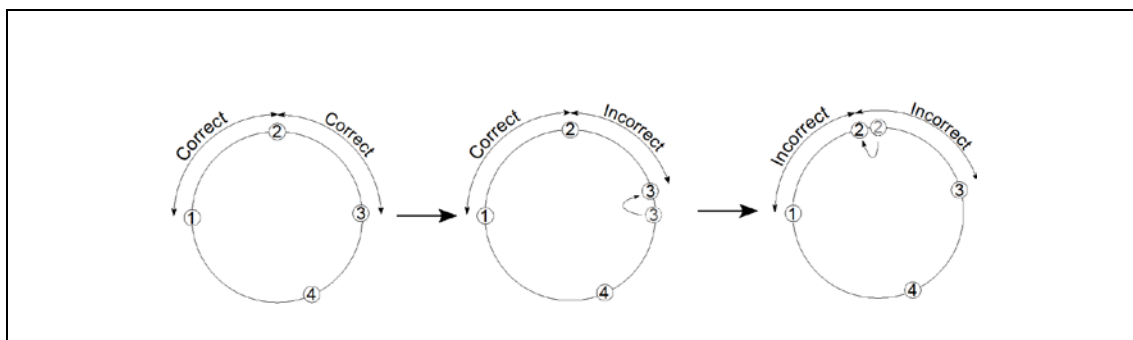


ภาพที่ 2.3 สมการและรูปการปรับตำแหน่งเวลาของ DESYNC

สมการที่ 1 คือสมการที่ใช้หาจุดกึ่งกลางช่วงเวลา โดยมีความถูกต้องเป็นหน่วยเวลาเล็กสุดที่โหนดจะสามารถตรวจจับได้ และสมการที่ 2 คือสมการที่ใช้หาดำแหน่งใหม่ของโหนดที่  $i$  โดย  $\alpha$  คือสัดส่วนของค่าใหม่ที่จะมาใช้ปรับ เพื่อให้การปรับนั้นมีคุณสมบัติสู่เข้าการไม่ประสานเวลา

ผลที่ได้จากงานวิจัยข้างต้นจะทำให้โหนดแต่ละตัวมีการไม่ประสานเวลาต่อกัน ทำให้ได้ปริมาณการส่งข้อมูล(Throughput)ที่มากขึ้น และทำให้มีโอการการชนกันของข้อมูล (Collision)น้อยลง โดยมีความเท่าเทียม(Fairness) ในการส่งข้อมูล โดยระบบจะทำการปรับเพื่อเข้าสู่การไม่ประสานเวลาเสมอ แม้ว่าจะมีการเพิ่มหรือลดโหนดเข้ามาในระบบก็ตาม

สังเกตได้ว่ากระบวนการของอัลกอริทึมนี้จะทำการแบ่งช่วงเวลาที่ไม่เท่ากันระหว่างโหนดแต่ละคู่โดยการกระจายออกไปทางซ้ายและขวาในแต่ละรอบซึ่งหากเป็นเช่นนั้นแล้วจะทำให้ระบบเข้าสู่สภาวะการไม่ประสานเวลาโดยมีความแปรปรวนของความผิดพลาดอยู่ในระดับที่สูง เนื่องจากหากในระบบมีโหนดใด ๆ ที่มีความผิดพลาดสูง โหนดนั้น ๆ ก็จะมีการกระจายความผิดพลาดไปให้โหนดเพื่อนบ้านและทำให้โหนดเพื่อนบ้านตัวที่ถูกต้องอยู่แล้วมีความผิดพลาดตาม ดังในภาพที่ 2.4

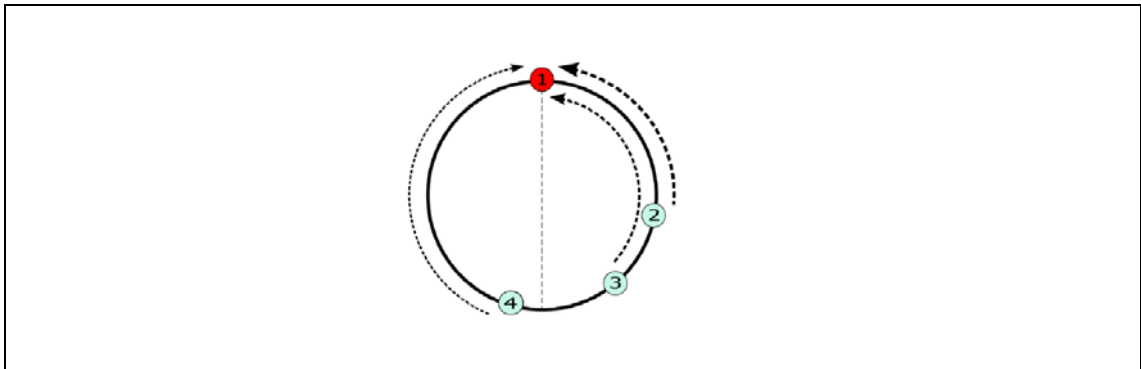


ภาพที่ 2.4 ภาพแสดงการกระจายตัวของความผิดพลาด

ซึ่งผู้เขียนได้ตระหนักถึงปัญหาจึงได้นำเสนออัลกอริทึมแบบจัดฟันเพื่อมาแก้ไขปัญหาที่มีอยู่ของอัลกอริทึมเดิมต่อไป

### 2.2.2 DWARF: Desynchronization with artificial force field [4]

เนื่องจาก DESYNC นั้นได้ใช้ข้อมูลของโหนดเพื่อนบ้านเท่านั้นในการจัดการทำให้ระบบเข้าสู่สถานะการไม่ประสานเวลา แต่ทว่าระบบจะอยู่ในสถานะการไม่ประสานเวลาได้ก็ต่อเมื่อระยะห่างของช่วงเวลาระหว่างทุกโหนดต้องเป็นค่าที่ถูกต้องดังนั้น DWARF จึงได้ใช้หลักการของสนามแม่เหล็กไฟฟ้าสมมุติให้ทุกโหนดมีแรงกระทำต่อกันตามภาพที่ 2.13 เพื่อให้ในการปรับตำแหน่งเวลาให้เข้าสู่สถานะการไม่ประสานเวลาได้ถูกต้องยิ่งขึ้น



ภาพที่ 2.5 ภาพจำลองแรงของ DWARF

โดยลูกศรเส้นประโค้งจะแสดงถึงแรงทั้งหมดที่กระทำต่อโหนดที่ 1 หากระยะของช่วงเวลาของโหนดมีขนาดกว้างแรงที่กระทำก็จะลดน้อยลงตามไปด้วย แรงที่กระทำต่อโหนดทั้งหมดจะสามารถหักล้างและเสริมกัน และโหนดจะปรับตำแหน่งเวลาของตนตามผลรวมของแรงที่ได้รับทั้งหมด สุดท้ายแล้วระบบจะเข้าสู่สถานะการไม่ประสานเวลาเมื่อผลรวมของแรงที่กระทำที่โหนดทุก ๆ โหนดมีค่าเป็นศูนย์

### 2.2.3 Towards Desynchronization of Multihop Topologies [5]

เป็นงานวิจัยที่พยายามนำการไม่ประสานเวลาเดิม มาปรับปรุงให้ดีขึ้น โดนทำให้สามารถทำการไม่ประสานเวลาในระยะที่ไกลขึ้นโดยสามารถทำให้ใช้ในการติดต่อสื่อสารในระยะมากกว่า 1 ช่วงของการส่งได้ โดยใช้วิธีการแต่มสึบนจุดของกราฟให้ไม่ซ้ำกัน โดยสีที่ไม่ซ้ำกันนี้จะแทนช่องเวลาของ TDMA เป็นการพยายามนำ Desync มาใช้บน Multi-hop Topology

ซึ่งงานวิจัยที่จะทำการปรับปรุงอัลกอริทึมนี้ จะสนใจเฉพาะการไม่ประสานเวลาเพียง 1 ช่วงการส่งเท่านั้นโดยจะนำไปทดลองใน Multi-hop Topology ในภายหลัง

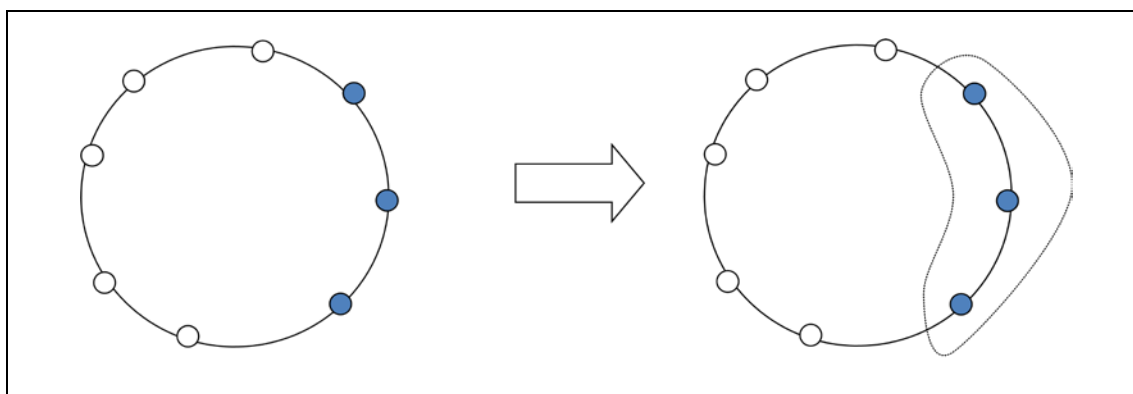
### บทที่ 3

#### การออกแบบอัลกอริทึมการไม่ประสานเวลาแบบจัดฟัน

วิทยานิพนธ์นี้ ได้นำเสนอ อัลกอริทึมการไม่ประสานเวลาแบบจัดฟัน โดยมีวัตถุประสงค์เพื่อปรับปรุงอัลกอริทึมการไม่ประสานเวลาที่มีอยู่เดิมให้สามารถทำให้ระบบเข้าสู่สถานะการไม่ประสานเวลาได้อย่างถูกต้องรวดเร็วยิ่งขึ้น โดยจะกระทำโดยพยายามให้โหนดที่มีความถูกต้องสูงเมื่อเทียบกับเพื่อนบ้านมีการปรับค่าของช่วงเวลาที่น้อยลงหรือไม่ปรับเลย เช่นเดียวกับการจัดฟันที่ทันตแพทย์จะมัดฟันที่มีระยะที่ถูกต้องเข้าด้วยกัน

ซึ่งหลักการเดียวกันนี้ได้นำมาใช้ในการออกแบบอัลกอริทึมโดยเราต้องการมัดโหนดที่มีความถูกต้องอยู่แล้วเข้าด้วยกันเพื่อให้ช่องว่างที่ถูกต้องอยู่แล้วไม่เปลี่ยนแปลง ดังในภาพที่

#### 3.1



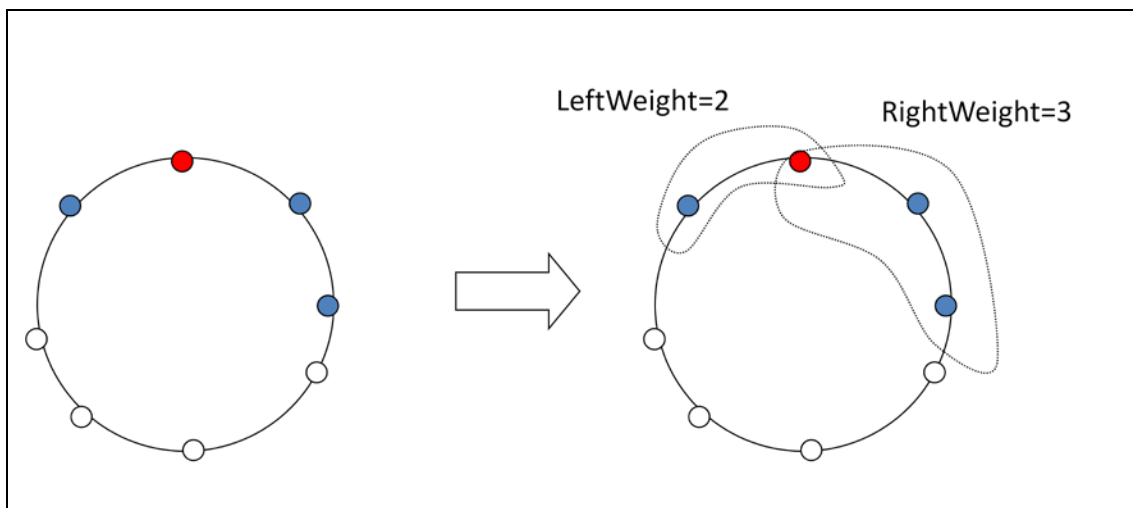
ภาพที่ 3.1 แนวคิดของอัลกอริทึม

จากภาพที่ 3.1 เราจะเห็นได้ว่าโหนดกลุ่มสีฟ้ามีความถูกต้องอยู่แล้ว เราจึงมัดรวมเข้าด้วยกันและพยายามรักษาระยะที่ถูกต้องระหว่างกันไว้เพื่อให้ความผิดพลาดไม่กระจายไปยังโหนดในกลุ่มนี้

การทำงานของอัลกอริทึมจะแบ่งออกเป็น 2 ขั้นตอนหลัก คือ ขั้นตอนแรกจะเป็นการตรวจสอบความถูกต้องของโหนดเมื่อเทียบกับเพื่อนบ้านข้างเคียงหลายๆตัวโดยแต่ละโหนดจะเก็บค่าน้ำหนักของตนเองซึ่งเปรียบเสมือนค่าความถูกต้องเมื่อเทียบความต่างของช่วงเวลากับโหนดเพื่อนบ้านข้างเคียง และขั้นตอนที่สองก็จะนำค่าน้ำหนักที่ได้ทำการหาค่าการปรับช่วงเวลาที่เหมาะสมต่อไป

### 3.1 ขั้นตอนการตรวจสอบความถูกต้องของโหนด

การทำงานของอัลกอริทึมนี้จะแบ่งออกเป็นสองส่วน โดยจะทำการหาน้ำหนักของตนเองในฝั่งซ้ายจนกระทั่งถึงเวลาที่ตนเองจะต้อง broadcast ข้อความจึงจะทำการหาน้ำหนักในฝั่งขวาต่อไป ดังในภาพที่ 3.2



ภาพที่ 3.2 การหาน้ำหนักของโหนดทั้งสองฝั่ง

#### 3.1.1 การคำนวณหาน้ำหนักฝั่งซ้าย

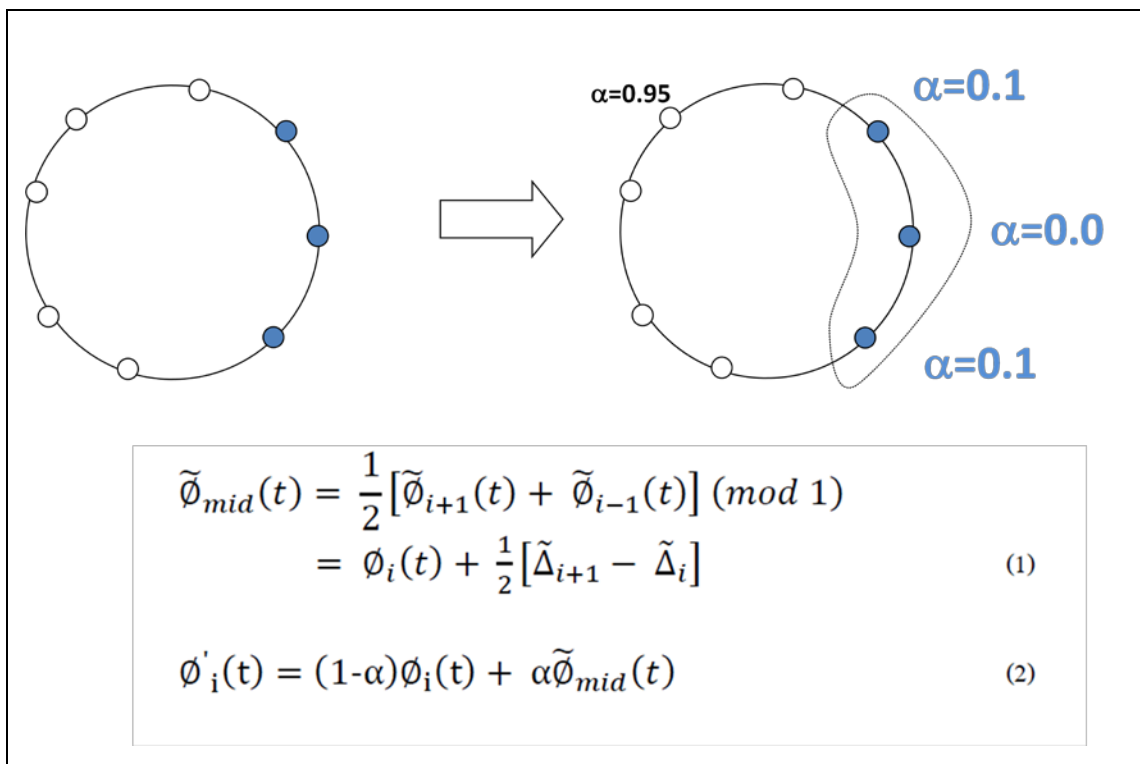
น้ำหนักทางฝั่งซ้ายจะสามารถคำนวณโดยคอยตรวจสอบเวลา broadcast ของเพื่อนบ้านที่ทำการ broadcast ไปก่อนหน้าเราตั้งช่วงตำแหน่งเวลา 6 – 12 นาฬิกาของตนเอง โดยหากมีระยะที่เพื่อนบ้านห่างจากเรามีค่าน้อยกว่า  $\in$  แล้วโหนดก็จะเพิ่มน้ำหนักทางฝั่งซ้ายของตนเองไปที่ละหนึ่งหน่วย หากเพื่อนบ้านตัวถัดมามีระยะที่ไม่ถูกต้องก็จะปรับน้ำหนักตัวเองเป็นหนึ่ง ทำเช่นเดียวกันนี้ไปจนถึงเวลาที่ตนเองต้อง broadcast ข้อความก็จะเป็นอันเสร็จสิ้นการคำนวณหาน้ำหนักฝั่งซ้าย

#### 3.1.2 การการคำนวณหาน้ำหนักฝั่งขวา

น้ำหนักทางฝั่งขวาจะสามารถคำนวณโดยคอยตรวจสอบเวลา broadcast ของเพื่อนบ้านที่ทำการ broadcast ไปต่อจากเราตั้งช่วงตำแหน่งเวลา 12 – 6 นาฬิกาของตนเอง โดยหากมีระยะที่เพื่อนบ้านห่างจากเรามีค่าน้อยกว่า  $\in$  แล้วโหนดก็จะเพิ่มน้ำหนักทางฝั่งขวาของตนเองไปที่ละหนึ่งหน่วย หากเพื่อนบ้านตัวถัดมามีระยะที่ไม่ถูกต้องก็จะเป็นอันเสร็จสิ้นการคำนวณหาน้ำหนักฝั่งขวา

### 3.2 หาค่าการปรับช่วงเวลาที่เหมาะสม

ในขั้นตอนนี้เราจําหน่าย้หน้าหนักที่คำนวณได้ทั้งสองฝั่งมาหาค่าการปรับช่วงเวลา  $\alpha$  ซึ่งในส่วนของอัลกอริทึมเดิมที่นำเสนอมาแล้วนั้นได้ทำการตั้งค่า  $\alpha$  ไว้ที่ 0.95 โดยไม่มีการเปลี่ยนแปลงค่าใดๆ ทำให้โหนดที่มีความผิดพลาดสูงสามารถกระจายความผิดพลาดไปยังเพื่อนบ้านได้ ดังนั้นเราจึงเปลี่ยนค่า  $\alpha$  ไปตามน้ำหนักที่คำนวณดังในภาพที่ 3.2

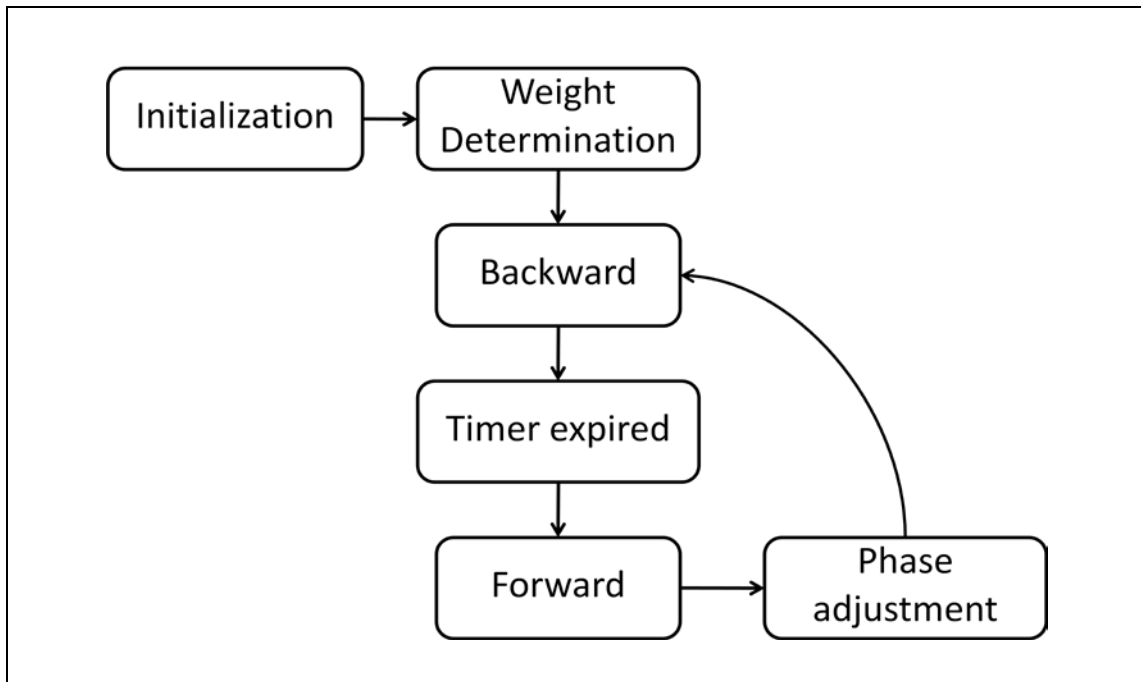


ภาพที่ 3.3 การปรับค่า  $\alpha$  ตามน้ำหนัก

เนื่องจากเราต้องการให้โหนดที่มีกลุ่มอยู่แล้วไม่ขยับ เราจึงต้องการให้โหนดที่มีระยะทั้งทางซ้ายและทางขวาถูกต้องอยู่แล้วไม่ขยับเราจึงให้ค่า  $\alpha=0$  หากโหนดมีน้ำหนักทั้งสองฝั่งมากกว่า 1 และให้โหนดที่อยู่ตรงปลายของกลุ่มสามารถปรับได้เล็กน้อยเพื่อให้สามารถเกิดการรวมกลุ่มได้เราจึงให้ค่า  $\alpha=0.1$  หากโหนดมีน้ำหนักฝั่งใดฝั่งหนึ่งมากกว่า 1

### 3.3 การทำงานของอัลกอริทึม

ภาพรวมขั้นตอนการทำงานของอัลกอริทึมจะสามารถเขียนเป็นแผนภูมิสถานะ (State Diagram) ได้ดังภาพที่ 3.4



ภาพที่ 3.4 แผนภูมิสถานะ

โดยแต่ละโหนดจะเริ่มจากสถานะ Initialization เพื่อทำการตั้งค่าเวลาในแต่ละรอบให้ตรงกันและจะเข้าสู่สถานะ Weight Determination เพื่อหาน้ำหนักทั้งฝั่งซ้าย (Backward) และฝั่งขวา (Forward) โดยมีสถานะ Timer expired ขึ้นกลางในช่วงที่โหนดต้องการบรอดแคส และหลังจากนั้นโหนดก็จะไปสู่สถานะ Phase adjustment เพื่อทำการปรับค่า  $\alpha$  ต่อไป



### 3.4 รหัสเทียมของอัลกอริทึม

จากแผนภูมิสถานะเราสามารถนำมาเขียนเป็นรหัสเทียมได้ 3 ส่วนดังภาพที่ 3.4.1-

#### 3.4.3

<b>Algorithm 1 Initialization</b>
1: $T \leftarrow TimePeriod$ {Configurable Time Period}
2: $PGAP \leftarrow T/NumNeighbors$ {Perfect phase diff}
3: $nextNbrTime, prevNbrTime, lastReceiveTime \leftarrow 0$
4: $lastFiringTime, bckwdWeight, fwdWeight \leftarrow 0$
5: $chaining \leftarrow BACKWARD$
6: call $SetFiringTimer(T)$

ภาพที่ 3.5 รหัสเทียมในส่วนของ Initialization

ในส่วนของ Initialization นั้น โหนดทุกตัวจะทำการเซตค่าพารามิเตอร์ต่างๆโดยมีค่าที่สำคัญดังนี้

- $T$  = ค่าเวลาในแต่ละรอบ
- $PGAP = T/N$  = ระยะห่างระหว่าง 1 โหนดที่ต้องการ

ส่วนค่าอื่นๆนั้นจะถูกเซตเป็น 0 เพื่อคอยรับตัวแปรต่างๆต่อไป

---

**Algorithm 2** Weight Determination Algorithm
 

---

```

1: Upon receiving a firing message
2:  $receiveTime \leftarrow currentTime$ 
3: if  $lastReceiveTime \leq lastFiringTime$  then
4:    $nextNbrTime \leftarrow receiveTime$ 
5: end if
6: if chaining is BACKWARD then
7:    $diff \leftarrow T - (receiveTime - lastFiringTime)$ 
8:    $neighborDiff \leftarrow receiveTime - lastReceiveTime$ 
9:   if  $(neighborDiff \leq \epsilon)$ 
     AND  $((diff \bmod PGAP \leq \epsilon)$  OR  $(PGAP - (diff$ 
      $\bmod PGAP) \leq \epsilon))$  then
10:     $bckwdWeight \leftarrow bckwdWeight + 1$ 
11:   else
12:     $bckwdWeight \leftarrow 0$ 
13:   end if
14: else {chaining is FORWARD}
15:    $diff \leftarrow receiveTime - lastFiringTime$ 
16:   if  $|diff - (PGAP * (fwdWeight + 1))| \leq \epsilon$  then
17:     $fwdWeight \leftarrow fwdWeight + 1$ 
18:   else {chain is broken}
19:    call Phase adjustment
20:   end if
21: end if
22:  $lastReceiveTime \leftarrow receiveTime$ 

```

ภาพที่ 3.6 รหัสเทียมในส่วนของ Weight Determination

ในส่วนของ Weight Determination นั้น โหนดทุกตัวจะทำการคำนวณความถูกต้องของตนเองเพื่อใช้ในการปรับต่อไป

---

**Algorithm 3** Phase adjustment algorithm
 

---

```

1: if bckwdWeight  $\geq$  1 AND fwdWeight  $\geq$  1 then
2:    $\alpha \leftarrow 0$ 
3: else if bckwdWeight  $\geq$  1 XOR fwdWeight  $\geq$  1 then
4:    $\alpha \leftarrow 0.1$ 
5: else
6:    $\alpha \leftarrow 0.95$ 
7: end if
8: goalTime  $\leftarrow (((1 - \alpha) * lastFiringTime) + (\alpha * (prevNbrTime + nextNbrTime)/2)) + T$ 
9: call SetFiringTimer(goalTime - currentTime)
10: bckwdWeight, fwdWeight  $\leftarrow 0$ 
11: chaining  $\leftarrow BACKWARD$ 

```

---

ภาพที่ 3.7 รหัสเทียมในส่วนของ Phase adjustment

และในส่วนในส่วนของ Phase adjustment นั้น โหนดทุกตัวจะทำการเซตค่า  $\alpha$  ที่เหมาะสมในการปรับ พร้อมทั้งคำนวณตำแหน่งเวลาที่ตัวเองจะต้องปรับต่อไป

## บทที่ 4

### ผลการทดลองและวิเคราะห์ผล

เนื่องด้วยในปัจจุบัน ตัวอุปกรณ์โหมตที่ใช้ทำการทดลอง มีราคาจำหน่ายที่ค่อนข้างสูงพอสมควร อีกทั้งในงานวิจัยนี้มีความจำเป็นที่จะต้องใช้อุปกรณ์จำนวนมากในการทดสอบการทำงานของอัลกอริทึม ดังนั้นการวัดผลจะกระทำทดลองบนโปรแกรมจำลอง (Simulator) ที่ชื่อว่าทอสซิม (TOSSIM) [1] แทนการใช้อุปกรณ์จริง โปรแกรมจำลองทอสซิมนี้สามารถ จำลองการทำงานของระบบปฏิบัติการไทนีโอเอส (TinyOS) [6] ที่ใช้ทำงาน บนแพลตฟอร์ม ตัวโหมตจริงได้ โปรแกรมประยุกต์ที่ใช้ในการทดสอบจะเขียนขึ้น ด้วยภาษาโปรแกรมเนสซี (nesC) [7] ซึ่งเป็นภาษาเดียวกันกับที่ใช้เขียนโปรแกรมเพื่อทำงานบนระบบไทนีโอเอส ดังนั้นในทางปฏิบัติแล้วตัวโปรแกรมทดสอบการทำงานของอัลกอริทึมนี้จะสามารถนำไปใช้งานในอุปกรณ์โหมตรวมไปถึงระบบบนสภาพแวดล้อมจริงได้

อัลกอริทึมการไม่ประสานเวลาที่นำมาทดสอบจะมีทั้งสิ้น 2 อัลกอริทึม โดยที่อัลกอริทึมแรกจะเป็นอัลกอริทึมจากงานวิจัยก่อนหน้าที่นำมาเปรียบเทียบผล ส่วนอัลกอริทึมหลังจะเป็นอัลกอริทึมใหม่ที่น่าเสนอในงานวิจัยนี้

ในหัวข้อที่ 4.1 จะกล่าวถึงรายละเอียดเครื่องมือต่างๆที่ใช้ในการทดลอง จากนั้นในหัวข้อที่ 4.2 จะอธิบายถึงวิธีการประเมินประสิทธิภาพของอัลกอริทึมการไม่ประสานเวลา 4.3 จะอธิบายถึงค่าพารามิเตอร์และวิธีการต่างๆที่ใช้ในการทดลองและ 4.3 จะเป็นผลการทดลองเปรียบเทียบระหว่างการทำงานของทั้งสองอัลกอริทึม

#### 4.1 เครื่องมือที่ใช้ในการทดลอง

##### 4.1.1 ระบบปฏิบัติการไทนีโอเอส (TinyOS) [6]

ไทนีโอเอส (TinyOS) [6] เป็นชื่อของระบบปฏิบัติการหนึ่งที่ถูกออกแบบมาเพื่อใช้ทำงานบนตัว โหมตโดยเฉพาะ การทำงาน ของระบบ จะเป็น อยู่บนพื้นฐานการเกิดขึ้นของเหตุการณ์เป็นหลัก ( Event-Based) โดยที่ ณ ขณะใดขณะหนึ่งจะมีเพียงแค่ 1 กระบวนการ (Process) เท่านั้นที่จะทำงานในระบบ หน่วยความจำจะถูกจัดสรร ตั้งแต่ตอน แปลโปรแกรม (Compile time) ซึ่งรหัสคำสั่งไทนีโอเอส (TinyOS code) และรหัสคำสั่ง ตัวโปรแกรมประยุกต์ (Application code) ที่เขียน จะถูกแปลรวมด้วยกันแล้วค่อยทำงาน

ไทนีโอเอสและโปรแกรมประยุกต์ ที่ทำงานจะแบ่งส่วนของการทำงานออกเป็น ส่วนโปรแกรม (Component) แต่ละส่วนโปรแกรมจะมีความสามารถในการทำงานที่แตกต่างกัน โดยที่ในแต่ละส่วนโปรแกรม จะมีส่วนที่เรียกว่า ส่วนต่อประสาน (Interface) ซึ่งประกอบไปด้วย ฟังก์ชันให้เรียกใช้เพื่อเชื่อมต่อการทำงานของส่วนโปรแกรมเข้าด้วยกัน ส่วนโปรแกรมที่เรียกใช้ฟังก์ชันจะอยู่ในฐานะเป็นผู้ให้บริการฟังก์ชัน ในทางตรงข้าม ส่วนโปรแกรมที่กำหนดการทำงานของฟังก์ชันจะเป็นผู้ให้บริการฟังก์ชัน

#### 4.1.2 โปรแกรมจำลองทอสซิม (TOSSIM) [1]

ทอสซิม (TOSSIM) [1] เป็นโปรแกรมจำลอง ที่ถูกเขียนขึ้นด้วยภาษา ซีและเนสซี เพื่อจำลองรูปแบบโครงสร้างการทำงานจริงของระบบไทนีโอเอส โดยการทำงาน จะเริ่มตั้งแต่ฟังก์ชัน main() ในแฟ้มชื่อ nido.nc ซึ่งมีขั้นตอนดังนี้

- 1) กำหนดค่าพารามิเตอร์ (Parameters) ตามที่ได้รับจากบรรทัดคำสั่ง (Command line)
- 2) กำหนดสถานะเริ่มต้นของทอสซิม
- 3) กำหนดการทำงานของคำสั่งภายนอก (External command)
- 4) กำหนด สถานะ เริ่มต้นของแต่ละ โหมด จาก การเรียก ฟังก์ชัน StdControl.init() และ StdControl.start()
- 5) เริ่มต้นวงวนเพื่อจัดการเกี่ยวกับเหตุการณ์ (Events) และภารกิจ (Tasks)

#### 4.1.3 ภาษาโปรแกรมเนสซี (nesC Language) [7]

เนสซี (nesC) [7] คือภาษา โปรแกรม ที่ใช้ในการเขียนโปรแกรม ประยุกต์ สำหรับทำงานบนโหมดสำหรับ ระบบเครือข่ายตัวรับรู้ โดยที่รูปแบบการเขียนจะเป็นลักษณะการนำเอา ส่วนของโปรแกรมหลายๆส่วนมาเชื่อมต่อกันเพื่อทำงานตามที่โปรแกรมประยุกต์ต้องการ ลักษณะการเขียนแบบนี้จะเรียกว่าลักษณะการเขียนโปรแกรมโดยอาศัยส่วนโปรแกรมประกอบ (Component-Based) นอกจากนี้เนสซียังถูกออกแบบมาให้มีการทำงานแบบเน้นประหยัดพลังงานด้วย ซึ่งเหมาะกับคุณลักษณะของโหมดที่มีพลังงานจำกัด

#### 4.2 วิธีการประเมินประสิทธิภาพของอัลกอริทึมการไม่ประสานเวลา

ในการวัดผลอัลกอริทึมการไม่ประสานเวลานั้นเราจะวัดโดยการนำช่องว่างเวลาระหว่างโหนดทุกๆคู่มาหักลบกับค่าที่ควรจะเป็นเพื่อหาค่าความผิดพลาดของแต่ละคู่ หลังจากนั้นเราจะนำค่าความผิดพลาดที่ได้ทั้งหมดไปหา ค่ารากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง (RMSE) ตามสมการในภาพที่ 4.1

$$ERR_i = \Delta\phi_{ij} - \frac{T}{N}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^N ERR_i^2}{N}}$$

ภาพที่ 4.1 สมการค่าความผิดพลาดและรากที่สองของค่าเฉลี่ยความคลาดเคลื่อนกำลังสอง

แต่หากเราต้องการเปรียบเทียบค่าความผิดพลาดที่ความหนาแน่นของโหนดต่างกัน เราจำเป็นต้องหารค่าความผิดพลาดด้วยค่า PGAP หรือค่าช่วงเวลาที่ควรจะเป็นเพื่อเทียบค่าความผิดพลาดเป็นอัตราส่วนจึงจะสามารถนำผลการทดลองที่ได้มาเปรียบเทียบกันได้ดังภาพที่ 4.2

$$NRMSE = \frac{RMSE}{\left(\frac{T}{N}\right)}$$

ภาพที่ 4.2 สมการค่าความผิดพลาดที่ทำให้เป็นมาตรฐานแล้ว

### 4.3 การทดลองทดสอบอัลกอริทึมการไม่ประสานเวลา

ในการทดลองเราจะทำการทดสอบเปรียบเทียบอัลกอริทึมทั้งสองแบบ โดยจะทำค่าความผิดพลาดมาเขียนเป็นกราฟเมื่อเทียบกับเวลาการปรับในแต่ละรอบเป็นเวลาทั้งหมด 300 รอบ หลังจากนั้นจะทำการทดลองเพิ่มเติมโดยทดสอบพารามิเตอร์ดังต่อไปนี้

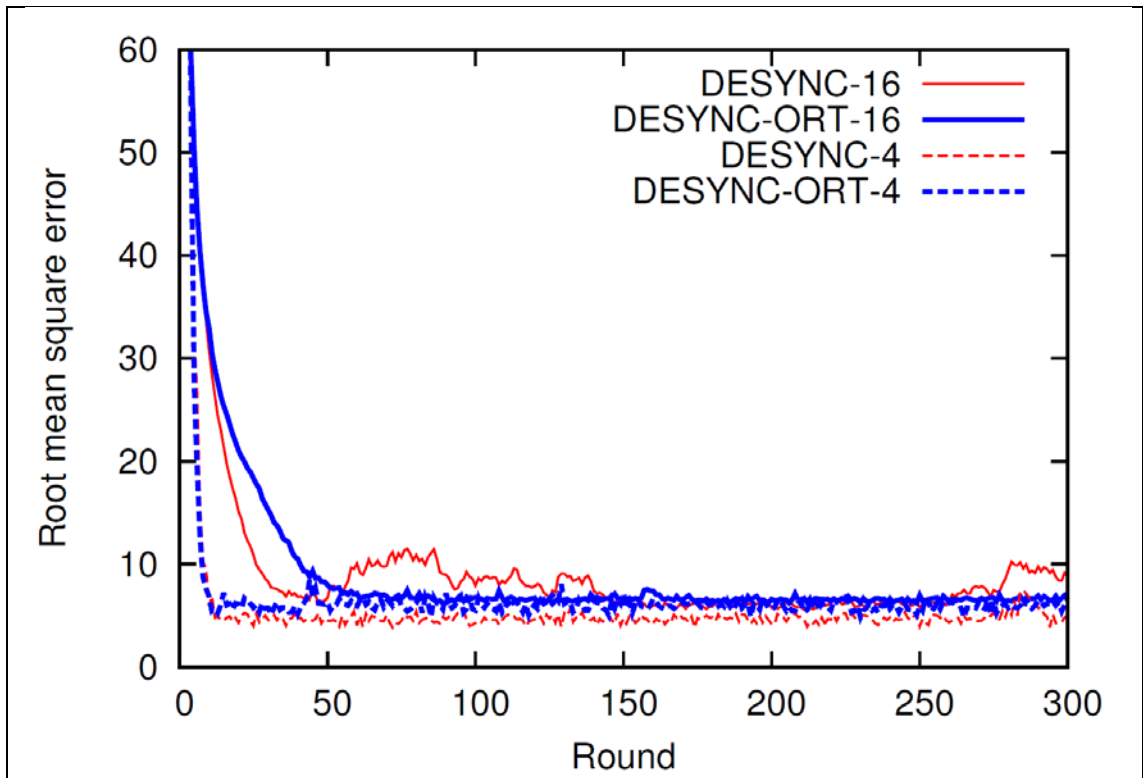
- จำนวนโหนด (N) จะทดลองในช่วง {4,16,32,48,64,80,96}
- คาบเวลา (T) จะทดสอบสองค่า 1000ms และ 2000ms
- ค่าความผิดพลาดที่ยอมรับได้ ( $\epsilon$ ) จะทดสอบ 3ms 6ms และ 9ms
- ค่าการปรับ  $\alpha$  จาก 0.95 ไปเป็น 0.05 0.1 และ 0.2

โดยการทดสอบนั้นเมื่อเราจะให้จำนวนโหนดเป็นตัวแปรตามเสมอและในส่วนของตัวแปรที่เหลือนั้นหากตัวใดตัวหนึ่งเป็นตัวแปรตามแล้วอีกตัวจะเป็นตัวแปรควบคุมเสมอเพื่อให้เห็นผลของพารามิเตอร์ต่างๆที่มีผลต่อค่าความผิดพลาดในการเข้าสู่สถานะการไม่ประสานเวลา

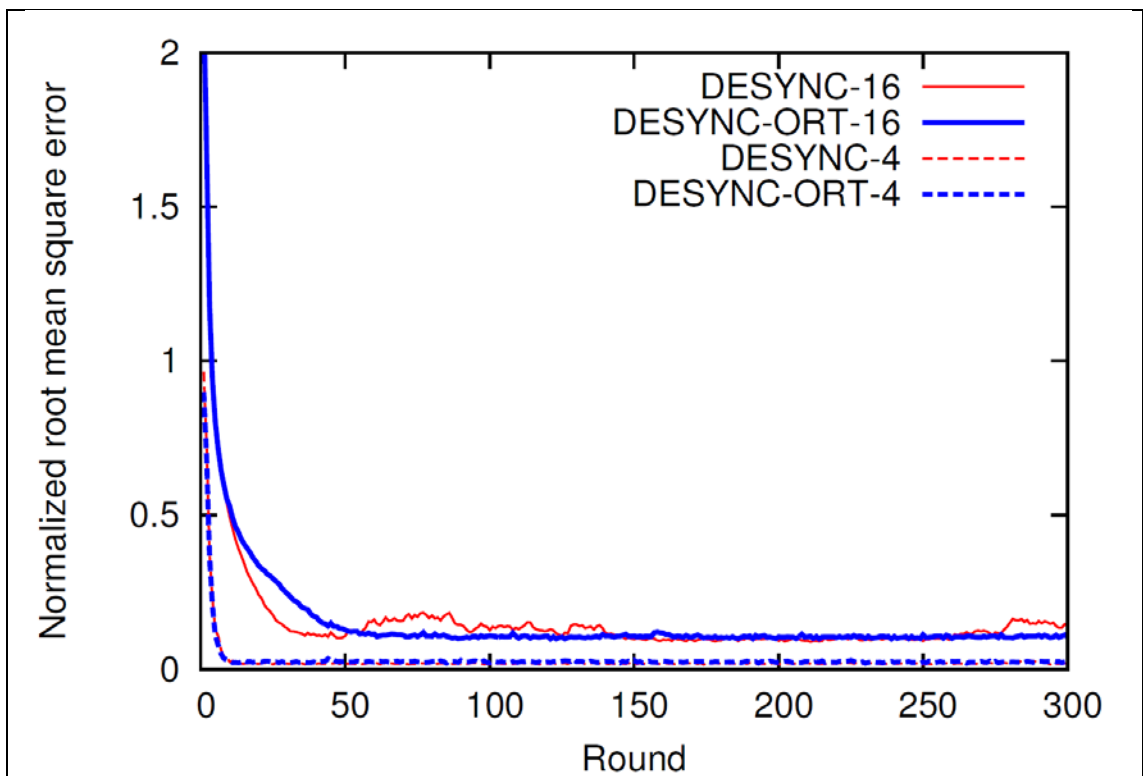
### 4.4 ผลการทดลองเปรียบเทียบอัลกอริทึมการไม่ประสานเวลา

#### 4.4.1 การเปรียบเทียบค่าความผิดพลาดในจำนวนโหนดที่ต่างกัน

เราได้ทำการทดสอบเปรียบเทียบอัลกอริทึมทั้งสองในจำนวนที่ต่างกันโดยใช้คาบเวลา(T)และความผิดพลาดที่ยอมรับได้ ( $\epsilon$ ) เป็น 1000ms และ 6ms ตามลำดับ ได้ผลการทดลองดังในภาพที่ 4.3 ถึง 4.8

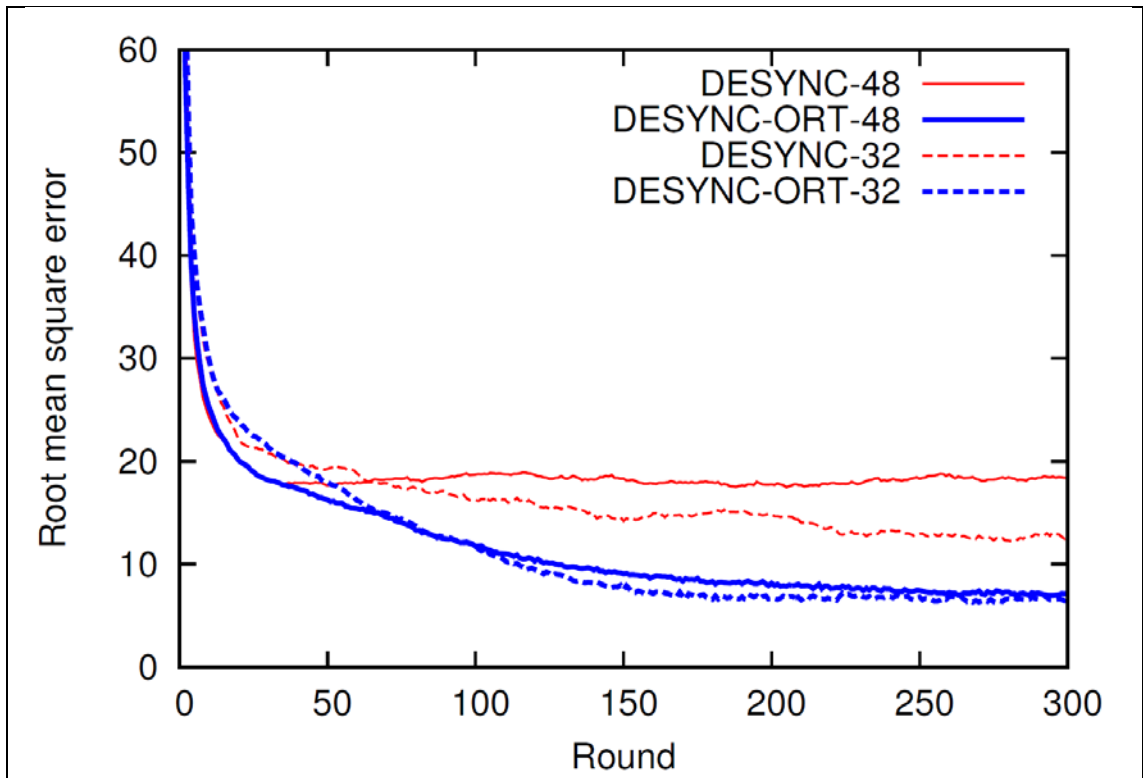


ภาพที่ 4.3 กราฟแสดงค่าความผิดพลาดในความหนาแน่นต่ำ

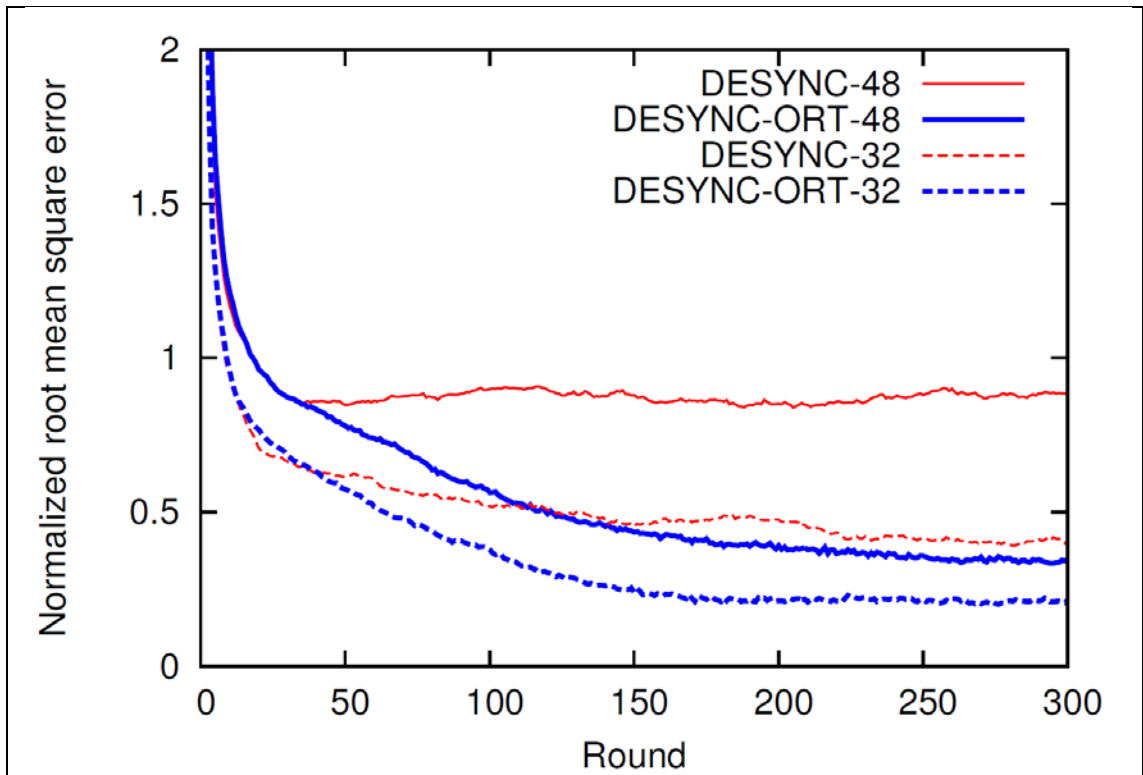


ภาพที่ 4.4 กราฟแสดงค่าความผิดพลาดในความหนาแน่นต่ำที่ทำให้เป็นมาตรฐาน

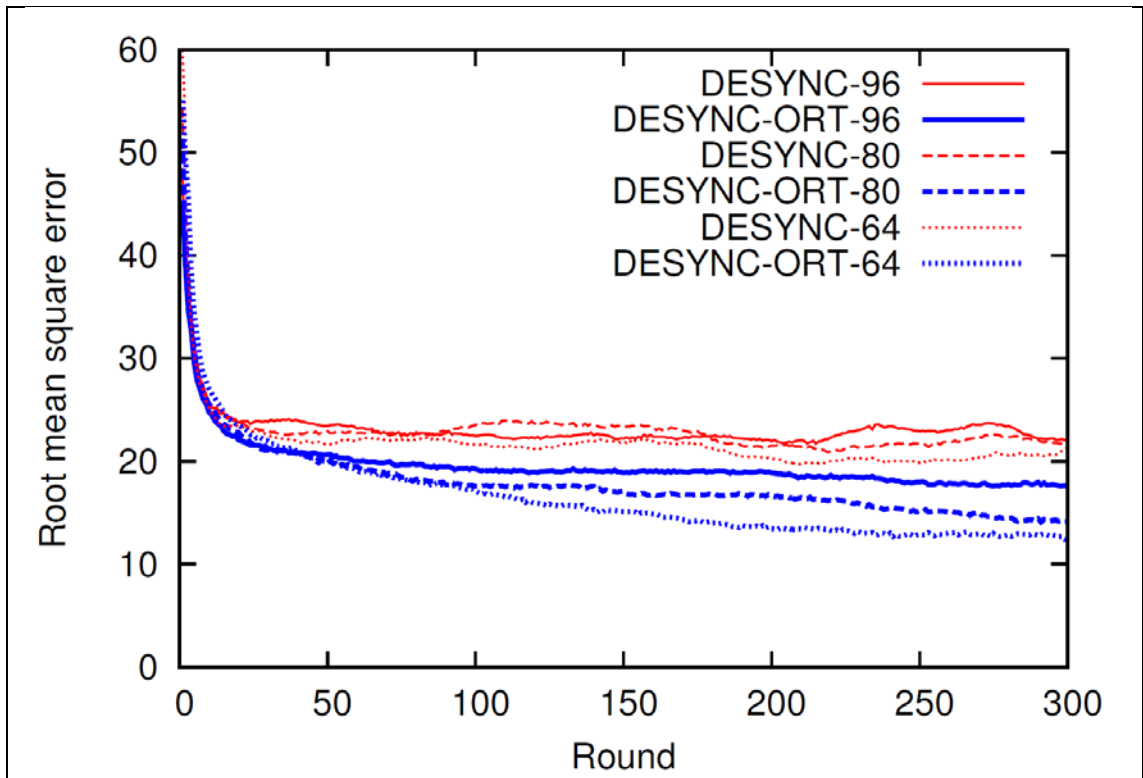




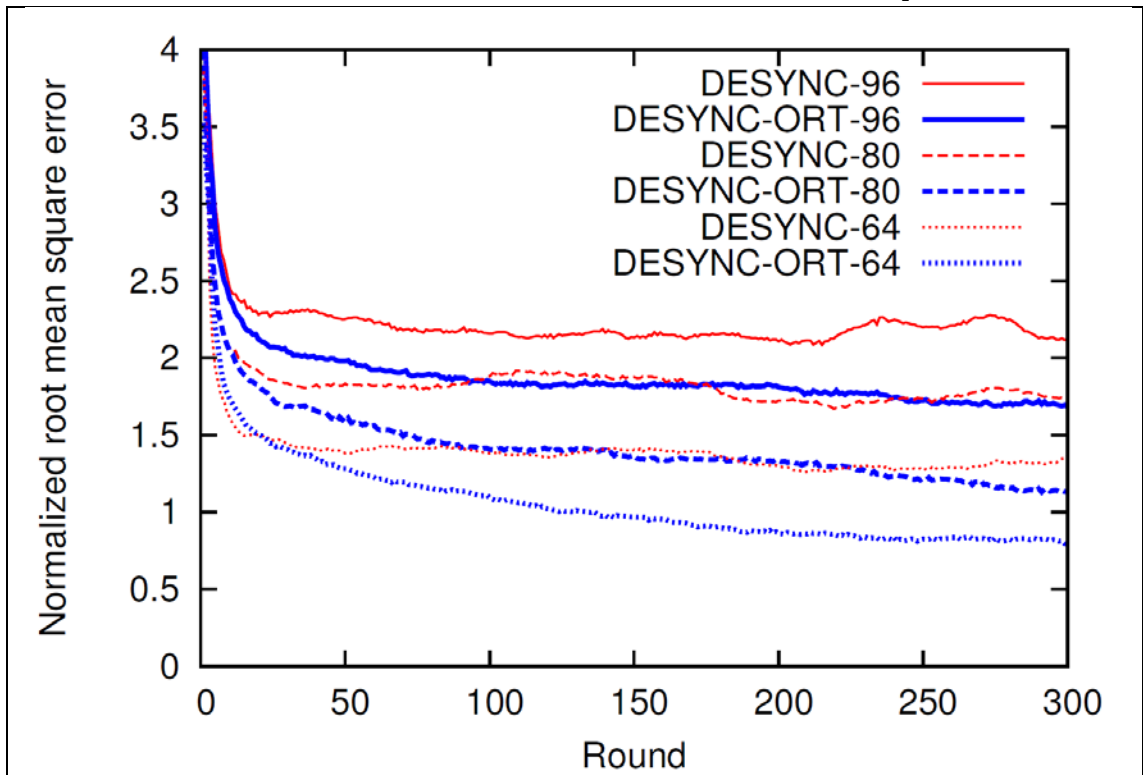
ภาพที่ 4.5 กราฟแสดงค่าความผิดพลาดในความหนาแน่นกลาง



ภาพที่ 4.6 กราฟแสดงค่าความผิดพลาดในความหนาแน่นกลางที่ทำให้เป็นมาตรฐาน

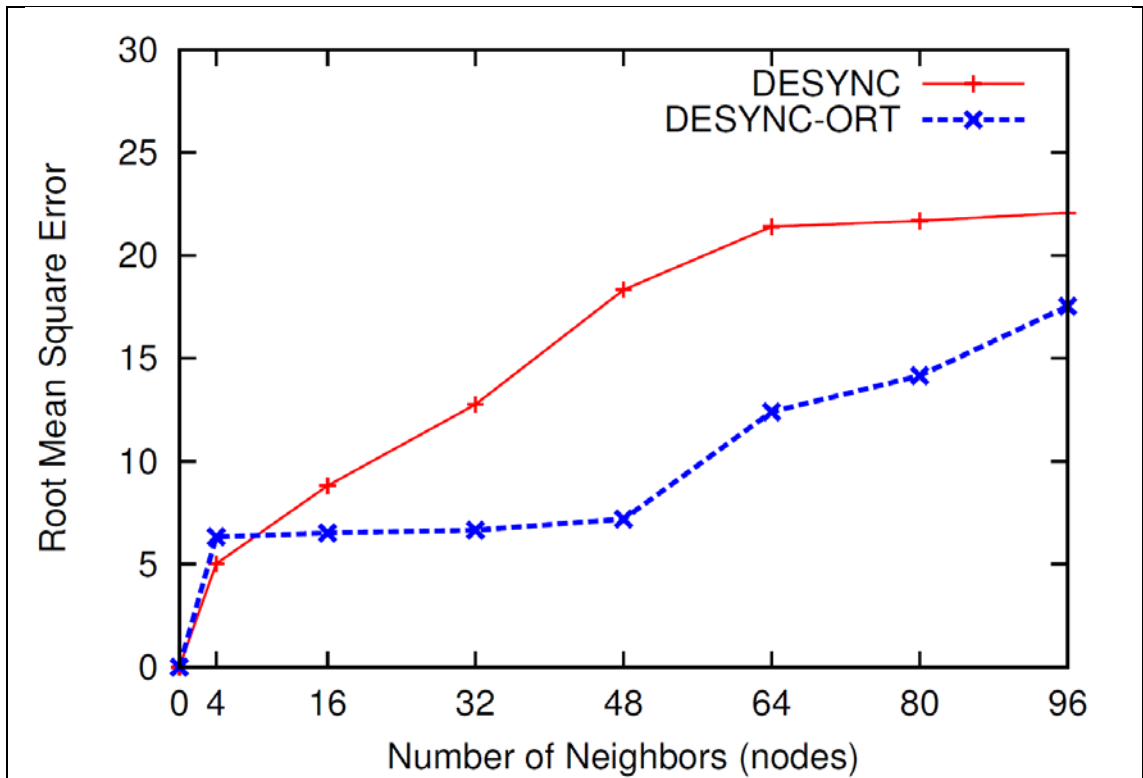


ภาพที่ 4.7 กราฟแสดงค่าความผิดพลาดในความหนาแน่นสูง

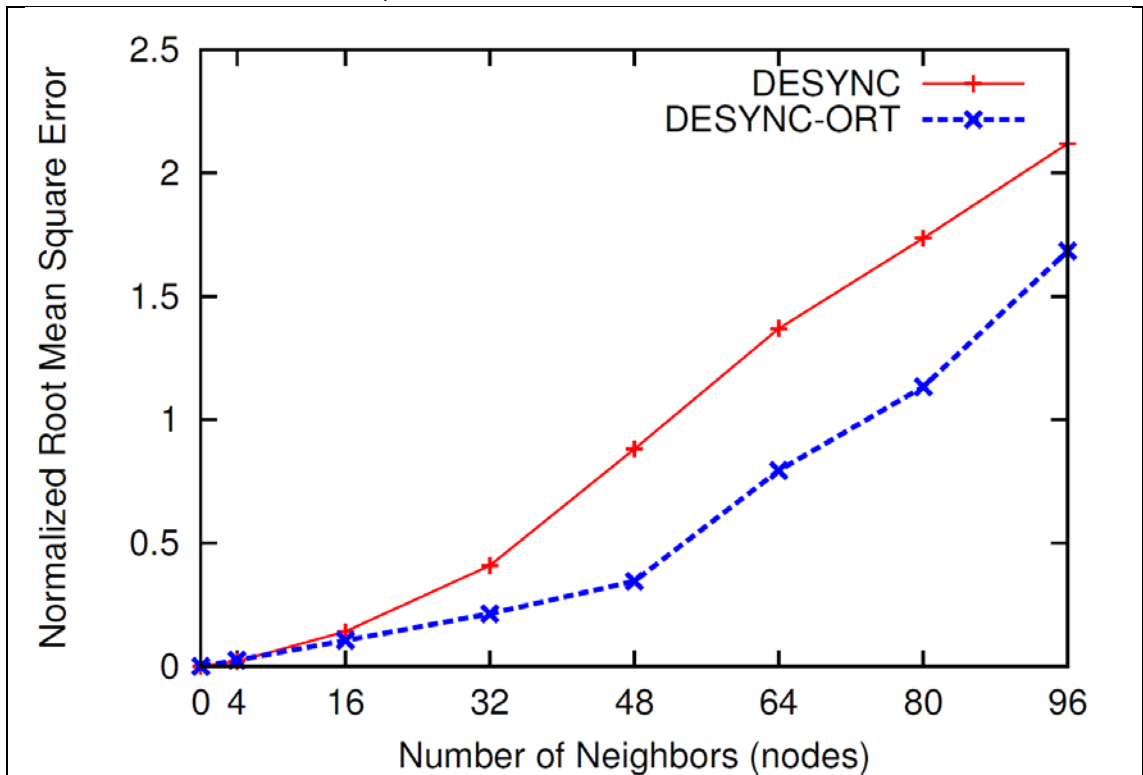


ภาพที่ 4.8 กราฟแสดงค่าความผิดพลาดในความหนาแน่นสูงที่ทำให้เป็นมาตรฐาน

จากภาพที่ 4. 3 – 4.8 เราได้เห็นได้ว่าอัลกอริทึมที่นำเสนอสามารถช่วยให้ อัลกอริทึมเดิมนั้นเข้าสู่สถานะการไม่ประสานเวลาโดยมีความผิดพลาดที่ต่ำลง โดยค่าความ ผิดพลาดนั้นจะสูงขึ้นเมื่อจำนวนโหนดมากขึ้นเพราะเมื่อจำนวนโหนดมากขึ้นแล้วจะทำให้ ข้อความที่บรอดแคสต์ออกไปมีโอกาสที่จะชนกันมากขึ้น จากผลการทดลองจะเห็นได้ว่าในช่วงที่ โหนดมีความหนาแน่นต่ำนั้นอัลกอริทึมการไม่ประสานเวลาแบบจัดฟันจะสามารถช่วยให้ความ ผิดพลาดไม่แปรปรวนได้เพียงอย่างเดียวแต่ในช่วงที่โหนดมีความหนาแน่นในระดับกลางนั้น อัลกอริทึมที่นำเสนอสามารถช่วยให้ระบบเข้าสู่สถานะการไม่ประสานเวลาโดยมีค่าความ ผิดพลาดต่ำกว่าอย่างเห็นได้ชัด ส่วนในช่วงที่มีความหนาแน่นสูงนั้นจะเห็นได้ว่าทั้งสอง อัลกอริทึมจะมีความผิดพลาดสูงเนื่องจากการส่งข้อมูลที่มีโอกาสชนกันได้สูงแต่ทว่าอัลกอริทึมที่ นำเสนอก็ยังสามารถช่วยรักษาระดับของความผิดพลาดให้ต่ำกว่าอัลกอริทึมเดิมได้ โดยเรา สามารถสรุปผลของสถานะการไม่ประสานเวลาเมื่อเวลาผ่านไป 300 รอบได้ดังภาพที่ 4.9-4.10



ภาพที่ 4.9 กราฟสรุปค่าความผิดพลาดหลังจากทำการปรับไป 300 รอบ



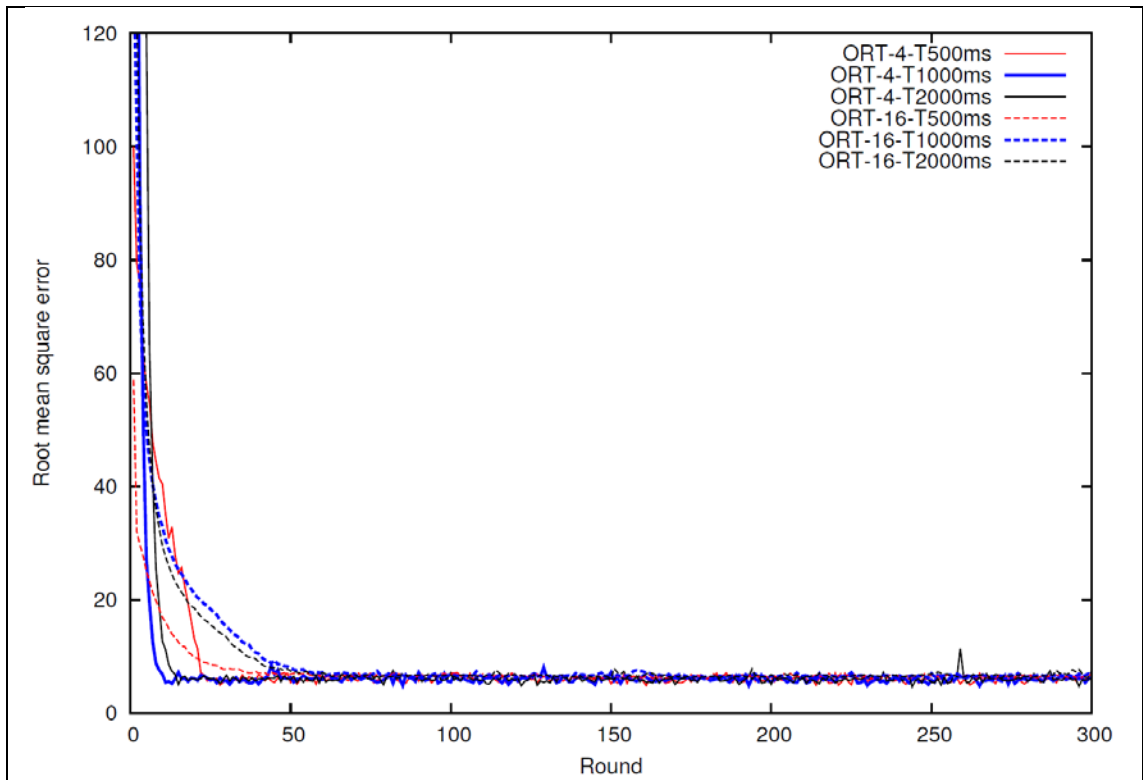
ภาพที่ 4.10 กราฟสรุปค่าความผิดพลาดหลังจากทำการปรับไป 300 รอบ ที่ทำให้เป็นมาตรฐาน

จากกราฟภาพที่ 4.9 และ 4.10 เราจะเห็นได้ว่าอัลกอริทึมที่นำเสนอ นั้นสามารถช่วยให้อัลกอริทึมการไม่ประสานเวลามีความผิดพลาดต่ำลงได้ในทุกช่วงของความหนาแน่นที่ต่างกัน แต่จะให้ผลที่ดีที่สุดในช่วงที่ความหนาแน่นอยู่ในระดับกลาง-สูง

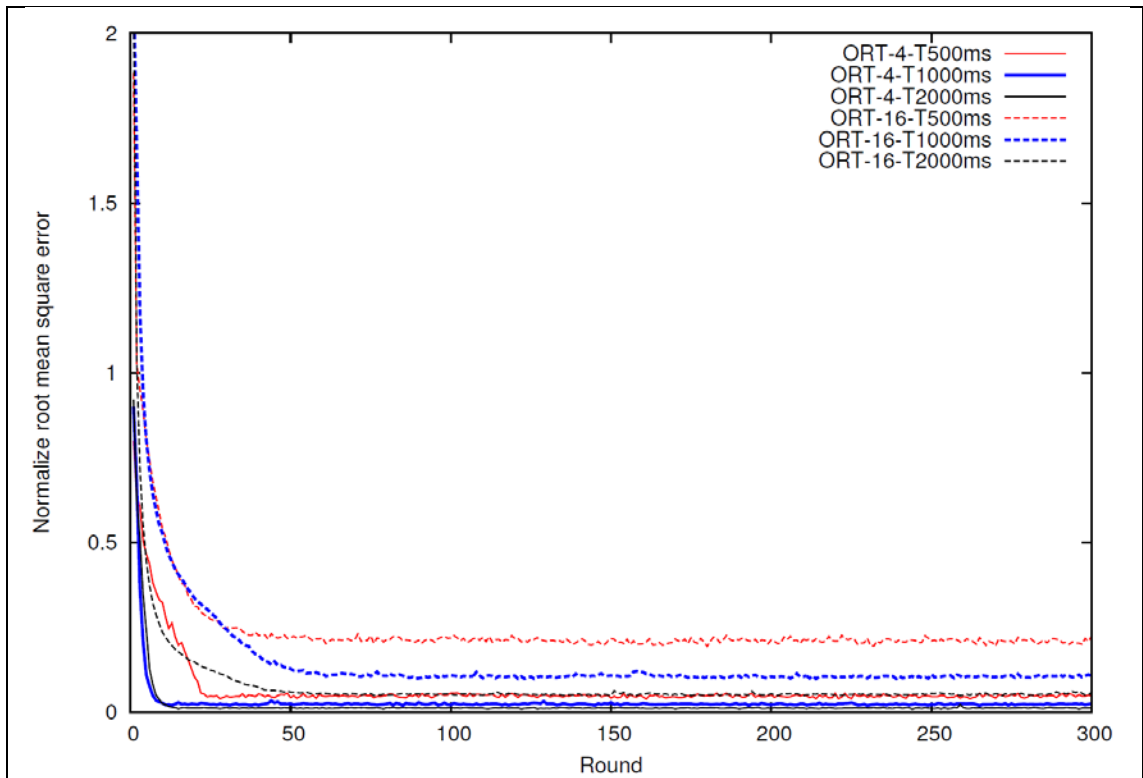
#### 4.4.2 การเปรียบเทียบค่าความผิดพลาดในคาบเวลาที่ต่างกัน

เพื่อเป็นการตรวจสอบว่าคาบเวลาที่มีผลต่ออัลกอริทึมหรือไม่เราได้ทำการทดสอบเปรียบเทียบอัลกอริทึมทั้งสองเช่นเดียวกับ 4.4.1 แต่เพิ่มคาบเวลาเป็น 2 เท่า โดยเราจะนำผลที่ได้ไปเปรียบเทียบกับ 4.4.1 เพื่อดูผลกระทบ ได้ผลการทดลองดังในภาพที่

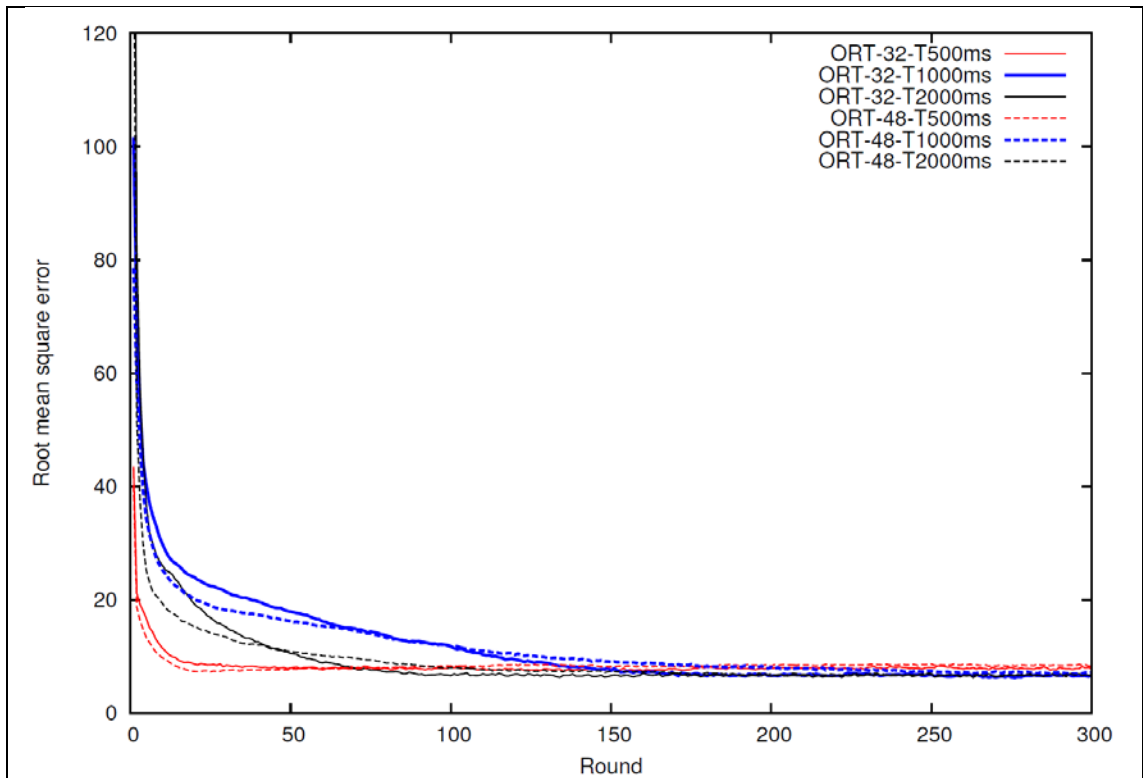
โดยพิจารณาจากกราฟสามารถสรุปได้ว่าเมื่อคาบเวลาน้อยลง ความหนาแน่นเพิ่มขึ้นแล้วโหนดจะมีโอกาสส่งข้อมูลชนกันมากขึ้นทำให้คาบที่เล็กจะมีความผิดพลาดสูงกว่าคาบที่ใหญ่ ทั้งนี้ผลการทดลองกับตรงข้ามในส่วนที่ความหนาแน่นสูงจนใกล้เคียงค่าดีเลย์ในรูปกราฟความหนาแน่นสูง ทั้งนี้เนื่องจากความหนาแน่นจะมีมากจนค่าที่วัดความผิดพลาดนั้นสูงเกินที่จะนำมาใช้จริงได้



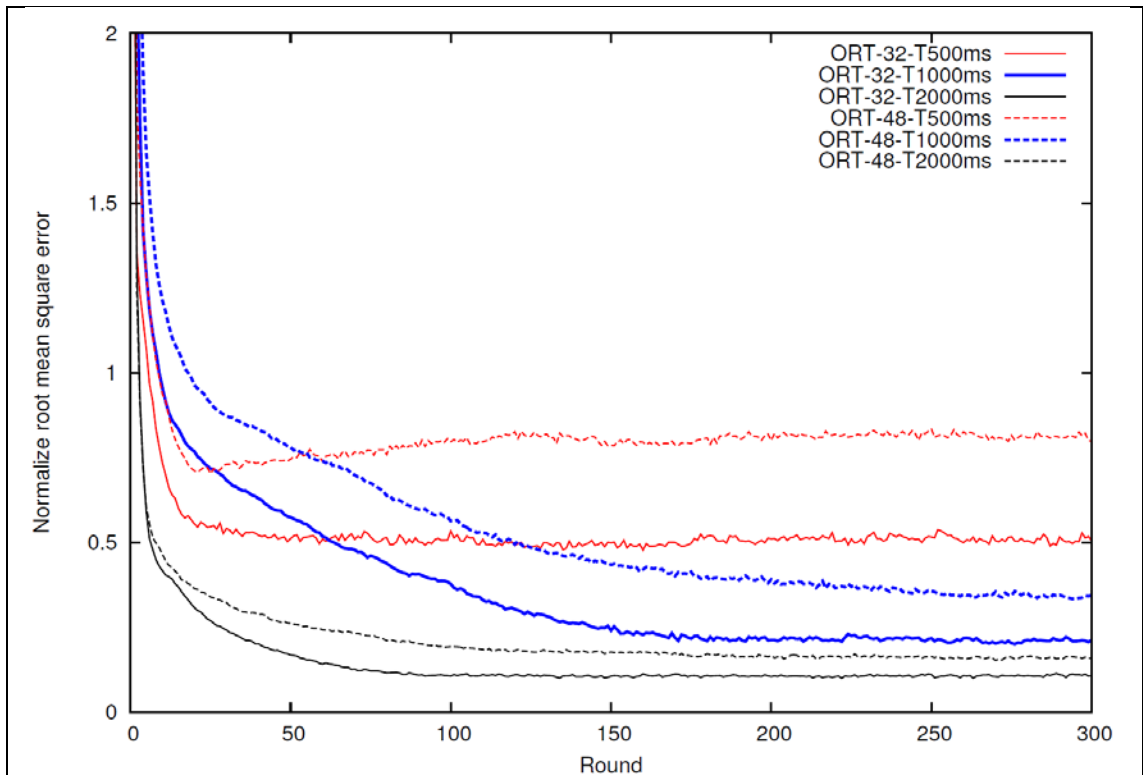
ภาพที่ 4.11 กราฟความผิดพลาดเทียบกับคาบเวลาในความหนาแน่นต่ำ



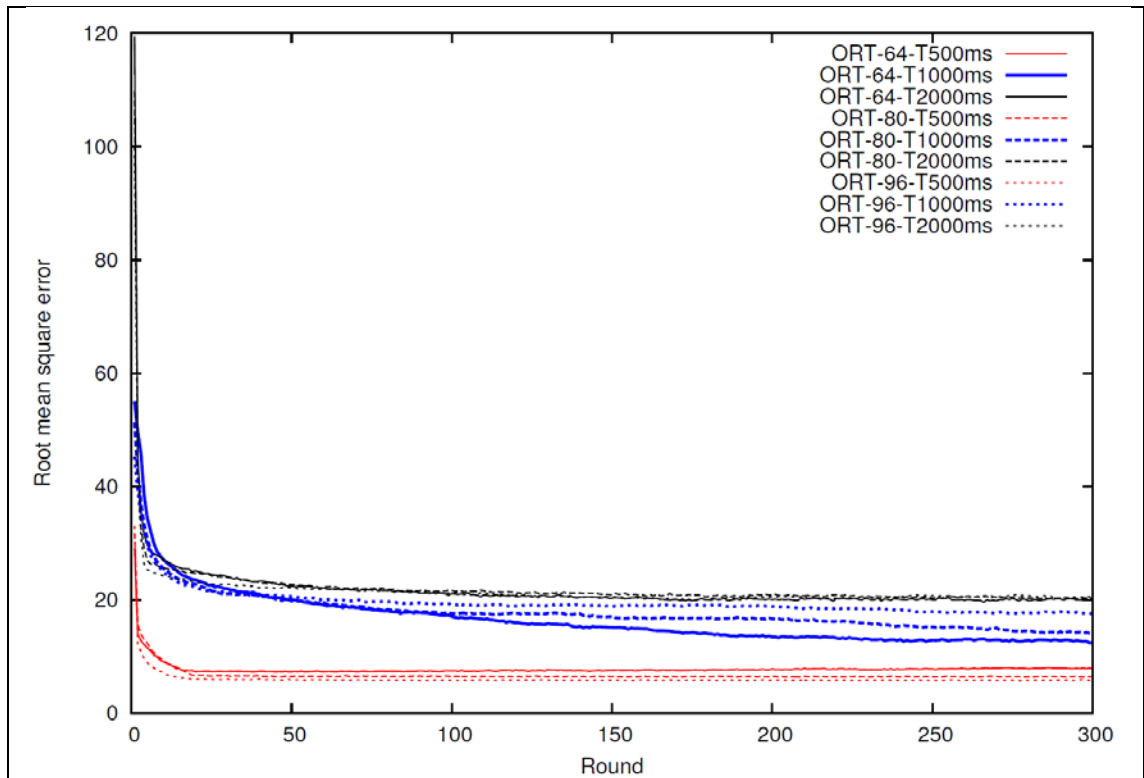
ภาพที่ 4.12 กราฟความผิดพลาดเทียบกับคาบเวลาในความหนาแน่นต่ำที่ทำให้เป็นมาตรฐาน



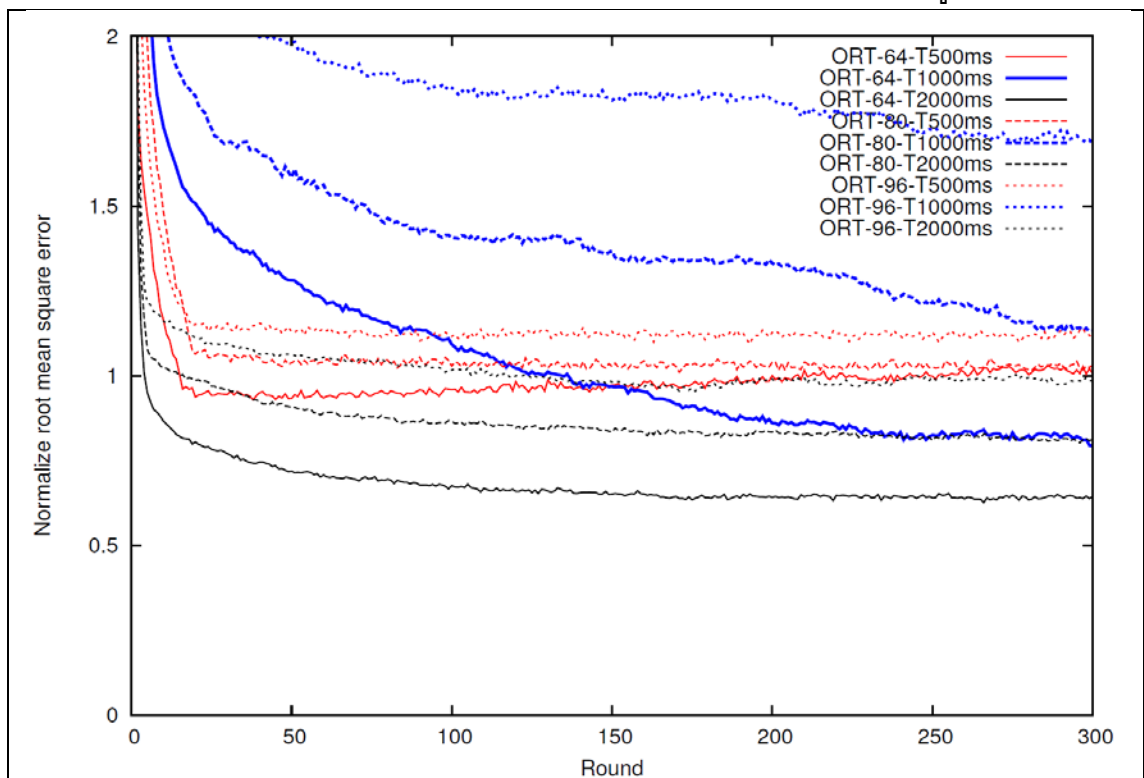
ภาพที่ 4.13 กราฟความผิดพลาดเทียบกับเวลาในความหนาปานกลาง



ภาพที่ 4.14 กราฟความผิดพลาดกับเวลาในความหนาแน่นปานกลางที่ทำเป็นมาตรฐาน



ภาพที่ 4.15 กราฟความผิดพลาดเทียบกับคาบเวลาในความหนาสูง



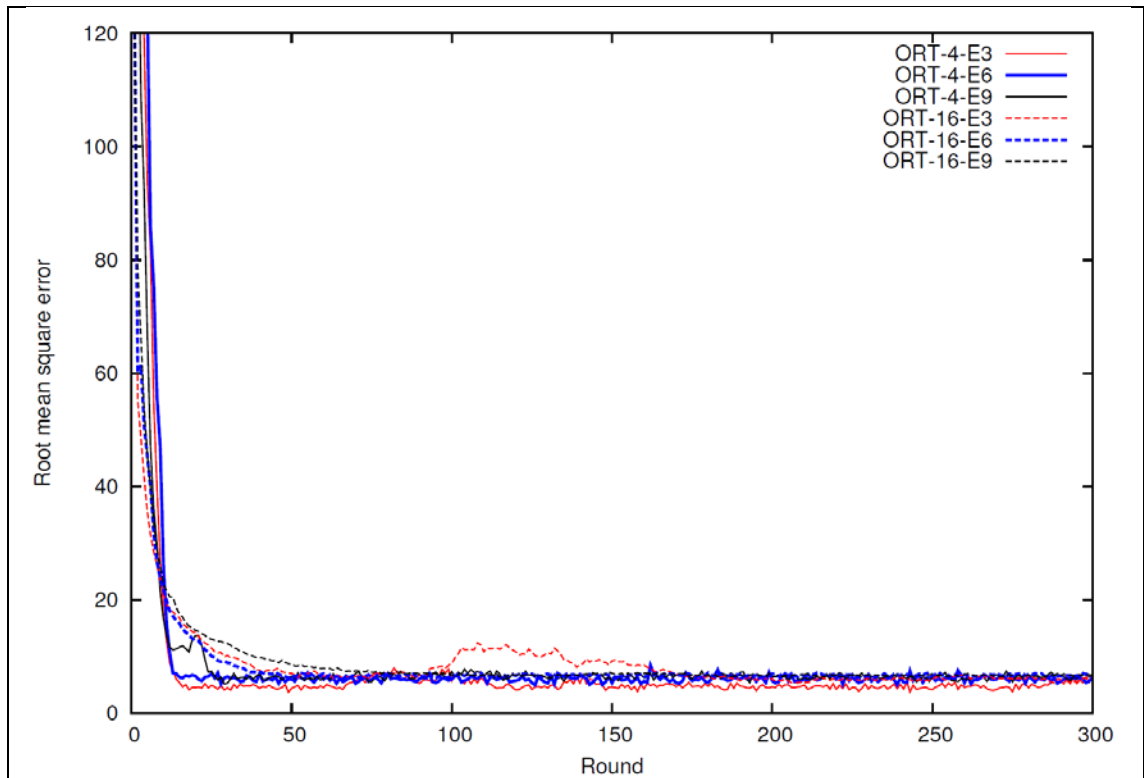
ภาพที่ 4.16 กราฟความผิดพลาดเทียบกับคาบเวลาในความหนาแน่นสูงที่ทำให้เป็นมาตรฐาน



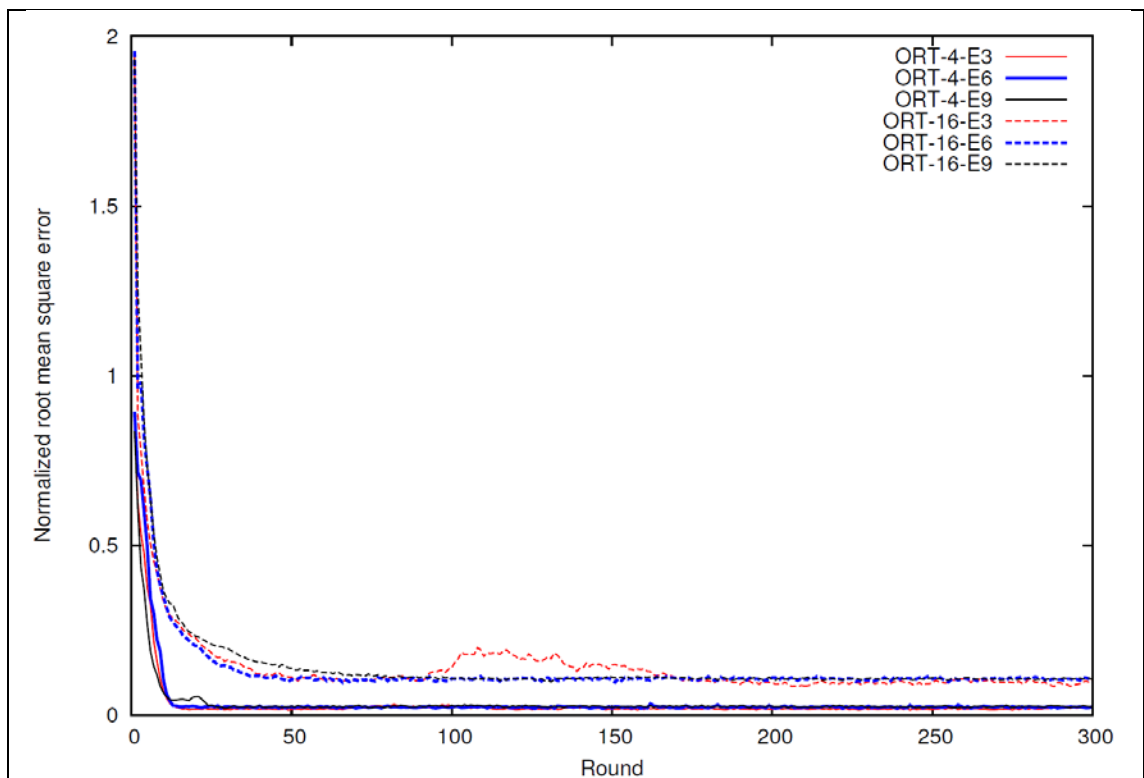
#### 4.4.3 การเปรียบเทียบค่าความผิดพลาดในความผิดพลาดที่รับได้ที่ต่างกัน

เพื่อเป็นการตรวจสอบว่าค่าความผิดพลาดที่รับได้มีผลต่ออัลกอริทึมหรือไม่เราได้ทำการทดสอบเปรียบเทียบอัลกอริทึมการไม่ประสานเวลาแบบจัดพื้นที่เปลี่ยนค่าความผิดพลาดที่รับได้ไปเป็น 3 ms และ 9ms โดยนำไปเทียบกับค่าเดิมคือ 6 ms ได้ผลการทดลองดังในภาพที่ 4.17-4.22

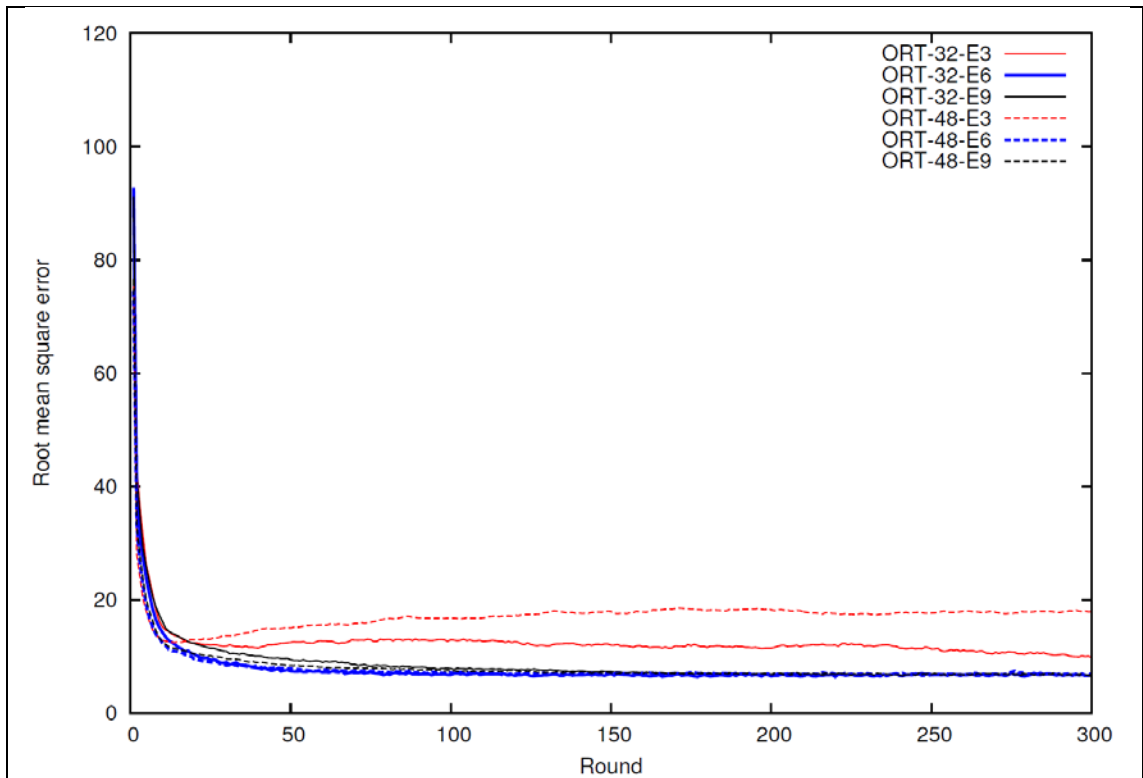
ค่าความผิดพลาดที่รับได้นั้นหมายถึงค่าที่ความผิดพลาดที่ยอมให้มีมากที่สุดในการจัดกลุ่มโดยหากค่านี้น้อยเกินไปโหนดก็จะไม่สามารถจับกลุ่มกันได้ดีเท่าที่ควรทำให้ค่าความผิดพลาดมีค่าสูง และหากมีค่ามากเกินไปโหนดก็จะมีการจัดกลุ่มได้ง่ายเกินไป ทำให้ค่าความผิดพลาดลดลงซ้ำเมื่อเทียบกับค่าที่เหมาะสมคือ 6 ms ที่คำนวณได้จากค่า Average Delay = 4ms และบวกเพิ่มไปอีก 2ms



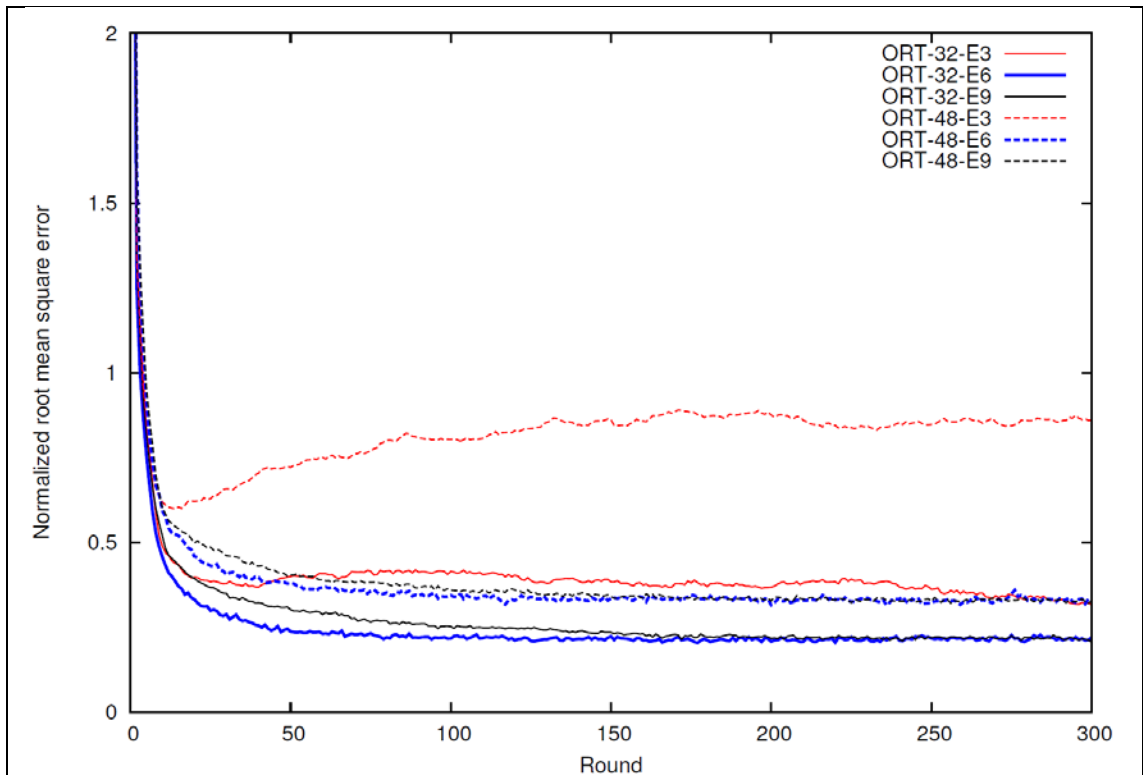
ภาพที่ 4.17 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นต่ำ



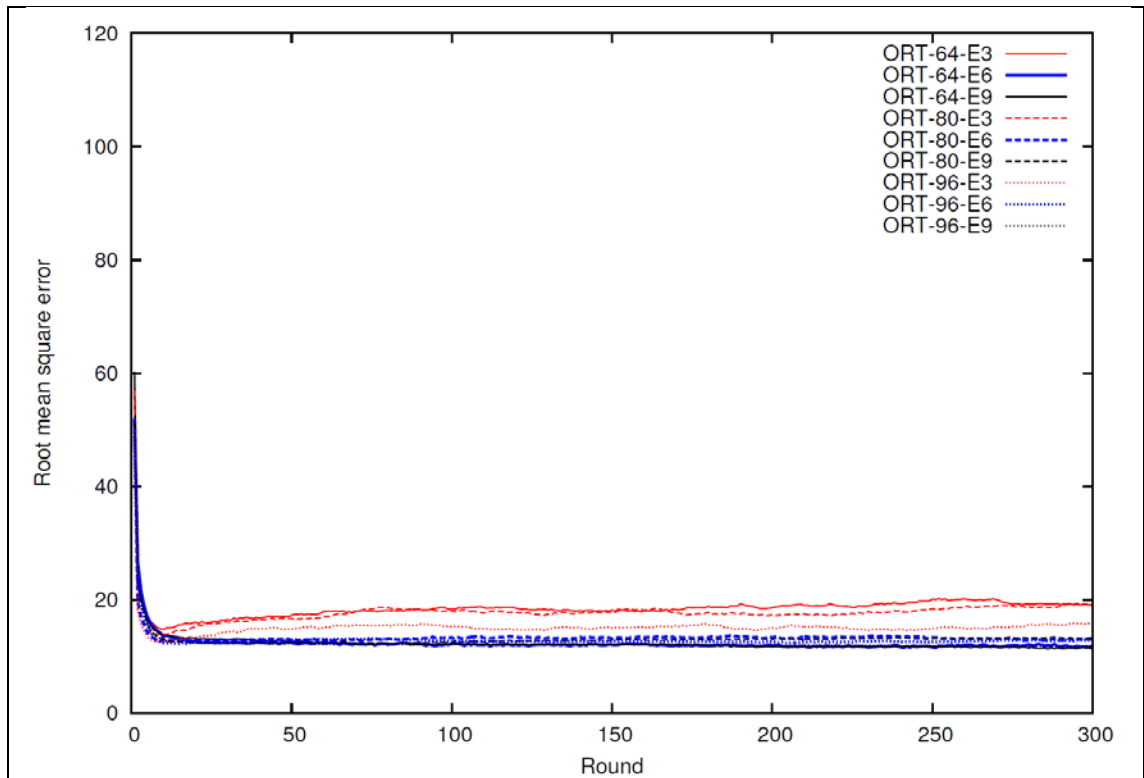
ภาพที่ 4.18 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นต่ำที่ทำเป็นมาตรฐาน



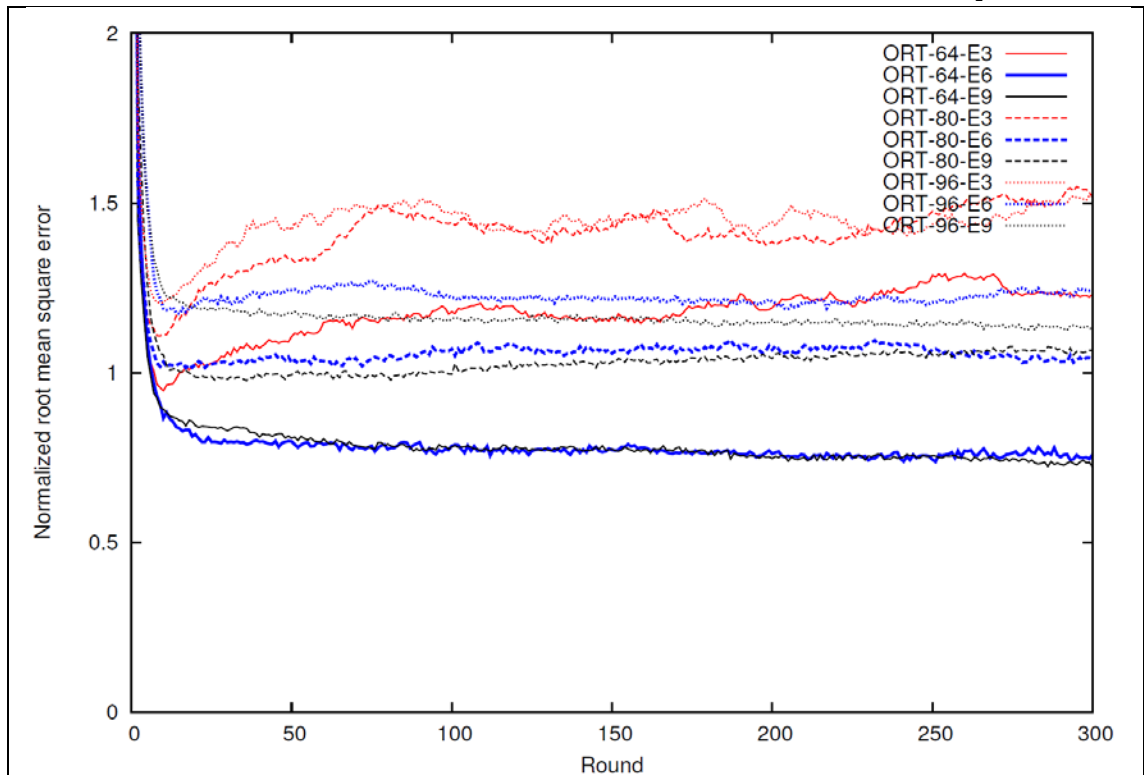
ภาพที่ 4.19 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นกลาง



ภาพที่ 4.20 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นกลางที่ทำให้เป็นมาตรฐาน



ภาพที่ 4.21 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นสูง

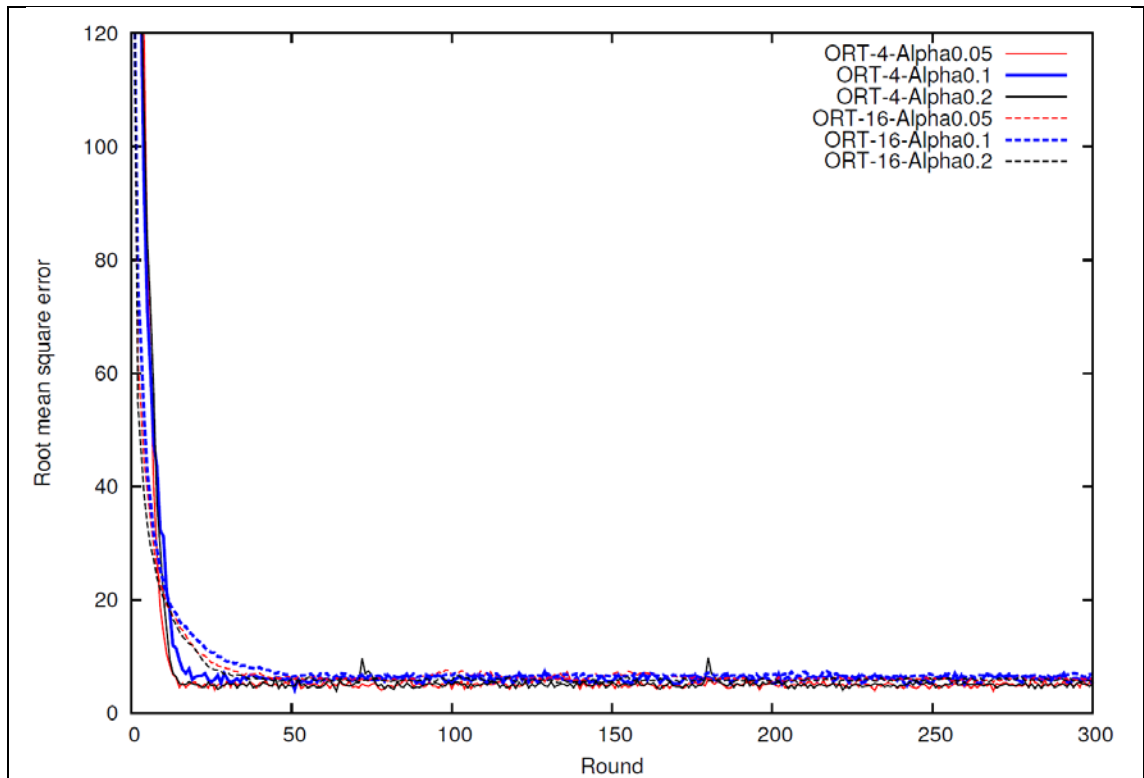


ภาพที่ 4.22 กราฟเปรียบเทียบความผิดพลาดที่รับได้ในความหนาแน่นต่ำที่ทำเป็นมาตรฐาน

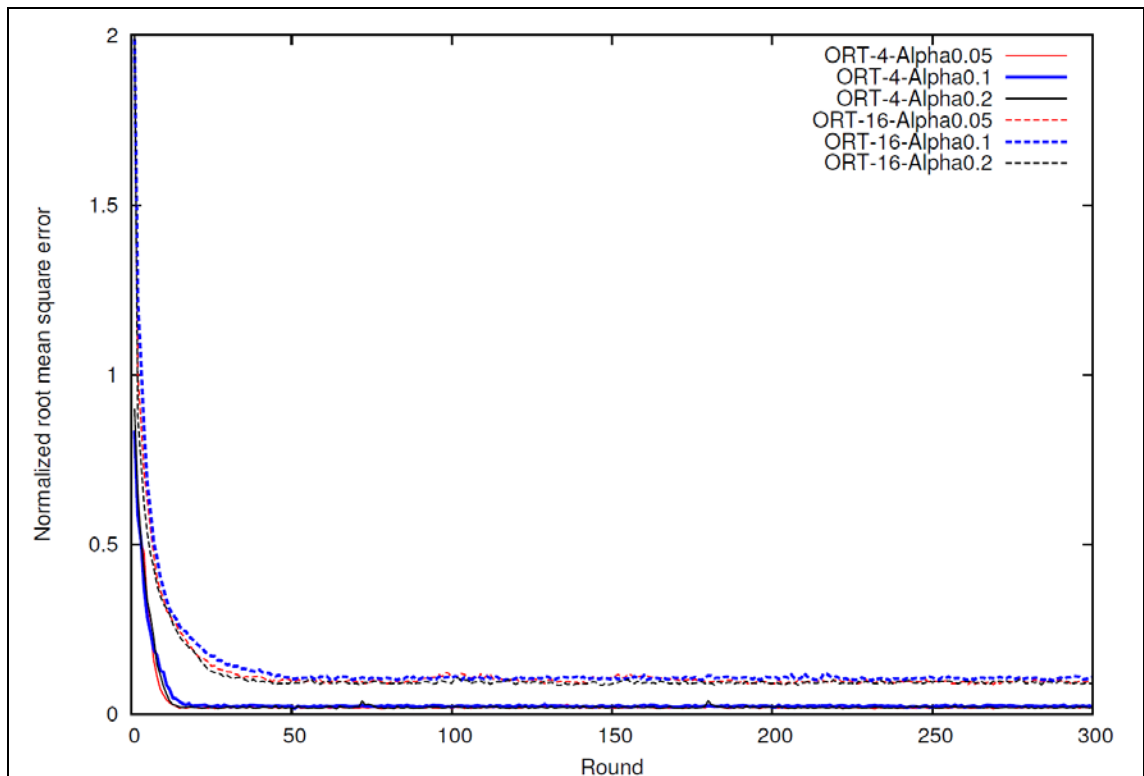
#### 4.4.4 การเปรียบเทียบค่าความผิดพลาดในค่าการปรับ $\alpha$ ที่ต่างกัน

เพื่อเป็นการตรวจสอบว่าค่าการปรับ  $\alpha$  มีผลต่ออัลกอริทึมหรือไม่เราได้ทำการทดสอบเปรียบเทียบอัลกอริทึมการไม่ประสานเวลาแบบจัดพื้นที่เปลี่ยนค่าการปรับ  $\alpha$  ไปเป็น 0.05 และ 0.2 โดยนำไปเทียบกับค่าเดิมคือ 0.1 ได้ผลการทดลองดังในภาพที่ 4.23-4.28

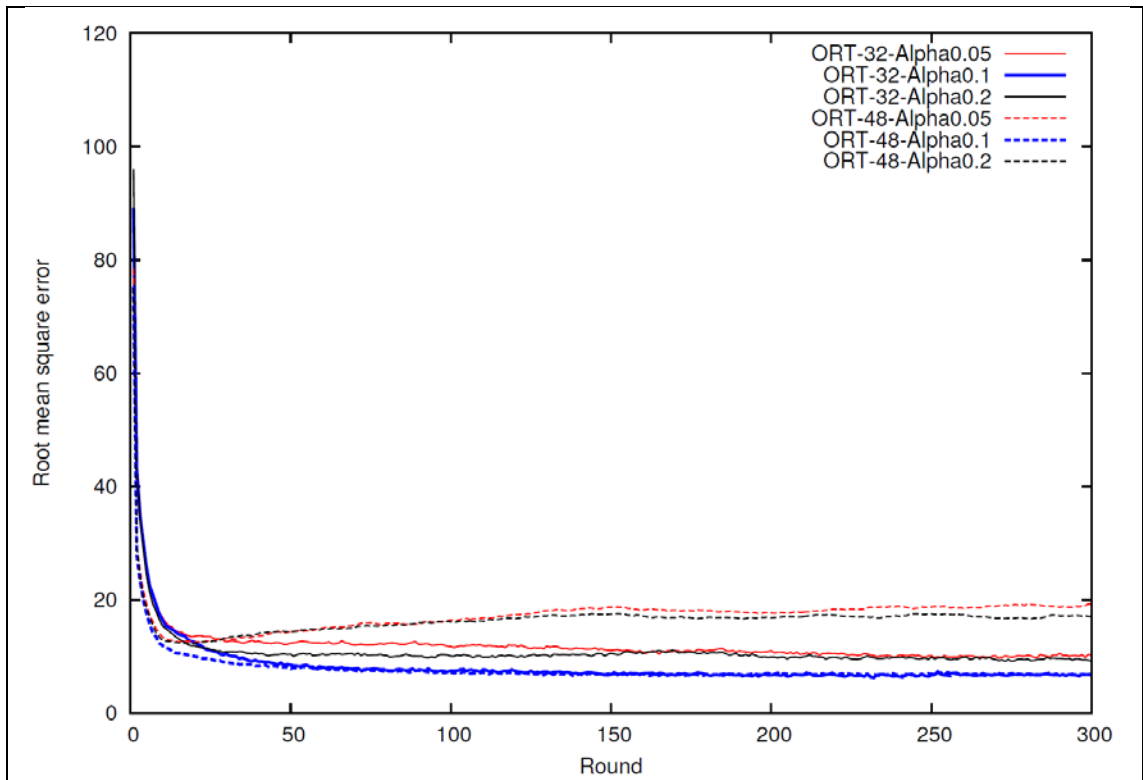
ค่าการปรับ  $\alpha$  คือค่าที่บ่งบอกว่าจะใช้ค่าที่คำนวณได้ใหม่ในอัตราส่วนเท่าใด โดยปกติเมื่อโหนดเริ่มจับกลุ่มได้หนึ่งฝั่งก็จะปรับลดลงมาจาก 0.95 เป็น 0.1 เราได้ทำการทดลองเพิ่มอีกสองค่าคือ 0.05 และ 0.2 เมื่อดูว่าค่าการปรับนี้มีผลต่ออัลกอริทึมการไม่ประสานเวลาแบบจัดพื้นที่หรือไม่ ซึ่งหากค่าการปรับมากไปก็จะทำให้โหนดปรับในปริมาณมากและถ้าหากน้อยไปก็จะทำให้การปรับนั้นเป็นไปได้ช้า สรุปจากการทดลองได้ว่าค่า 0.1 เป็นค่าที่ให้ ความผิดพลาดต่ำสุดและเป็นค่าที่เหมาะสมแล้ว



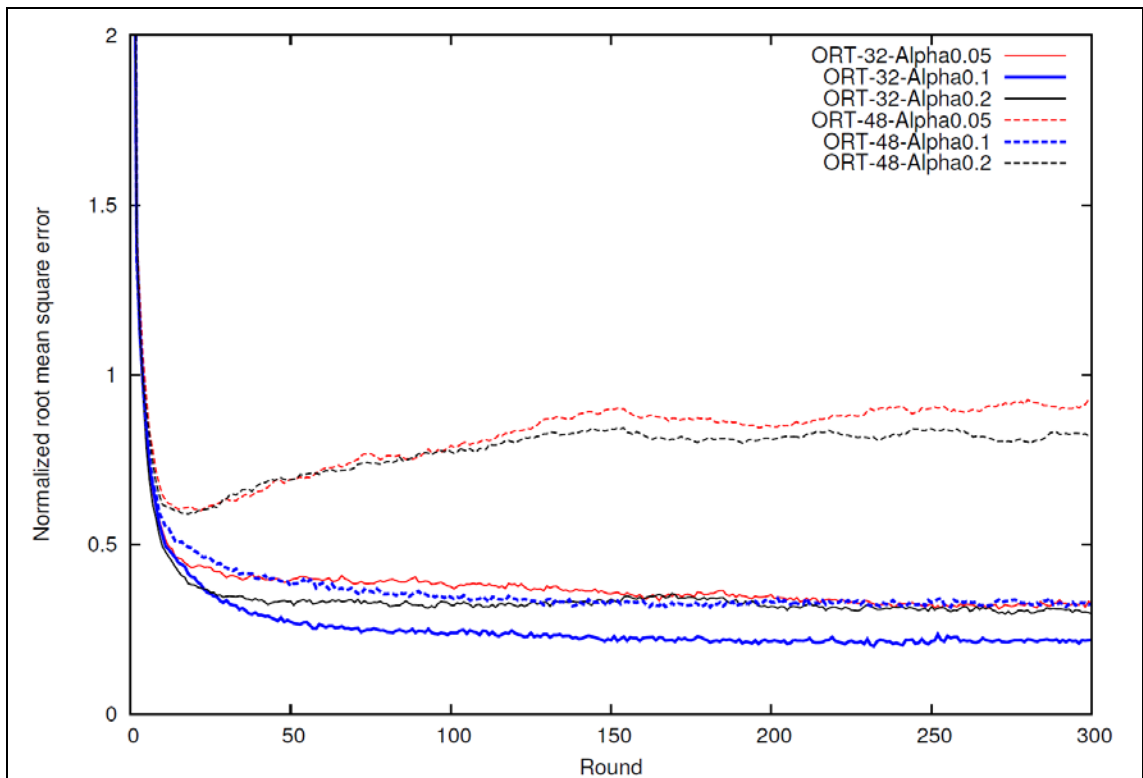
ภาพที่ 4.23 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นต่ำ



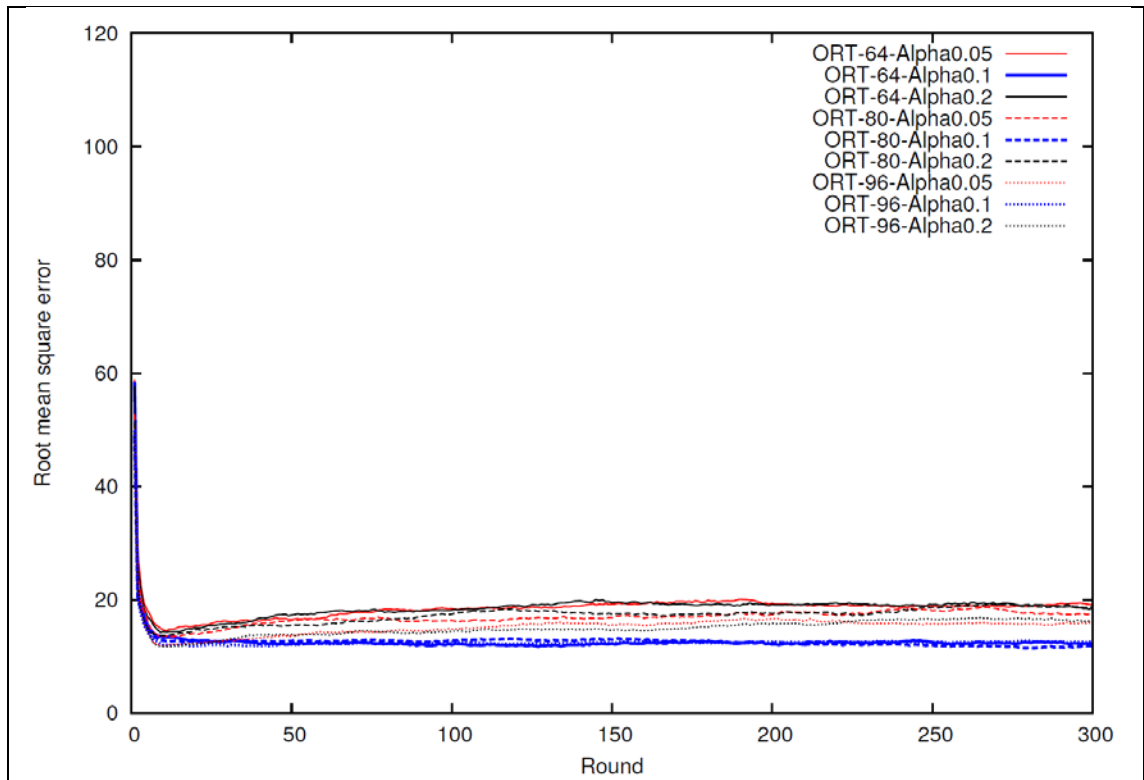
ภาพที่ 4.24 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นต่ำที่ทำให้เป็นมาตรฐาน



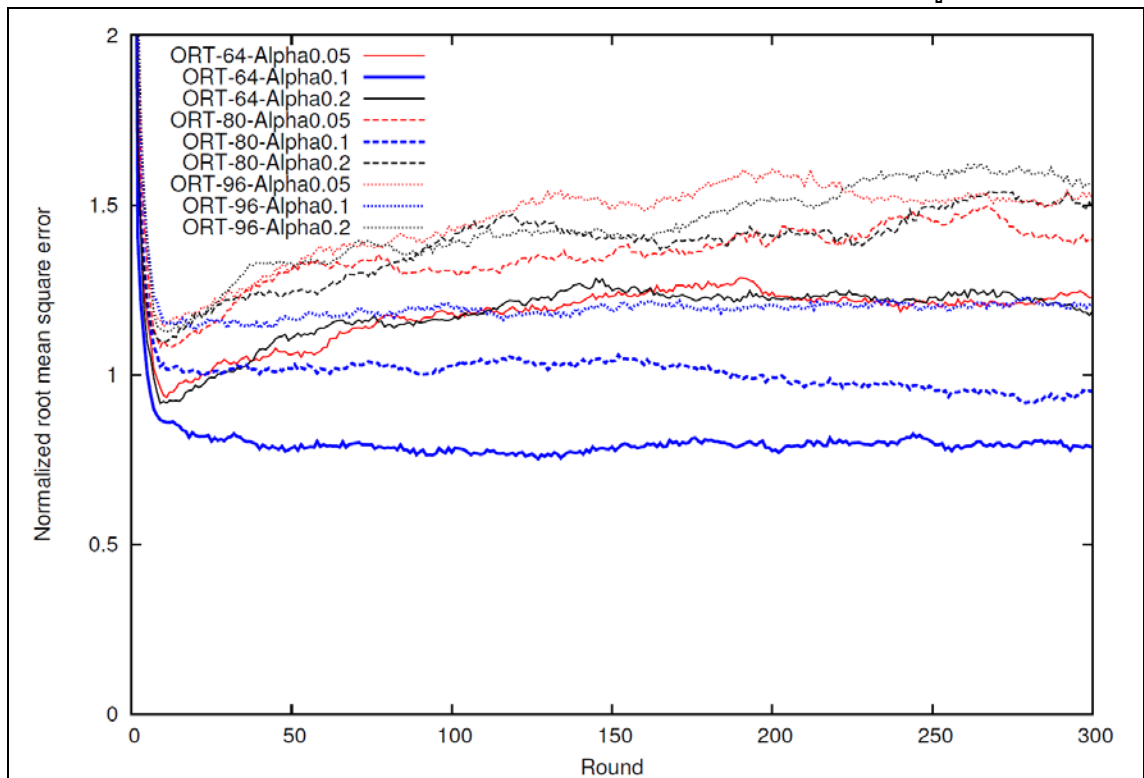
ภาพที่ 4.25 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นกลาง



ภาพที่ 4.26 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นกลางที่ทำให้เป็นมาตรฐาน



ภาพที่ 4.27 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นสูง



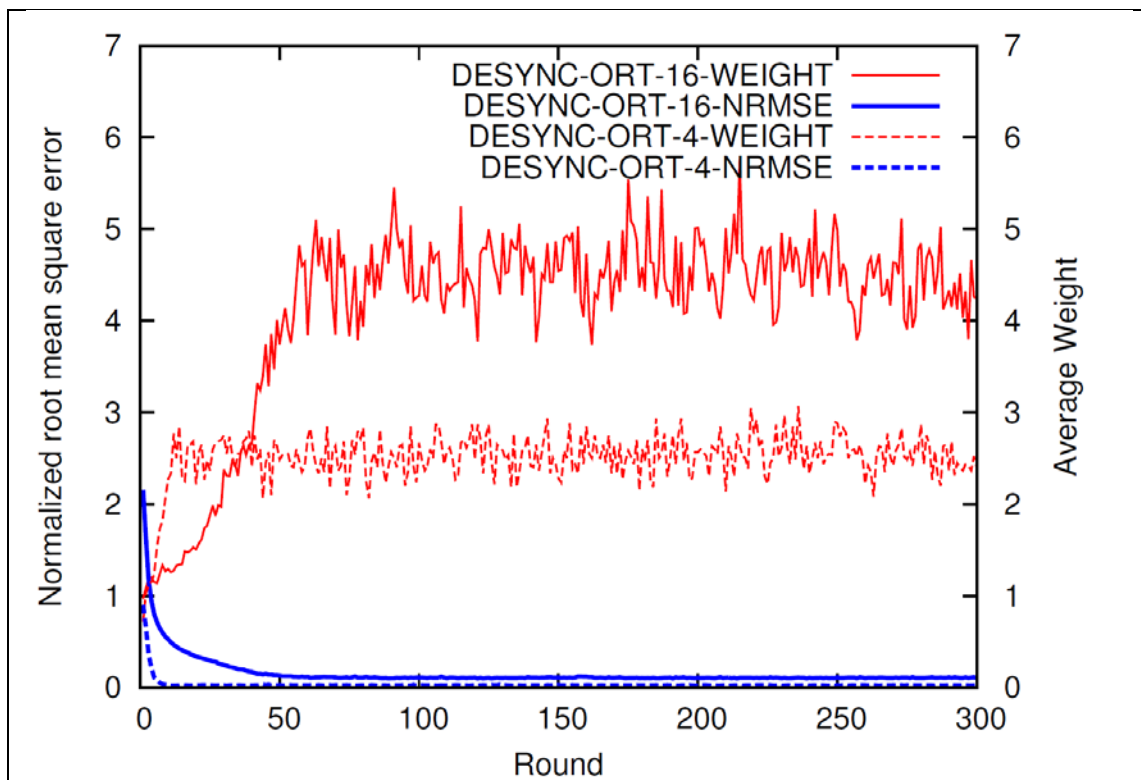
ภาพที่ 4.28 กราฟเปรียบเทียบค่าการปรับ  $\alpha$  ในความหนาแน่นสูงที่ทำให้เป็นมาตรฐาน



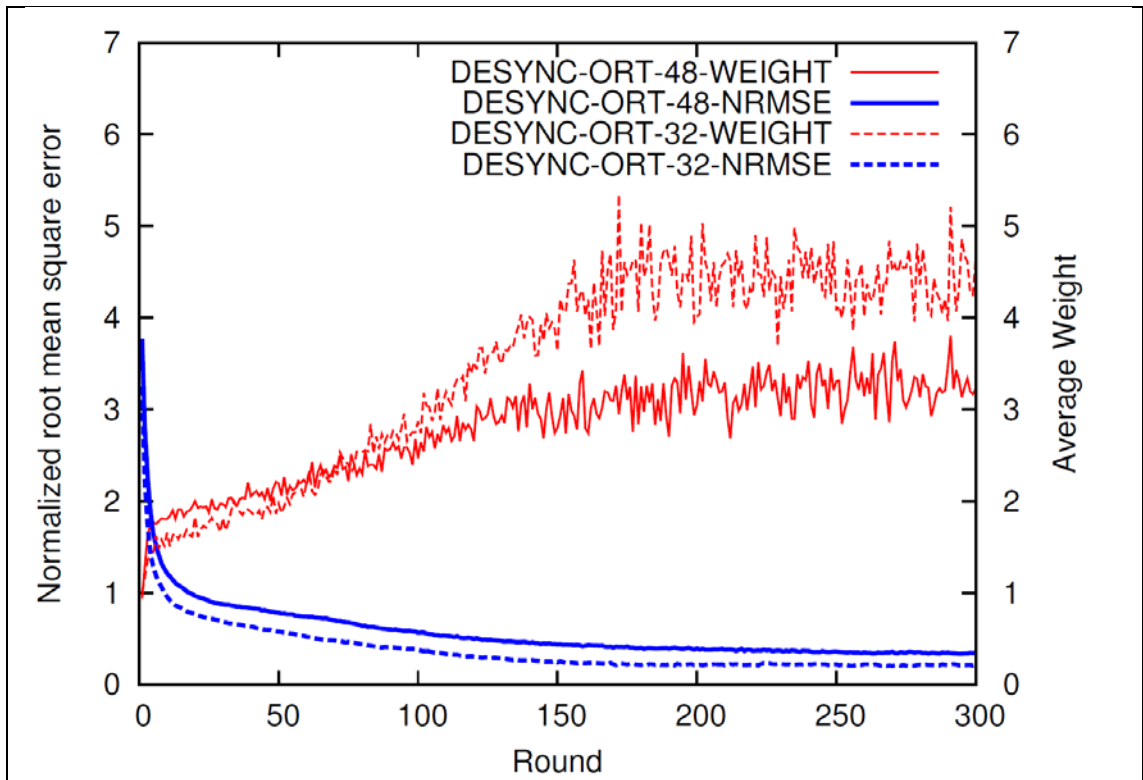
#### 4.4.5 การเปรียบเทียบค่าความผิดพลาดและน้ำหนัก

เพื่อตรวจสอบว่าอัลกอริทึมของเราทำงานได้ดีมากน้อยเพียงใด เราได้ทำการวัดน้ำหนักเฉลี่ยควบคู่ไปกับค่าความผิดพลาดได้ผลการทดลองดังในภาพที่ 4.29-4.31

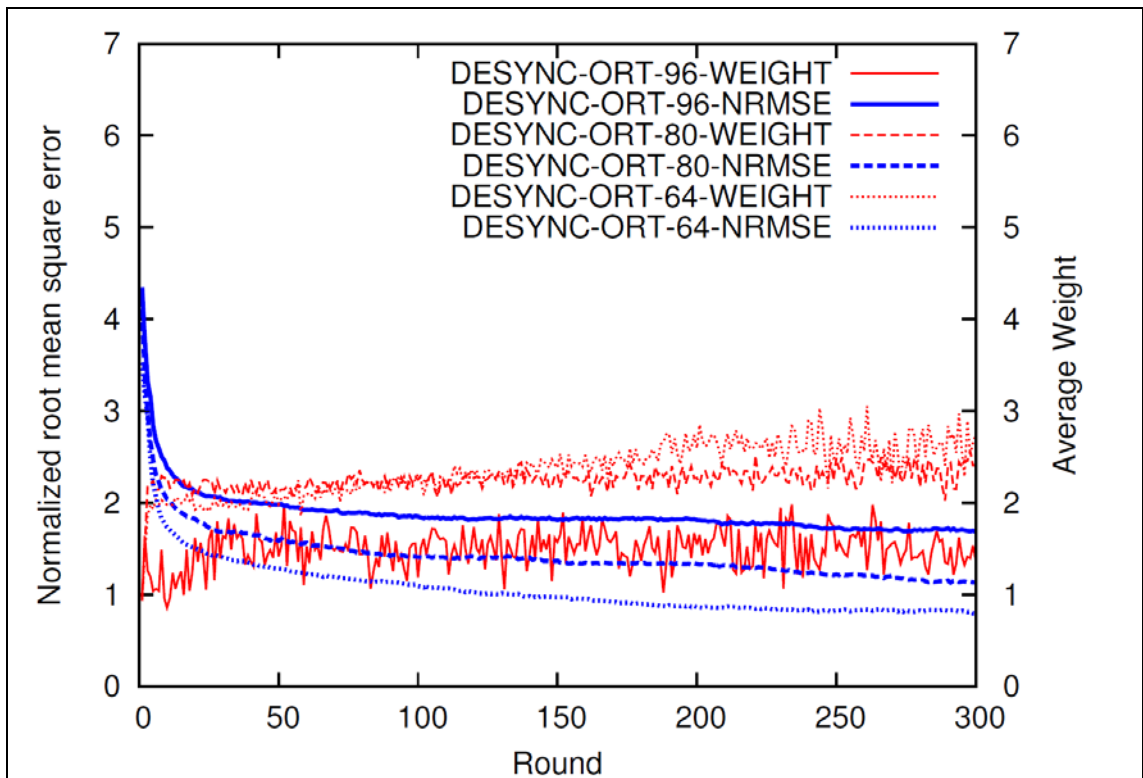
ซึ่งจะเห็นได้ว่าหากระบบมีความค่าความผิดพลาดที่ต่ำลง น้ำหนักเฉลี่ยก็จะเพิ่มมากขึ้น สังเกตว่าในช่วงความหนาแน่นต่ำระบบเข้าสู่การไม่ประสานเวลาได้รวดเร็วน้ำหนักก็จะเพิ่มขึ้นอย่างรวดเร็ว ในความหนาแน่นปานกลางกลาง น้ำหนักจะขึ้นมาช้ากว่าเดิม และในความหนาแน่นสูงนั้นแอลกอริทึมได้รับผลกระทบจากการที่ส่งข้อมูลชนกันสูงจึงทำให้น้ำหนักเพิ่มน้อย แต่อย่างไรก็ตามแอลกอริทึมการไม่ประสานเวลาแบบจัดฟันก็ยิ่งช่วยให้ระบบเข้าสู่สถานะการไม่ประสานเวลาที่ความผิดพลาดที่ต่ำกว่าแอลกอริทึมเดิม



ภาพที่ 4.29 กราฟเปรียบเทียบค่าน้ำหนักและความผิดพลาดในความหนาแน่นต่ำ



ภาพที่ 4.30 กราฟเปรียบเทียบค่าน้ำหนักและความผิดพลาดในความหนาแน่นกลาง



ภาพที่ 4.31 กราฟเปรียบเทียบค่าน้ำหนักและความผิดพลาดในความหนาแน่นสูง

## บทที่ 5

### สรุปผลการวิจัยและข้อเสนอแนะ

งานวิจัยนี้ได้นำเสนออัลกอริทึมการไม่ประสานเวลาแบบจัดฟันโดยนำเอาอัลกอริทึมการไม่ประสานเวลาเดิมมาเพิ่มเติมในส่วนของการจัดการการปรับที่เหมาะสมเข้าไป เพื่อให้สามารถเข้าสู่สถานะการไม่ประสานเวลาได้อย่างถูกต้องแม่นยำมากยิ่งขึ้น โดยอาศัยหลักการเช่นเดียวกับการจัดฟัน ซึ่ง อัลกอริทึมที่ออกแบบจะมีการทำงานเป็นแบบกระจาย และสามารถนำไปใช้งานได้ในระบบและสภาพแวดล้อมจริง จากผลการทดลองและวิจัยสามารถสรุปได้ดังนี้

#### 5.1 สรุปผลการวิจัย

- 1) อัลกอริทึมการไม่ประสานเวลาแบบเดิมนั้นไม่มีอัลกอริทึมควบคุมการปรับที่เหมาะสม ทำให้ความผิดพลาดของโหนดใดๆโหนดหนึ่งมีผลต่อทุกๆโหนดในระบบต่างจากอัลกอริทึมการไม่ประสานเวลาแบบจัดฟันซึ่งมีการควบคุมการปรับที่เหมาะสมเพื่อแก้ไขปัญหาดังกล่าวจึงทำให้ค่าความผิดพลาดต่ำกว่ามาก
- 2) ในกรณีที่ความหนาแน่นของโหนดสูงมาก อัลกอริทึมทั้งสองจะได้รับผลกระทบอย่างรุนแรงเนื่องจากข้อความที่ส่งนั้นมีอัตราการสูญหายค่อนข้างสูงแต่อย่างไรก็ตามอัลกอริทึมการไม่ประสานเวลาแบบจัดฟันก็ยังให้ผลลัพธ์ที่ดีกว่า
- 3) ค่าความผิดพลาดที่ยอมรับได้ ( $\epsilon$ ) นั้นหากมีค่าน้อยเกินไปก็จะทำให้น้ำหนักไม่สามารถจับกันได้เท่าที่ควรทำให้อัลกอริทึมไม่มีประสิทธิภาพ หากมีค่ามากไปก็จะทำให้จับกันง่ายเกินไป ทำให้ความค่าผิดพลาดลดลงได้ไม่ต่ำเท่าที่ควร
- 4) ค่าการปรับ  $\alpha$  ที่ใช้ปรับลดลงมานั้นหากมีค่าน้อยเกินไปก็จะทำให้ระบบสับสน หากมีค่ามากไปก็จะทำให้เกิดการปรับที่มากเกินไปทำให้ค่าความผิดพลาดสูง
- 5) เนื่องจากอัลกอริทึมการไม่ประสานเวลาแบบจัดฟันต้องใช้ข้อมูลของโหนดเพื่อนบ้านทุกตัวอาจทำให้ต้องใช้พลังงานในการรับข้อมูลมากกว่าอัลกอริทึมเดิม

#### 5.2 ข้อจำกัดและข้อเสนอแนะ

- 1) อัลกอริทึมที่นำเสนอเหมาะสำหรับการทำงานบนระบบเครือข่ายตัวรับรู้ที่มีความหนาแน่นในระดับกลาง เนื่องจากโอเวอร์เฮดด้านปริมาณข้อมูลที่จำเป็นต้องมีการรับข้อมูลเพื่อ

ใช้ในการคำนวณนั้นแปรผันตามจำนวนโหนดเพื่อนบ้าน ทำให้อัลกอริทึมนี้อาจยังไม่เหมาะสำหรับการใช้งานในระบบที่มีความหนาแน่นสูงมาก

2) อัลกอริทึมที่นำเสนอจะยังไม่ได้นำจำนวนน้ำหนักมาคิด ดังนั้นอัลกอริทึมนี้จึงยังคงไม่ใช่ อัลกอริทึมที่ดีที่สุด (Optimized algorithm) ที่จะสามารถปรับปรุงอัลกอริทึมการไม่ประสานเวลา แต่แนวคิดของวิทยานิพนธ์นี้ สามารถที่จะนำไป ใช้เป็นต้นแบบแนวคิดในการพัฒนา งานวิจัยการไม่ประสานเวลาอื่นๆต่อไปได้

3) โปรแกรมจำลอง TOSSIM ไม่สามารถจำลองการสูญหายของข้อมูลได้ ดังนั้นหากจะนำ อัลกอริทึมนี้ไปใช้งานจริงอาจต้องทำการทดสอบเมื่อหาค่าที่เหมาะสมในการจัดการข้อมูลที่สูญหาย

4) กระบวนการทำงานของอัลกอริทึมการไม่ประสานเวลาแบบจัดฟันจำเป็นต้องมีการส่ง ข้อมูลแบบกระจายไปยังทุกๆโหนดในแต่ละรอบ ซึ่งเมื่อพิจารณาเวลาที่นำเอาไปใช้งานใน สภาวะแวดล้อมจริงที่มีสี่รอบกวนอยู่ตลอดเวลาแล้ว กลุ่มข้อมูลบางส่วนอาจเกิดการสูญหาย ได้ ดังนั้นสำหรับในกรณีที่ระบบมีการสูญหายของข้อมูลในระหว่างรับส่งจะเป็นสิ่งจำเป็นที่ต้อง พิจารณาแก้ไขต่อไปสำหรับงานวิจัยในอนาคต

### 5.3 ปัญหาและอุปสรรค

1) เนื่องด้วยเทคโนโลยีระบบเครือข่ายไร้สายถือได้ว่า เป็นเทคโนโลยีใหม่ที่เกิดขึ้นมาไม่นานนัก สิ่งดีพิมพ์หรือข้อมูล สำคัญที่มีให้ศึกษาจึงยังคงมีอยู่น้อย ในประเทศไทย อีกทั้งวิธีการ และกระบวนการทำงานของระบบบางส่วนยังอยู่ในขั้นตอนกระบวนการวิจัย ข้อมูลส่วนใหญ่จะเป็นข้อมูลที่รับทาง อินเทอร์เน็ต หรือรายงานการวิจัยก่อนหน้าเกือบทั้งหมด ทำให้การศึกษา ให้เข้าใจและดำเนินการวิจัยนั้นทำได้ยาก

2) การขาดแคลนตัวอุปกรณ์จริงที่ใช้ทำการทดลองจริง เนื่องจากราคาที่ค่อนข้างสูงและยัง ไม่มีจำหน่ายภายในประเทศไทย ในการทดลองเบื้องต้นจึง ยังคงต้องทดลองบนโปรแกรมจำลอง ทำให้ยังไม่สามารถสรุปผลจากการทำงานบนระบบจริงได้

## รายการอ้างอิง

- [1] Levis, P., Lee, N., Welsh, M., and Culler, D. (2003). TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, Los Angeles, California, USA.
- [2] A. Patel, J. Degeysys, and R. Nagpal, "Desynchronization: The Theory of Self-Organizing Algorithms for Round-Robin Scheduling," in Self-Adaptive and Self-Organizing Systems, 2007. SASO '07. First International Conference on, 2007, pp. 87–96
- [3] J. Degeysys, I. Rose, A. Patel, and R. Nagpal, "DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks," in Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on, 2007, pp. 11–20.
- [4] Supasate Choochaisri, Kittipat Apicharttrisorn, Kittiporn Korprasertthaworn and Chalermek Intanagonwiwat, DRAWF: Decynchronicity with Artificial Force Field for Wireless Networks
- [5] J. Degeysys and R. Nagpal, "Towards Desynchronization of Multi-hop Topologies," in Self-Adaptive and Self-Organizing Systems, 2008. SASO '08. Second IEEE International Conference on, 2008, pp. 129–138.
- [6] Levis, P. (2003). TinyOS Tutorial, The tutorial introduces the basic concepts of TinyOS through a set of simple applications and exercises [Online]. Available from: <http://www.tinyos.net/tinyos-1.x/doc/tutorial/index.html> [10/8/2009]
- [7] Gay, D., Levis, P., Behren, R.V., Welsh, M., Brewer, E., and Culler, D. (2003). The nesC Language: A Holistic Approach to Networked Embedded Systems. Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation, San Diego, California, USA.

## ภาคผนวก

### บทความทางวิชาการ

ส่วนหนึ่งของงานวิทยานิพนธ์ที่ได้รับการตีพิมพ์เป็นบทความวิชาการในหัวเรื่อง “An Orthodontic Desynchronization Algorithm for Wireless Sensor Network” โดย พงษ์ภักดิ์ เตชะเลิศไพศาล, ศุภเสฏฐ์ ชูชัยศรี และ เฉลิมเอก อินทนากรวิวัฒน์ ในงานประชุมวิชาการ “The 13 IEEE International Conference on Communication and Technology(ICCT2011)” ซึ่งจัดขึ้น ณ เมืองจีหนาน( Jinan) มณฑลซางตง( Shandong) ประเทศจีน ระหว่างวันที่ 25-28 กันยายน 2554

## ประวัติผู้เขียนวิทยานิพนธ์

นายพงษ์ภักดิ์ เตชะเลิศไพศาล เกิดเมื่อวันที่ 23 มีนาคม พ.ศ. 2527 ที่จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้นจากโรงเรียน เซนต์ดอมินิก จังหวัด กรุงเทพมหานคร และมัธยมศึกษาตอนปลายจากโรงเรียนเตรียมอุดมศึกษา กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2549 และเข้าศึกษาในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2552