วิทยานิพนธ์

ระบบเครือข่ายมิเตอร์น้ำแบบแอ๊ดฮอกไร้สาย

WIRELESS AD-HOC DIGITAL FLOW METER NETWORK

นายเอกลักษณ์ เวศนารัตน์

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

พ.ศ. 2550

ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์)

ปริญญา

วิศวกรรมคอมพิวเตอร์	วิศวกรรมคอมพิวเตอร์
สาขา	**ภาควิชา**

เรื่อง ระบบเครือข่ายมิเตอร์น้ำแบบแอ๊ดฮอกไร้สาย

Wireless Ad-hoc Digital Flow Meter Network

นามผู้วิจัย นายเอกลักษณ์ เวศนารัตน์

ได้พิจารณาเห็นชอบโดย

ประธานกรรมการ ....

(ผู้ช่วยศาสตราจารย์ยอดเยี่ยม ทิพย์สุวรรณ์, Ph.D)

กรรมการ ....

(รองศาสตราจารย์อนันต์ ผลเพิ่ม, Ph.D)

กรรมการ ....

(ผู้ช่วยศาสตราจารย์จิตร์ทัศน์ ฝักเจริญผล, Ph.D)

หัวหน้าภาควิชา ....

(ผู้ช่วยศาสตราจารย์เขมะฑัต วิภาตะวนิช, Ph.D)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

....

(รองศาสตราจารย์วินัย อาจคงหาญ, M.A.)

คณบดีบัณฑิตวิทยาลัย

วันที่............ **เดือน**.................. **พ.ศ.**..............

วิทยานิพนธ์

เรื่อง

ระบบเครือข่ายมิเตอร์น้ำแบบแอ๊ดฮอกไร้สาย

Wireless Ad-hoc Digital Flow Meter Network

โดย

นายเอกลักษณ์ เวศนารัตน์

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์
เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์)
พ.ศ. 2550

เอกลักษณ์ เวศนารัตน์ 2550: ระบบเครือข่ายมิเตอร์น้ำแบบแอ๊ดฮอกไร้สาย
ปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมคอมพิวเตอร์) สาขาวิชาวิศวกรรมคอมพิวเตอร์
ภาควิชาวิศวกรรมคอมพิวเตอร์ ประธานกรรมการที่ปรึกษา: ผู้ช่วยศาสตราจารย์
ยอดเยี่ยม ทิพย์สุวรรณ์, Ph.D. 80 หน้า

ปัจจุบันในบางประเทศ ได้มีการพัฒนาระบบอ่านมิเตอร์อัตโนมัติ หรือ AMR (automatic meter reading) ขึ้นมาใช้งานเก็บข้อมูลจากมิเตอร์น้ำประปาตามที่อยู่อาศัย เพื่อประหยัดค่าใช้จ่ายและลดความผิดพลาดในการเก็บข้อมูล ระบบ wireless ad-hoc sensor network เป็นระบบที่เหมาะแก่การประยุกต์ใช้เพื่องานนี้ระบบหนึ่ง แต่ยังมีปัญหาเรื่องอัลกอริทึมการรวบรวมข้อมูลที่สิ้นเปลืองพลังงานและอายุใช้งานของระบบสั้น เพื่อแก้ปัญหาในการนำระบบเช่นนี้มาใช้ในการจัดเก็บข้อมูลจากมาตรน้ำประปาให้เหมาะสมกับสภาพแวดล้อมในประเทศไทย ผู้วิจัยได้ค้นคว้าเปรียบเทียบอัลกอริทึมที่ใช้เก็บข้อมูลในเครือข่ายเซนเซอร์ไร้สายที่เป็นที่นิยมใช้กันในปัจจุบัน พบว่า อัลกอริทึมที่เน้นการประหยัดพลังงานที่เหมาะสมกับเครือข่ายชนิดนี้ได้แก่ LEACH, PEGASIS และ PEDAP โดย PEDAP เป็นอัลกอริทึมที่ประหยัดพลังงานที่สุด แต่อัลกอริทึมดังกล่าวไม่สามารถนำมาใช้กับงานนี้ได้ทันที เนื่องจากถูกออกแบบมาใช้กับระบบที่ตำแหน่งของเซนเซอร์ทุกตัวต้องกำหนดตำแหน่งที่แน่นอนได้ และข้อมูลที่จัดส่งจะต้องสามารถรวมเข้าด้วยกัน(fused)เป็น packet ที่มีความยาวคงที่ได้ แต่ในระบบเครือข่ายจัดเก็บข้อมูลจากมาตรวัดน้ำประปาทั่วไปไม่สามารถทำเช่นนี้ได้

งานวิจัยนี้ได้พยายามพัฒนาอัลกอริทึมที่สามารถแก้ปัญหาที่กล่าวมาแล้ว โดยได้ตั้งชื่อระบบนี้ว่า EnForced Multiple Sub-Tree Algorithm (EFMST) ผลจากการจำลองระบบที่พัฒนาขึ้นพบว่า ระบบ EFMST สามารถยืดอายุการทำงานของระบบเครือข่ายมาตรวัดน้ำโดยรวมได้มากกว่าการใช้อัลกอริทึม PEDAP ภายใต้สภาพแวดล้อมเดียวกัน และผลจากการทดลองสร้างระบบด้วยบอร์ดเซนเซอร์ไร้สาย ยืนยันว่า วิธีการดังกล่าวสามารถประหยัดพลังงานของภาคส่งสัญญาณวิทยุของตัว first-level child node ได้มากกว่าอัลกอริทึม PEDAP ซึ่งจะช่วยเพิ่มรอบการทำงานของเครือข่ายรวบรวมข้อมูล(data gathering round) ให้สามารถทำงานตามจำนวนครั้งของการรวบรวมข้อมูลที่สูงกว่าได้

เนื่องจากจำนวนบอร์ดเซนเซอร์ไร้สาย ที่ใช้ในการทดลองวัดผลการใช้พลังงานส่งข้อมูลที่ได้ทำในงานวิจัยนี้มีจำนวนค่อนข้างน้อย และทำการทดสอบภายในอาคารที่มีระยะห่างระหว่างเครื่องรับส่งสั้นกว่าระยะใช้งานจริง และไม่มีสิ่งกีดขวางคลื่นวิทยุ ผลการทดลองที่ได้จึงอาจมีข้อผิดพลาดจากสภาวะแวดล้อมที่แท้จริงอยู่บ้าง งานต่อไปจึงควรมีการทดลองสร้างเครือข่ายเซนเซอร์ที่มีจำนวนมากขึ้น และนำไปทดสอบในบริเวณอาคารที่อยู่อาศัยจริงๆ เพื่อจะได้ข้อมูลการทำงานของระบบที่ถูกต้องแม่นยำยิ่งขึ้น

____________________ ____________________ ___ / ___ / ___

ลายมือชื่อนิสิต ลายมือชื่อประธานกรรมการ

Akekaluck Wesnarat 2007: Wireless Ad-hoc Digital Flow Meter Network. Master of Engineering (Computer Engineering), Major Field: Computer Engineering, Department of Computer Engineering. Thesis Advisor: Assistant Professor Yodyium Tipsuwan, Ph.D. 80 pages.

Automatic meter reading systems have been developed for data gathering from water meter networks in residential areas in some countries. Ad-hoc sensor network methods have been adapted for such applications. An underlined problem in implementing a wireless network system for this purpose is the power consumption of a wireless unit that entails the network short life and high cost. In order to address problems of implementing wireless-sensor networks for data gathering from residential water meters according to the environment and standard in Thailand, Researcher have investigated existing data gathering algorithms and found that energy sufficient algorithms possible for implementing a water meter network are LEACH, PEGASIS and PEDAP. Among them, PEDAP is the most efficient method. However, all of these algorithms can not be used directly on the targeted system since they need exact locations of the sensors and data from all sensors must be able to be fused. This requirement is not suitable for water meter networks.

The objective of this research is to develop an algorithm that can solve to all previous mentioned problems. This algorithm is denoted as EnForced Multiple Sub-Tree Algorithm (EFMST). A simulation result shows that EFMST is capable to extend the water meter network life time longer than PEDAP under the same environment. In addition, the experimental result from an implementation of EFMST algorithm using wireless-sensor boards shows that this algorithm can save battery power on the transmitting unit of first-level child nodes more than PEDAP. This implires that the network life-time at a higher number of data gathering rounds can be achived.

Due to a limited number of sensor boards used in the implementation of this algorithm, the result of power consumption measured may posses errors. A further experiment with real water meter sensor boards and the installation in actual residential area should be conducted. The number of sensor node to be tested should be increased to yield more accurate testing results.

________________________ ________________________ ___ / ___ / ___

Student's signature Thesis Advisor's signature

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จได้ด้วยความกรุณาช่วยเหลือ และความร่วมมือจากบุคคลหลายท่าน บุคคลสองท่านแรกที่ผู้วิจัย ขอกราบขอบพระคุณเป็นอย่างสูงในความกรุณาคือ ผศ.ดร.ยอดเยี่ยม ทิพย์สุวรรณ ประธานกรรมการที่ปรึกษาวิทยานิพนธ์ และคุณพ่อณรงค์ เวศนารัตน์ ที่กรุณาถ่ายทอดความรู้ประสิทธิประสาทวิชา และให้คำปรึกษาแนะนำตลอดจนให้ความเมตตาช่วยเหลือ ปรับปรุงแก้ไขข้อบกพร่องต่าง ๆ ในการทำวิทยานิพนธ์จนเสร็จสมบูรณ์ ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาเป็นอย่างยิ่งและขอจดจำคำแนะนำคำสั่งสอนอันมีค่าเพื่อนำไปใช้ในชีวิตในอนาคตต่อไป ขอกราบขอบพระคุณ รศ.ดร.อนันต์ ผลเพิ่ม ผศ.ดร.จิตรภัทร์ ฝักแก้ว และ รศ.ดร.มงคล รักษาพัชรวงศ์ ที่ให้ความกรุณาเป็นคณะกรรมการสอบและแก้ไขข้อบกพร่องในการทำวิทยานิพนธ์ให้มีความสมบูรณ์ยิ่งขึ้น นอกจากนี้ผู้วิจัยขอกราบขอบพระคุณคณาจารย์ทุกท่านที่ให้การอบรมสั่งสอน ตลอดจนปลูกฝังความรู้ที่มีคุณค่าให้กับผู้วิจัย ขอขอบคุณเจ้าหน้าที่คณะวิศวกรรรมคอมพิวเตอร์ สำหรับน้ำใจที่ให้ความช่วยเหลือทุกสิ่งทุกอย่างและการประสานงานจนงานวิจัยนี้สำเร็จได้ด้วยดี

ขอขอบคุณ บริษัทอาซาฮี-ไทย อัลลอย ที่ได้กรุณาให้เวลาในการทำวิจัย ขอขอบคุณ คุณรวีวรรณ เล็กศรีสกุล ที่ให้ความรัก ความปรารถนาดี ให้ความช่วยเหลือและให้กำลังใจกับผู้วิจัยในหลายๆ เรื่อง และขอบคุณสำหรับมิตรภาพ คำแนะนำดี ๆ ที่ได้รับจาก พี่ๆ และน้อง ๆ ในห้อง Lab Intelligent Mechatronics Laboratory (IML) ทุกคนที่ทำให้การทำวิทยานิพนธ์ให้สำเร็จลุล่วงไปได้ เหนือสิ่งอื่นใดผู้วิจัยกราบขอบพระคุณบุคคลผู้มีพระคุณที่สุดในชีวิตคือ คุณพ่อณรงค์ และคุณแม่ประหยัด เวศนารัตน์ที่ ให้ความรัก การอบรมเลี้ยงดู และ ความห่วงใยพร้อมกับกำลังใจในทุกเรื่องเป็นอย่างดี และขอบคุณพี่สาวที่แสนดีคือ คุณพิมพ์ใจ เวศนารัตน์ สำหรับความรักและกำลังใจที่ช่วยสนับสนุนในการทำวิทยานิพนธ์ในครั้งนี้

สุดท้ายนี้ผู้วิจัยขอระลึกถึงความปรารถนาดีของทุกท่านที่กล่าวมาข้างต้นและผู้ที่เกี่ยวข้องที่มิได้เอ่ยนาม และหากงานวิจัยครั้งนี้มีประโยชน์กับบุคคลหรือองค์การใดก็ตามผู้วิจัยขออุทิศส่วนกุศลให้กับ คุณย่าของข้าพเจ้า แม่อุ๊ยจันทร์ติ๊บ สิงห์แก้ว ผู้ล่วงลับ สู่สุขติ อยู่ในภพ ภูมิที่ดี

เอกลักษณ์ เวศนารัตน์

ตุลาคม 2550

สารบัญ

หน้า

สารบัญ	(1)
สารบัญตาราง	(2)
สารบัญภาพ	(3)
คำอธิบายสัญลักษณ์และคำย่อ	(6)
คำนำ	1
วัตถุประสงค์	2
การตรวจเอกสาร	3
อุปกรณ์และวิธีการ	22
อุปกรณ์	22
วิธีการ	26
ผลการทดลองและการอภิปรายผลที่ได้	50
ผลการทดลอง	50
วิจารณ์ผลการวิจัย	59
สรุปและข้อเสนอแนะ	61
สรุป	61
ข้อเสนอแนะ	61
แนวทางการพัฒนาต่อ	63
เอกสารและสิ่งอ้างอิง	65
ภาคผนวก	69

สารบัญตาราง

ตารางที่ **หน้า**

1	ค่าพลังงานที่วัดได้จากการ experiment ที่วางตำแหน่งของ Sensor แบบ Grid	53
2	ค่าพลังงานที่วัดได้จากการ experiment ที่วางตำแหน่งของ Sensor แบบ Random	55
3	ค่าพลังงานที่วัดได้จากการ experiment ที่วางตำแหน่งของ Sensor แบบ Diagonal	57

สารบัญภาพ

ภาพที่		หน้า
1	การวางระบบแบบเส้นตรงแบบง่าย	6
2	Dynamic clustering ใน LEACH รอบที่ 1 และ 2 โดยจะมีการเลือก head ที่ต่างกันในแต่ละรอบเพื่อกระจายการใช้พลังงานของตัว head ในการส่งข้อมูลไปยัง base station	8
3	แสดงการสร้างเส้นทางแบบ (a) address-centric routing และ (b) data-centric routing	8
4	ระบบการทำงานของ directed diffusion	11
5	การสร้าง Chain ใน PEGASIS	12
6	การรวมข้อมูลใน Chain	13
7	แสดงการสร้าง Chain แบบ 2-level	14
8	ปัญหาข้อมูลซ้ำกันและข้อมูลบางส่วนเหมือนกันในการส่งข้อมูลแบบ flooding	15
9	การทำงานของโปรโตคอล SPIN	16
10	อัลกอริทึมของ GEAR	18
11	การกระจายข้อมูลภายใน Region R	19
12	การสิ้นสุดการกระจาย packet ภายใน region R	19
13	รูปแบบการเลือกเส้นทางของ PEDAP ใน simple network	20
14	Tmote Sky ด้านบน	23
15	Tmote Sky ด้านล่าง	24
16	แสดงการเชื่อมต่อแบบ Provide แทนด้วยสี่เหลี่ยมจตุรัสลายแนวนอน และการเชื่อมต่อแบบ Used แทนด้วยสี่เหลี่ยมจตุรัสสีดำ	25
17	วิธีการเก็บข้อมูลจากมิเตอร์น้ำทั่วไป	26
18	โครงสร้างของ Tree	29
19	การเลือก first-level child nodes	32
20	ขั้นตอนในการสร้าง Tree ในงานวิจัย	34
21	แสดงตัวอย่างของ fist-level child node และเส้นทางการส่งข้อมูลแบบกริด (a) PEDAP spanning tree algorithm. (b) วิธีการของงานวิจัย	38

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

22 แสดงลักษณะการว่างตำแหน่งแบบวงกลม 40

23 รูปแบบการวางตำแหน่งที่ใช้ในการทดลอง 42

24 ตัวอย่างเส้นทางที่ได้จากการ Simulation ด้วยวิธีการสร้างของ PEDAP 44

25 ตัวอย่างเส้นทางที่ได้จากการ Simulation ด้วยวิธีการ Enforced Multiple sub-tree 45

26 การวางตำแหน่ง Sensor แบบ Grid 46

27 ตัวอย่างการวางตำแหน่ง Sensor แบบ Random 47

28 การวางตำแหน่ง Sensor แบบ Diagonal 47

29 โปรแกรมที่ใช้ในการเก็บข้อมูลและแสดงเส้นทางที่สร้างได้ 48

30 ตัวอย่างเส้นทางที่ได้จากการสร้างเส้นทางจากวิธีการ Enforce-Multiple Sub-Tree 49

31 ผลที่ได้จากการ Simulation เมื่อระยะทางมากที่สุดในการส่งข้อมูลคือ 200 m, จำนวนรอบที่ต้องการคือ 60 รอบ โดยทำการเปลี่ยนแปลงค่าพลังงานเริ่มต้นที่ 1, 0.5 และ 0.25 J 50

32 ผลที่ได้จากการ Simulation เมื่อระยะทางมากที่สุดในการส่งข้อมูลอยู่ที่ 200 m., พลังงานเริ่มต้นของแต่ละมิเตอร์คงที่ที่ 1 J แต่ทำการเปลี่ยนแปลงจำนวนรอบที่ต้องการเป็น 60, 250 และ 500 รอบ 51

33 ผลการทดลองการวางตำแหน่ง Sensor แบบ Grid 52

34 ผลการทดลองการวางตำแหน่ง Sensor แบบ Random 54

35 ผลการทดลองการวางตำแหน่ง Sensor แบบ Diagonal 56

36 การเพิ่มตัว repeater เข้าในระบบ 62

ภาพผนวกที่

1 Module ที่สร้างเพิ่มขึ้นใน TinyOS 71

2 วิธีการตรวจสอบ Packet ในการส่ง Packet ซ้ำ 72

3 Header และ Type ของ packet ที่ใช้ในการสร้าง, รายงาน และเก็บ ข้อมูล 73

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

ภาพที่		หน้า
4	สถานะการทำงานของ node ในการสร้างเส้นทางใน EFMST	74
5	สถานะการทำงานของ node ในการเก็บข้อมูลและรายงานเส้นทางใน EFMST	75
6	สถานะการทำงานของ base station ใน EFMST	76
7	สถานะการทำงานของ node ในการสร้างเส้นทางใน MST	77
8	สถานะการทำงานของ node ในการรายงานเส้นทางใน MST	78
9	สถานะการทำงานของ node ในการเก็บข้อมูลใน MST	79
10	สถานะของ base station ใน MST	80
11	วงจรวัดกระแสของ root node ที่ใช้ในการทดลองกับบอร์ดจริง	80

คำอธิบายสัญลักษณ์และคำย่อ

BS	=	Base station
RF	=	Radio frequency
PDA	=	Personal Digital Assistants
LCD	=	Liquid Crystal Display
FSK	=	Frequency shift keying
FD	=	Frequency Division
FH	=	Frequency Hopping
SMS	=	Short Message Service
GPRS	=	General Packet Radio Service
LEACH	=	Low-Energy Adaptive Clustering Hierarchy
E_{elec}	=	พลังงานที่ใช้ในส่วนของภาคอิเล็กทรอนิค
e_{amp}	=	พลังงานที่ใช้ในภาคขยายสัญญาณวิทยุ
PEGASIS	=	Power-efficient Gathering in Sensor Information System
SPIN	=	Sensor Protocols for Information via Negotiation
GEAR	=	The Geographical and Energy-Aware Routing
E_{Tx}	=	พลังงานที่ใช้ในการส่งข้อมูลจำนวน 1 hop
E_{Rx}	=	พลังงานที่ใช้ในการรับข้อมูลจำนวน 1 hop
k	=	จำนวนบิตที่ส่งออกไป
d	=	ระยะทางไกลที่สุดที่สามารถส่งข้อมูลระหว่างตัวรับและตัวส่งได้
PEDAP	=	Power efficient data aggration protocol
First-level child nodes	=	มิเตอร์ที่อยู่ใกล้กับ Base station และสามารถส่งและรับข้อมูลกับ Base station ได้
EFMST	=	Enforce Multiple Sub-Tree
E_P	=	พลังงานเริ่มต้นของ nodes
R_D	=	จำนวนครั้งที่ต้องการในการเก็บข้อมูลมายัง Base station
N_{max}	=	จำนวน child node ที่มากที่สุดที่สามารถส่งข้อมูลทั้งหมดออกไปได้
N_T	=	จำนวน nodes ทั้งหมด
N_R	=	จำนวน First-level child nodes

คำอธิบายสัญลักษณ์และคำย่อ (ต่อ)

m_{Oi}	=	จำนวน child node ในเส้นทางที่มี node i เป็น First-level child node ในระบบที่ใช้วิธีของงานวิจัยนี้
m_{Ti}	=	จำนวน child node ในเส้นทางที่มี node i เป็น First-level child node ในระบบที่ใช้วิธีของ PEDAP MST
L_{SO}	=	Life Time ของระบบที่ใช้วิธีของงานวิจัยนี้
L_{ST}	=	Life Time ของระบบที่ใช้วิธีของ PEDAP MST
L_i	=	Life Time ของ node i
E_p^i	=	พลังงานเริ่มต้นของ node i
E_{Tx}^i	=	พลังงานที่ใช้ในการส่งข้อมูลของ node i
E_{Rx}^i	=	พลังงานที่ใช้ในการรับข้อมูลของ node i
d_{Ti}	=	ระยะทางที่มากที่สุดที่ node i สามารถส่งข้อมูลได้ในระบบที่ใช้วิธีของ PEDAP MST
d_{Oi}	=	ระยะทางที่มากที่สุดที่ node i สามารถส่งข้อมูลได้ในระบบที่ใช้วิธีของงานวิจัยนี้

ระบบเครือข่ายมิเตอร์น้ำแบบแอ๊ดฮอกไร้สาย

Wireless Ad-hoc Digital Flow Meter Network

คำนำ

งานวิจัยนี้มีเป้าหมายเพื่อพัฒนาระบบอ่านปริมาณการใช้น้ำประปาโดยอัตโนมัติ เพื่อแก้ปัญหาข้อมูลผิดพลาดที่เกิดจากการอ่านค่าโดยคน ลดค่าใช้จ่ายในการจ้างคนไปอ่านข้อมูลและลดระยะเวลาตั้งแต่การเก็บข้อมูลจนถึงเวลาที่วางใบเก็บเงินที่ผู้ให้บริการประสบอยู่ในปัจจุบัน โดยระบบที่พัฒนาขึ้นจะต้องมีราคาถูก รวดเร็วและคงทนต่อความผิดพลาด ในเกณฑ์ที่สามารถนำไปใช้ได้ในทางปฏิบัติได้ เทคโนโลยีหลักที่นำมาใช้ในการศึกษาและพัฒนาระบบนี้ คือเทคโนโลยีทางด้าน sensor network ซึ่งในการนำเอามิเตอร์พร้อม sensor ไปใช้งานนั้นจำเป็นต้องใช้แหล่งพลังงานอย่างใดอย่างหนึ่ง ซึ่งแหล่งพลังงานจาก battery เป็นสิ่งที่ใช้ได้ง่ายที่สุดในขณะนี้ ทำให้ระยะเวลาการใช้งานถูกจำกัดด้วยปริมาณพลังงานที่ได้จาก battery เป็นหลัก

พลังงานที่สูญเสียไปให้กับ RF module ยังแบ่งได้เป็น 2 ส่วนด้วยกันคือ ส่วนของวงจรอิเล็กทรอนิค และส่วนของการรับส่งสัญญาณ RF เพื่อรับส่งข้อมูล ซึ่งงานวิจัยนี้จะพยายามลดพลังงานที่สูญเสียไปในการรับส่งข้อมูล เพื่อให้ระยะเวลาที่มิเตอร์สามารถทำงานได้ยาวนานขึ้น โดยทางผู้จัดทำได้นำเทคโนโลยี sensor network มาศึกษา จากนั้นทำการวิจัยและพัฒนาโปรโตคอลและอัลกอริทึมให้เหมาะสม เพื่อให้สามารถนำไปแก้ไขปัญหาดังที่กล่าวมาแล้วได้ ซึ่งในงานวิจัยนี้ได้ออกแบบวิธีการที่เรียกว่า Enforce multiple sub-tree (EFMST) เพื่อใช้การยืดอายุการใช้งานของ battery ภายในตัวมิเตอร์ และได้ทำการเปรียบเทียบกับวิธีการของ sensor network ที่มีอยู่แล้วในปัจจุบัน (PEDAP) จากผลการทดลองสามารถบอกได้ว่าวิธีการที่คิดขึ้นนี้ ช่วยลดพลังงานที่สูญเสียไปในตัวมิเตอร์ที่ติดต่อกับ base station ได้ดีกว่าวิธีการที่มีอยู่แล้วด้วย พร้อมทั้งได้ทำการทดลองจริงลงบนบอร์ด tmote sky ซึ่งผลที่ได้ก็เป็นอัลกอริทึมที่ได้ออกแบบไว้และสามารถนำมาใช้งานเก็บข้อมูลได้จริง

ระบบที่พัฒนาขึ้นนี้ สามารถนำไปประยุกต์ใช้กับมิเตอร์น้ำประปาแบบดิจิตอล ซึ่งมีระบบสื่อสารด้วยคลื่นวิทยุที่ผลิตขึ้นในประเทศได้

วัตถุประสงค์

1. เพื่ออำนวยความสะดวกและลดค่าใช้จ่ายในการรับส่งข้อมูลจากมิเตอร์วัดการไหลของน้ำไปยังฐานข้อมูลกลางของผู้ให้บริการ

2. เพื่อค้นคว้าวิจัยและปรับปรุง routing protocol ในการค้นหาเส้นทางการส่งข้อมูลที่ดีที่สุดผ่านทาง wireless sensor network ให้สามารถนำมาใช้กับระบบอ่านค่าน้ำประปาตามบ้านเรือนทั่วไปได้ โดยรับส่งข้อมูลกับศูนย์กลางจัดเก็บข้อมูลหรือ server ได้อย่างรวดเร็ว และผิดพลาดน้อยที่สุด

3. เพื่อหาวิธีประหยัดพลังงานจากแบตเตอรี่ในมาตรวัดน้ำแบบดิจิตอลที่ใช้ระบบสื่อสารข้อมูลแบบคลื่นวิทยุ กับศูนย์จัดเก็บข้อมูลของผู้ให้บริการ ให้สามารถใช้งานให้ได้นานที่สุด

การตรวจเอกสาร

ในหัวข้อนี้จะกล่าวถึงความรู้พื้นฐานของวิธีการรวบรวมข้อมูลจากมิเตอร์น้ำ ซึ่งโดยทั่วไปแล้วการเก็บค่าน้ำจากมิเตอร์ในปัจจุบันอาศัยคนเดินเท้าไปอ่านที่ตัวมิเตอร์ จดลงบน PDA และเก็บข้อมูลเหล่านี้ลงใน Database เพื่อนำไปใช้งานต่อไป จากวิธีการที่กล่าวมานี้จะต้องเสียค่าใช้จ่ายในการจ้างพนักงานเก็บข้อมูล และยังเกิดการผิดพลาดจากการอ่านของพนักงาน ที่เกิดจากปัจจัยหลาย ๆ อย่างเช่น พนักงานอ่านค่าตัวเลขผิดพลาดเนื่องจากเกิดความเหนื่อยล้า หน้าจอมิเตอร์สกปรก หรือในกรณีที่หน้าปัดเป็น LCD อาจเกิดการที่สีตัวแสดงผลจางเกินไป เป็นต้น หลาย ๆ วิธีการได้ถูกคิดขึ้นเพื่อที่จะปรับปรุงวิธีการเก็บข้อมูลเพื่อลดค่าใช้จ่าย และความผิดพลาดที่เกิดขึ้น (Neptune, 2006; Neptune, 2007) โดยวิธีการที่ใช้อยู่ในปัจจุบันสามารถแบ่งได้ 3 กลุ่มใหญ่ๆ ดังนี้

1. การใช้สายต่อตรง (direct wire connection)

วิธี direct wire connection เป็นวิธีการเก็บข้อมูลผ่านทางสายรับส่งสัญญาณที่ต่อตรงเข้าหาตัวมิเตอร์ ซึ่งมีสองโปรโตคอลใหญ่ ๆ ที่ใช้กันมากได้แก่

1.1 Meter bus (M-Bus) วิธีนี้ใช้สายสัญญาณข้อมูล 2 เส้นในการรับส่งข้อมูล โดยใช้โปรโตคอล M-BUS ในการส่งและรับข้อมูลจากมิเตอร์แต่ละตัวไปยังศูนย์ควบคุม (base station) (Ziegler and Physik, 2007) การเดินสายสัญญาณข้อมูลชนิดนี้แบบ outdoor ในบริเวณหมู่บ้านที่บ้านแต่ละหลังไม่ได้อยู่ติดกันแบบอาคารชุดหรือทาวน์เฮาส์ จะมีค่าใช้จ่ายในการติดตั้งที่สูงมาก

1.2 การใช้สายไฟฟ้ากำลัง (Power line link) วิธีนี้ใช้สายไฟฟ้าที่ใช้แจกจ่ายกำลังไฟฟ้าไปยังบ้านของผู้ใช้น้ำเป็นตัวรับส่งข้อมูล โดยใช้วิธีการที่เรียกว่า modulation แบบ Frequency Shift Keying (FSK), Frequency Diversity (FD) และ Frequency Hopping (FH) อย่างใดอย่างหนึ่งเป็นตัว Modulate ข้อมูลที่ได้จากมิเตอร์เข้าไปในสายไฟฟ้า และส่งข้อมูลไปยังศูนย์ควบคุม(base station) (Park *et al.*, 2002; Tan and Moghavvemi, 2002; Wong and Moghavvemi, 2002) โดยทั่วไปแล้ววิธีการแบบนี้จะใช้ค่าใช้จ่ายอุปกรณ์ในการติดตั้งสูงและยังยากต่อการบำรุงรักษา

2. การใช้คลื่นวิทยุ (Radio Frequency (RF))

วิธี RF คือการใช้คลื่นวิทยุรับส่งข้อมูลจากมิเตอร์ไปยังศูนย์ควบคุม โดยสามารถแบ่งวิธีการตามลักษณะการติดตั้งได้ 2 รูปแบบคือ

2.1 การใช้คลื่นวิทยุแบบสถานีควบคุมอยู่กับที่ (fixed base station) วิธีนี้มิเตอร์แต่ละตัวส่งข้อมูลผ่านทาง RF ไปยัง Base station ที่อยู่กับที่ (Koay et al., 1999) ซึ่งจะใช้ได้ในพื้นที่ที่มิเตอร์อยู่ในรัศมีคลื่นรับส่งที่ติดต่อกับ Base station ได้โดยตรง หรือใช้วิธีสร้างเครือข่ายคลื่นวิทยุ โดยการส่งต่อจากมิเตอร์แต่ละตัวเป็นทอด ๆ ไปยัง Base station ได้

2.2 การใช้คลื่นวิทยุแบบสถานีควบคุมเคลื่อนที่ (mobile ration station) วิธีการในกลุ่มนี้อาจเรียกอีกอย่างได้ว่า **"walk-by"** หรือ **"drive-by"** (Heinzelman *et al.*,2000) ซึ่งเป็นวิธีการที่ใช้คนงานถือเครื่องมือที่เรียกว่า mobile station เดินผ่านหรือขับรถผ่านบริเวณที่มีมิเตอร์ โดยเมื่อตัว mobile station เคลื่อนที่ผ่านบริเวณที่สามารถติดต่อสื่อสารกับมิเตอร์ได้ มิเตอร์ก็จะส่งข้อมูลมาให้กับตัว mobile station ทันที ซึ่งวิธีนี้ยังจำเป็นต้องเสียค่าใช้จ่ายในการจ้างคนงานในการเก็บข้อมูลอยู่เช่นเดิม แต่อาจจะเก็บข้อมูลได้รวดเร็วขึ้น

3. การใช้เครือข่ายโทรศัพท์มือถือ (mobile telephone network)

ในวิธีนี้ ตัวมิเตอร์จะใช้ประโยชน์จากเครือข่ายโทรศัพท์มือถือในการรับส่งข้อมูล โดยส่งผ่านทาง SMS หรือ GPRS ไปให้กับ base station (Shiwei *et al.,* 1996; Ando *et al*., 2002) วิธีนี้สามารถใช้ได้ในระยะทางที่ไกลมาก ๆ และไม่จำเป็นต้องใช้พนักงานเดินทางไปยังตัวมิเตอร์ แต่อุปกรณ์ในการติดต่อและค่าบริการโทรศัพท์ที่ต้องเสียไปให้กับมิเตอร์แต่ละตัวนั้นมีราคาที่สูงมาก ดังนั้นวิธีการนี้จึงเหมาะกับการใช้งานในระบบที่มีจำนวนของมิเตอร์ไม่มากนัก และเป็นมิเตอร์สำหรับผู้ใช้น้ำรายใหญ่ ๆ เช่น ในโรงงานอุตสาหกรรม ศูนย์การค้า เป็นต้น

การใช้ Ad-hoc Wireless Sensor Network ในการเก็บค่าน้ำประปา

ด้วยปัญหาและค่าใช้จ่ายของระบบที่กล่าวมาทั้งหมด ทำให้ความสามารถในการเก็บข้อมูลจากพื้นที่ไปยัง base station โดยใช้วิธีการที่เรียกว่า Ad-hoc Wireless Sensor Network เป็น

ทางเลือกในการเก็บข้อมูลจากมิเตอร์น้ำอีกทางเลือกหนึ่ง ซึ่งวิธีนี้สามารถใช้ประโยชน์ในการเก็บข้อมูลจาก sensor node จำนวนมากไปยัง base station ได้เร็วและใช้พลังงานน้อย(Heinzelman et al., 2000; Lindsey and Raghavendra, 2002; Tan and Korpeoglu, 2003; Manjeshwar and Agrawal, 2001; Manjeshwar and Agrawal, 2002)

ในการประยุกต์วิธี Ad-hoc wireless sensor network มาใช้เก็บข้อมูลจากมิเตอร์น้ำ เราสามารถใช้ตัวมิเตอร์น้ำแบบดิจิตอลที่มีอยู่แล้วเป็นพื้นฐาน โดยแก้ไขให้มิเตอร์ทำหน้าที่คล้ายกับ sensor node โดยเพิ่มส่วนของ RF เข้าไปในตัวมิเตอร์ (Dugas, 2005) ซึ่งทำให้ตัวมิเตอร์สามารถส่งข้อมูลผ่านทาง RF ได้พร้อมทั้งสามารถใช้เทคนิคการเลือกเส้นทางในระบบ sensor network บางตัว (Wang and Arslan, 2003; Moore and Sekercioglu, 2005; Hussain and Matin, 2005; Perkins and Royer, 1999; Rabaey *et al.*, 2000) ได้ด้วย ซึ่งการรวบรวมจะอาศัยวิธีการเลือกเส้นทางจากการศึกษาพบว่ามีความเป็นไปได้ที่จะนำวิธีการหาเส้นทางมาประยุกต์ใช้กับระบบการอ่านค่าน้ำประปาได้ แต่เนื่องจากไม่สามารถนำเอาวิธีการเลือกเส้นทางในระบบ sensor network มาใช้ได้โดยตรงเพราะแต่ละวิธีการออกแบบมาให้เหมาะกับงานแต่ละงาน ทำให้วิทยานิพนธ์นี้จึงเสนอวิธีการที่เรียกว่า EnForce Multiple Sub-Tree เพื่อให้เหมาะสมกับงานเก็บค่าน้ำประปา โดยมีรายละเอียดดังนี้

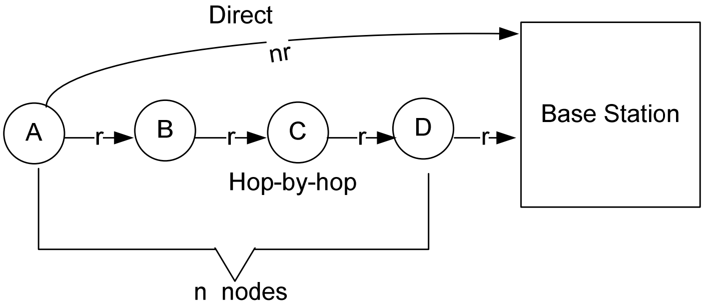
เทคนิคการค้นหาเส้นทางการรวบรวมข้อมูลของ sensor network

เทคนิคในการสร้างเส้นทางในการรวบรวมข้อมูลของระบบ Sensor network ในปัจจุบันได้มีงานวิจัยออกมามากมาย ซึ่งผู้ทำการวิจัยจะอธิบายบางวิธีการที่ได้พิจารณาแล้วว่ามีความเป็นไปได้ที่จะนำมาประยุกต์ใช้ในการอ่านค่าปริมาณการใช้น้ำประปา

1. LEACH

คำว่า LEACH ย่อมาจาก Low-Energy Adaptive Clustering Hierarchy (Heinzelman *et al.*, 2000) ซึ่ง LEACH จะไม่ใช้วิธีการส่งข้อมูลต่อเป็นทอด ๆ (hop-by-hop routing) แต่จะใช้ลักษณะของการส่งข้อมูลโดยตรง (direct) ไปยังbase station โดยการส่งข้อมูลแบบ direct นั้น node จะต้องใช้พลังงานในการส่งมากถ้าตัวรับอยู่ไกล ซึ่ง LEACH จะใช้วิธีการแบ่งกลุ่มของ node ออกเป็นกลุ่ม ๆ และทำการสุ่ม node ใด node หนึ่งในกลุ่มนั้น ๆให้เป็น head ซึ่ง node ที่ถูกเลือกเป็น head นั้นจะ

เป็นตัวส่งข้อมูลที่ node ในกลุ่มของมันส่งมาไปยัง base station โดยการที่มี head นั้นจะทำให้พลังงานของ node ที่เป็น head หมดอย่างรวดเร็ว ทำให้ LEACH ได้ใช้วิธีการสุ่ม (random) ตัว head ขึ้นมาเพื่อที่จะยืดอายุการใช้งานของระบบให้นานขึ้น



ภาพที่ 1 การวางระบบแบบเส้นตรงแบบง่าย

ตัวอย่าง network ที่เป็น simple linear network มีลักษณะเป็นเส้นตรงดังภาพที่ 1 ซึ่งผู้เขียนในเรื่อง LEACH ได้พิสูจน์ว่าการส่งข้อมูลโดยตรงจะใช้พลังงานน้อยกว่าวิธีการส่งแบบ hop-by-hop routing ในกรณีที่พลังงานการส่งข้อมูล (E_{Tx}) กับรับข้อมูล (E_{Rx}) มีค่าเท่ากัน (E_{Tx}=E_{Rx}) โดยมีระยะทางในการส่งข้อมูลระหว่าง node (r) สั้นและพลังงานที่ใช้กับวงจรส่งสัญญาณ (e_{amp}) มีค่ามาก ซึ่งเงื่อนไขดังกล่าวเขียนเป็นสมการได้ดังนี้

$$E_{elec}\frac{E_{elec}}{e_{amp}} > \frac{r^2 n}{2} \tag{1}$$

เมื่อ n เป็นจำนวนของ node

r เป็นระยะทางระหว่าง node

E_{elec} เป็นพลังงานที่ใช้ในทำงานของวงจรรับหรือส่งข้อมูล

e_{amp} เป็น พลังงานที่ใช้กับตัว amplifier

การทำงานของ LEACH จะแบ่งเป็นรอบ ๆ โดยแต่ละรอบจะแบ่งเป็น 2 เฟส คือ setup phase และ steady-state phase ใน setup phase จำนวนของกลุ่มจะถูกกำหนดขึ้นโดยอัตโนมัติ และ cluster head ของแต่ละ cluster จะถูกสุ่มขึ้นมา โดยแต่ละ node ทำการเลือกตัวเลขแบบสุ่มระหว่าง

0 ถึง 1 ขึ้นมาและนำมาเทียบกับค่า threshold ซึ่งถ้าค่าที่เลือกมาได้มีค่ามากกว่า threshold จะไม่ต้องเป็น cluster-head ในรอบนั้น โดยค่า threshold *T*(*n*) แสดงไว้ในสมการที่ (2)

$$T(n) = \frac{P}{1 - P * (r \bmod \frac{1}{P})} \qquad if\ n \in G \qquad (2)$$

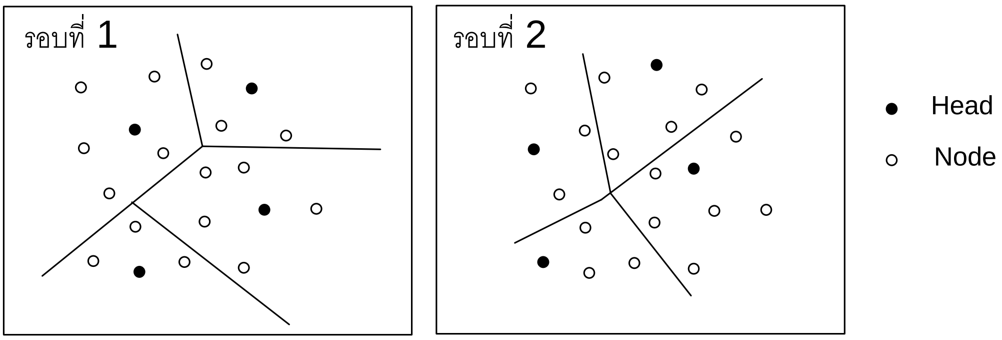
P = ความต้องการให้ nodes ใน network เป็น cluster-head ต่อ 100 nodes ตัวอย่างเช่น 0.05 หมายความว่าเป็น cluster-head 5 ตัวจาก nodes ทั้งหมด 100 ตัว
(ใน paper นี้ได้ทำการทดลองหาค่า P โดยค่าที่ 5% จะช่วยลดพลังงานได้มากที่สุด)

r = รอบ

G = set ของ nodes ที่ยังไม่ได้เป็น head

หลังจากการหา cluster-head ได้แล้วจะทำการกำหนดว่า node ใดจะมาอยู่ในกลุ่มของ head โดยพิจารณาจากระดับความแรงของสัญญาณที่ node นั้นได้รับจาก cluster head นั้น ความแรงสัญญาณของ cluster-head ไหนมากที่สุด node นั้นก็จะอยู่ในกลุ่มนั้น

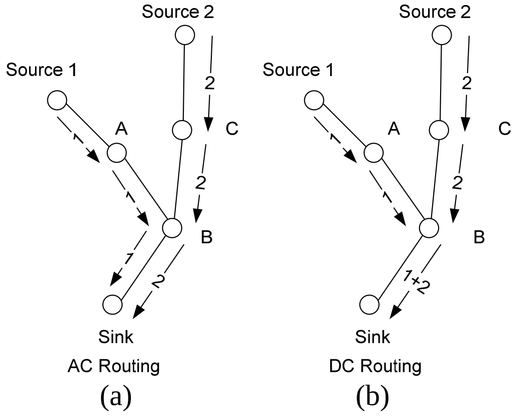
เมื่อเสร็จสิ้น setup phase แล้วระบบจะเข้าสู่ steady-state phase คือการที่ node ในกลุ่มส่งข้อมูลไปให้กับ cluster head เมื่อ cluster head ได้รับข้อมูลมาครบแล้วจะทำการรวมข้อมูล และส่งข้อมูลไปยัง base station (โดยวิธีการ direct communication) ซึ่งในภาพที่ 2 แสดงถึงผลของการจัดกลุ่มของ node โดยกลุ่มจะเปลี่ยนแปลงเป็นรอบ ๆ เพื่อที่เปลี่ยนตัว head ของกลุ่ม (เพื่อกระจายการใช้พลังงานของตัว head และช่วยเพิ่มอายุการใช้งานของระบบ)



ภาพที่ 2 Dynamic clustering ใน LEACH รอบที่ 1 และ 2 โดยจะมีการเลือก head ที่ต่างกันในแต่ละรอบเพื่อกระจายการใช้พลังงานของตัว head ในการส่งข้อมูลไปยัง base station

2. Directed Diffusion

Directed Diffusion [29] เป็นรูปแบบการติดต่อสื่อสารแบบหนึ่งที่ใช้ในการแพร่กระจายข้อมูลใน sensor network โดยใช้วิธีการที่เรียกว่า "data-centric routing (Park *et al.*, 2002) " โดยมีการแบ่งลักษณะการเก็บข้อมูลออกเป็น 2 แบบ



ภาพที่ 3 แสดงการสร้างเส้นทางแบบ (a) address-centric routing และ (b) data-centric routing

Address-centric routing เป็นการส่งข้อมูลจาก source ไปยัง sink โดยใช้เส้นทางจาก shortest path จาก source ไป sink เป็นคู่ ๆ โดยไม่มีการรวมข้อมูลเข้าด้วยกันหรือตัดข้อมูลที่ซ้ำกันออก ตัวอย่างเช่นในภาพที่ 3(a) source 1 ได้มีการส่งข้อมูลไปให้กับ sink โดยผ่าน A เนื่องจาก

ระยะทางจาก source 1 ไปยัง A สั้นกว่าระยะทางจาก source 1 ไปยัง B ส่วน source 2 ส่งไปหา sink ผ่าน C กับ B

Data-centric routing เป็นการส่งข้อมูลโดยยอมให้มีการรวมข้อมูลหรือตัดข้อมูลที่ซ้ำกันออกในระว่างทางได้ จากภาพที่ 3(b) source 1 และ 2 ส่งข้อมูลไปยัง sink โดยจะมีการร่วมข้อมูลที่ node B ก่อนเพื่อทำการรวมข้อมูลหรือตัดข้อมูลที่ซ้ำกันออก (เพื่อลดขนาดของข้อมูล)

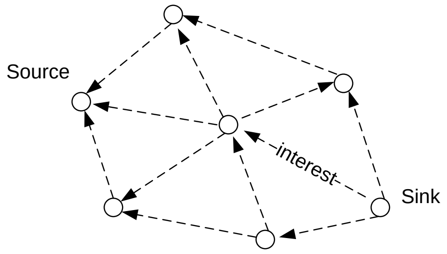
การทำงานของ directed diffusion เลือกวิธีการเก็บข้อมูลแบบ data-centric ซึ่งจะใช้วิธีการส่งคำถามและรอคำตอบ (query-response operation model) โดยรูปแบบของคำถาม (query หรือ interest) เช่น “ต้องการทราบว่าในพื้นที่ R มีสัตว์อยู่อาศัยอยู่หรือไม่ โดยให้รายงานทุก ๆ I วินาที เป็นระยะเวลา T วินาที” ซึ่ง query หรือ interest จะถูกส่งจาก node ใด node หนึ่งใน network ซึ่งจะเรียก node นั้นว่า sink node และจะกระจายต่อไปทั่ว network โดย interest จะประกอบไปด้วย

Type	= four-legged animal	// ชนิดของการตรวจสอบ
Interval	= 1 s	// รายงานทุก I วินาที
Duration	= 10 s	// เป็นระยะเวลา T วินาที
Rect	= {-100,100,200,200}	// จาก sensor ในพื้นที่ R

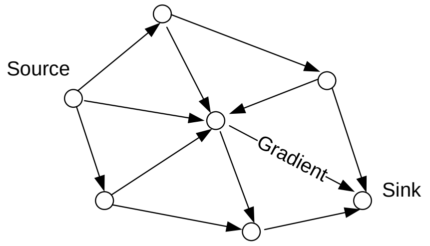
ในกรณีที่ source node ยังมีข้อมูลพร้อมที่จะส่ง sink node จะทำการกระจาย interest ออกไปอยู่เป็นระยะ ๆ โดยจะกำหนดความถี่ของอัตราความต้องการข้อมูลให้ต่ำ (low data rate) ซึ่งจะระบุไว้ใน interval (ซึ่งในที่นี้จะเป็น 1 วินาที) โดยแต่ละ node จะมี cache ที่ใช้ในการเก็บข้อมูลของ interest ไว้ ซึ่งจะประกอบด้วยเวลาที่เข้ามา timestamp field และ gradient field โดย gradient field จะเป็นตัวบอกว่า Node ที่ต้องการข้อมูลนั้นต้องการ data rate เท่าไรและ node ไหนเป็นตัวส่ง interest มาให้ โดยเมื่อ node ได้รับ interest แล้ว node จะ update ค่าที่อยู่ใน cache ของตัวเองให้ตรงกับ interest ที่ได้รับมา ว่า interest ที่ได้นี้มาจาก node ไหนมี timestamp เท่าไรและ duration ของ interest เป็นเท่าไร ซึ่งเมื่อเสร็จการ update แล้วก็จะส่ง interest ต่อให้กับ neighbors ของมันต่อไป ซึ่ง node ที่ได้รับก็จะทำในลักษณะเดียวกันจนครบทุก node ที่อยู่ใน network โดยลักษณะของการส่ง interest และ gradient จะเป็นดังภาพที่ 4(a) และ 4(b)

เมื่อ node ในพื้นที่ที่กำหนดไว้ตรวจจับเหตุการณ์ได้ตาม interest ที่ได้รับมา node จะส่งคำตอบกลับไปให้กับ node ที่มันได้รับ interest (ซึ่งดูได้จาก gradient ที่เก็บไว้และอัตราความต้องการข้อมูล(data rate) ที่มากที่สุด) และจะส่งต่อไปจนถึงตัว sink node โดยอาจจะส่งออกเป็นหลาย ๆ path ได้ดังภาพที่ 4(c) ซึ่งในภาพที่ 4(c) นั้นเป็นขั้นตอนเริ่มต้น ทำให้ค่าอัตราความต้องการข้อมูล (data rate) มีค่าเท่ากันหมดทำให้มีเส้นทางหลายเส้นทาง และเมื่อ node ได้รับ data จะทำการเก็บค่าของ data ไว้ ซึ่งถ้า data ที่ได้รับมาใหม่ตรงกับของเดิมก็จะตัดทิ้ง

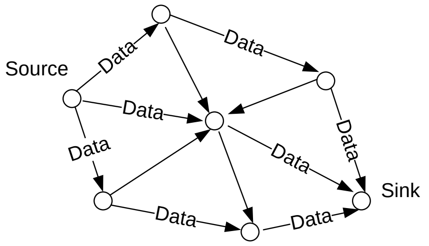
หลังจากที่ sink node ได้รับข้อมูล (data) ที่ส่งมาจาก source แล้วตัว sink node จะส่ง interest ไปให้กับตัว neighbor ตัวใดตัวหนึ่งเท่านั้นดังภาพที่ 4(d) ซึ่งพิจารณาจากความล่าช้าของข้อมูลที่ส่งมา โดยคำนวณจากค่า timestamp ของข้อมูลกับเวลาที่ได้รับ โดย sink node จะส่ง interest ที่จะมีอัตราความต้องการข้อมูล (data rate) ที่สูงขึ้น ซึ่งเรียกวิธีการนี้ว่า reinforcement ซึ่งถ้า node ไหนได้รับ data rate ที่มากกว่าค่า gradient ที่มีอยู่แล้ว node นั้นจะทำการ reinforce กับ neighbor ตัวนั้นเท่านั้นโดยดูจากข้อมูลที่เก็บไว้ใน memory ว่า node ที่ส่งข้อมูลมาหามันตัวแรกเป็นของ node ไหนแล้วก็จะส่ง Interest ไปให้ โดยส่งต่อตามเส้นทางที่ได้รับมาไปจนถึง source node ซึ่งผลที่ได้จากการ simulation ของ paper นี้ได้แสดงว่า directed diffusion มีประสิทธิภาพที่ดีกว่า flooding และ omniscient multicast ในด้านของพลังงานที่ใช้ในการกระจายข้อมูล (energy dissipated) อย่างไรก็ตามการที่ directed diffusion ส่ง low-data rate path และ broadcast interest บ่อย ๆ ทำให้ lifetime ของ network ลดลง



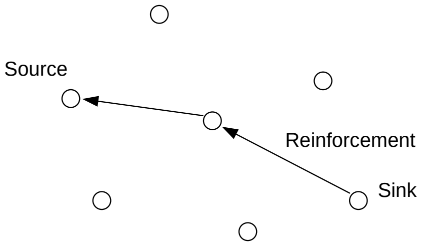
(a) ลักษณะการกระจายของ interest



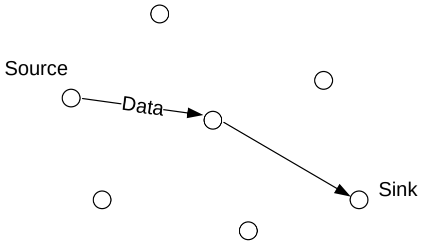
(b) ลักษณะทิศทางของ Gradient โดยจะชี้ไปยัง node ที่มันได้รับ interest



(c) ลักษณะการส่งข้อมูลจาก source ไป sink โดยพิจารณาจาก Gradient



(d) ลักษณะการส่ง Reinforcement จาก sink ไปยัง source ซึ่งเลือก path โดยพิจารณาจากความล่าช้าของข้อมูล (Delay)



(e) ลักษณะการส่งข้อมูลจาก source ไป sink โดยส่งผ่านเส้นทางการ reinforcement

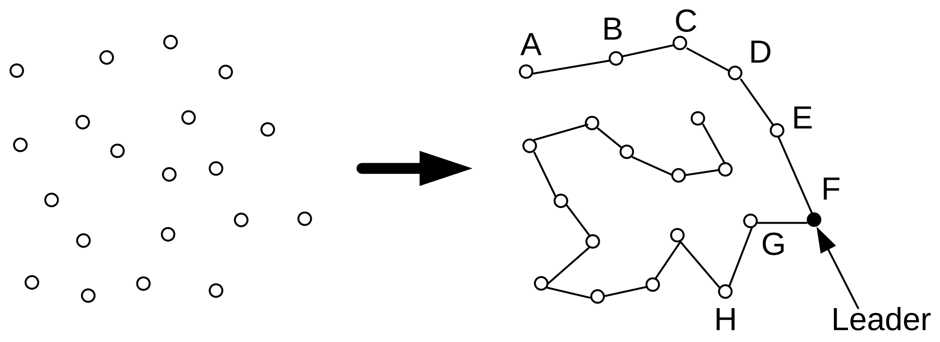
ภาพที่ 4 ระบบการทำงานของ directed diffusion

3. PEGASIS

PEGASIS (Power-Efficient Gathering in Sensor Information Systems) (Lindsey *et al.*, 2002) เป็นโปรโตคอลที่กำหนดให้เพียง node เดียวเท่านั้นที่สามารถส่งข้อมูลไปให้กับ base station (BS) ได้ในแต่ละรอบ โดยรูปแบบของ radio model ที่ใช้ใน PEGASIS จะมีลักษณะเช่นเดียวกันกับ LEACH

PEGASIS จะทำงานสร้าง chain ให้กับทุก node ใน network โดยแต่ละ node จะส่งข้อมูลได้เฉพาะกับ node รอบข้างเท่านั้น และจะมีเพียง node เดียวเท่านั้นใน chain ที่สามารถส่งข้อมูลไปให้กับ base station ได้ ซึ่งขั้นตอนในการทำงานของระบบ PEGASIS นั้นแบ่งเป็น 2 ขั้นตอน

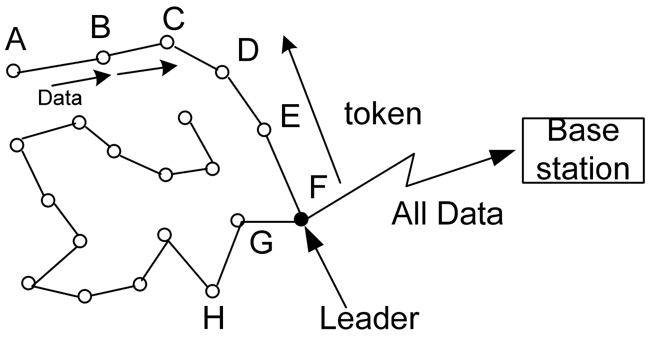
3.1 การสร้างโซ่ (chain) เป็นการสร้างโซ่โดยกำหนดเส้นทางการส่งข้อมูลและกำหนดตัว leader ที่จะส่งข้อมูลไปให้กับ base station ซึ่งการสร้างเส้นทางนี้มีลักษณะของปัญหาของ traveling salesman เพื่อหาเส้นทางที่สั้นที่สุดที่จะวิ่งไปทุก ๆ node ใน network หรืออาจจะใช้ greedy algorithm ในการสร้างก็ได้เพื่อให้สามารถสร้างเส้นทางได้อย่างรวดเร็ว(แต่อาจจะมีระยะทางไกลเมื่อเทียบกับใช้ traveling salesman) โดยจะเริ่มสร้างจาก node ที่ไกลจาก base station มากที่สุด (ซึ่งสามารถแน่ใจได้ว่า node นี้จะต้องมี node รอบข้าง) ซึ่งการสร้าง chain จะได้ผลดังภาพที่ 5 โดยเมื่อมี node ไหนใน chain พลังงานหมดหรือเกิดการ error ไม่สามารถส่งข้อมูลได้ระบบต้องทำการสร้าง chain ใหม่ ซึ่งจะตัด node นั้นออกไป โดยเมื่อทำการสร้าง chain เสร็จแล้วจึงจะเริ่มทำการเก็บข้อมูล



ภาพที่ 5 การสร้าง Chain ใน PEGASIS

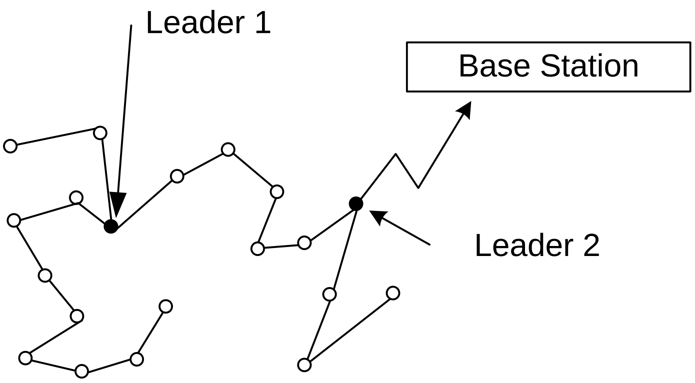
3.2 การรวบรวมข้อมูล (gathering data) จะทำงานเป็นรอบ ๆ โดยแต่ละรอบจะเลือกตัว leader มารอบละตัวโดยจะเลือกให้ไม่ซ้ำกันจนกว่าจะครบทุก node เมื่อครบทุก node แล้วจึงจะเริ่มต้นใหม่ (ดังภาพที่ 6 node F เป็น leader เมื่อทำงานครบรอบแล้ว node F จะไม่สามารถเป็น leader ได้ต้องรอจนกว่าจะเป็น leader ครบทุก node แล้วจึงจะเป็นใหม่ได้ โดยในรอบต่อไปอาจจะเป็น node E เป็นต้น) ซึ่งตัว node ที่เป็น leader ในแต่ละรอบจะส่ง token ไปตาม chain ไปหาท้ายสุดของ chain เพื่อเป็นตัวบอกเริ่มทำการเก็บข้อมูล ดังภาพที่ 6 leader จะส่ง token ไปให้หา A

โดยผ่านแต่ละ node คือ F, E, D, C, B ตามลำดับ เมื่อถึง A แล้ว A จะส่งข้อมูลมาให้กับ B และ Bจะรวมข้อมูลของ A กับข้อมูลของตัวมันเข้าด้วยกันละส่งไปให้กับ C ซึ่ง C ก็จะทำในลักษณะเดียวกันจนถึง leader เมื่อ leader ได้รับก็จะส่ง token ไปอีกทางหนึ่ง โดยในที่นี้จะส่งไปให้กับ G ซึ่งจะทำในลักษณะเดียวกัน ซึ่งเมื่อ leader ได้รับข้อมูลครบแล้วก็จะทำการส่งข้อมูลทั้งหมดไปให้กับ base station



ภาพที่ 6 การรวมข้อมูลใน Chain

กรณีที่ตัว leader node นั้นไม่สามารถติดต่อกับ base station (เป็นตัวที่ทำหน้าที่รวบรวมข้อมูลส่งให้กับผู้ใช้) โดยตรงตัว leader ก็สามารถที่จะใช้การส่งข้อมูลแบบ hop-by-hop routing แทนได้ ซึ่งการที่ PEGASIS กำหนดให้แต่ละ node ต้องมีข้อมูลที่อยู่ (location) ของทุก node ใน network จึงทำให้วิธีการ PEGASIS นี้มีปัญหาในด้านของ scalability ของระบบ โดยเฉพาะอย่างยิ่งปัญหาของเวลาที่ใช้ในการรวบรวมข้อมูล (long latency) ซึ่งการรวบรวมข้อมูลของ PEGASIS นี้จะขึ้นอยู่กับจำนวนของ node ใน network โดยปัญหานี้สามารถที่จะแก้ได้โดยการใช้วิธี multi-level chaining (เป็นการส่งข้อมูลแบบลูกโซ่เป็นทอด ๆ ต่อเป็นลำดับชั้นดังภาพที่ 7 Leader 1 มีระยะห่างจาก base Station มากจนรัศมีของสัญญาณไม่สามารถส่งไปถึง Base Station ได้ทำให้ต้องมีการแบ่งเป็น level โดย leader 1 จะเป็น node ใน chain ของ leader 2) และการที่ทุก node ต้องมีข้อมูลที่อยู่ของทุก ๆ node ใน network ทำให้ node ต้องใช้พลังงานเพิ่มขึ้นเพื่อที่จะหา neighbor ที่อยู่ใกล้กับตัวมันเอง



ภาพที่ 7 แสดงการสร้าง Chain แบบ 2-level

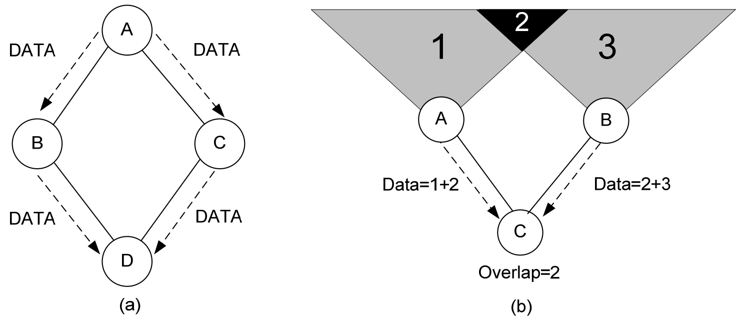
4. SPIN

SPIN (Sensor Protocol for Information via Negotiation) (Heinzelman et al., 1999) เป็นโปรโตคอล ที่ใช้สำหรับระบบ sensor network ที่แต่ละ node ทำงานแบบอัตโนมัติไม่มี central point (sink node) และแต่ละ node ทำงานที่แตกต่างกันออกไป โดยอาจจะหมายถึงการที่แต่ละ node มีลักษณะข้อมูลที่ต้องการไม่เหมือนกัน แต่จะต้องตรวจจับจากหลาย ๆ ที่ทำให้เมื่อ node ใด node หนึ่งมีข้อมูลมาใหม่ต้องทำการกระจายข้อมูลนี้ให้กับ node ใน network ทั้งหมด ซึ่งการทำงานในลักษณะนี้ทำให้เกิดปัญหาขึ้นคือ implosion, overlap และ resource-blindness โดยใน SPIN จะเน้นในการแก้ปัญหาทั้ง 3 นี้

1) Implosion เป็นปัญหาที่เกิดขึ้นเมื่อ node D รับข้อมูลที่เหมือนกันมาจาก 2 neighbor ดังภาพที่ 8(a) node A ทำการส่งข้อมูลให้กับ neighbors (node B และ C) ซึ่งเมื่อ neighbors ได้รับจะทำการส่งข้อมูลออกไปให้กับ neighbors ของมันอีกซึ่งคือ node D ดังนั้น node D จะได้รับข้อมูลที่เหมือนกันจาก node B และ C

2) Overlap เป็นปัญหาที่เกิดขึ้นเมื่อ sensor ตรวจสอบครอบคลุมพื้นที่ที่เดียวกัน ซึ่งจะทำให้ข้อมูลที่ได้มีการซ้อนทับกันบางส่วนดังภาพที่ 8(b) ซึ่งการส่งข้อมูลที่ซ้ำกันออกไปนั้นทำให้สิ้นเปลือง bandwidth และพลังงานอย่างไม่จำเป็น

3) Resource-blinding เป็นปัญหาที่ node ไม่สามารถที่จะรู้ถึงพลังงานที่มีอยู่ในตัวมัน ทำให้ไม่สามารถเปลี่ยนแปลงการทำงานของตัวมันเองได้ ซึ่งจะทำให้การข้อมูลที่ส่งขาดหายได้ในกรณีที่พลังงานของ node หมดขณะส่งหรือรับข้อมูล



ภาพที่ 8 ปัญหาข้อมูลซ้ำกันและข้อมูลบางส่วนเหมือนกันในการส่งข้อมูลแบบ flooding

วิธีการของ SPIN จะใช้ meta-data ในการกำหนดข้อมูลที่ต้องการส่งออกไป ซึ่ง meta-data นี้เป็นการสรุปหรืออธิบายลักษณะของข้อมูล (DATA) ที่ได้รับมา โดยขนาดของ meta-data จะเล็กกว่าขนาดของข้อมูล ซึ่งถ้าข้อมูลที่ได้มามีลักษณะเหมือนกัน ตัว meta-data ก็จะเหมือนกันด้วย โดย meta-data จะส่งไปพร้อมกับ message ซึ่งใน SPIN จะแบ่ง message ออกเป็น 3 ชนิดคือ

1). Advertising new data (ADV) เป็น message ที่ประกอบด้วย meta-data อย่างเดียวที่ใช้ในการประกาศให้กับ neighbors ว่าได้มีข้อมูลใหม่เข้ามา

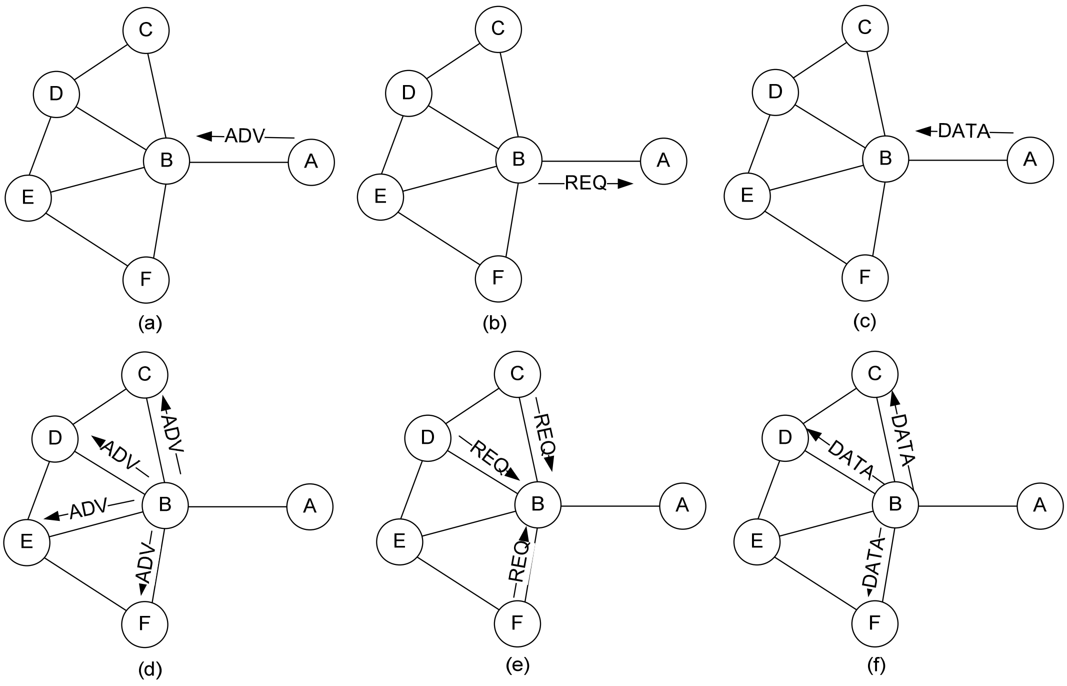
2). Requesting data (REQ) เป็น message ที่ประกอบด้วย meta-data อย่างเดียวที่ใช้เมื่อ node ได้รับ ADV แล้วจะทำการส่ง REQ กลับไป เพื่อบอกว่ามันต้องการจะรับข้อมูลที่ประกาศมา

3). Data message (DATA) เป็น message ทีประกอบด้วย meta-data และข้อมูลที่ได้มาจาก sensor ของ node นั้น

ในงานวิจัยของ SPIN ได้ทำการทดลองโดยแบ่งออกเป็น 2 Protocol คือ SPIN-1, SPIN-2 ใน SPIN-1 จะเริ่มต้นการส่งข้อมูลเมื่อมี node ใด node หนึ่งได้รับข้อมูลมาใหม่ node นั้นจะส่ง ADV ออกไป เพื่อแจ้งให้ neighbors ของมันทราบว่าข้อมูลที่ได้มาใหม่นี้เป็นข้อมูลลักษณะแบบ

ไหนดังภาพที่ 9(a) และถ้า neighbors ตัวใดสนใจข้อมูลนี้มาก็จะส่ง REQ ไปให้กับ node ที่ส่ง ADV มาดังภาพที่ 9(b) และ node A ได้รับ REQ มาก็จะส่ง DATA ให้ดังภาพที่ 9(c) และ node B ได้รับก็จะได้รับข้อมูลใหม่มาก็จะทำการกระจายข้อมูลต่อไปอีกดังภาพที่ 9(d)-9(f) โดยจะทำไปเรื่อย ๆ ไปจนกระทั่งทุก ๆ node ได้รับข้อมูลใหม่ทั้งหมด ซึ่งในภาพที่ 9(d)จะพบว่า node B ไม่ส่ง ADV ไปให้กับ node A เนื่องจาก node A เป็น node ที่ส่งข้อมูลนี้มาให้กับ B ซึ่งการใช้ mete-data ทั้ง 3 ตัวนี้จะลดปัญหาการเกิด implosion และ overlap ได้ และในภาพที่ 9(e) node E ไม่ต้องการข้อมูลจาก node B จึงไม่ส่ง REQ ให้กับ node B ซึ่ง node B ก็จะไม่ส่งข้อมูลไปให้กับ node E ดังภาพที่ 9(f)

ใน SPIN-2 จะเหมือนกับ SPIN-1 แต่จะเพิ่ม low-energy threshold เข้าไปคือ ถ้า node ได้รับ new data หรือได้รับ ADV จะไม่ทำตาม โปรโตคอลนั้น ถ้าตัวของมันมีพลังงานไม่เพียงพอ (น้อยกว่า low-energy threshold) สำหรับการทำงานจนครบตามโปรโตคอล (การทำงานครบนั้นคือการที่ node ใด node หนึ่งสามารถที่จะส่งข้อมูล ADV หรือ REQ และรับ DATA ได้ครบโดยที่พลังงานไม่หมดก่อน)



ภาพที่ 9 การทำงานของโปรโตคอล SPIN

5. GEAR

GEAR (The Geographical and Energy-Aware Routing)(Koay *et al.*, 1999) ระบบจะทำงานแบบถามไปตอบกลับ(query-response model) โดยจะมีข้อกำหนดว่าแต่ละ node รู้ถึง location, energy level ของตัวมันและ neighbor ของมันด้วย ซึ่งวิธีการของ GEAR นี้จะแบ่งเป็น 2 ขั้นตอนคือ

5.1 การหาเส้นทาง

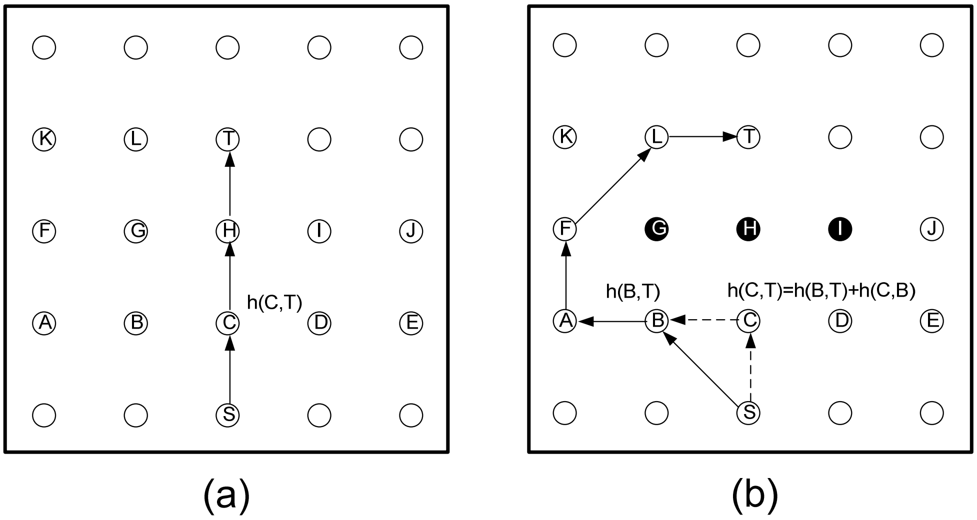
เป็นการหาเส้นทางเพื่อที่จะส่ง query ไปหาพื้นที่ที่กำหนดไว้โดยใช้พลังงานเป็นตัวเลือก node ที่อยู่รอบข้าง (energy-aware) ร่วมกับ geographically-informed neighbor selection heuristics โดยวิธีการ geographically-informed neighbor selection heuristics เป็นวิธีการกระจายข้อมูลไปหา region R ใน phase แรก โดยแต่ละ node N จะมีค่า learned cost ไปยัง neighbor i ใน region R เอาไว้ซึ่งแทนด้วย $c(N_i,R)$ และยังเก็บ learned cost ของ neighbor $h(N_i,R)$ ไปยัง neighbor i ด้วย โดยถ้า node ใดไม่มีค่า learned cost ของ neighbor มันจะคำนวณค่าโดยประมาณ cost $c(N_i,R)$ ตามค่า $h(N_i,R)$ ที่ได้กำหนดไว้ตั้งแต่เริ่มต้น (default) ซึ่งจะคำนวณได้จาก

$$c(N_i,R)=\alpha d(N_i,R) + (1-\alpha)e(N_i) \qquad (3)$$

α	เป็นค่า weight
$d(N_i,R)$	เป็นระยะห่างจาก N_i ไปยัง centroid D ของ R โดย normalized ด้วยระยะทางที่มากที่สุดในจำนวนของ neighbors ทั้งหมดของ N
$e(N_i)$	เป็นค่าพลังงานที่ใช้มากที่สุดที่ node N_i โดย normalized ด้วยค่าพลังงานที่ใช้มากที่สุดของ neighbors ของ N

จากภาพที่ 10(a) เป็นการส่งข้อมูลจาก node S ไปยัง node T ซึ่งค่า cost จาก node C ไป T มีค่าเท่ากับ h(C,T) แต่ว่าเมื่อ node C ไม่สามารถส่งข้อมูลไปให้กับ node T ได้เพราะ node G,H,I พลังงานหมดดังภาพที่ 10(b) ทำให้ node C ต้องหาเส้นทางในการส่งข้อมูลไปยัง T ใหม่โดยจะทำการเลือกว่าจะส่งไปยัง neighbors ไหนดี(พิจารณาจากค่า cost ที่น้อยที่สุด) โดยสมมุติว่าเป็น node B ดังภาพที่ 10(b) ซึ่งเมื่อ node C รู้ว่าจะส่งข้อมูลไปให้ node B แล้ว จะทำการเปลี่ยนแปลงค่า cost ของตัวมันโดยจะบวกค่า cost ระหว่าง node C ไป B ด้วย h(C,T) ใหม่เป็น h(B,T)+C(C,B) ซึ่งใน

เทอมหลังเป็น cost ของการส่ง packet จาก C ไปยัง B โดยเมื่อมีการส่งข้อมูลครั้งต่อไป node S ทำการเลือกโดยดูจากค่า cost ที่น้อยที่สุดซึ่งในขณะนี้ค่า cost ของ C ได้เปลี่ยนไปแล้วดังภาพที่ 10(b) โดยจะมีค่ามากว่าค่า cost ที่ S ส่งไปยัง B ทำให้ node S ไม่ส่งข้อมูลไปให้กับ C แต่จะส่งข้อมูลไปทาง B แทน

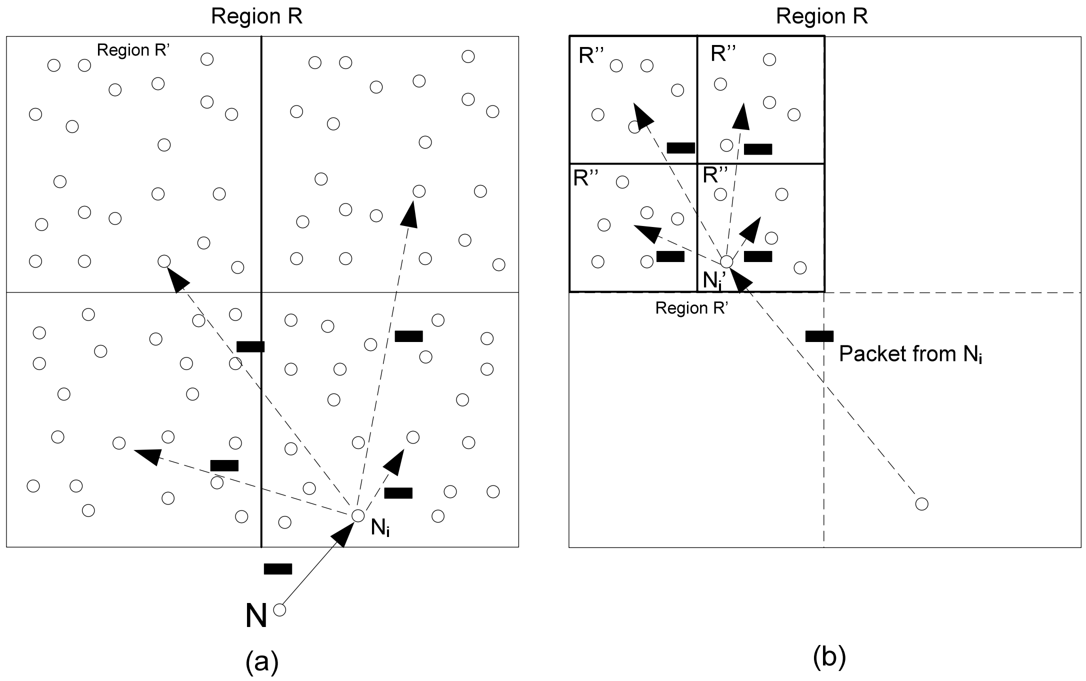


ภาพที่ 10 อัลกอริทึมของ GEAR

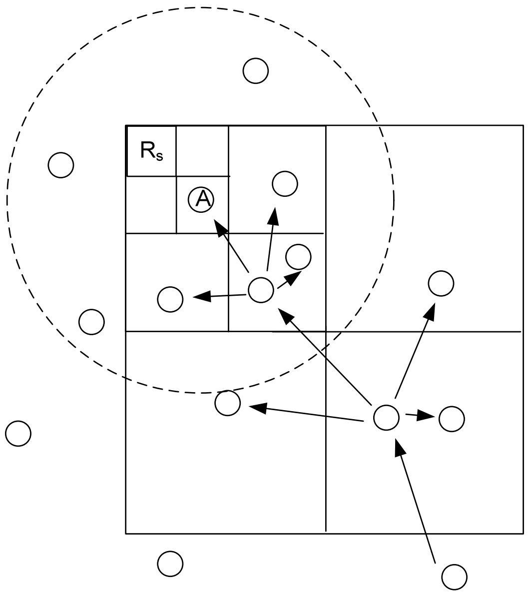
5.2 การกระจาย

การกระจาย query ให้ทั่วทั้ง region R ซึ่งจะใช้วิธีการส่งแบบ recursive geographic forwarding หรือ restricted flooding algorithm เป็นวิธีการส่งข้อมูลโดยการแบ่งพื้นที่ออกเป็นส่วนๆ และส่งข้อมูลเข้าไปในแต่ละส่วนนั้น โดยในแต่ละส่วนก็จะทำการแบ่งพื้นที่ย่อยออกเป็นส่วน ๆ อีกดังภาพที่ 11(a)-(b) ในภาพที่ 11(a) เมื่อ node N ได้รับข้อมูลและกระจายต่อให้กับ N_i ซึ่งเป็น node ที่อยู่ใน Region R โดย node N_i จะส่งข้อมูลต่อไปโดยจะแบ่ง Region R ออกเป็นส่วน ๆ ในที่นี้สมมุติว่าได้แบ่งออกเป็น 4 ส่วนและเมื่อแต่ละส่วนได้รับข้อมูลมาแล้วก็จะทำการแบ่งพื้นที่ออกเป็นส่วน ๆ ต่อไปอีกดังภาพที่ 11(b) และจะทำต่อไปเรื่อย ๆ จนกระทั่งสัญญาณของ node ที่ได้รับ packet นั้นครอบคลุมพื้นที่ย่อย (sub-region R) ที่ถูกแบ่งออกแต่ไม่มี neighbors ของ node นั้นอยู่ในพื้นที่ย่อยนั้น ซึ่งภาพที่ 11 จะพบว่า node A ได้รับ packet และแบ่ง region ออกเป็นย่อย ๆ (R_S) โดยสัญญาณวิทยุของ node A (แสดงด้วยเส้นประวงกลม) นี้ครอบคลุมพื้นที่ย่อยนั้นแต่ไม่มี

neighbors ของ node A อยู่ในพื้นที่นั้นเลย (R_S) ทำให้ไม่ต้องทำการส่งข้อมูลต่อและสิ้นสุดวิธีการกระจาย packet ใน region R



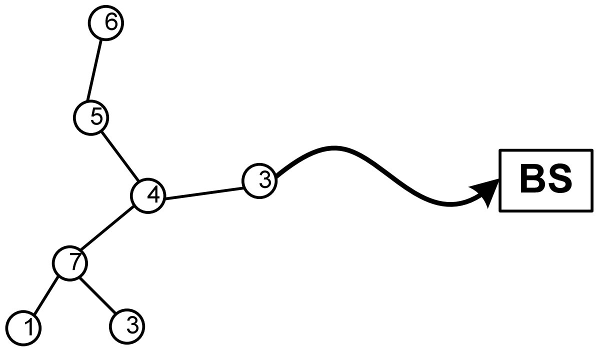
ภาพที่ 11 การกระจายข้อมูลภายใน Region R



ภาพที่ 12 การสิ้นสุดการกระจาย packet ภายใน region R

6. PEDAP

Power Efficient Data Gathering and Aggregation Protocol (PEDAP) (Tan and Korpeoglu, 2003) เป็นวิธีการที่มีเป้าหมายที่จะต้องการให้ระบบทำงานอยู่ได้นานที่สุดเมื่อตัว base station อยู่ตำแหน่งศูนย์กลางของระบบ โดยได้เสนอแนวความคิด 2 แนวด้วยกัน แนวความคิดที่หนึ่งจะใช้วิธีการของ minimum spanning tree (MST) ในการสร้างเส้นทางที่จะส่งข้อมูลไปให้กับ base station ดังภาพที่ 13



ภาพที่ 13 รูปแบบการเลือกเส้นทางของ PEDAP ใน simple network

เริ่มต้นจะกำหนดให้ base station เป็น root ของ MST ซึ่งจะคำนวนด้วย Prim's minimum spanning tree โดยแต่ละรอบจะทำการเพิ่ม node เข้าไปใน tree ทีละตัว ซึ่งดูจากเส้นทางที่มี weight หรือ cost น้อยที่สุด ที่คิดจาก node ที่อยู่ใน tree ไปยัง node ที่ไม่ได้อยู่ใน tree และสมการ cost ที่ใช้คือ

$$C_{ij}(k) = 2kE_{elec} + kd_{ij}^2 E_{amp} \quad (4)$$

$$C_i^{'}(k) = kE_{elec} + kd_{ib}^2 E_{amp} \quad (5)$$

C_{ij} เป็น cost ที่ใช้ในการส่งข้อมูลระหว่าง node i ไปยัง node j k เป็นจำนวนข้อมูลในการส่ง มีหน่วยเป็นบิต (bits) E_{elec} เป็นพลังงานที่ใช้ในส่วนประมวลผลข้อมูล (Joules/bit) และ E_{amp} เป็นพลังงานที่ใช้ในการส่งข้อมูล (Joules/bit/m^2) ส่วน C'_i จะเป็น cost ที่ใช้สำหรับการส่งข้อมูลระหว่าง node i กับ base station ซึ่งสมการ cost ที่ทาง PEDAP ใช้ได้พิจารณาจากสมการพลังงานตามงานวิจัยของ LEACH และ PEGASIS

กระบวนการเพิ่ม node เข้าไปใน tree นี้จะทำตามลักษณะดังกล่าวไปเรื่อย ๆ จนกว่า node ทุกตัวจะอยู่ใน tree และเมื่อ node ทุกตัวถูกเพิ่มเข้าไปอยู่ใน tree แล้วตัว base station จะทำการเก็บข้อมูลจาก node ทุกตัวในระบบ โดยเมื่อมีการเก็บข้อมูลครบตามจำนวนรอบที่ต้องการแล้ว (ในที่นี้กำหนดไว้ที่ 100 รอบ)ตัว base station จะทำการคำนวนหาเส้นทางในการเก็บข้อมูลใหม่ แต่ในเส้นทางใหม่นี้จะตัด node ที่พลังงานหมดแล้วออกไป ซึ่งการสร้างเส้นทางใหม่นี้ตัว base station จะทำการส่งคำสั่งไปกำหนดเส้นทางของแต่ละ node เอง ทำให้ cost ในการสร้างเส้นทางใหม่นี้จะเท่ากับ cost ที่ต้องเสียให้กับการส่งสัญญาณของ node ทั้งหมดรวมกัน

ส่วนในแนวทางที่สองจะใช้วิธีการสร้างเส้นทางในลักษณะเดียวกันแต่จะเพิ่มตัวแปรตัวหนึ่งในสมการ cost นั้น คือพลังงานที่เหลืออยู่ในแต่ละ node (e_i) เพื่อช่วยในการทำ load balance ให้กับtree ซึ่งเสามารถขียนสมการ cost ใหม่จากสมการที่ (4) และ (5) เป็นดังสมการที่ (6) และ (7)

$$C_{ij}(k) = \frac{2kE_{elec} + kd_{ij}^2 E_{amp}}{e_i} \tag{6}$$

$$C_i^{'}(k) = \frac{kE_{elec} + kd_{ib}^2 E_{amp}}{e_i} \tag{7}$$

ในขณะที่พัฒนาอัลกอริทึมที่ใช้ในวิทยานิพนธ์นี้ PEDAP เป็นวิธีการที่ประหยัดพลังงานและยืดอายุการทำงานของระบบได้มากกว่าทุกวิธีที่กล่าวมาแล้ว

อุปกรณ์และวิธีการ

อุปกรณ์

1. ฮาร์ดแวร์ระบบประกอบด้วย

1.1. เครื่องคอมพิวเตอร์พีซี (x86)
1.2. บอร์ดเซ็นเซอร์เน็ตเวิร์คและระบบสื่อสารโดยใช้คลื่นวิทยุ (Tmote sky)

Tmote sky

เพื่อให้สามารถทดสอบอัลกอริทึมที่พัฒนาขึ้นมาว่าสามารถใช้ได้ในทางปฏิบัติ จำเป็นต้องมีการทดลองการทำงานโดยใช้ฮาร์ดแวร์สำหรับ sensor network ซึ่งในที่นี้ได้ใช้บอร์ดของ Tmote Sky ของห้องปฏิบัติการจักรกลอัจฉริยะ Tmote Sky (Moteiv, 2006; Moteiv, 2007) เป็นอุปกรณ์สำหรับเครือข่ายเซ็นเซอร์ไร้สายที่พัฒนาและออกแบบขึ้นโดย University of California, Berkeley ซึ่งรองรับมาตรฐาน IEEE 802.15.4 สามารถทำงานในระยะสื่อสารที่กว้าง รวมทั้งการใช้พลังงานที่น้อยกว่า สามารถโปรแกรมผ่าน USB port ได้ การพัฒนาโปรแกรมจะต้องใช้ระบบปฏิบัติการ TinyOS โดยใช้ภาษาที่เรียกว่า NesC โดยมีส่วนประกอบที่สำคัญดังนี้

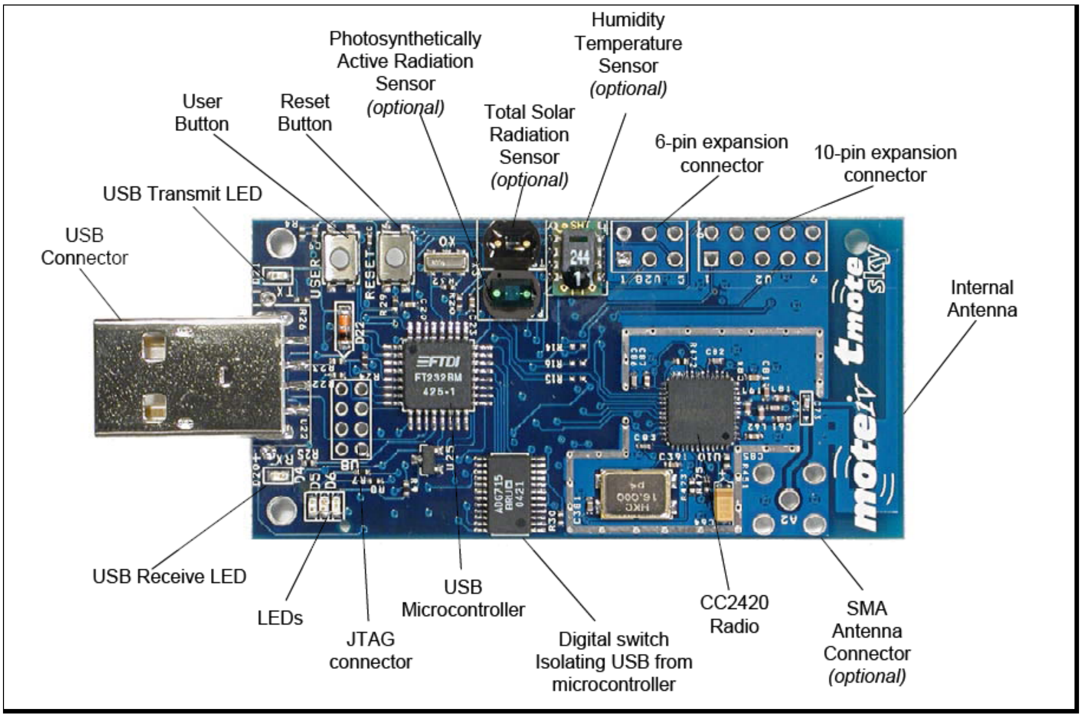
1. ส่วนประมวลผล บอร์ด Tmote Sky ใช้ชิป Texas Instruments MSP430 microcontroller ทำงานที่ความเร็ว 8 MHZ RAM 10 Kbytes มี Flash Memory สำหรับเก็บโปรแกรม 48 Kbytes โดยส่วนประมวลผลนี้จะทำหน้าที่ควบคุมการเชื่อมต่อต่างๆ เช่น ส่วนเซ็นเซอร์ ส่วนการสื่อสาร นอกจากนี้ยังทำหน้าที่จัดการเกี่ยวกับการใช้พลังงานของตัวบอร์ดด้วย

2. ส่วนเซ็นเซอร์ เป็นส่วนที่ทำหน้าที่น้อยที่สุด แต่มีความสำคัญมากเนื่องจากจะต้องตรวจจับข้อมูลภายนอกได้อย่างถูกต้อง โดยบอร์ด Tmote Sky มีเซ็นเซอร์ติดตั้งมาแล้ว 3 ชนิดคือ เซ็นเซอร์แสง เซ็นเซอร์อุณหภูมิ เซ็นเซอร์ความชื้น และยังมีพอร์ต UART ไว้ให้สามารถติดตั้งเซ็นเซอร์ภายนอกได้ตามความต้องการ

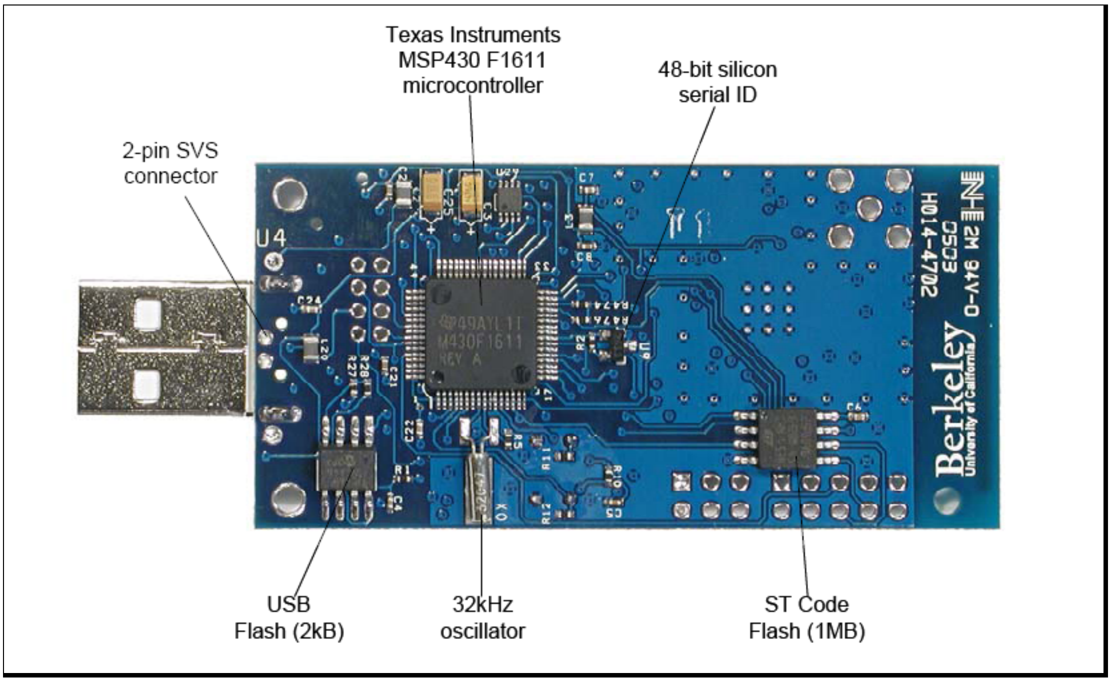
3. ส่วนเก็บข้อมูล ในการทำงานนั้นข้อมูลที่ได้รับจากส่วนเซ็นเซอร์จะถูกเก็บไว้ในหน่วยความจำของตัวเอง หรือ ส่งไปให้โหนดตัวอื่นๆ ขึ้นอยู่กับจุดประสงค์ในการใช้งาน ซึ่งสามารถโปรแกรมได้อย่างอิสระ

4. ส่วนการสื่อสาร ส่วนนี้เป็นการสื่อสารด้วยคลื่นวิทยุ โดยใช้ชิป Chipcon CC2420 Wireless Transceiver (Chipcon AS SmartRF, 2004) ซึ่งทำงานที่ความถี่ 2.4 GHz ความเร็วสูงสุด 250 kbps รองรับมาตรฐาน IEEE 802.15.4 ส่งข้อมูลในอาคารได้ 50 เมตร และบริเวณที่โล่งนอกอาคารได้ 150 เมตร

ลักษณะของบอร์ด Tmote Sky และส่วนประกอบต่างๆ แสดงอยู่ในภาพที่ 14 และ 15



ภาพที่ 14 Tmote Sky ด้านบน



ภาพที่ 15 Tmote Sky ด้านล่าง

2. ซอฟต์แวร์ระบบ ซอฟต์แวร์ระบบปฏิบัติการและระบบพัฒนาโปรแกรมที่ใช้มีดังต่อไปนี้

2.1. ระบบปฏิบัติการ Tiny OS

2.2. ระบบปฏิบัติการ Linux

2.3. ระบบปฏิบัติการ Windows

2.4. NesC Compiler และ ภาษา nesC

ภาษา nesC

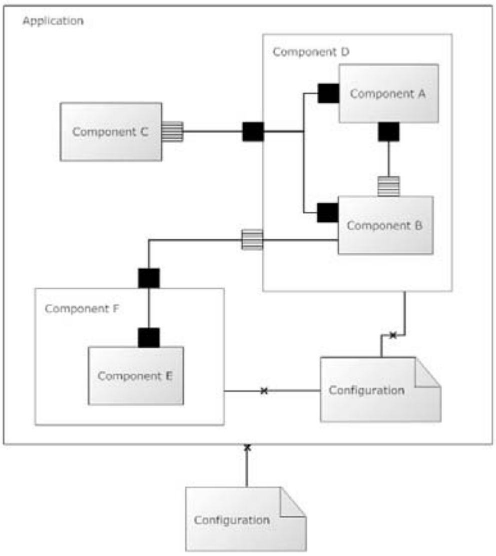
ผู้ผลิตบอร์ด tmote sky ได้ออกแบบบอร์ดมาให้เหมาะสมกับการใช้ภาษา nesC ในการพัฒนาโปรแกรมโดยเฉพาะ และออกแบบมามี module ที่ใช้งานเฉพาะกับบอร์ดมาให้ด้วย ซึ่งภาษา nesC เป็นภาษาที่ปรับปรุงเพิ่มเติมจากภาษา C เพื่อให้ทำงานบน tinyOS ที่มีลักษณะดังนี้

1) มีการแยกโครงสร้าง (construction) และองค์ประกอบต่าง ๆ (composition) ออกจากกัน โดยโปรแกรมจะถูกแบ่งให้เป็นโปรแกรมย่อย ๆ (component) แยกออกจากกัน จากนั้นจึงทำการเชื่อมส่วนย่อยเข้าด้วยกัน (wire) ในการโปรแกรมจึงทำให้สะดวกต่อการใช้งานมากขึ้น

2) ในการเชื่อมต่อส่วนย่อยจะมีการกำหนดลักษณะการ interface โดยแบ่งเป็น software module 2 ชนิด คือ

2.1) Provide Interface คือ interface module ส่วนที่ใช้การเชื่อมต่อนี้เป็นส่วนที่ให้บริการแก่ส่วนอื่นมาเชื่อมเข้าด้วยกัน

2.2) Used Interface คือ interface module ส่วนที่กำหนดว่าส่วนที่มีการเชื่อมต่อนี้เป็นส่วนที่ขอใช้บริการจากส่วนอื่น



ภาพที่ 16 แสดงการเชื่อมต่อแบบ provide แทนด้วยสี่เหลี่ยมจตุรัสลายแนวนอน และการเชื่อมต่อแบบ used แทนด้วยสี่เหลี่ยมจตุรัสสีดำ

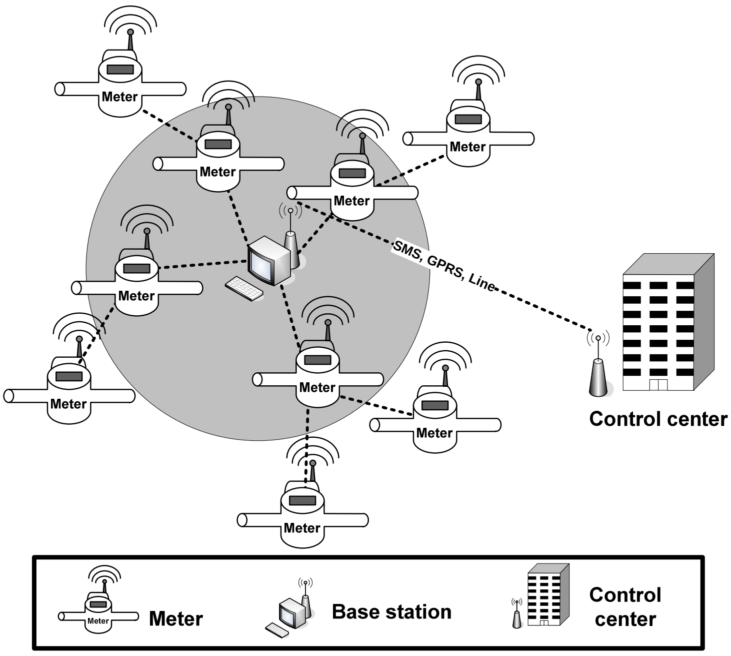
3) ส่วนที่ต่อเชื่อมที่มีการติดต่อกันสองทิศทาง คือส่วนที่สั่งการ (command) และส่วนที่ส่งสัญญาณกลับมา (event)

4) ภาษา nesC ถูกพัฒนามาใช้กับตัวแปลภาษา (compiler) ที่แปลโปรแกรม (compile) ทีเดียวทั้งโปรแกรม ทำให้ได้ assembly code ที่เหมาะสมที่สุด (optimized)

วิธีการ

1. ออกแบบ

การวิจัยนี้ได้สร้างแบบจำลองระบบขึ้นมา ซึ่งแบบจำลองของระบบ (system modeling) ที่ใช้ในการวิจัยจะมีลักษณะที่ใกล้เคียงกับระบบการอ่านค่ามิเตอร์น้ำทั่ว ๆ ไป ซึ่งสามารถแสดงได้ดังภาพที่ 17



ภาพที่ 17 วิธีการเก็บข้อมูลจากมิเตอร์น้ำทั่วไป

ลักษณะการทำงานของระบบในภาพที่ 17 คือ มิเตอร์แต่ละตัวจะส่งข้อมูลให้กับ base station เมื่อ base station ได้รับข้อมูลเสร็จสิ้นแล้ว ข้อมูลที่ได้ทั้งหมดจะถูกส่งไปยัง control center เพิ่มนำข้อมูลไปดำเนินการต่อไป โดยทั่ว ๆ ไปแล้วระยะทางระหว่างมิเตอร์จะประมาณ 10-100 เมตร ซึ่งเพียงพอที่บอร์ด RF ทั่วไปจะส่งข้อมูลระหว่างมิเตอร์ได้ ถ้าระยะทางระหว่างมิเตอร์

มากกว่านี้สามารถใช้ตัวขยายสัญญาณ (repeater) ในการขยายสัญญาณเพื่อให้รัศมีการส่งข้อมูลระหว่างมิเตอร์ได้ไกลขึ้น

ในงานวิทยานิพนธ์ฉบับนี้จะวิเคราะห์เฉพาะในส่วนการเก็บข้อมูลจากมิเตอร์แต่ละตัวมายัง base station เท่านั้น โดยประกอบด้วยสมมุติฐานดังต่อไปนี้

1. ตำแหน่งของมิเตอร์ (sensor node) และ base station อยู่กับที่ แต่ตำแหน่งที่แน่นอนไม่สามารถเลือกได้โดยสะดวก
2. ระยะทางระหว่างมิเตอร์ที่อยู่ติดกัน และระหว่างมิเตอร์กับ base station จะถูกจำกัดไว้ไม่เกินค่าสูงสุดค่าหนึ่ง เพื่อให้การสื่อสารด้วย RF เป็นไปได้
3. พลังงานของ base station ไม่มีวันหมด แต่ของมิเตอร์มีจำกัด
4. พลังงานเริ่มต้นของมิเตอร์แต่ละตัวมีค่าเท่ากัน
5. แต่ละมิเตอร์มี ID ที่แน่นอน
6. ข้อมูลจากมิเตอร์แต่ละตัวไม่มีส่วนที่เหมือนกัน ทำให้ไม่สามารถนำข้อมูลมารวมกันเพื่อให้ขนาดของข้อมูลเล็กลงได้
7. Software และ hardware ของมิเตอร์แต่ละตัวเหมือนกัน

สมการพลังงาน (energy model) ในตัวมิเตอร์แต่ละตัวในระบบจะใช้เหมือนกับในงานวิจัย (Heinzelman *et al.,* 2000) ซึ่งแสดงได้ดังสมการที่ (8) และ (9) ดังนี้

$$E_{Tx} = kE_{Elec} + kd^2 E_{Amp}, \tag{8}$$

$$E_{Rx} = kE_{Elec}, \tag{9}$$

โดยที่ E_{Tx} เป็นพลังงานในการส่งข้อมูล, E_{Rx} เป็นพลังงานในการรับข้อมูล, E_{Elec} เป็นพลังงานที่สูญเสียไปในภาคของอิเล็คทรอนิคส์ในการจัดการข้อมูล (Joulse/bit), E_{Amp} เป็นพลังงานที่สูญเสียไปในภาคขยายสัญญาณในการส่งข้อมูล 1 บิต (Joule/bit/m^2), k เป็นจำนวนบิตที่จะส่งออกไป และ d เป็นระยะทางไกลสุดที่สามารถส่งข้อมูลระหว่างตัวรับและตัวส่งได้ (เมตร)

งานวิจัยของ LEACH และ PEGASIS ไม่เหมาะกับสภาพแวดล้อมของการอ่านค่าปริมาตจากมิเตอร์น้ำผ่านทาง RF เพราะไม่สามารถออกแบบให้ base station สามารถส่งข้อมูลติดต่อกับ

มิเตอร์หรือ node ทุกตัวในระบบได้ ซึ่งในการศึกษาและวิจัยที่ผ่านมาแล้ว (Tan and Korpeoglu, 2003) วิธีการของ PEDAP จะมีประสิทธิภาพมากกว่า LEACH และ PEGASIS

เพราะฉะนั้นในงานวิจัยนี้จะเปรียบเทียบกับงานวิจัยของ PEDAP เป็นหลัก ซึ่งจากสมมุติฐานที่ 6) ข้อมูลของมิเตอร์ไม่สามารถนำมารวมกันได้ เนื่องจากข้อมูลของมิเตอร์แต่ละตัวมีค่าที่ไม่เหมือนกัน แต่มิเตอร์บางตัวจะต้องส่งข้อมูลมีขนาดที่มากขึ้นเพราะต้องรับช่วงทำหน้าที่เป็นตัวส่งข้อมูลของมิเตอร์อื่นไปยัง base station ด้วย

ใน PEDAP ตัว base station จะทำหน้าที่เป็น root ในการสร้างเส้นทางแบบ minimum spanning tree ทำให้ตัวมิเตอร์ที่ส่งข้อมูลให้กับ base station นั้นมีจำกัด โดยอาจจะมีอยู่เพียงตัวเดียวก็ได้ ซึ่งทำให้มิเตอร์ตัวนั้นใช้พลังงานในการส่งข้อมูลมากที่สุด และหมดพลังงานก่อนตัวอื่น เพราะต้องส่งผ่าน (forward) ข้อมูลของมิเตอร์ทั้งหมดไปให้กับ base station

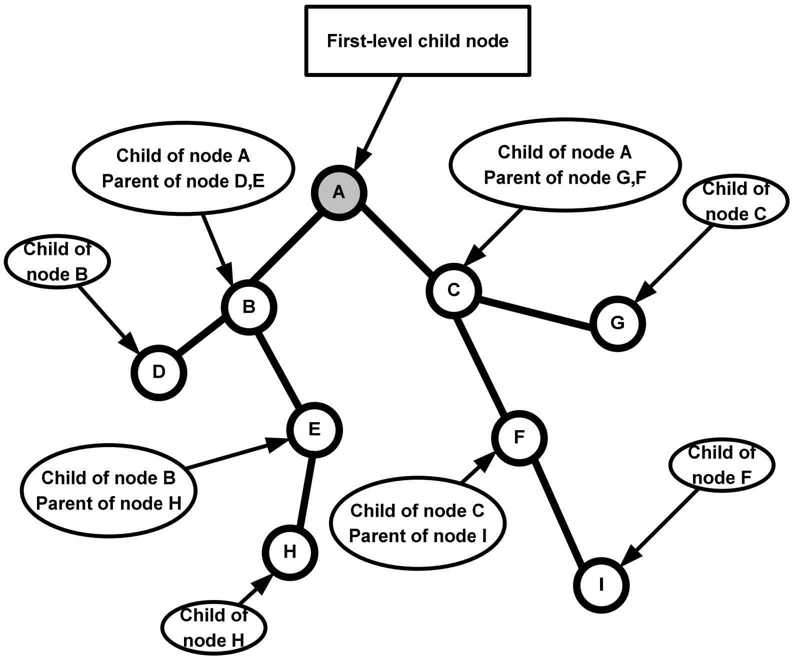
ต่อจากนี้ไป ผู้วิจัยจะใช้คำว่า node แทนคำว่ามิเตอร์ และคำว่า first-level child nodes หมายถึง nodes หรือมิเตอร์ที่อยู่ใกล้กับ base station ที่สามารถที่จะรับส่งข้อมูลให้กับ base station ได้โดยตรง นั่นคืออยู่ในรัศมีการสื่อสารด้วยสัญญาณวิทยุได้ ซึ่งในที่นี้จะถือว่า first-level child node แต่ละตัวเป็น root ของแต่ละ sub-tree ซึ่งตามทฤษฎีการสื่อสารด้วยคลื่นวิทยุแล้วจะเห็นว่าพลังงานของพวก first-level child nodes จะหมดเร็วกว่า nodes อื่น ๆ

งานวิจัยนี้ ผู้วิจัยได้พยายามเสนอวิธีการเพิ่มประสิทธิภาพให้กับอัลกอริทึมรับส่งข้อมูลในเครือข่ายมิเตอร์น้ำที่ใช้ระบบรับส่งข้อมูลแบบ wireless-sensor networks โดยการสร้าง tree แบบใหม่ที่ลดพลังงานที่ใช้ในการส่งข้อมูลจำนวนมาก ๆ เพื่อทำให้ระบบมีระยะเวลาใช้งานได้นานขึ้น และได้ตั้งชื่อวิธีการสร้าง tree ในลักษณะนี้ว่า *Enforced multiple sub-tree generation data gathering* (EFMST)

วิธี *enforced multiple sub-tree generation data gathering* เป็นวิธีการที่นำมาใช้ในการลดพลังงานที่ first-level child nodes โดยระบบจะพยายามเพิ่มจำนวนของ first-level child nodes และให้แต่ละ nodes ทำหน้าที่เป็น root เพื่อใช้ในการสร้าง tree ของตันเองขึ้นมา ซึ่งวิธีการนี้ใช้เพื่อกระจายจำนวนของข้อมูล (load) ที่ตัว first-level child nodes ดังนั้นพลังงานที่ใช้ในการเก็บข้อมูลที่ตัว first-level child nodes จึงมีค่าน้อยลง ซึ่งจะทำให้ระยะเวลาการทำงานของระบบโดยรวม

ยาวนานขึ้นด้วย โครงสร้างของ tree แสดงอยู่ในภาพที่ 18 ซึ่งข้อมูลจาก node ทั้งหมดจะถูกรวบรวมและส่งผ่านตามเส้นทางของ tree ผ่าน first-level child nodes ไปยัง base station

ในภาพที่ 18 จะเห็นว่า node A เป็น first-level child node ซึ่งสามารถส่งข้อมูลไปยัง base station ได้โดยตรง และ parent node จะหมายถึง node ที่รับข้อมูลจาก node อื่น ยกตัวอย่างเช่น node C เป็น parent node ของ node G และ F ส่วน child node หมายถึง node ที่ส่งข้อมูลไปให้ parent node ซึ่งถ้าพิจารณาภาพที่ 18 จะเห็นว่า node G และ F เป็น child nodes ของ node C ด้วยเหตุนี้ทำให้ node ใดก็ตามเป็นได้ทั้ง parent หรือ child node ได้ยกเว้น node A



ภาพที่ 18 โครงสร้างของ Tree

วิธีการสร้าง tree แบบ enforced multiple sub-tree data gathering

ขั้นตอนและวิธีการของ *enforced multiple sub-tree generation data gathering* ที่ใช้ในงานวิจัยนี้ เป็นไปตามขั้นตอนดังต่อไปนี้

ขั้นที่ 1 การหาจำนวนของ first-level Child nodes ของระบบ

จำนวนของ first-level child node สามารถหาได้จากจำนวนของ node ทั้งหมดในระบบกับจำนวนรอบของการเก็บข้อมูลที่ต้องการ ซึ่งจะทำให้แน่ใจได้ว่าระบบสามารถทำงานได้ตามระยะเวลาที่ต้องการ จากการทำงานของการอ่านค่าน้ำประปาทั่ว ๆ ไปจะทำการอ่านเดือนละครั้ง หรือ 12 ครั้งต่อปี

หลักการคำนวณเบื้องต้นเป็นดังนี้คือ กำหนดให้มิเตอร์หรือ node ทุกตัวมีพลังงานเริ่มต้นที่ $E_P = 1$ J (Joule), กำหนดให้พลังงานที่ใช้ในส่วนของภาคอิเล็กทรอนิค $(E_{Elec}) = 50$ *nJ/bit*, พลังงานที่ใช้ในภาคขยายสัญญาณวิทยุ $E_{Amp} = 100$ $pJ/bit/m^2$, จำนวนของข้อมูล $k = 2{,}000$ bits และระยะทางที่มากที่สุดที่สามารถส่งข้อมูลได้ $d = 50\,m$. ดังนั้นพลังงานที่ต้องการในการส่งข้อมูล k บิต จาก node เริ่มต้นไปยัง node ปลายทางที่ระยะทางมากที่สุดสามารถหาได้จากสมการที่ (8) คือ

$$E_{Rx} = 2000\left(50\times10^{-9}\right) = 0.1mJ \tag{10}$$

และ

$$E_{Tx} = 2000\left(50\times10^{-9}\right) + \left(2000\times50^2\times100\times10^{-12}\right) = 0.6mJ. \tag{11}$$

เนื่องจากสมมุติฐานที่ว่า hardware และ software ของทุก ๆ node มีลักษณะที่เหมือนกันทำให้ E_{Elec} และ E_{Amp} ของทุก nodes มีค่าเท่ากัน และที่สำคัญคือข้อมูลของมิเตอร์แต่ละตัวไม่สามารถนำมารวมกันได้ เพราะฉะนั้นถ้า first-level child node มีการส่งข้อมูลจาก node ทั้งหมดจำนวนข้อมูลจึงถูกรวมสะสมกันเพิ่มขึ้นตามจำนวนของ child node

ยกตัวอย่างเช่น ถ้า first-level child node หนึ่งตัวมีจำนวนของ child nodes ทั้งหมด N ตัว จำนวนข้อมูลทั้งหมดที่ first-level child node จะต้องส่งออกไปยัง Base station คือ $(N+1)k$ และจำนวนของข้อมูลที่จะต้องรับมาคือ Nk ซึ่งจากสมการที่ (8) และ (9) สามารถหาจำนวนของ child node ที่มากที่สุดที่ first-level child node หนึ่งตัวสามารถส่งของมูลทั้งหมดออกไปได้ในการส่งหนึ่งครั้ง ได้ดังสมการที่ (12) โดยที่ R_D เป็นจำนวนครั้งที่ต้องการในการเก็บข้อมูล

$$N_{\max} = \frac{\left(E_P/R_D\right) - E_{Tx}}{E_{Tx} + E_{Rx}} \tag{12}$$

จากสมการนี้ $(E_P/R_D)-E_{Tx}$ จะต้องมีค่าเป็นบวก เพราะถ้ามีค่าเป็นลบแสดงว่า battery ไม่สามารถจ่ายพลังงานให้กับ node ได้ตามจำนวนครั้งที่ต้องการได้

โดยทั่วไปแล้วระยะเวลาในการทำงานของ battery ของ electronic water meter จะขึ้นอยู่กับชนิดมิเตอร์ที่ผลิตขึ้นในตลาด ซึ่งโดยทั่วไปจะประมาณ 3-10 ปี (AMCO, 2003; Hydrometer, 2007) ถ้าเรากำหนดให้ระยะเวลาการทำงานของ Electronic water meter ที่เหมาะสมในทางการค้าคือ 5 ปี ก็หมายความว่า first-level child node ต้องสามารถทำงานส่งข้อมูลให้ได้อย่างน้อยสุด 60 ครั้ง ซึ่งถ้าพลังงานของ first-level child node มีค่าคงที่ เราจะคำนวณหาจำนวนของ child nodes ได้โดยใช้สมการที่ (12) คือ $N_{\max} = 22.9524 \approx 23$ ตัว

ใน Enforced Multiple sub-tree data gathering นี้ภาระที่ First-level child nodes จะถูกกระจายออกไปให้มากที่สุด เพื่อยืดอายุการทำงานของระบบให้ใกล้เคียงกับระยะเวลาที่ต้องการให้ได้ ซึ่งถ้าคิดในกรณีที่เลวร้ายที่สุดคือ ระยะทางในการส่งข้อมูลแต่ละครั้งมีค่ามากที่สุด จำนวนของ first-level child nodes ที่จำเป็นต้องมีในระบบจะหาได้จากจำนวนของ node ทั้งหมดและจำนวนครั้งในการเก็บข้อมูลที่ต้องการ ซึ่งสามารถเขียนได้ดังสมการที่ (13)

$$N_R = \frac{N_T}{N_{\max}}, \tag{13}$$

ในสมการที่ (13) นี้ N_R เป็นจำนวนของ First-level child nodes ที่ต้องการและ N_T เป็นจำนวนของ node ทั้งหมดในระบบ

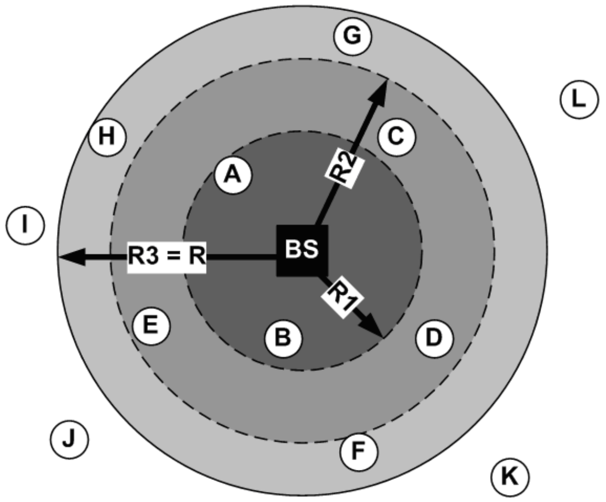
ในการหาจำนวนของ node ทั้งหมดที่ใช้ได้ในระบบ สามารถหาได้จากการนับมิเตอร์ที่ติดตั้งไปแล้วและอยู่ในรัศมีสื่อสารเลยก็ได้ หรือจะใช้อัลกอริทึม เช่น flooding algorithm (Tanenbaum, 1996) ก็ได้ ซึ่งไม่ใช่เป้าหมายหลักในงานวิจัยชิ้นนี้

ขั้นที่ 2 การเลือก first-level child node

ในการเลือก first-level child nodes หรือ root node ตัว base station คำนวนหาจำนวนของ first-level child nodes โดยใช้สมการที่ (13) จากนั้น base station จะกระจายสัญญาณออก

(broadcast) ด้วยระดับพลังงานที่ต่ำสุด และรอคำตอบจาก node ที่อยู่ในรัศมีสัญญาณ ซึ่ง node ที่ได้รับสัญญาณจะส่งสัญญาณตอบกลับให้กับ base station แต่ถ้า base station ไม่ได้รับสัญญาณตอบกลับ base station จะเพิ่มความแรงของการกระจายสัญญาณขึ้นอีกระดับหนึ่ง ทำให้รัศมีของการติดต่อสื่อสารเพิ่มขึ้น เพื่อหา node ที่ตอบรับกลับมาได้

เมื่อ base station ได้รับสัญญาณตอบกลับแล้ว base station จะเก็บค่า ของ ID ของ node ที่ส่งมาพร้อมด้วยระดับความแรงของสัญญาณ ซึ่งวิธีการนี้จะทำต่อไปเรื่อย ๆ จนกระทั่งได้จำนวนของ node ที่ส่งสัญญาณตอบกลับมา มีค่ามากกว่าค่าที่ได้คำนวนไว้แล้วคือ N_R โดย base station จะกำหนดให้ node ทั้งหมดที่ส่งสัญญาณมานี้เป็น first-level child nodes หรือ root nodes ของระบบทั้งหมด ซึ่งขั้นตอนที่กล่าวมาสามารถแสดงได้ดังภาพที่ 19



ภาพที่ 19 การเลือก first-level child nodes

จากตัวอย่างระบบที่มีลักษณะดังภาพที่ 19 และจำนวนของ root node (N_R) ที่คำนวณได้คือ 6 ตัว การทำงานของระบบจะเริ่มหา first-level child node โดย base station เริ่มส่งสัญญาณที่ระดับพลังงานต่ำสุดซึ่งจะครอบคลุมรัศมี R_1 โดยจะได้รับสัญญาณตอบกลับจาก node A และ B ซึ่งจำนวนของ first-level child node ขณะนี้คือ 2 ซึ่งยังน้อยกว่าค่า N_R เพราะฉะนั้น base station จะเพิ่มพลังงานในการส่งขึ้นไปอีกระดับหนึ่ง จนได้รัศมี R_2 โดยจะทำในลักษณะเดียวกันกับตอนแรก และตอนนี้จำนวนของ first-level child node ที่อยู่ในรัศมีคือ 5 เพราะจะรวม node C,D และ E เข้าไปด้วย

เนื่องจาก 5 มีค่าน้อยกว่า N_R Base station จึงเพิ่มระดับความแรงของสัญญาณขึ้นไปอีกจนกว่าจะมีจำนวน node ที่ส่งสัญญาณตอบกลับมากกว่าหรือเท่ากับ N_R โดยจากตัวอย่างนี้เมื่อเพิ่มระดับสัญญาณไปอีกระดับหนึ่ง จะมีจำนวน node ที่ตอบกลับมารวมทั้งสิ้นกับของเดิมเป็น 8 ตัว ซึ่งจะมากกว่า N_R นั่นก็คือ node A ถึง H

ในกรณีที่ Base station เพิ่มระดับสัญญาณสูงสุดแล้ว แต่จำนวน node ที่ตอบกลับยังมีค่าน้อยกว่าค่า N_R เนื่องจากไม่มี node ที่สามารถส่งสัญญาณตอบรับมาถึง BS ได้อีกแล้ว ตัว Base station จะยอมให้เฉพาะแค่ node ทั้งหมดเท่าที่ตอบกลับมาแล้ว เป็น first-level child node ถึงแม้จะมีจำนวนน้อยกว่า N_R ก็ตาม

ขั้นที่ 3 วิธีการสร้างเส้นทางภายใน sub-tree

วิธีการสร้าง tree ของแต่ละ sub-tree จะกระทำโดยกำหนดสถานะ (status) ของแต่ละ node เป็น 4 สถานะคือ

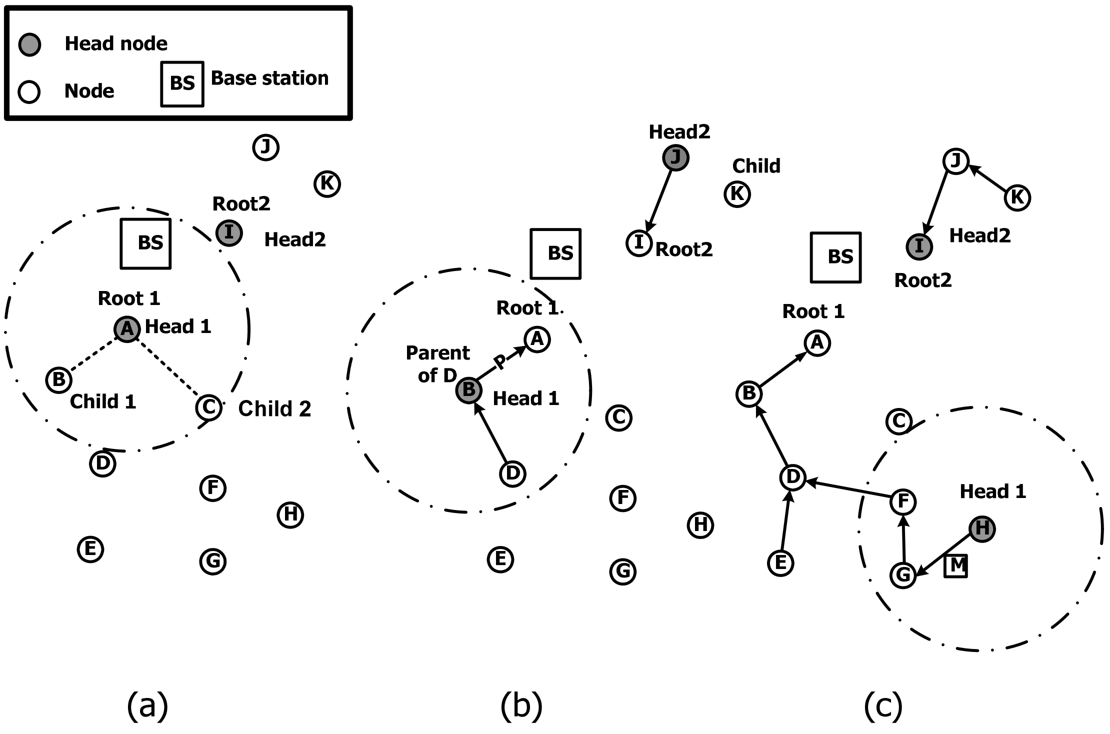
1. Root state คือ สถานะของ node ที่ทำหน้าที่เป็น fist-level child node ใน PEDAP หรือในงานวิจัยนี้จะเรียกว่า Root ของ Sub-tree

2. Head state คือ สถานะของ node ที่กำลังทำการค้นหา child nodes โดย node ในสถานะนี้จะเปลี่ยนเป็น parent state ถ้าเจอ child nodes

3. Parent state คือสถานะของ node ที่ทำหน้าที่เป็น parent คือมี node ที่อยู่ถัดไปเป็น child node

4. Child state คือสถานะของ node ที่ทำหน้าที่เป็น child node ของ parent node ตัวใดตัวหนึ่ง

ขั้นตอนการสร้าง tree จะแสดงได้ดังภาพที่ 20



ภาพที่ 20 ขั้นตอนในการสร้าง Tree ในงานวิจัย

เมื่อเริ่มต้นทำงาน สถานะของ node ทั้งหมดจะอยู่ใน child state หลังจากทำการหา first-level child node เสร็จสิ้นแล้ว node ที่เป็น first-level child nodes จะอยู่ในสถานะของ root state และจะเรียกว่า root node โดยแต่ละ root node จะสลับกันส่ง setup message ออกไปดังภาพที่ 20(a) ซึ่งในรูปจะเห็นว่า root nodes คือ node A และ I และถ้า root node ส่งสัญญาณออกไปแต่ไม่ได้รับ reply message ตอบรับภายในระยะเวลาที่กำหนด (timeout) root node ตัวนี้จะจบขั้นตอนในการสร้าง tree

Node ที่อยู่ใน child state ที่ได้รับ setup message จะส่ง reply message กลับไปให้ node ที่ส่ง setup message มา ตัวอย่างเช่นในภาพที่ 20(a) เมื่อ root node A ได้รับ reply message จาก node B และ C , root node A จะทำการเลือก node เพียงหนึ่งตัวมาเป็น child node โดยดูจากค่าของ signal strength หรือ routing cost ที่คำนวนได้จาก Radio Signal Strength Indicator (RSSI) และตัว

แปรอื่น ๆ เพื่อที่จะนำมาใช้ในการยืนยัดว่าเมื่อมีการส่งข้อมูลแล้วจะใช้พลังงานน้อยที่สุด ในกรณีนี้ child node คือ node B เพราะว่าระยะทางจาก node A ไป B สั้นกว่าระยะทางจาก A ไป C

เมื่อ root node ทำการเลือก child node แล้วจะส่ง set parent message ให้กับ child node ที่ถูกเลือก เมื่อ child node ที่ถูกเลือกได้รับ set parent message แล้วจะทำการบันทึกเลข ID ของตัวที่ส่งมาลงใน memory ที่เก็บค่า ID ของ parent node จากนั้น ตัว child node จะเปลี่ยนสถานะเป็น head node state ดังที่แสดงไว้ในภาพที่ 20(b)

แต่ละ head node จะทำการค้นหา node ที่ใกล้ที่สุดมาเป็น child node อีกทอดหนึ่งโดยการส่งสัญญาณ setup message ออกในลักษณะเดียวกันกับ parent-node ของตัวมัน ซึ่งขั้นตอนนี้จะทำไปเรื่อย ๆ จนกระทั่ง sub-tree ทั้งหมดได้ถูกสร้างขึ้นจนถึง node สุดท้ายที่สื่อสารกันได้

ตัวอย่างในภาพที่ 20(b), node B จะส่งสัญญาณ setup message ออกไปและจะรอรับ reply message ถ้าได้รับ reply message จะเปลี่ยนสถานะเป็น parent node และส่ง set parent message ให้กับ child node ที่ถูกเลือก ในภาพที่ 20(b) node D รับ Set parent message จาก node B ดังนั้น node B เป็น parent ของ node D

ถ้าหาก head node ไม่ได้รับ reply message ภายในเวลาที่กำหนด head node จะส่ง end message ให้กับ parent ของมัน ซึ่งสามารถบอกได้ว่า node นี้เป็น node สุดท้ายภายใน sub-tree ดังแสดงไว้ในภาพที่ 20(c), node H ส่งสัญญาณ setup message แต่ไม่ได้รับ reply message ในระยะเวลาที่กำหนด เพราะฉะนั้น node H จะส่ง end message ให้กับ parent ของมันและเป็น node สุดท้ายใน tree

ถ้า Parent node ได้รับ end message แล้ว ตัว parent node จะเปลี่ยนสถานะเป็น head node อีกครั้งและจะทำการค้นหา child node ต่อไปอีก เช่น node D ในภาพที่ 20(c) รับ end message จาก node E ก่อนเพราะ E อยู่ใกล้ D มากกว่า F จากนั้นตัว node D จะเปลี่ยนสถานะจาก parent state เป็น head state และทำการส่ง setup message เพื่อหา child node ตัวใหม่อีกครั้งหนึ่ง ซึ่งได้คำตอบจาก F จากนั้นกระบวนการดำเนินต่อไปจนถึง node H ในขณะที่ E เป็นตัวสุดท้ายในกิ่งหนึ่งของต้นไม้นี้นอกจาก H เมื่อไม่มี node เพิ่มแล้ว D จะส่ง end Message ไปให้ root node เพื่อแสดงว่าจบ

การสร้าง tree ของ root node นี้เสร็จสิ้นแล้ว ขั้นตอนการสร้าง Tree ของระบบทั้งหมดจะเสร็จสิ้นเมื่อตัว base station ได้รับ end message จาก root node ทุกตัว

ขั้นตอนในการเพิ่ม node เข้าไปใน Tree จะเริ่มขึ้นทีละ root node สลับกันไปจนกระทั่งทุก node ที่อยู่ในระบบถูกเพิ่มเข้าไปใน Sub-tree โดยวิธีการนี้ได้พยายามสร้าง minimum spanning tree จากแต่ละ root node ของ sub-tree ให้ประหยัดพลังงานที่สุด และวิธีการนี้ยังสามารถที่จะเพิ่ม node เข้าไปในระบบได้ในภายหลังโดยการติดตั้งมิเตอร์วัดน้ำเพิ่ม จากนั้นสั่งให้ restart การสร้างเส้นทาง ซึ่งต้องทำทุกครั้งที่มีการเพิ่ม node เข้าไปในระบบ base station สามารถส่งคำสั่งเพื่อที่จะทำการ restart ระบบที่เวลาใดก็ได้

ขั้นที่ 4 การเก็บข้อมูล (Data gathering)

วิธีการเก็บข้อมูลจากมิเตอร์มายัง base station เริ่มต้นที่ตัว base station จะทำการส่ง gather message ให้กับ ตัว root node เมื่อ root node ได้รับจะส่งต่อให้กับตัวลูก (child node) ของมัน และตัวลูกก็จะส่งต่อให้กับตัวลูกของมันอีก เป็นทอด ๆ ซึ่งเมื่อ node ตัวไหนได้รับ message แล้วไม่มีตัวลูกต่อไปอีก node ตัวนั้นจะส่งข้อมูล (data message) ของตัวเองให้กับตัวแม่ (parent node) เมื่อ node ไหนได้รับข้อมูล (data message) จากตัวลูก, node นั้นจะนำข้อมูลที่ได้มารวมเข้ากับข้อมูลของตัวเอง และส่งข้อมูลที่มีขนาดยาวมากขึ้นให้กับตัวแม่ แต่ละ node จะทำเช่นนี้ไปเรื่อยๆ จนกระทั่งถึง root node ซึ่งจะเป็นข้อมูลที่ยาวที่สุด ประกอบด้วยข้อมูลของทุก node ภายใน sub-tree จากนั้นแต่ละ root node จะส่งข้อมูลทั้งหมด รวมทั้งข้อมูลของตัวมันเองให้กับ base station เป็นขั้นตอนสุดท้าย

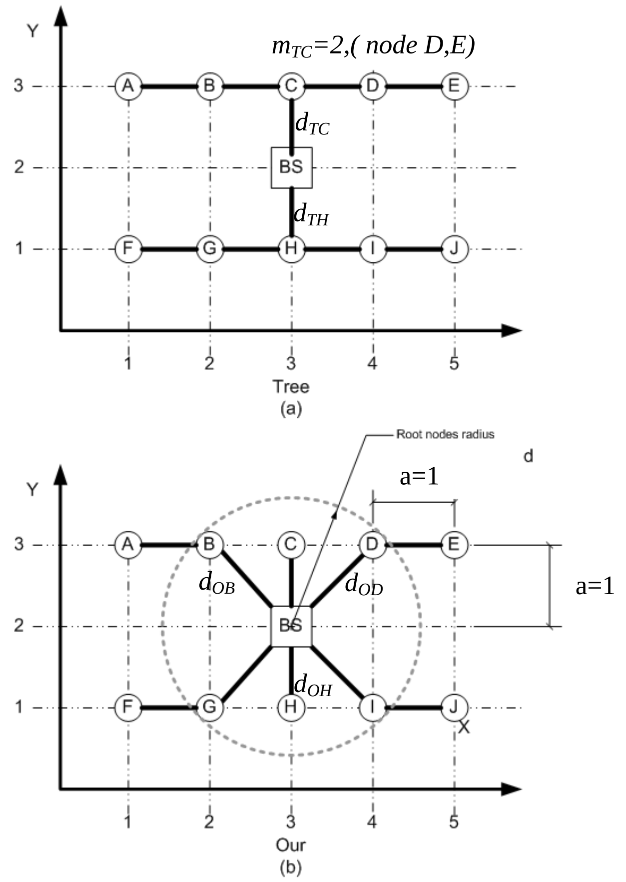
2. พิสูจน์เปรียบเทียบประสิทธิภาพของอัลกอริทึมที่พัฒนาขึ้นกับระบบอื่น

ประสิทธิภาพของอัลกอริทึมที่ได้ของแต่ละอัลกอริทึมจะขึ้นอยู่กับลักษณะการวางตำแหน่งของ sensor ซึ่งเมื่อมีการวางตำแหน่งที่แตกต่างกัน ประสิทธิภาพที่ได้ก็จะแตกต่างกันด้วย ทำให้ในการวิเคราะห์ระบบ เราจะกำหนดให้ตำแหน่งของแต่ละ node โดยการวางตำแหน่งแบบกริด (grid) 1 หน่วย และแบบ วงกลม (circle) 1หน่วย เพื่อให้ง่ายในการวิเคราะห์เปรียบเทียบ

เมื่อพิจารณาการวางระบบแบบกริดดังแสดงในภาพที่ 21 ผู้วิจัยได้กำหนดตำแหน่งให้มีลักษณะคล้ายกับระบบการเก็บค่าน้ำให้มากที่สุด โดยสมมุติว่า Base station แบบเคลื่อนที่ได้ (เช่น รถเก็บข้อมูล) เคลื่อนที่มาหยุดอยู่บนถนนเพื่อเก็บค่าจากมิเตอร์ในบริเวณบ้านใกล้เคียง ในสภาวะแวดล้อมนี้ เส้นทางในการเก็บข้อมูลที่ได้จากอัลกอริทึมของ PEDAP minimum spanning tree (MST) แสดงได้ดังภาพที่ 21(a) และเส้นทางในการเก็บข้อมูลโดยใช้อัลกอริทึมตามงานวิจัยนี้จะได้ดังภาพที่ 21(b)

ในการเก็บข้อมูลด้วยวิธีการตามงานวิจัยเริ่มต้น ระบบจะต้องหาจำนวนของ root node หรือ first-level child node ก่อน ซึ่งถ้ารัศมีการติดต่อสื่อสารของ base station เป็นดังเส้นปะในภาพที่ 21(b) จะพบว่าจำนวนของ root node หรือ first-level child node ที่ได้มีจำนวน 6 ตัวนั้นคือ node B, C, D, G, H และ I ซึ่ง lifetime ของระบบจะขึ้นอยู่กับจำนวนของ child node ของ first-level child node i (m_i) ที่มากที่สุดในระบบ เพราะต้องส่งข้อมูลไปยัง base station มากที่สุด จากภาพที่ 21(b) m_i ที่มากที่สุดคือ 2 ที่ first-level child node B, D, G และ I ต่อจากนั้นเมื่อทำการสร้างเส้นทางการเก็บข้อมูลจาก first-level child node เหล่านี้แล้วจะแสดงเส้นทางที่ได้ด้วยเส้นหนาในภาพที่ 21 (b) ในการวิเคราะห์จะกำหนดให้

m_{Oi}	=	จำนวน child node ในเส้นทางที่มี node i เป็น First-level child node ในระบบที่ใช้วิธีของงานวิจัยนี้
m_{Ti}	=	จำนวน child node ในเส้นทางที่มี node i เป็น First-level child node ในระบบที่ใช้วิธีของ PEDAP MST
L_{SO}	=	Life Time ของระบบที่ใช้วิธีของงานวิจัยนี้
L_{ST}	=	Life Time ของระบบที่ใช้วิธีของ PEDAP MST
L_i	=	Life Time ของ node i
E^i_p	=	พลังงานเริ่มต้นของ node i
E^i_{Tx}	=	พลังงานที่ใช้ในการส่งข้อมูลของ node i
E^i_{Rx}	=	พลังงานที่ใช้ในการรับข้อมูลของ node i
d_{Ti}	=	ระยะทางที่มากที่สุดที่ node i สามารถส่งข้อมูลได้ในระบบที่ใช้วิธีของ PEDAP MST
d_{Oi}	=	ระยะทางที่มากที่สุดที่ node i สามารถส่งข้อมูลได้ในระบบที่ใช้วิธีของงานวิจัยนี้



ภาพที่ 21 แสดงตัวอย่างของ fist-level child node และเส้นทางการส่งข้อมูลแบบกริด (a) PEDAP spanning tree algorithm. (b) วิธีการของงานวิจัย

จากภาพที่ 21(a) node C และ H มีจำนวน child node มากที่สุดและมีค่าเท่ากัน ($m_{TC}= m_{TH}$) ทำให้ระยะเวลาในการเก็บข้อมูลของระบบที่ได้จากการใช้อัลกอริทึมของ PEDAP minimum spanning tree จะขึ้นอยู่กับ 2 node นี้ ซึ่งสามารถคำนวณ lifetime ของระบบจาก lifetime ของ node C และ H ซึ่งเขียนได้ดังสมการที่ (14) ในที่นี้จึงวิเคราะห์ที่ node C เท่านั้น

$$Ls_T = L_C = L_H \tag{14}$$

$$\begin{aligned} Ls_T &= \frac{E_P^C}{E_{Tx}^C + E_{Rx}^C} \\ &= \frac{E_P^{TC}}{E_{elec} m_{TC} k + E_{amp} m_{TC} k\, d_{TC}^2 + E_{elec}(m_{TC}-1)k} \end{aligned} \tag{15}$$

ถ้าใช้อัลกอริทึมในงานวิจัยนี้เส้นทางเก็บข้อมูลจะได้ดังภาพที่ 21(b) นั้นคือ lifetime ของระบบจะขึ้นอยู่กับ node B, D, G และ I ซึ่งมีจำนวนของ Child node เท่ากันและมากที่สุดในระบบ ($m_{OB} = m_{OD} = m_{OG} = m_{OI}$) เพราะฉะนั้นสามารถคำนวณหาระยะเวลาในการเก็บข้อมูล (system lifetime) ที่ได้ดังนี้

$$Ls_O = L_B = L_D = L_G = L_I \tag{16}$$

$$\begin{aligned} Ls_O &= \frac{E_P^B}{E_{Tx}^B + E_{Rx}^B} \\ &= \frac{E_P^B}{E_{elec} m_{OB} k + E_{amp}\ m_{OB} k\ d_{OB}^2 + E_{elec}(m_{OB} - 1)k} \end{aligned} \tag{17}$$

เมื่อนำเอาค่าระยะเวลาของทั้งสองอัลกอริทึมมาเปรียบเทียบกันโดยนำสมการที่ (15) กับ (17) มาหารกันจะได้ $\frac{Ls_T}{Ls_O}$

$$\frac{Ls_T}{Ls_O} = \frac{E_{elec}(2m_{OB} - 1)k + E_{amp} m_{OB} k d_{OB}^2}{E_{elec}(2m_{TC} - 1)k + E_{amp} m_{TC} k\ d_{TC}^2} \tag{18}$$

ระยะทางและตำแหน่งของแต่ละ node ที่จะนำมาคำนวนสามารถแสดงได้ดังภาพที่ 21 โดยสามารถหาระยะทางของแต่ละเส้นทางได้คือ $d_{TC} = d, d_{OB} = d\sqrt{2}, m_{OB} = 2$ และ $m_{TC} = 5$ นำมาแทนค่าในสมการ (18) จะได้สมการที่ (19)

$$\begin{aligned} \frac{Ls_T}{Ls_O} &= \frac{E_{elec}(2m_{OB} - 1)k + E_{amp}\ m_{OB} k d_{OB}^2}{E_{elec}(2m_{TC} - 1)k + E_{amp} m_{TC} k d_{TC}^2} \\ &= \frac{E_{elec}(4-1)k + E_{amp}\ 2\,k\left(d\sqrt{2}\right)^2}{E_{elec}(10-1)k + 5\,E_{amp} k\ d^2} \end{aligned} \tag{19}$$

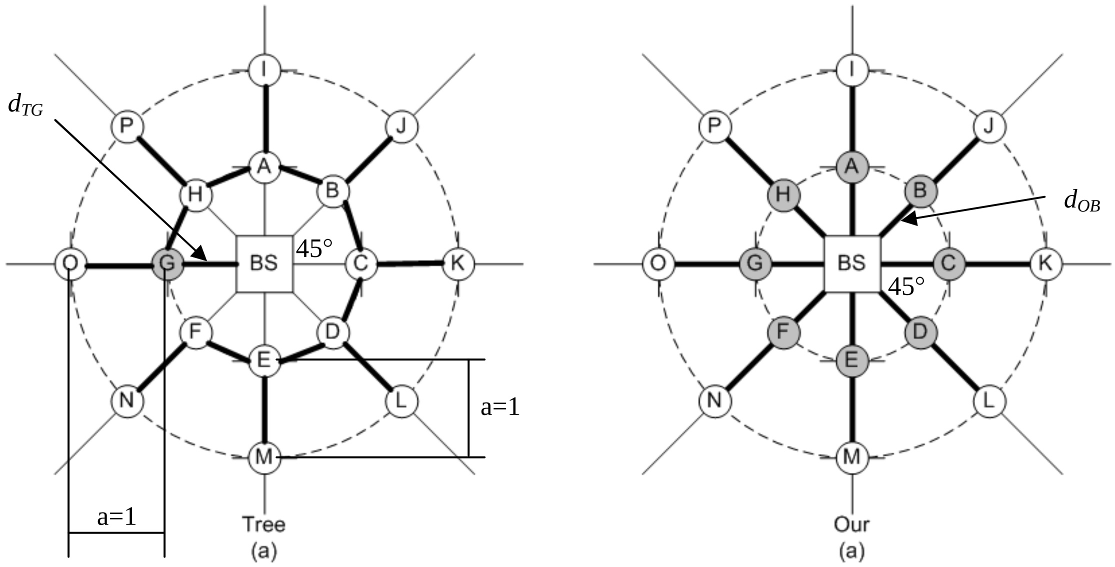
ทั้งสองอัลกอริทึมใช้ค่าตัวแปรในสมการพลังงานค่าเดียวกันนั้นคือ $k=1, E_{elec}=50$ *nJ/bit*, E_{amp}=100 pJ/bit/m^2 และ d=1 ทำให้เมื่อนำมาแทนค่าในสมการที่ (19) แล้วจะได้

$$\frac{Ls_T}{Ls_O} = 0.334148$$
$$\approx \frac{1}{3}$$

จากผลที่ได้พบว่าค่าระยะเวลาการทำงานของระบบที่ใช้วิธีการหาเส้นทางแบบ PEDAP minimum spanning tree ตามรูปแบบดังแสดงในภาพที่ 21(a) จะให้ระยะเวลาที่น้อยกว่าการใช้วิธีการหาเส้นทางตามงานวิจัยฉบับนี้ประมาณ 3 เท่า ซึ่งสามารถเขียนให้อยู่ในรูปทั่วไปได้ ดังนี้

$$Ls_T = 3Ls_O$$

ในการวิเคราะห์ระบบโดยกำหนดให้ค่าตัวแปรพลังงานในระบบเหมือนเดิม แต่ใช้การวางตำแหน่งของ node เป็นแบบวงกลม ดังภาพที่ 22 ในการวิเคราะห์ โดยใช้อัลกอริทึมของงานวิจัยฉบับนี้ จะได้จำนวนของ root node เป็น 7 ตัวดังภาพที่ 22(b) นั้นคือ node A, B, C ,D ,E ,F และ G แต่ในระบบที่ใช้วิธีการ PEDAP minimum spanning tree จะได้ root node เพียงตัวเดียวดังแสดงในภาพที่ 22(a)



ภาพที่ 22 แสดงลักษณะการว่างตำแหน่งแบบวงกลม

พิจารณาในการวางตำแหน่งดังรูปที่ 22 ด้วยวิธีการเดียวกันกับที่กล่าวมาและหาค่าตัวแปรต่าง ๆ ที่ใช้ในการคำนวนหา lifetime ของระบบได้ดังนี้คือ $d_{TC} = d_{OB} = d$, $m_{OB} = 2$ และ $m_{TC} = 16$ นำค่าระยะเวลาของทั้งสองอัลกอริทึมมาแทนค่าลงในสมการที่ (18) จะได้

$$\begin{aligned}\frac{Ls_T}{Ls_O} &= \frac{E_{elec}(2m_{OB}-1)k + E_{amp}m_{OB}k\,d_{OB}^2}{E_{elec}(2m_{TC}-1)k + E_{amp}m_{TC}k\,d_{TC}^2} \\ &= \frac{E_{elec}(4-1)k + E_{amp}2\,k\,d^2}{E_{elec}(32-1)k + 16E_{amp}k\,d^2} \\ &= 0.096897 \\ &\approx 0.1\end{aligned}$$

ในกรณีนี้ค่าจำนวนของระยะเวลาในการเก็บข้อมูลของงานวิจัยนี้จะดีกว่าระบบที่ใช้การหาเส้นทางแบบ PEDAP spanning tree ประมาณ 10 เท่า

$$\therefore Ls_O = 10Ls_T$$

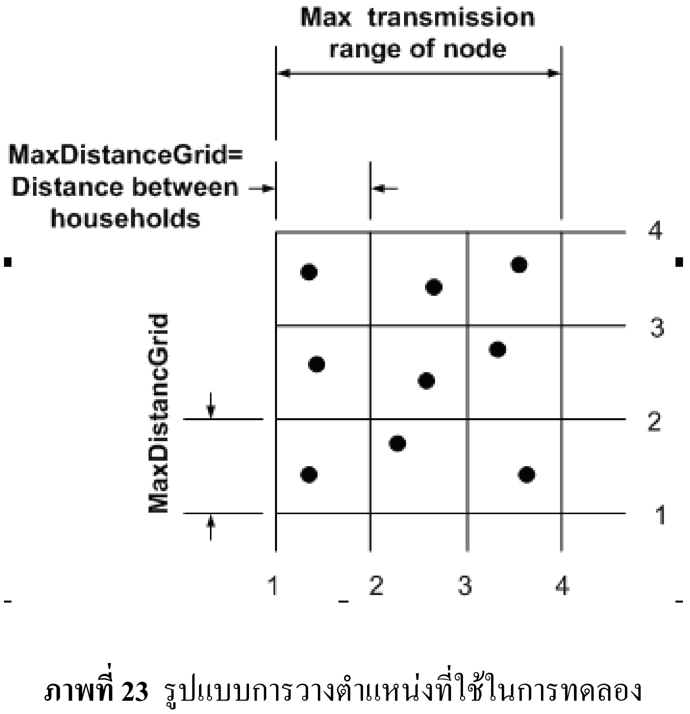
โดยค่าที่ได้จากทั้งสองตัวอย่างขั้นต้นนี้ แสดงให้เห็นว่าระบบที่ใช้การหาเส้นทางการเก็บข้อมูลที่ใช้อัลกอริทึมของงานวิจัยนี้ให้ผลที่ดีกว่าการใช้อัลกอริทึมของ PEDAP minimum spanning tree ในการวางตำแหน่งของ node แบบภาพที่ 21 และ 22 ซึ่งจากการวิเคราะห์พบว่าผลของระยะเวลาการเก็บข้อมูลมีผลมาจากเส้นทางที่ใช้ในการเก็บข้อมูลและลักษณะตำแหน่งของ node ในระบบด้วย ดังนั้นในงานวิจัยชิ้นต่อไปควรจะทำการหาประสิทธิภาพของการเก็บข้อมูลจากระบบหลาย ๆ ลักษณะ (topology) เพื่อให้วิธีการที่ได้มีประสิทธิภาพใกล้เคียงกับความเป็นจริงยิ่งขึ้น

3. วิธีการจำลองระบบ

3.1. การจัดระบบเพื่อการจำลองการทำงาน (simulation setup) มีขั้นตอนดังนี้

ในการจำลอง (Simulation) ได้ใช้ Matlab 7.0 เพื่อจำลองระบบและหาประสิทธิภาพการทำงาน โดยใน simulation นั้นได้ออกแบบสิ่งแวดล้อมให้คล้ายกับระบบอ่านค่าน้ำประปาตามหมู่บ้านจัดสรร แบบบ้านเดี่ยวในประเทศไทยให้มากที่สุด ซึ่งผลของการ simulation ที่นำมาเปรียบเทียบระหว่างวิธีการที่คิดขึ้นกับวิธีการของ PEDAP นั้นเป็นผลที่วิเคราะห์จากสถานะการณ์เดียวกัน นั้นก็คือข้อมูลที่ได้จากมิเตอร์ (sensor node) ไม่สามารถนำมารวมเพื่อย่อขนาดของข้อมูลได้ ดังที่แสดงไว้ในสมมุติฐานที่ 6)

โครงสร้างในการวางมิเตอร์ (sensor node) ในการ simulation นั้นจะวางเป็นลักษณะแบบกริด (grid) ดังภาพที่ 23 กำหนดให้ distance between house holds เท่ากับ 10 m



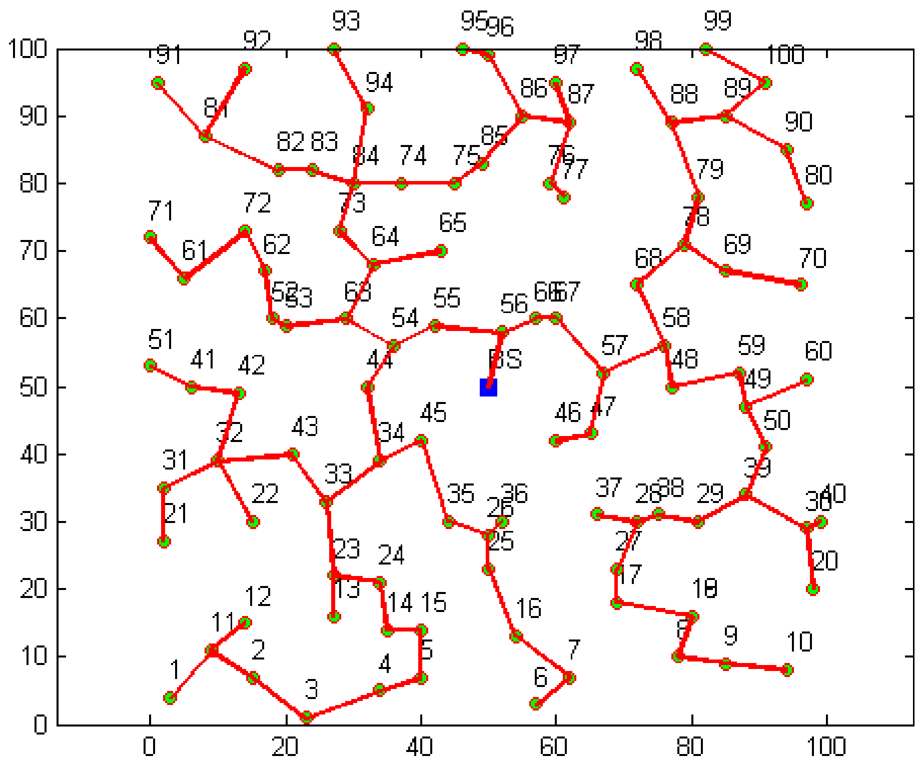
ภาพที่ 23 รูปแบบการวางตำแหน่งที่ใช้ในการทดลอง

ขนาดของแต่ละ Block ภายในกริดนั้นจะมีพื้นที่ประมาณ 10x10 m^2 (ประมาณบ้านเดี่ยว 1 หลัง) และแต่ละ Block จะมีมิเตอร์อยู่ โดยจะทำการสุ่ม ตำแหน่งขึ้นมาภายในพื้นที่ของ Block นั้น

เพื่อให้สมจริงกับตำแหน่งมิเตอร์ของแต่ละบ้าน ซึ่งในการทดลองจะทดลองที่ขนาดของกริดเพิ่มขึ้นจาก 10x10 $block^2$ ไปจะถึง 25x25 $block^2$ เพราะฉะนั้นทำให้จำนวนของมิเตอร์ในระบบมีจำนวนเพิ่มขึ้นจาก 100 ตัวไปจะกระทั่งถึง 625 ตัว โดยได้กำหนดรัศมีการส่งข้อมูลสูงสุดไว้ที่ 50 เมตร พลังงานเริ่มต้นของแต่ละมิเตอร์อยู่ที่ 1, 0.5 และ 0.25 *J* ตามงานวิจัยที่ (Heinzelman *et al.*, 2000). เพื่อดูแนวโน้มว่าถ้าพลังงานมากขึ้นผลลัพธ์ที่ได้จะเป็นอย่างไร และค่าของตัวแปรที่ใช้ในสมการที่ (8) และ (9) คือ พลังงานสูญเสียในระบบอิเล็กทรอนิคส์ E_{Elec}=50 *nJ/bit*, พลังงานสูญเสียในส่วนของภาคขยายสัญญาณ E_{Amp}= 100 $pJ/bit/m^2$ และจำนวนของข้อมูล *k*=2,000 bits

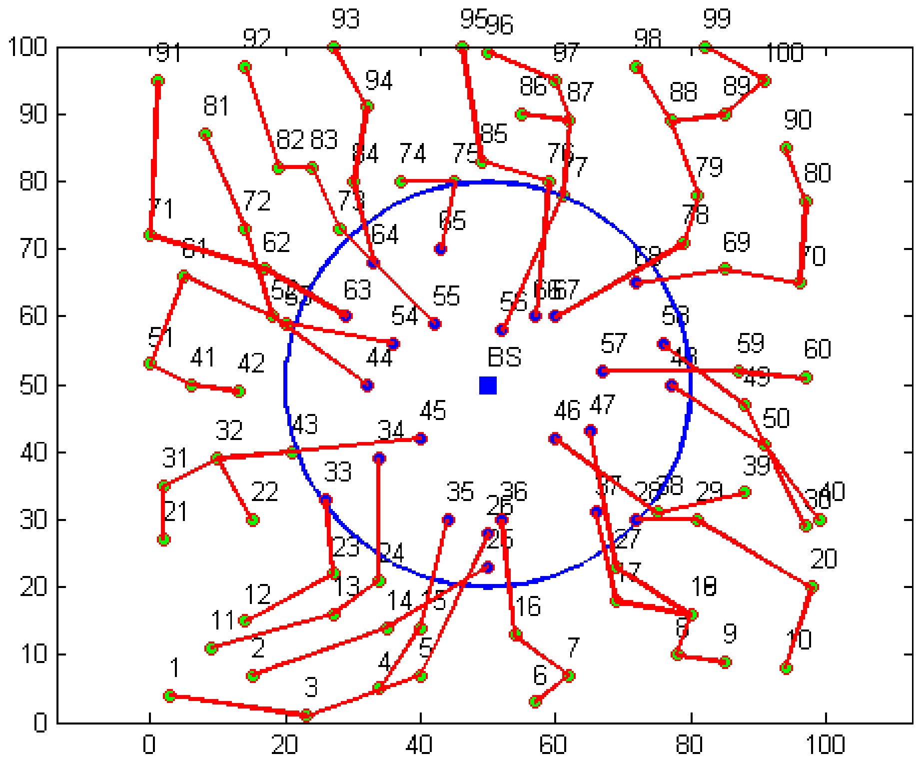
การรวบรวมข้อมูลหนึ่งรอบ (Round) หมายความถึง base station ได้รับข้อมูลจากมิเตอร์ในระบบจนครบทุกตัว ซึ่งเทียบได้กับการเก็บค่าน้ำในพื้นที่ให้บริการ 1 เดือน จากนั้นจะทำการคำนวนค่าเฉลี่ยของจำนวนครั้งที่ base station ได้รับข้อมูลมาเปรียบเทียบกันระหว่าง 2 อัลกอริทึม ซึ่งจำนวนรอบ (round)ในการ simulation แต่ละครั้ง จะนับถึงครั้งสุดท้ายก่อนครั้งที่ base station ได้รับข้อมูลไม่ครบถ้วน แสดงว่ามิเตอร์ตัวใดตัวหนึ่งในระบบไม่สามารถส่งข้อมูลมาให้กับ Base station ได้เนื่องจากพลังงานแบตเตอรี่ไม่พอ ถือว่าสิ้นสุดอายุใช้งานของระบบ sensor network ในการจำลองครั้งนั้น

ผู้วิจัยได้ทำการจำลองในลักษณะเดียวกัน 5 ครั้งต่อหนึ่ง algorithm และนำเอาค่าจำนวน round ที่ได้ทั้ง 5 ครั้งมาหาค่าเฉลี่ยเพื่อใช้ในการเปรียบเทียบ



ภาพที่ 24 เส้นทางข้อมูลหรือ tree ที่ได้จากวิธีการของ PEDAP

จากภาพที่ 24 จะเห็นว่าเส้นทางที่ได้จากการใช้ PEDAP อัลกอริทึมนี้ มีโหนดที่เชื่อมต่อกับ base station อยู่แค่ 2 ตัวทำให้ node สองตัวนี้ต้องรับภาระในการส่งข้อมูลที่มีจำนวนมากให้กับ base station ซึ่งทำให้ node สองตัวนี้พลังงานหมดเร็วกว่า node อื่นๆ เป็นผลให้อายุใช้งานของระบบสั้นลง



ภาพที่ 25 เส้นทางหรือ tree ที่ได้จากวิธีการ enforced Multiple sub-tree ของผู้วิจัย

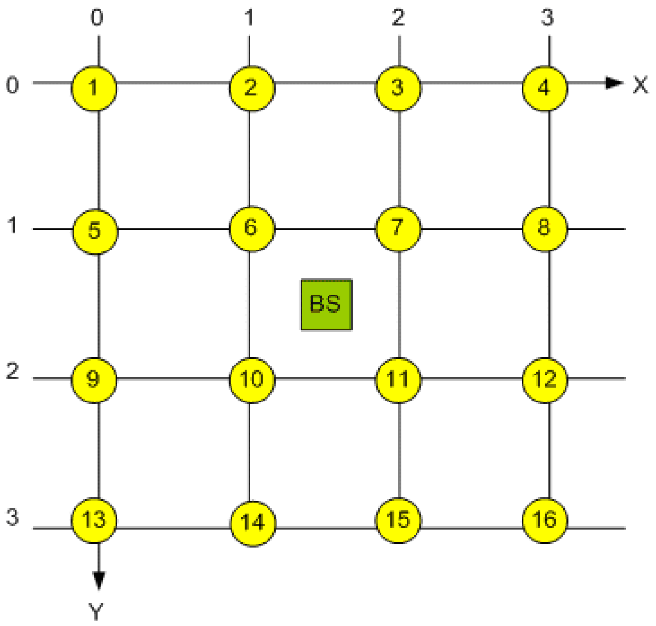
จากภาพที่ 25 เป็นเส้นทางที่ได้จากการใช้ enforced multiple sub-tree ในการเลือกเส้นทาง โดยวงกลมใหญ่คือรัศมีการรับส่งข้อมูลของ base station กับ nodes รอบข้าง ที่เท่ากับการจำลองครั้งแรก แต่จะเห็นว่าแตกต่างกับของ PEDAP ตรงที่จำนวน node ที่ส่งข้อมูลให้กับ base station นี้มีจำนวนมากขึ้น โดยจำนวนข้อมูลที่ส่งนี้ก็จะเล็กลงทำให้สามารถยืดอายุการใช้งานของระบบได้มากขึ้น

3.2. การจัดระบบเพื่อการทดลองระบบที่พัฒนาขึ้นกับ sensor boards ของจริง (Experiment Setup)

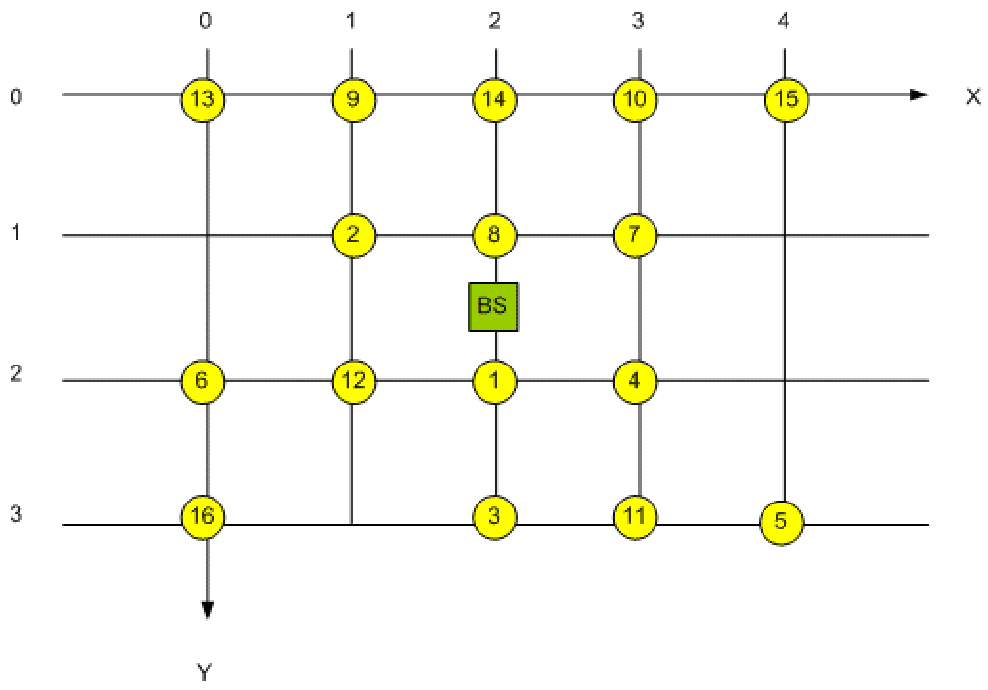
ในการทดลอง experiment กับ hardware จริงนั้น เนื่องจากพลังงานที่ได้จาก battery จริงๆ มีพลังงานที่สูงมาก ทำให้ถ้าจะให้การคำนวณเป็นไปตามสมการที่ (8) และ (9) จำนวน Sensor ที่ต้องใช้ในการทดลองจำเป็นต้องมีจำนวนมาก แต่เนื่องจากจำนวน Sensor ที่มีอยู่มีจำนวนจำกัดเพียง 16 ตัว ทำให้ผู้วิจัยจำเป็นต้องทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมตามจำนวน sensor ที่

มีอยู่ และทำการทดลองโดยการบังคับเพิ่มจำนวนของ first-level child node จาก 1,2 และ 3 node เพื่อเปรียบเทียบพลังงานที่ใช้ไปในการส่งข้อมูลของ first-level child node จากจำนวนน้อยไปสู่มาก และบังคับสร้างเส้นทางหรือ tree ให้มีลักษณะที่สามารถเป็นตัวแทนคุณลักษณะของระบบจริงได้ เพื่อแสดงให้เห็นว่าวิธีการที่เสนอมานี้ใช้พลังงานลดลงจริงหรือไม่ เมื่อจำนวน first-level child node มากขึ้น โดยมีขั้นตอนการทดลองดังนี้

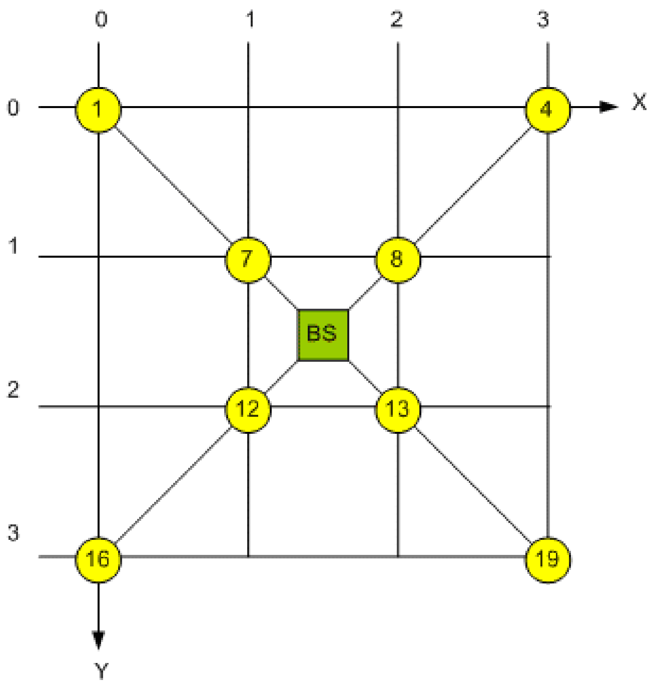
3.2.1 ทำการทดลองกับบอร์ด TmoteSky ซึ่งทำงานด้วยระบบปฏิบัติการ TinyOS จำนวน 16 ตัว โดยวางตัวบอร์ดในลักษณะต่าง ๆ 3 แบบคือ กริด (grid) , สุ่ม (random) และ ทแยง (diagonal) ดังรูป



ภาพที่ 26 การวางตำแหน่ง sensor แบบกริด



ภาพที่ 27 ตัวอย่างการวางตำแหน่ง sensor แบบสุ่ม



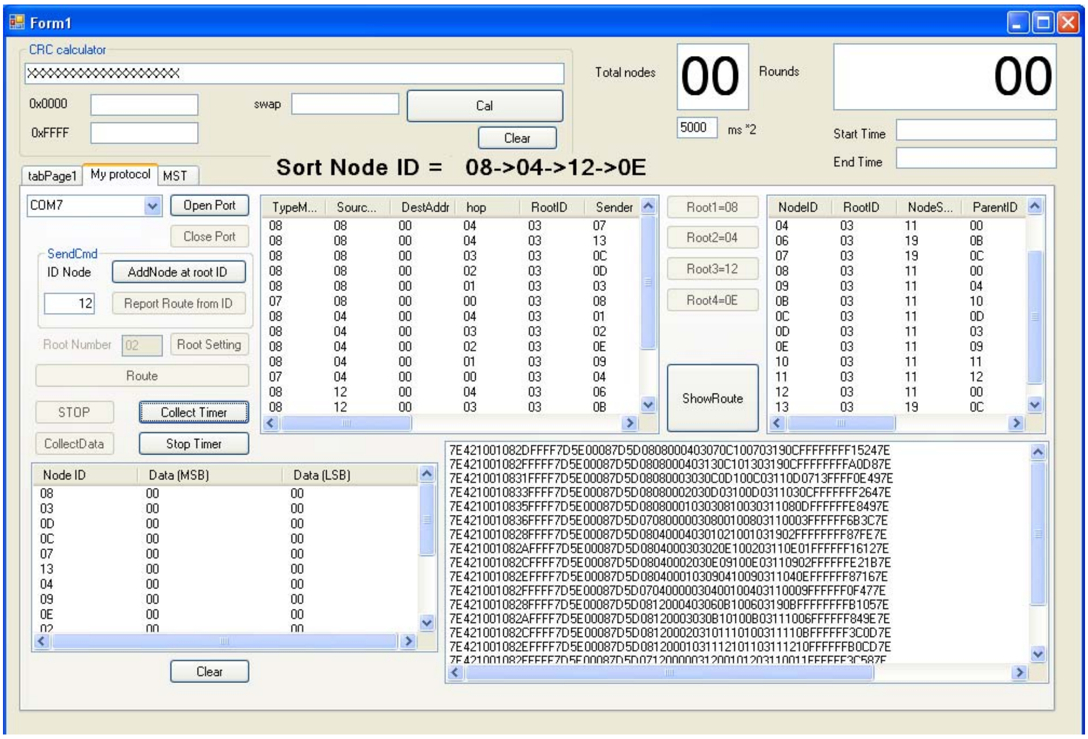
ภาพที่ 28 การวางตำแหน่ง Sensor แบบทแยง

3.2.2 เลือกตำแหน่งการวางบอร์ดมาหนึ่งรูปแบบ (Grid, Random หรือ Diagonal)

3.2.3 Download โปรแกรมที่เขียนลงบนบอร์ด TmosteSky ให้ครบ โดยเลือกว่าจะใช้โปรโตคอล แบบไหนก่อนก็ได้ (PEDAP MST หรือ EFMST)

3.2.4 Download โปรแกรมของ Base station ให้ตรงกับโปรโตคอล ที่ใช้ (PEDAP MST หรือ EFMST) ลงบอร์ดที่วางตำแหน่งดังรูป

3.2.5 เปิดโปรแกรมบนเครื่อง PC ซึ่งผู้วิจัยพัฒนาขึ้นมาเพื่อตรวจสอบการทำงานของระบบ ซึ่งมีลักษณะหน้าจอภาพ ดังนี้

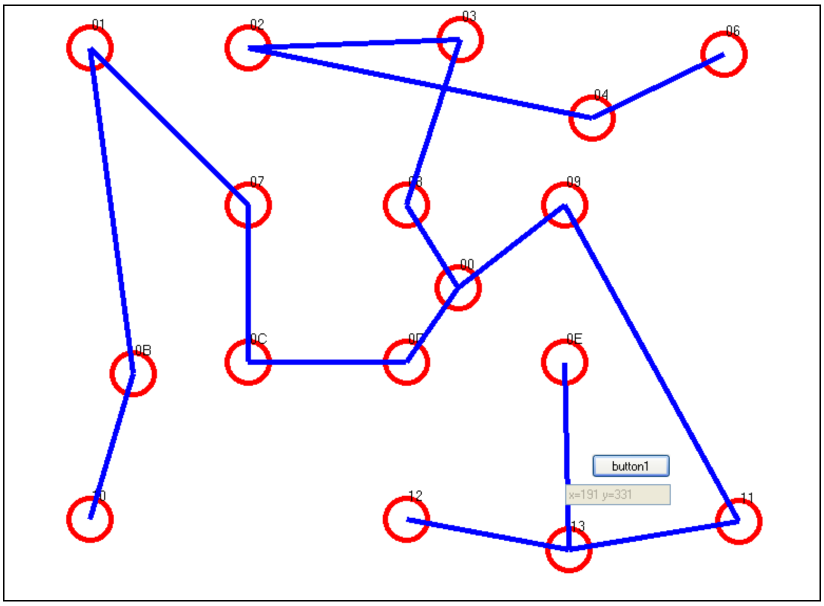


ภาพที่ 29 โปรแกรมที่ใช้ในการเก็บข้อมูลและแสดงเส้นทางที่สร้างได้

3.2.6 ทำการสร้าง network จากโปรแกรมบน PC

3.2.7 เมื่อตัวโปรแกรมทำการสร้าง network เสร็จเรียบร้อยให้ส่งคำสั่ง report route โดยกดปุ่ม report route บนโปรแกรม

3.2.8 เมื่อจำนวนของ node ที่ส่ง report มาครบแล้วให้กด showroute โปรแกรมจะแสดงเส้นทางที่สร้างขึ้นออกมาดังรูปข้างล่าง



ภาพที่ 30 ตัวอย่างเส้นทางที่ได้จากการสร้างเส้นทางจากวิธีการ enforce-multiple sub-tree

3.2.9 เมื่อทราบเส้นทางแล้วนำ Oscilloscope มาวัดกระแสใช้งานของ node ที่เชื่อมต่อกับ base station หรือ first-level child node เพื่อเก็บค่าแรงดันคร่อม Rs เพื่อหาค่ากระแสใช้งานของ node นี้ไปใช้คำนวณหาค่าพลังงานที่ใช้ในการส่ง packet ข้อมูลแต่ละครั้ง โดยใน EFMST จะต้องกำหนดจำนวน first-level child node (root node) ให้กับ algorithm ด้วย ซึ่งในที่นี้จำกำหนดไว้ที่ 1,2 และ 3 nodes วงจรที่ใช้ในการวัดกระแสแสดงอยู่ในภาพผนวกที่ 11

3.2.10 กดที่ปุ่ม gather data ในโปรแกรม

3.2.11 บันทึกค่าสัญญาณกระแสใช้งานขณะส่งข้อมูลจาก oscilloscope โดยเอาค่าที่ใช้เวลาส่งมากที่สุดเป็นเกณฑ์ในการเปรียบเทียบ เพราะจะเป็น node ที่ใช้พลังงานแบตเตอร์รี่มากสุด และอายุใช้งานสั้นที่สุดในระบบ อายุการทำงานของ network จึงขึ้นอยู่กับ root node ตัวนี้

3.2.12 เมื่อเก็บค่าจาก oscilloscope เสร็จแล้ว ทำการ reset network และสร้าง network ใหม่และเก็บค่าจาก oscilloscope ซ้ำอีก โดยในแต่ละ algorithm จะทำการสร้าง network ใหม่ 10 ครั้ง เพื่อนำเอาค่าที่ได้ทั้งหมดมาหาค่าเฉลี่ยเพื่อทำการเปรียบเทียบกันระหว่าง 2 algorithms

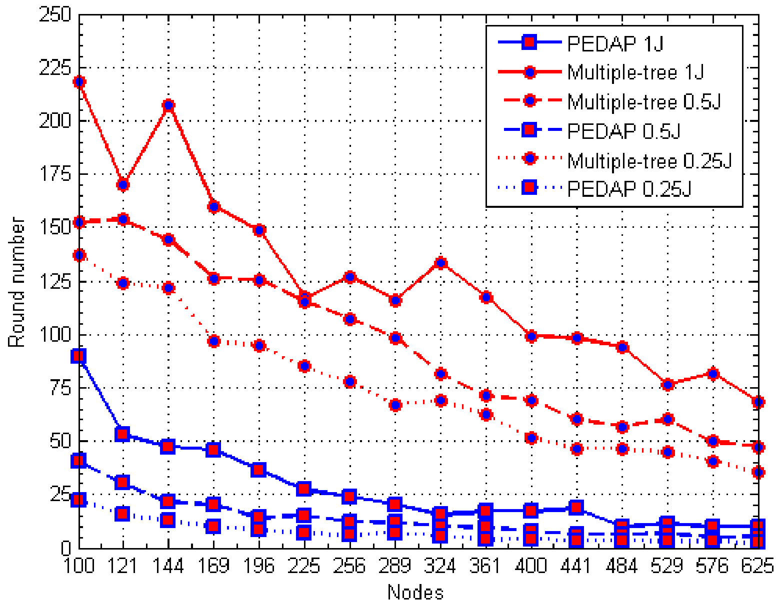
3.2.13 เมื่อเก็บค่าเรียบร้อยแล้ว เปลี่ยน algorithm เป็นอีกแบบหนึ่งแล้วทำการเก็บค่าอีกครั้ง เมื่อเก็บค่าครบทั้ง 2 algorithm แล้ว เปลี่ยนการวางตำแหน่งของ node เป็นแบบอื่น โดยทำให้ครบทั้ง 3 รูปแบบได้แก่ Grid, Random และ Diagonal

ผลการทดลองและการอภิปรายผลที่ได้

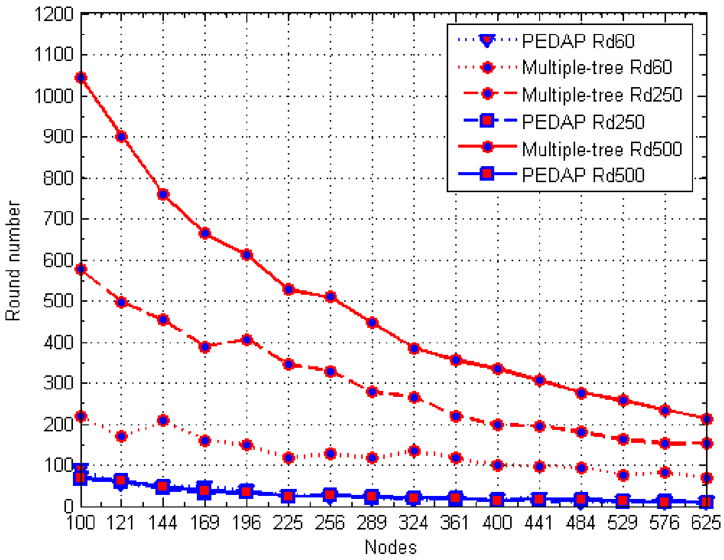
ผลการทดลองจาก Simulation

ภาพที่ 31 แสดงถึงจำนวนรอบที่ได้สูงสุด จากผลของการ simulation ที่แปรผันตามจำนวน node ที่พลังงานเริ่มต้นของแบตเตอรี่แตกต่างกัน โดยกำหนดรัศมีการส่งข้อมูลสูงสุดที่ 200 m, จำนวนรอบที่ต้องการ 60, และพลังงานเริ่มต้นของแต่ละมิเตอร์ 1, 0.5, และ 0.25 *J*

ส่วนในภาพที่ 32 แสดงผลการจำลองจำนวนรอบ ที่ได้สูงสุดที่แปรผันตามจำนวน node และจำนวนรอบที่ต้องการ โดยใช้พลังงานเริ่มต้นของมิเตอร์แต่ละตัวคงที่ที่ 1 Joule และเปลี่ยนแปลงค่าของจำนวนรอบที่ต้องการจาก 60, 250, และ 500 ตามลำดับ จำนวน Round number สูงสุดที่ได้ นับจากครั้งแรกถึง round สุดท้ายก่อนที่ node ตัวใดตัวหนึ่งจะไม่สามารถส่งข้อมูลถึง node ข้างเคียงได้ นั้นคือ network fail ซึ่งค่านี้เป็นตัวบ่งชี้ว่า อายุใช้งานของระบบมีมากน้อยเพียงใด ค่ามากกว่า แสดงว่าใช้งานได้นานกว่า



ภาพที่ 31 ผลการจำลองระบบที่ได้เมื่อระยะทางสูงสุดในการรับส่งข้อมูลกำหนดไว้ที่ 200 m จำนวนรอบที่ต้องการคือ 60 รอบ โดยทำการเปลี่ยนแปลงค่าพลังงานแบตเตอรี่เริ่มต้นที่ 1, 0.5 และ 0.25 *J*



ภาพที่ 32 ผลการจำลองเมื่อระยะทางสูงสุดในการส่งข้อมูลกำหนดไว้ที่ 200 m., พลังงานเริ่มต้นของแต่ละมิเตอร์คงที่ที่ 1 *J* แต่ทำการเปลี่ยนแปลงจำนวนรอบที่ต้องการเท่ากับ 60, 250 และ 500 รอบ

อภิปรายผลการ simulation

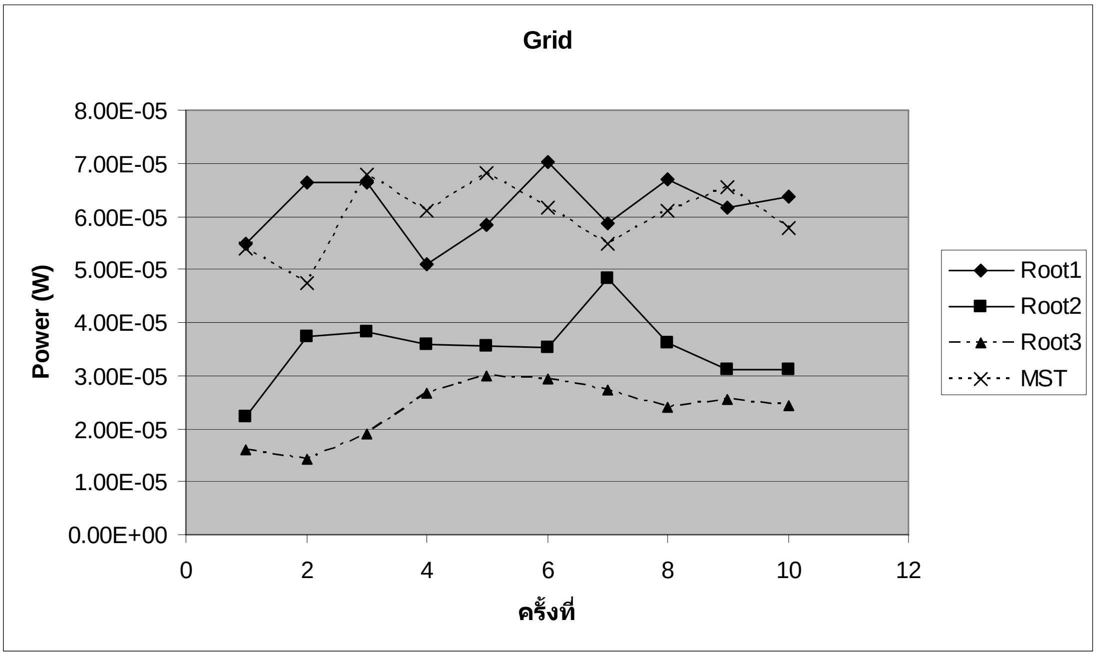
จากผลที่แสดงอยู่ในกราฟจะเห็นว่า algorithm ที่พัฒนาขึ้น สามารถทำให้ network รับส่งข้อมูลได้จำนวน round สูงกว่า PEDAP ในทุกกรณี แสดงว่าจะทำให้อายุการใช้งานของ network ยาวกว่าโดยใช้แบตเตอรี่ที่จ่ายพลังงานได้เท่ากัน

ผลการทดลองจากระบบจริง

ผลการทดลองจากระบบจริง (experiment) ระบบแสดงอยู่ในภาพต่อไปนี้ จากกราฟ Root1, Root2 และ Root3 คือเส้นกราฟพลังงานสูงสุดที่ใช้ส่งข้อมูลให้ base station ของ root node จากการกำหนดจำนวน first-level child node ให้เป็น Root node ใน algorithm แบบ enforced multiple sub-tree ที่จำนวน 1, 2 และ 3 ตัวตามลำดับ ส่วน MST เป็นพลังงานเฉลี่ยที่ได้จากการใช้ Algorithm แบบ PEDAP minimum spanning tree โดยค่าที่คำนวณได้มีหน่วยเป็น Joule ค่านี้ได้จากการทดลองซ้ำกัน 10 ครั้งภายใต้ค่าตัวแปรเดียวกัน

1. ผลการทดลองแบบ Grid

การทดลองแบบ grid ให้ผลดังกราฟเฉลี่ยพลังงานสูงสุดที่ได้ในแต่ละครั้งดังต่อไปนี้



ภาพที่ 33 ผลการทดลองการวางตำแหน่ง Sensor แบบ Grid เส้นกราฟ Root1, Root2 และ Root3 คือพลังงานที่ใช้ของ root node เมื่อเพิ่มจำนวนจาก 1 เป็น 2 และ 3 ตามลำดับ

จากภาพที่ 33 แสดงให้เห็นว่า เมื่อจำนวน root node เพิ่มขึ้น พลังงานที่ใช้ในการส่งข้อมูลที่รวบรวมจาก node ทุกตัวให้แก่ base station ในระบบของ root node ลดลงจริง เป็นไปตามที่คาดไว้จากผลของการ simulation

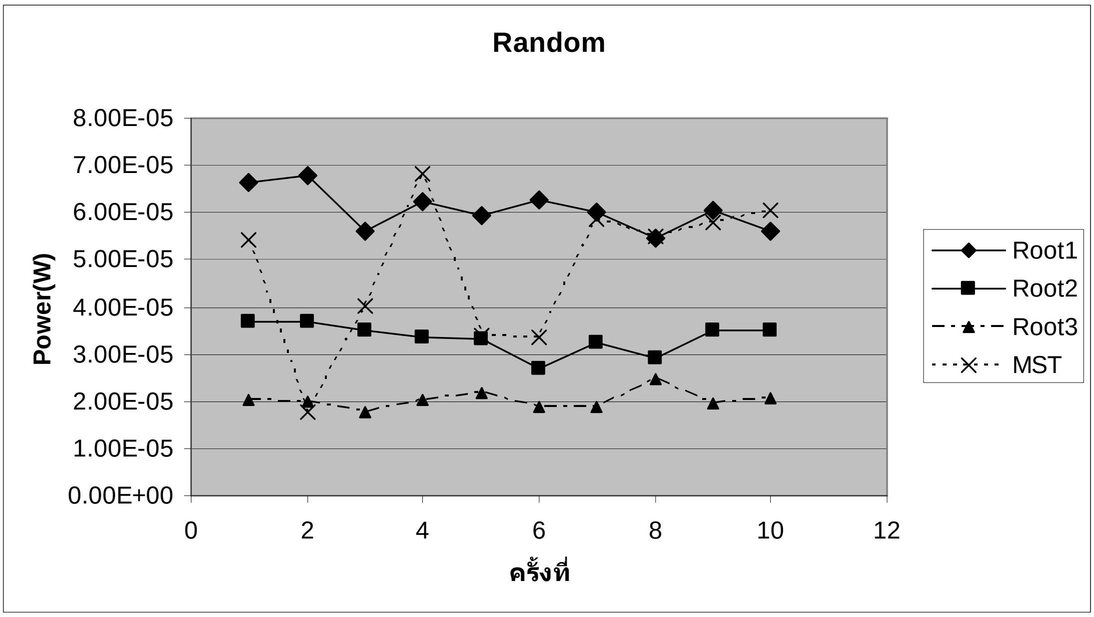
ตารางที่ 1 ค่าพลังงานที่วัดได้จากการ experiment ที่วางตำแหน่งของ Sensor แบบ Grid

Grid (16 nodes) (Joules)				
Number	EFMST			MST
	First 1	First 2	First 3	
1	5.49E-05	2.24E-05	1.60E-05	5.39E-05
2	6.65E-05	3.72E-05	1.43E-05	4.74E-05
3	6.65E-05	3.83E-05	1.91E-05	6.79E-05
4	5.09E-05	3.60E-05	2.67E-05	6.09E-05
5	5.84E-05	3.57E-05	2.99E-05	6.83E-05
6	7.02E-05	3.53E-05	2.92E-05	6.15E-05
7	5.88E-05	4.83E-05	2.73E-05	5.48E-05
8	6.69E-05	3.62E-05	2.41E-05	6.09E-05
9	6.16E-05	3.11E-05	2.54E-05	6.54E-05
10	6.36E-05	3.10E-05	2.43E-05	5.78E-05
AV	6.183E-05	3.51E-05	2.36E-05	5.99E-05

จากตารางที่ 1 First1, First2 และ First3 คือค่าพลังงานสูงสุดที่ใช้ในการส่งข้อมูลใน 1 รอบของ First-level child node เมื่อทดลองที่จำนวน First-level child node เท่ากับ 1,2 และ 3 ตัว ตามลำดับ โดยทำการทดลอง 10 ครั้ง นำไปสร้างกราฟและนำมาหาค่าเฉลี่ย AV จะพบว่าเมื่อมีการเพิ่มจำนวน first-level child node มากขึ้นพลังงานจะลดลง

2. ผลการทดลองแบบ Random

การทดลองวางตำแหน่ง node แบบ random ให้ผลการวัดพลังงานดังกราฟในภาพที่ 34



ภาพที่ 34 ผลการทดลองหาพลังงานของ root node ที่ใช้ส่งข้อมูลให้ base station จากการวางตำแหน่ง Sensor แบบ Random

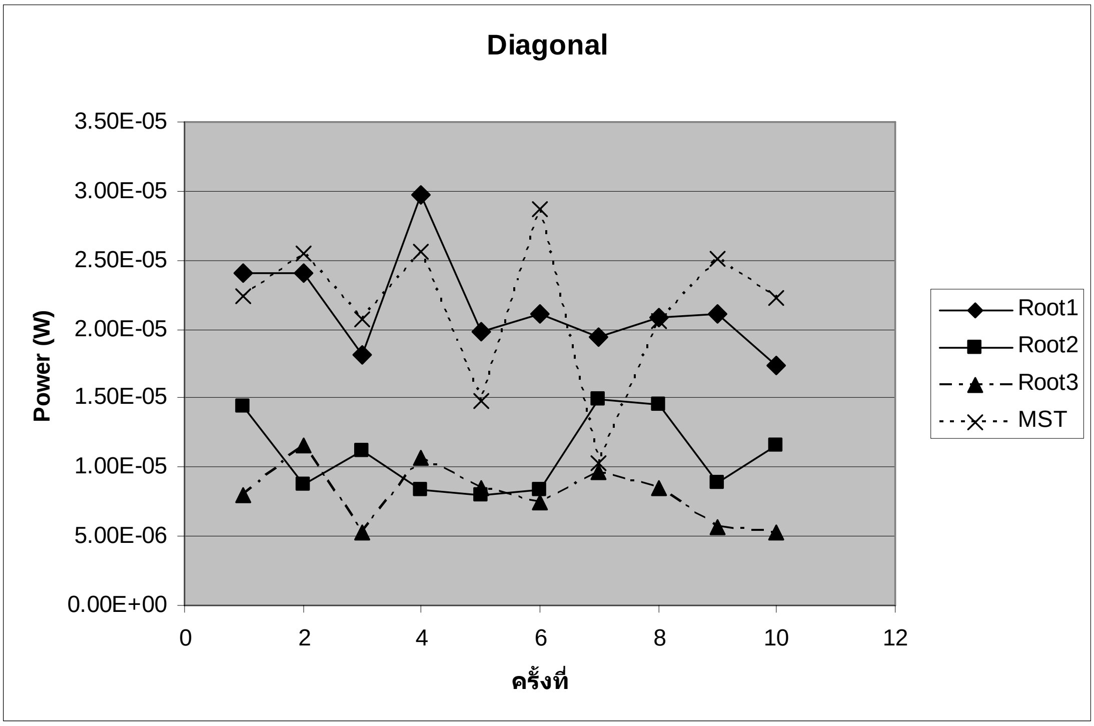
จากภาพที่ 34 ซึ่งแสดงให้เห็นว่า เมื่อจำนวน root node เพิ่มขึ้น พลังงานที่ใช้ในการส่งข้อมูลจาก node ทุกตัวในระบบของ root node ให้แก่ base station ลดลงจริงตามที่คาดไว้จากผลของการ simulation

ตารางที่ 2 ค่าพลังงานที่วัดได้จากการ experiment ที่วางตำแหน่งของ Sensor แบบ Random

Random (16 nodes) (Joules)				
Number	EFMST			MST
	First 1	First 2	First 3	
1	6.65E-05	3.70E-05	2.02E-05	5.41E-05
2	6.78E-05	3.69E-05	2.00E-05	1.78E-05
3	5.60E-05	3.51E-05	1.77E-05	4.02E-05
4	6.22E-05	3.37E-05	2.02E-05	6.82E-05
5	5.92E-05	3.32E-05	2.18E-05	3.39E-05
6	6.28E-05	2.69E-05	1.86E-05	3.36E-05
7	6.00E-05	3.24E-05	1.87E-05	5.87E-05
8	5.46E-05	2.93E-05	2.45E-05	5.51E-05
9	6.05E-05	3.50E-05	1.96E-05	5.79E-05
10	5.62E-05	3.52E-05	2.06E-05	6.05E-05
AV	6.06E-05	3.35E-05	2.02E-05	4.8E-05

3. ผลการทดลองแบบ Diagonal

การทดลองวางตำแหน่ง sensor แบบ Diagonal ให้ผลดังแสดงอยู่ในกราฟต่อไปนี้



ภาพที่ 35 ผลการทดลองการวางตำแหน่ง Sensor แบบ Diagonal เส้นกราฟ Root1, Root2 และ Root3

จากภาพที่ 35 พลังงานที่ใช้ของ root node เมื่อเพิ่มจำนวนจาก 1 เป็น 2 และ 3 ตามลำดับ ซึ่งแสดงให้เห็นว่า เมื่อจำนวน root node เพิ่มขึ้น พลังงานที่ใช้ในการส่งข้อมูลที่รวบรวมจาก node ทุกตัวให้แก่ base station ในระบบของ root node ลดลงจริง แต่ระยะทางการส่งข้อมูลอาจมีผลต่อพลังงานได้ ดังเส้นกราฟ Root3

ตารางที่ 3 ค่าพลังงานที่วัดได้จากการ experiment ที่วางตำแหน่งของ Sensor แบบ Diagonal

Pluz (8 nodes) (Joules)				
Number	EFMST			MST
	First 1	First 2	First 3	
1	2.40E-05	1.44E-05	7.98E-06	2.24E-05
2	2.40E-05	8.75E-06	1.16E-05	2.55E-05
3	1.81E-05	1.11E-05	5.25E-06	2.08E-05
4	2.97E-05	8.39E-06	1.07E-05	2.55E-05
5	1.98E-05	8.03E-06	8.46E-06	1.48E-05
6	2.11E-05	8.31E-06	7.44E-06	2.87E-05
7	1.94E-05	1.49E-05	9.70E-06	1.03E-05
8	2.08E-05	1.45E-05	8.44E-06	2.05E-05
9	2.12E-05	8.89E-06	5.65E-06	2.51E-05
10	1.73E-05	1.15E-05	5.25E-06	2.23E-05
AV	2.156E-05	1.09E-05	8.05E-06	2.16E-05

อภิปรายผลการ Experiment ที่ตำแหน่งของ Sensor node แตกต่างกัน

1. อภิปรายผลการทดลองวางตำแหน่งแบบ grid

จากผลการ experiment แบบ grid จะพบว่าเมื่อเพิ่มจำนวนของ first-level child node พลังงานที่ใช้จะลดลงเนื่องจากมีการแบ่ง load ออกเป็น 3 ส่วน ทำให้พลังงานของ first-level child node แต่ละตัวที่ใช้ลดลงที่เห็นได้ชัดเจนจากกราฟ แต่ในทางกลับกันใน algorithm แบบ MST จะพบว่าผลมีการกระโดดขึ้นๆ ลงๆ เนื่องจากการเลือกเส้นทางจะเกิดจากการเลือก weight ที่น้อยที่สุด ที่แปรผันไปตามภาวะแวดล้อม โดยในที่นี้จะวัดจาก link quality ทำให้จำนวนของ root node ไม่คงที่ขึ้นอยู่กับ weight ขณะนั้น

2. อภิปรายผลการทดลองวางตำแหน่งแบบ Random

จากผลการ experiment แบบ random จะพบว่ามีผลในลักษณะเดียวกันกับแบบ กริดและพลังงานที่ใช้ก็จะใกล้เคียงกันด้วย เพราะจากวิธีการของ EFMST แต่ละ Root node จะสลับกันหา node มาเพิ่มเข้าไปใน sub-tree ทำให้จำนวน nodes ของแต่ละ sub-tree ที่จำนวน node เท่ากันนั้นจะมีค่าใกล้เคียงกัน ทำให้พลังงานที่ใช้ไปในการส่งข้อมูลไปยัง base station ของแต่ละ root node มีค่าใกล้เคียงกันด้วย

3. อภิปรายผลการทดลองวางตำแหน่งแบบ Diagonal

จากผลการ experiment แบบ diagonal จะพบว่า ผลที่ได้ของการเพิ่มจำนวนของ root nodes และแบบ MST มีค่าไม่แตกต่างกันมากนัก เพราะว่าจำนวนของ nodes รวมทั้งหมดที่ใช้ในการทดลองมีจำนวนที่น้อยเกินไป ข้อมูลที่ส่งในไปให้กับ Base station จึงมีจำนวนบิตใกล้เคียงกัน ระยะทางการส่งที่ไกลกว่าการทดลอง 2 แบบแรกมากจึงมีผลต่อพลังงานที่ใช้ด้วย เนื่องจากการสร้าง tree นั้นไม่อาจคาดเดาและเจาะจงตัว node ได้จากระยะห่างของ sensor ในแนวราบ ผลการทดลองนี้จึงไม่สามารถบอกความแตกต่างได้ชัดเจนเท่าแบบ random และแบบ grid

วิจารณ์ผลการวิจัย

จากผลการจำลองระบบที่ได้ในภาพที่ 33, 34 และ 35 และยืนยันด้วยผลจากการทดลอง experiment แบบวางตำแหน่ง sensor แบบ grid และ random แสดงให้เห็นว่าในการใช้วิธีตามงานวิจัยฉบับนี้ ให้จำนวนรอบของการเก็บข้อมูลที่มากกว่าวิธีของ PEDAP หมายความว่าอายุใช้งานของเครือข่ายมาตรวัดน้ำที่ใช้วิธีนี้จะยาวนานกว่า ประหยัดพลังงานของ root node ได้มากกว่า

แต่อย่างไรก็ตาม เป็นที่คาดหมายได้ว่า จำนวนรอบ (round) ของการรวบรวมข้อมูลจาก node ทุกตัวในเครือข่ายที่ทำงานได้จริงในทางปฏิบัติ จะแตกต่างจากจำนวนรอบที่ต้องการที่กำหนดขึ้นในตอนคำนวณเล็กน้อย เพราะจำนวนรอบที่ต้องการนั้น ได้มาโดยการคำนวณจากกรณีที่แย่ที่สุด ซึ่งในกรณีที่จำนวนของมิเตอร์มีน้อย จำนวนรอบที่ใช้ได้จริงน่าจะมีค่าสูงกว่าจำนวนรอบที่ต้องการเสมอ เนื่องจากระยะทางการส่งข้อมูลระหว่างมิเตอร์ ไม่ใช่ค่าสูงสุดตามที่กำหนดไว้ในเงื่อนไขการ simulation

ในกรณีที่เมื่อจำนวนของมิเตอร์มีมากขึ้น จำนวนรอบที่ทำงานได้จริง น่าจะมีค่าที่น้อยกว่าที่ได้คำนวณไว้ เพราะว่าระยะทางสูงสุดที่มิเตอร์สามารถรับส่งข้อมูลให้กับ base station ได้โดยตรงจะมีขนาดจำกัดอยู่ค่าหนึ่ง ซึ่งเป็นผลให้จำนวนของ root nodes ในระบบที่ algorithm หาได้จริงจะได้น้อยกว่าที่คำนวณไว้ เพราะถูกจำกัดด้วยระยะทางสื่อสารสูงสุดที่ sensor node จะสามารถตอบรับสัญญาณกลับมาถึง base station ได้โดยตรง

แม้ว่า N_R จะไม่ใช้จำนวนของ Root node ที่ได้ในทางปฏิบัติจริง ๆ และค่าของจำนวนรอบที่ได้จากการทดลองจะไม่เป็นไปตามที่ได้จากการคำนวณ แต่ค่าตัวแปรเหล่านี้มีเป็นประโยชน์ในการออกแบบและวางตำแหน่งของมิเตอร์ ผลการ simulation และ experiment ยืนยันว่า ถ้าจำนวนของ root node มีค่าใกล้กับค่า N_R ระยะเวลาในการทำงานของระบบหรือจำนวนรอบที่ได้มีค่าใกล้เคียงกับค่าที่ได้คำนวณไว้

สิ่งหนึ่งที่ algorithm ที่คิดขึ้นทำได้ง่าย ก็คือการเพิ่มจำนวนของ Root node เข้าไปในระบบเมื่อมีการติดตั้งมิเตอร์เพิ่มขึ้น โดยการ restart สร้าง tree ใหม่

ในด้านของการทดลองจริง พบว่าจำนวนของ nodes ที่ใช้ในการทดลองยังมีจำนวนไม่มากพอที่จะเห็นผลอายุการใช้งานจริง คือทดลองต่อเนื่องจนถึงเวลาที่ network ล่มเพราะหมดพลังงาน battery ใน root node ตัวใดตัวหนึ่ง ซึ่งอาจจะทำให้ผลการทดลองที่ได้ยังไม่สามารถบอกผลข้างเคียงอื่นๆ ที่ไม่อาจคาดเดาได้ รวมทั้งระบบนี้ยังไม่ได้นำไปติดตั้งในสถานที่จริง ซึ่งอาจมีสิ่งแวดล้อมอื่นมากระทบต่อการสร้าง tree และหา root node ที่เหมาะสม เช่น สัญญาณรบกวนคลื่นวิทยุที่เกิดขึ้นอย่างไม่ต่อเนื่อง ที่ผู้วิจัยยังไม่สามารถคาดเดาได้

อนึ่งข้อมูลที่เก็บได้ก็ยังไม่ใช่ข้อมูลจริงในการเก็บค่าน้ำประปาแต่ค่าที่ได้จากการ experiment นั้นพอจะยืนยันได้ว่า วิธีการที่ได้นำเสนอมานี้มีผลลัพธ์ที่ถูกต้องตามการ simulation และน่าจะสามารถนำมาใช้ลดพลังงานที่ใช้ในการส่งข้อมูลของปริมาณน้ำจากมาตรวัดน้ำในบริเวณที่อยู่อาศัยที่บ้านเรือนหนาแน่นพอ คือระยะห่างระหว่างบ้านอยู่ในรัศมีการส่งข้อมูลระหว่าง sensor board ที่สร้างขึ้นสำหรับ digital electronic water meter เพื่อการเก็บค่าน้ำประปาอัตโนมัติ โดยใช้ระบบ wireless sensor network ด้วย algorithm ที่พัฒนาขึ้นมานี้ได้

สรุปและข้อเสนอแนะ

สรุป

งานวิจัยฉบับนี้ได้ทำขึ้น เพื่อพัฒนาอัลกอริทึมเพื่อใช้ในการเก็บข้อมูลจาก sensor board ที่ออกแบบพิเศษสำหรับมาตรวัดน้ำประปา โดยรวบรวมข้อมูลปริมาตรน้ำจากมาตรวัดทุกตัวที่อยู่ในรัศมีการสร้างเครือข่ายหรือ tree ส่งมายังศูนย์กลาง Base station เพื่อนำมาปรับปรุงใช้กับการเก็บค่าจากมิเตอร์น้ำประปาอัตโนมัติในอนาคต ซึ่งได้อัลกอริทึมที่ตั้งชื่อว่า enforced multiple sub-tree

เงื่อนไขของการพัฒนาอัลกอริทึมนี้ได้แก่ ข้อมูลจากมิเตอร์น้ำไม่สามารถรวมกันเป็นก้อนเดียวที่มีความยาวคงที่(fused)ได้ ซึ่งประสิทธิภาพที่ได้นั้นได้นำเสนอในรูปแบบของการ simulation โดยยึดลักษณะการวางตำแหน่งของตัวมิเตอร์ตามลักษณะของบ้านเดี่ยวในหมู่บ้านจัดสรรทั่วไปในกรุงเทพมหานคร ซึ่งผลที่ได้จากการใช้วิธีการ enforced multiple sub-tree ที่พัฒนามาใช้ในการสร้างเส้นทางรับส่งข้อมูลนี้ สามารถยืดอายุการใช้งานของระบบที่ sensor node ใช้พลังงานจาก battery ได้

จากผลการ simulation วิธีการที่คิดขึ้นนี้ แสดงว่าสามารถช่วยยืดอายุการทำงานของระบบเครือข่ายวิทยุของมิเตอร์น้ำ เพื่อการเก็บค่าน้ำประปาได้ และดีกว่าใช้ algorithm ที่มีอยู่แล้วบางตัวภายใต้ลักษณะสิ่งแวดล้อมเดียวกัน

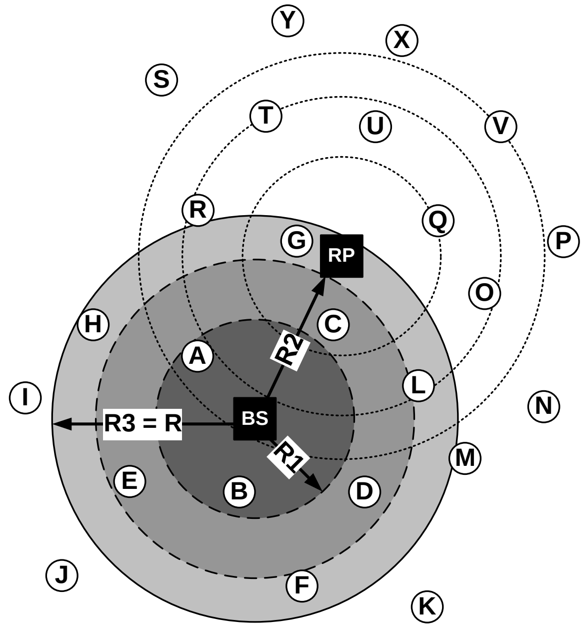
ในการ experiment จะพบว่า ผลที่ได้สามารถยืนยันได้ว่าในการเพิ่มจำนวนของ first-level child nodes เข้าไปในระบบ สามารถลดพลังงานที่ใช้ใน first-level child nodes ได้ พร้อมทั้งสามารถบอกได้ว่าวิธีการที่เสนอนี้สามารลดพลังงานที่ใช้ในการเก็บข้อมูลของ first-level child nodes ได้มากกว่าวิธีการที่ใช้การสร้างเส้นทางแบบ PEDAP MST

ข้อเสนอแนะ

เนื่องจากการทดลองของงานวิจัยฉบับนี้ ไม่ได้พยายามตอบปัญหาการหาจำนวนของ First-level child nodes ที่ดีที่สุด ว่าควรจะมีเท่าไรจึงจะทำให้ระยะเวลาการใช้งานของระบบยาวนานที่สุด

ซึ่งถ้าจำนวนของ first-level child nodes มีมากเกินไปจะทำให้ระยะทางระหว่าง first-level child nodes บางตัว กับ base station มีระยะทางที่ไกลมากขึ้น ทำให้พลังงานที่ใช้ส่งคลื่นวิทยุต้องมากขึ้นด้วย และอาจมีผลให้อายุการใช้งานของ battery ลดลง แต่ถ้ามีจำนวนน้อยเกินไป first-level child nodes ก็จะต้องทำการส่งข้อมูลที่มีจำนวนข้อมูลจาก node อื่นๆ มารวมกันมากขึ้น ทำให้สิ้นเปลืองพลังงานเพิ่มขึ้นอีก สิ่งที่ควรทำคือ หาจุดต่ำสุดของกราฟพลังงานสูญเสียบน first-level childnode ที่ลดลงเมื่อจำนวน first-level childnode เพิ่มขึ้น ก่อนที่กราฟจะตีกลับเป็นเพิ่มขึ้นเนื่องจากระยะทางมีผลต่อการใช้พลังงานมากขึ้น จะทำให้รู้จุดจำกัดของการเพิ่มจำนวน first-level childnode ที่ไม่ควรเพิ่มมากกว่านั้นอีกแล้ว ทั้งนี้ยังมีตัวแปรอื่นที่มีผลด้วยคือจำนวน sensor node ทั้งหมดในระบบที่มีผลในการเพิ่มจำนวน first-level childnode และพลังงานสูญเสียเนื่องจากความยาวของ packet ข้อมูลที่ผ่าน first-level childnode ซึ่งจะต้องหาวิธีวิจัยหาคำตอบที่เหมาะสมต่อไป

สำหรับกรณีที่ไม่สามารถติดตั้ง base station กลางกลุ่ม sensor node ได้ซึ่งอาจทำให้ไม่สามารถหา first-level child nodes ได้ตามจำนวนที่ต้องการ อาจจะแก้ไขได้ด้วยการใช้ตัว repeater ที่ทำงานด้วย algorithm ที่พัฒนาให้เหมาะสม ติดตั้งเพิ่ม เพื่อเพิ่มระยะการส่งสัญญาณของ base station ให้สามารถติดต่อกับ node ได้มากขึ้นดังภาพที่ 36



ภาพที่ 36 การเพิ่มตัว repeater เข้าในระบบ

จากภาพที่ 36 ถ้าพิจารณาตามหลักการเบื้องต้นของการสื่อสารด้วยคลื่นวิทยุ เมื่อมีการติดตั้งตัว repeater (RP) เพิ่มเข้าไปในระบบดังรูป จำนวนของ first-level child node ที่หาได้ของระบบจะเพิ่มขึ้นจากเดิมที่มี 8 ตัวเพิ่มขึ้นเป็น 13 ตัว นั้นคือ node O, Q, R, T และ U การเพิ่ม repeater นี้ อาจเพิ่มขึ้นหลาย ๆ ตัว เพื่อให้ระบบหา first-level child node ได้ตามจำนวนที่ต้องการ

ทางด้านของการ experiment จำนวนโหนดที่ใช้ยังมีจำนวนที่น้อยเกินไป เนื่องจากขาดอุปกรณ์และงบประมาณในการทดลอง รวมทั้งยังไม่มีการทดลองส่งข้อมูลที่อ่านได้จากตัวมิเตอร์น้ำจริงๆ และไม่ได้ออกแบบระบบ security ในรักษาข้อมูลที่ส่งออกไปยัง base station ซึ่งในงานวิจัยในครั้งต่อไปควรจะมีการเชื่อมต่อ sensor network เข้ากับมิเตอร์ที่ใช้ในการอ่านค่าน้ำประปาจริง และนำไปติดตั้งทดลองในสถานที่ใช้จริง หรืออาจจะวิธีเร่งขบวนการคือส่งข้อมูลให้ถี่มากขึ้นและข้อมูลอาจจะได้มาจาก sensor อื่น ๆ แทนเช่น อุณหภูมิ, ความชื่น หรือ แสง แทนข้อมูลที่ได้จากมิเตอร์น้ำจริง ๆ เพื่อให้ผลการทดลองที่มีค่าที่ถูกต้องที่สุด

แนวทางการพัฒนาต่อ

ในวิทยานิพนธ์ฉบับนี้ไม่ได้ครอบคลุมในด้านของระบบ security ที่จะนำมาใช้ป้องกันระบบ ซึ่งในการนำไปใช้จริง จำเป็นที่จะต้องมีวิธีการเพื่อป้องกันการโกงปริมาตรน้ำด้วยการแก้ไขข้อมูลที่ส่งผ่าน packet สื่อสาร ซึ่งจากข้อจำกัดทาง physical ของตัวมิเตอร์เอง ทำให้ระบบรักษาความปลอดภัยของข้อมูล ที่สามารถนำมาใช้ในระบบได้ ต้องมีการคำนวนที่ไม่ยากเกินไป เพราะจะทำให้สินเปลืองพลังงานมากขึ้นไปอีก

อีกประการหนึ่ง จำนวนของกุญแจ (key) ที่ใช้ในการเข้ารหัสข้อมูล ต้องสามารถเก็บอยู่ภายใน memory ของตัวมิเตอร์ได้ โดยอาจจะนำโปรโตคอลบางตัวที่มีอยู่ในปัจจุบันมาใช้ป้องกันข้อมูลในการส่งข้อมูลของของตัวมิเตอร์ได้เช่น IEEE 802.15.4 ที่ใช้อยู่ใน zigbee (Zigbee, 2006) หรืออาจจะใช้อีกวิธีหนึ่งคือ security protocol for senser network (SPINS) (Perrig et al., 2001) ที่ใช้อยู่ใน TinyOS

อัลกอริทึมที่ใช้ในวิทยานิพนธ์ฉบับนี้ มิได้ครอบคลุมวิธีการแก้ปัญหาข้อมูลผิดพลาดขณะส่ง ซึ่งหากขณะส่งข้อมูลมีข้อมูลผิดผลาดขึ้นด้วยประการใดๆ ก็ตาม เช่น อาจเกิดจากสัญญาณรบกวนในขณะมีฝนตกฟ้าผ่า ระบบที่พัฒนาขึ้นจะแก้ปัญหาด้วยการส่ง packet ซ้ำ ซึ่งทำให้สูญเสียพลังงานในตัวส่งมากขึ้น จนอาจทำให้ระบบไม่สามารถทำงานได้ตามอายุใช้งานที่ต้องการ

ดังนั้น ในงานวิจัยที่ต่อเนื่องจากงานนี้ ควรที่จะครอบคลุมถึงการวิเคราะห์ error rate ในการส่งข้อมูลด้วย โดยอาจเพิ่มตัวแปรบางตัวที่ใช้แทน error rate ของการส่งข้อมูลเข้าไปในสมการพลังงานที่ใช้ในการคำนวณหาจำนวนของ first-level child node เพื่อให้ได้จำนวนที่สามารถชดเชยกับพลังงานที่อาจสูญเสียไปเนื่องจากการส่งซ้ำที่เกิดจาก error rate ที่ประมาณการไว้ในสภาพแวดล้อมแต่ละแห่ง

เอกสารและสิ่งอ้างอิง

AMCO Water Metering Systems Inc., 2003. **Electronic Water Meter AquaMaster,** Available Source: www.amcometer.com/products/files/AMTS_ 1.pdf.

Ando N., M. Takashima, Y. Doi, Y. Sugitate and N. Menda, 2002. Automatic meter reading System adopting automatic routing technology. , vol. 3, pp. 2305-2309, **IEEE/PES Transmission and Distribution Conference and Exhibition : Asia Pacific**.

Chipcon AS SmartRF, 2004. **CC2420 Preliminary Datasheet (rev 1.2),** Available Source: http://www-inst.eecs.berkeley.edu/~cs150/Documents/CC2420.pdf.

Dugas C., 2005. Configuring and managing a large-scale monitoring network: solving real world challenges for ultra-low-powered and long-range wireless mesh networks. vol. 15, no. 4, pp. 269-282, **International journal of network management.**

_______, 2003 "WAVENIS® : Ultra Low Power & Long Range Wireless Technology Extending Bluetooth to Ultra Low Powered equipments. **Smart Objects Conference (SOC'03).**

Heinzelman W. R., A. Chandrakasan and H. Balakrishnan. 2000. Energyefficient communication protocol for wireless microsensor networks. vol. 2, pp. 3005-3014. **Proceedings of the 33rd Annual Hawaii International Conference on System Sciences**.

_______, J. Kulik, and H. Balakrishnan, 1999. Adaptive protocols for information dissemination in wireless sensor networks. , pp. 174-185, **ACM/IEEE International Conference on Mobile Computing and Networking.**

Hussain S., A. W. Matin, 2005. Energy Efficient Hierachical Cluster-Based Routing for Wireless Sensor Networks, **Technical Report – TR-2005-011, Jodrey School of Computer Science Acadia University.**

Hydrometer, 2007. **Data Sheet: Electronic Water Meter Models 742/743: Hydrometer**, Available Source: http://www.hydrometer.de., 7 Oct 2007

Koay B. S., S. S. Cheah, Y. H. Sng, P. H. J. Chong, P. Shum, Y. C. Tong, X. Y. Wang, Y. X. Zuo, and H. W. Kuek, 2003. Design and implementation of Bluetooth energy meter, , vol. 3, pp. 1474-1477, **Proceedings of the Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, and the Fourth Pacific Rim Conference on Multimedia**.

Lindsey S. and C. S. Raghavendra, 2002., PEGASIS: power-efficient gathering in sensor information systems, vol. 3, pp. 1125-1130, **IEEE Aerospace Conference Proceedings.**

Manjeshwar A, and D. P. Agrawal, 2001., TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, pp. 2009-2015, **International Parallel and Distributed Processing Symposium**.

_______, and D. P. Agrawal, 2002. APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, pp. 195-202, **International Parallel and Distributed Processing Symposium**.

Moore N., A. Sekercioglu, G. K. Egan. 2005. Virtual Localization for Mesh Network Routing, **Networks and Communication System**.

Moteiv Corporation, **Tmote Sky**, Available Source: *http://www.moteiv.com*

Moteiv Corporation, June 2006. **Tmote Sky Low Power Wireless Sensor ModuleDatasheet**, Available Source: http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf, 7 Oct 2007

Neptune Technology Group Inc, 2006 **A Hybrid Mobile and Fixed Network RF AMR System Approach Auburn,Alabama.,** Available Source: http://www.neptunetg.com/uploadedFiles/CS_20AUBURN20080420(print)secure.pdf., 10 Dec 2006

_______, 2007. **Utility cuts meter reading labor costs by more than 80% and improves efficiency by over 85% using a Neptune automated meter reading solution. Water accountability improves with a 60% reduction in lost water.** Available Source: http://www.neptunetg.com/uploadedFiles/NTG289%20CS%20PASADENA%2004.06.pdf , 7 Oct 2007

Park B. S., D. H. Hyun, and S. K. Cho, 2002. Implementation of AMR system using power line communication,vol. 1, pp. 18-21, **IEEE/PES Transmission and Distribution Conference and Exhibition: Asia Pacific**.

Perkins C. E., E. M. Royer. 1999. Ad-hoc On-Demand Distance Vector Routing, pp. 90-100, **IEEE Workshop on Mobile Computing Systems and Applications.**

Perrig, A. et al., 2006. SPINS: security protocols for network sensors, pp. 189, **Proc. 7th Int. Conf. Mobile Computing Networking (MOBICOM), ACM Press**

Rabaey J.M., M.J. Ammer, J.L. da Silva Jr., D. Patel, S. Roundy, 2000. PicoRadio Supports Ad Hoc Ultra-Low Power WirelessNetworking, pp.42-48, **IEEE Computer Magazine.**

Shiwei L., W. Chenshong, C. Mengshi, and W. Koutan, 1996. Design of an automatic meter reading system, , pp. 631-636**, Proceedings of the 1996 IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation.**

Tan S. Y. and M. Moghavvemi, July 2002. PIC-based automatic meter reading and control over the low voltage distribution network, , pp. 517-520, **SCOReD Student Conference on Research and Development Proceedings, Shah Alam, Malaysia.**

Tan H. and I. Korpeoglu, Dec 2003. Power efficient data gathering and aggregation in wireless sensor networks, vol. 32 No.4, **ACM SIGMOD Record**

Tanenbaum A. S., 1996. **Computer Networks**, 3ed: Prentice-Hall PTR.

Wang Y., T. Arslan, May 2003. An Adaptive Routing Algorithm for Longlifetime Wireless Picoradio Networks, **Smart Objects Conference (SOC'03).**

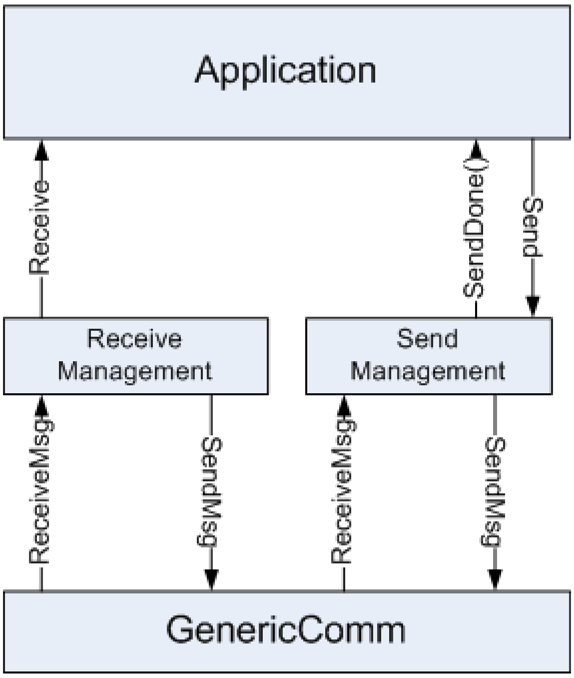
Wong S. K. and M. Moghavvemi, July 2002. A robust system for data transmission over the row voltage distribution network, pp. 399-402, **SCOReD Student Conference on Research and Development.**

Ziegler H. and F. Physik, 2007. **The M-Bus: A Documentation Rev. 4.8.** Available Source: http://www.m-bus.com., Oct 7, 2007

Zigbee Alliance, 2006. **Zigbee specification**. Available Source: http://www.zigbee.org, Sep 29,2007

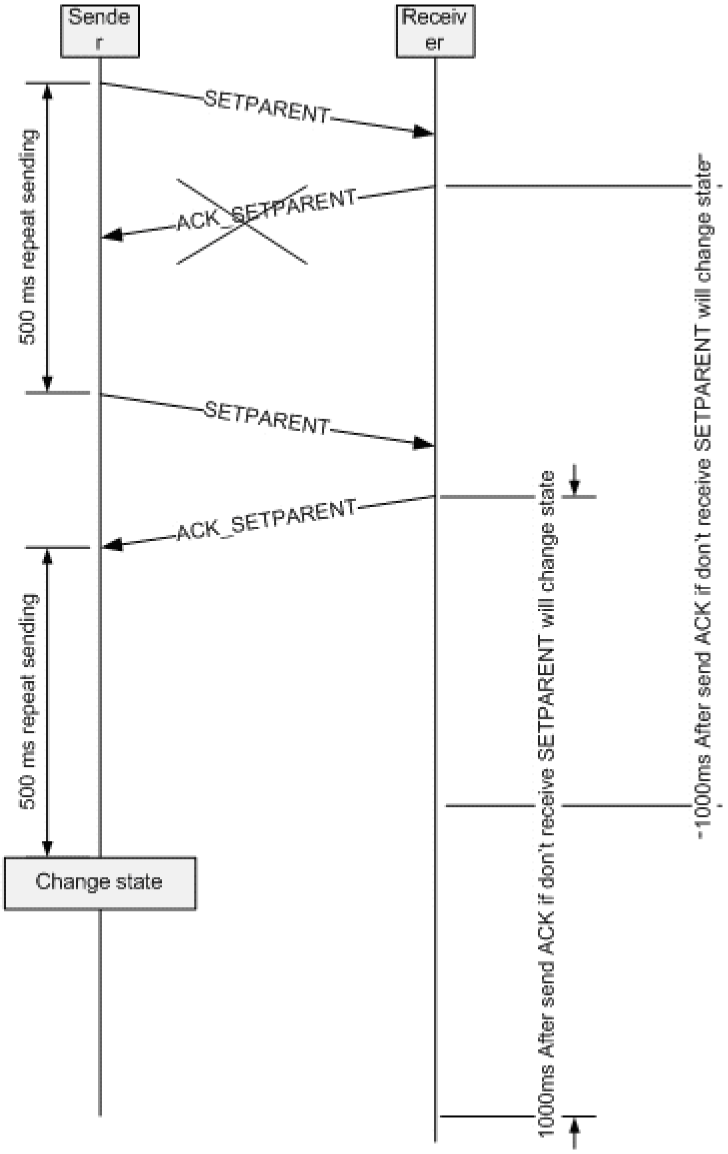
ภาคผนวก

State การสร้างเส้นทางและเทคนิคที่ใช้

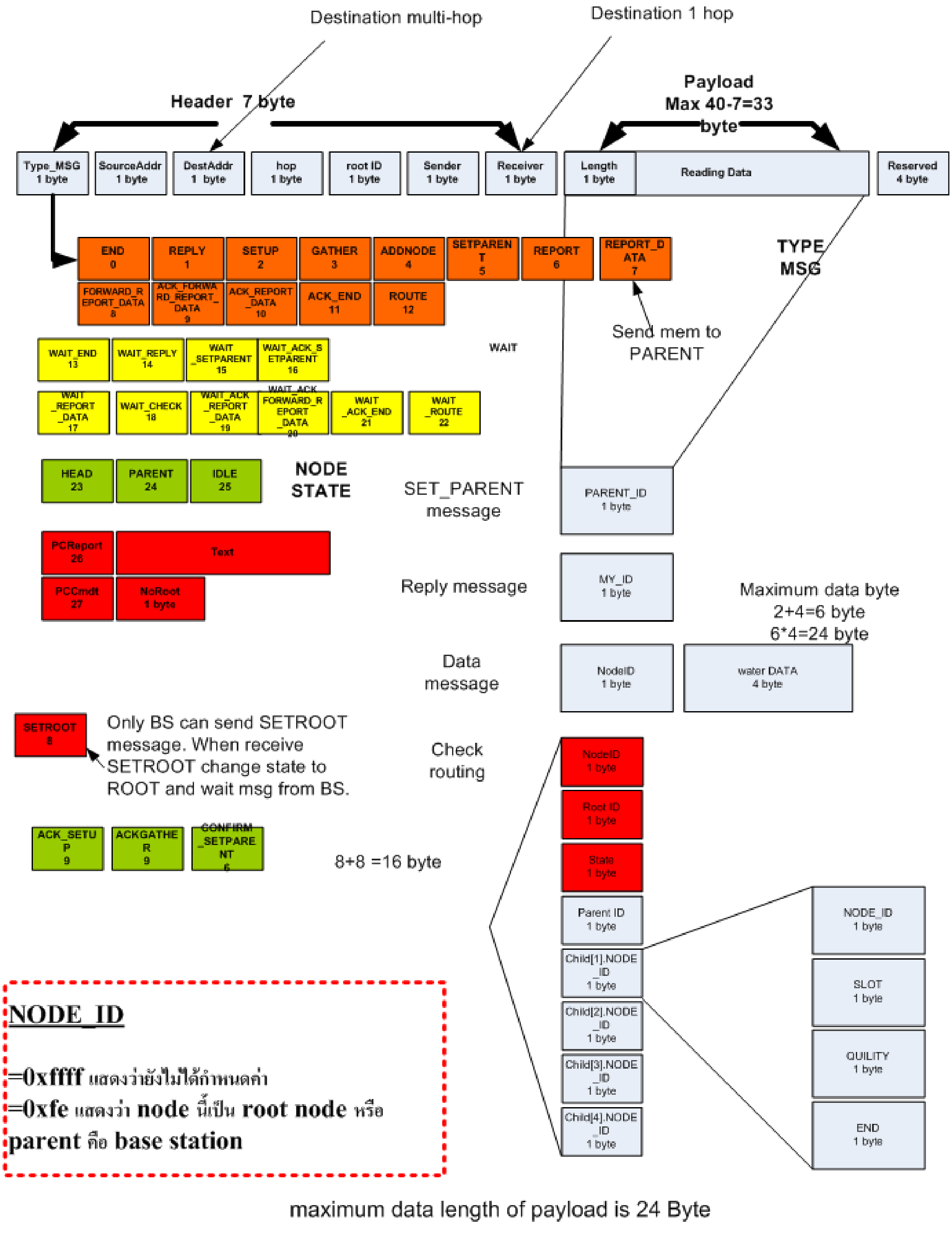


ภาพผนวกที่ 1 Module ที่สร้างเพิ่มขึ้นใน TinyOS

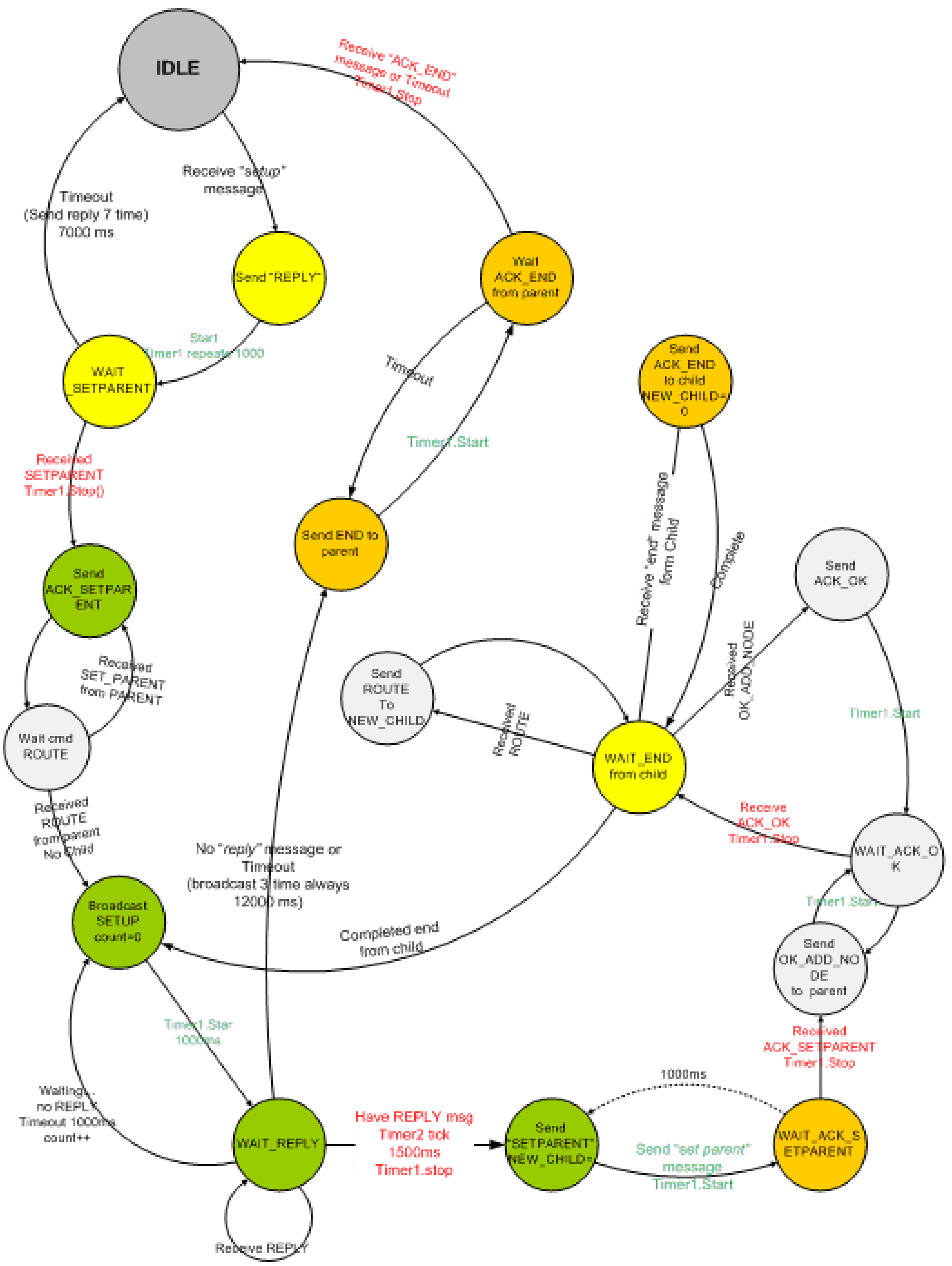
จากภาพผนวกที่ 1 เป็น Module ที่สร้างเพิ่มขึ้นใน TinyOS ประกอบด้วย 2 ส่วนคือ Receive Management และ Send Management โดย Receive Management ทำหน้าที่จัดการ Packet ที่เข้ามาบนบอร์ดพร้อมทั้งส่ง Acknowledgement กลับไปได้ ส่วนทางด้านของ Send Management จะทำหน้าที่จัดการ Packet ที่จะส่งออกจากบอร์ดพร้อมทั้งตรวจสอบ Acknowledgement ถ้าไม่ได้ก็จะส่งซ้ำออกไป



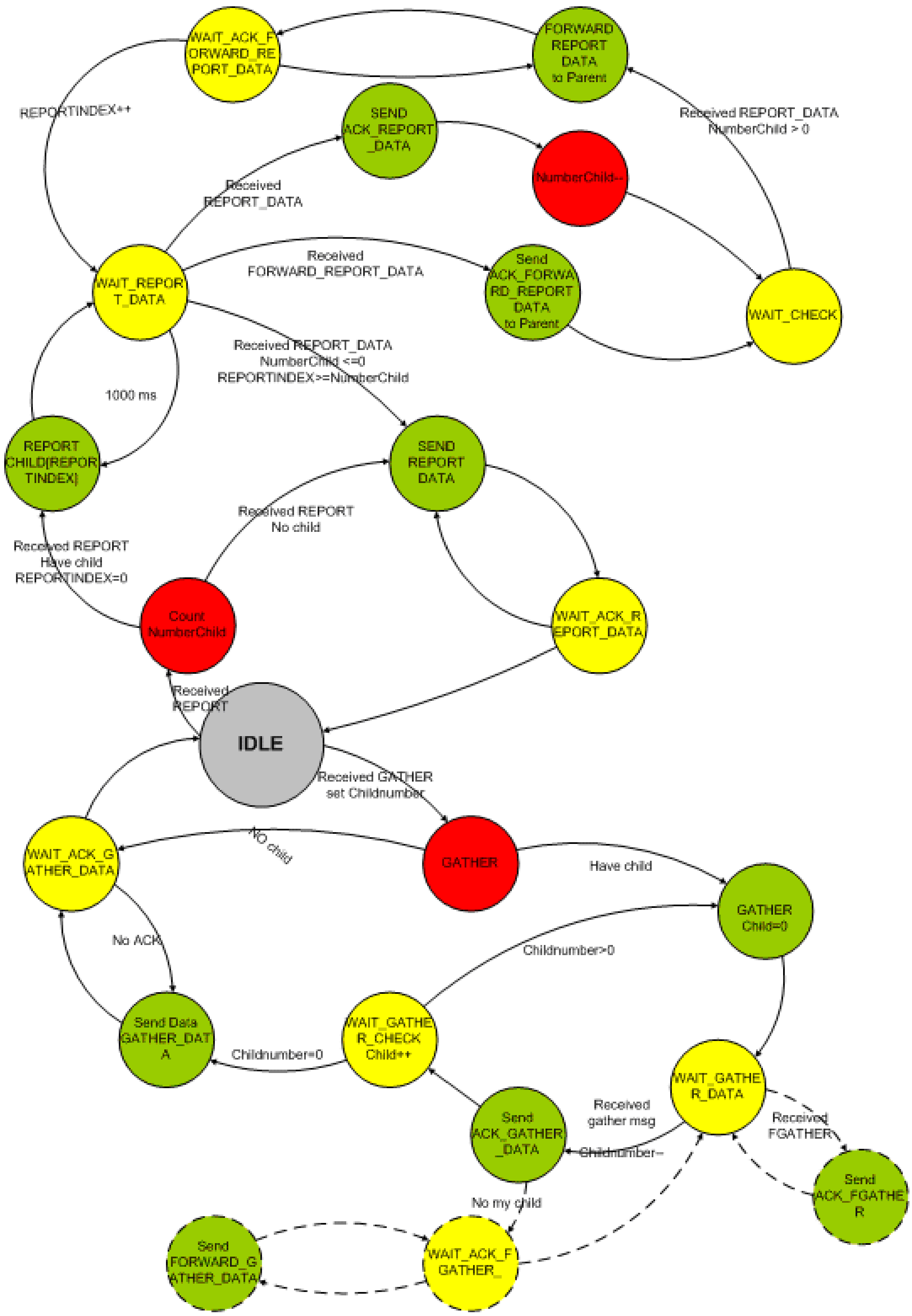
ภาพผนวกที่ 2 วิธีการตรวจสอบ Packet ในการส่ง Packet ซ้ำ



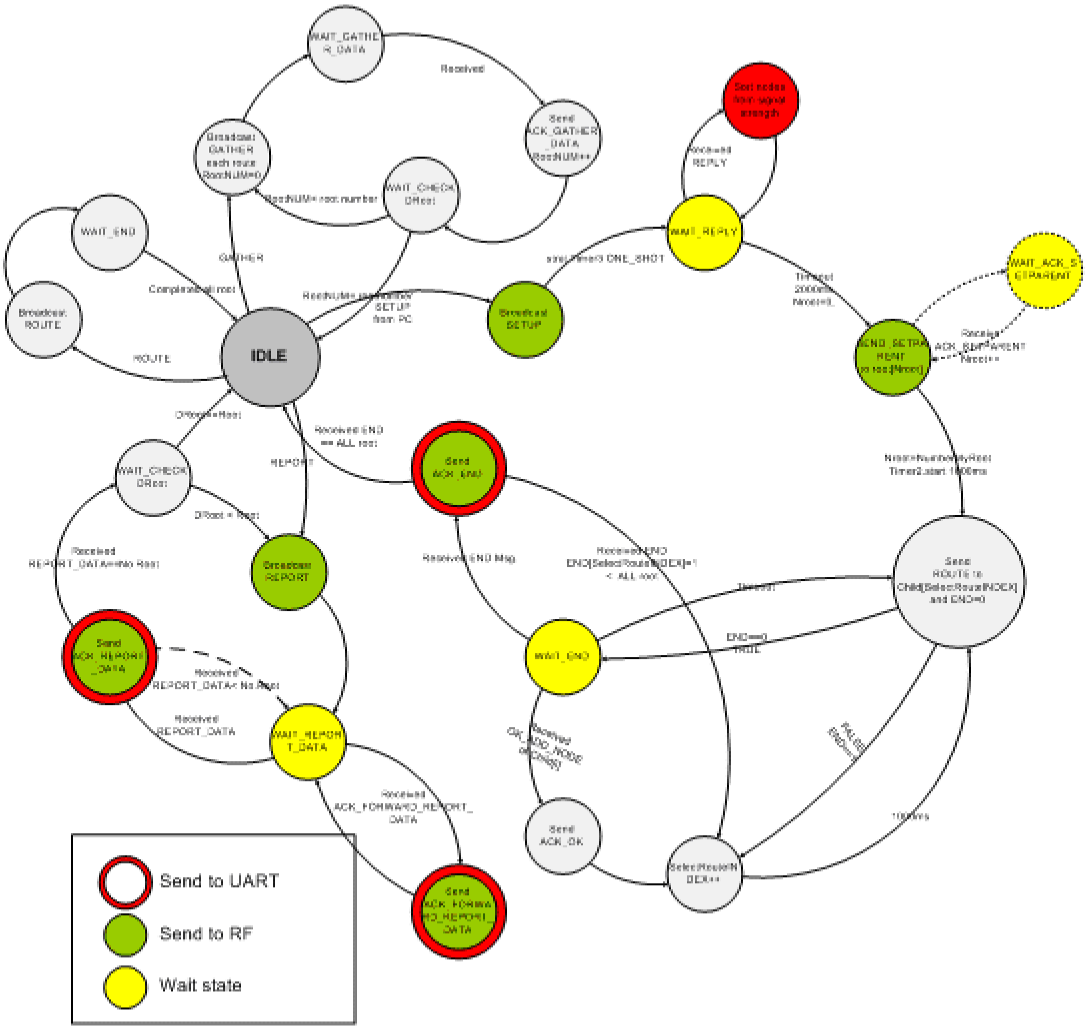
ภาพผนวกที่ 3 Header และ Type ของ packet ที่ใช้ในการสร้าง, รายงาน และเก็บ ข้อมูล



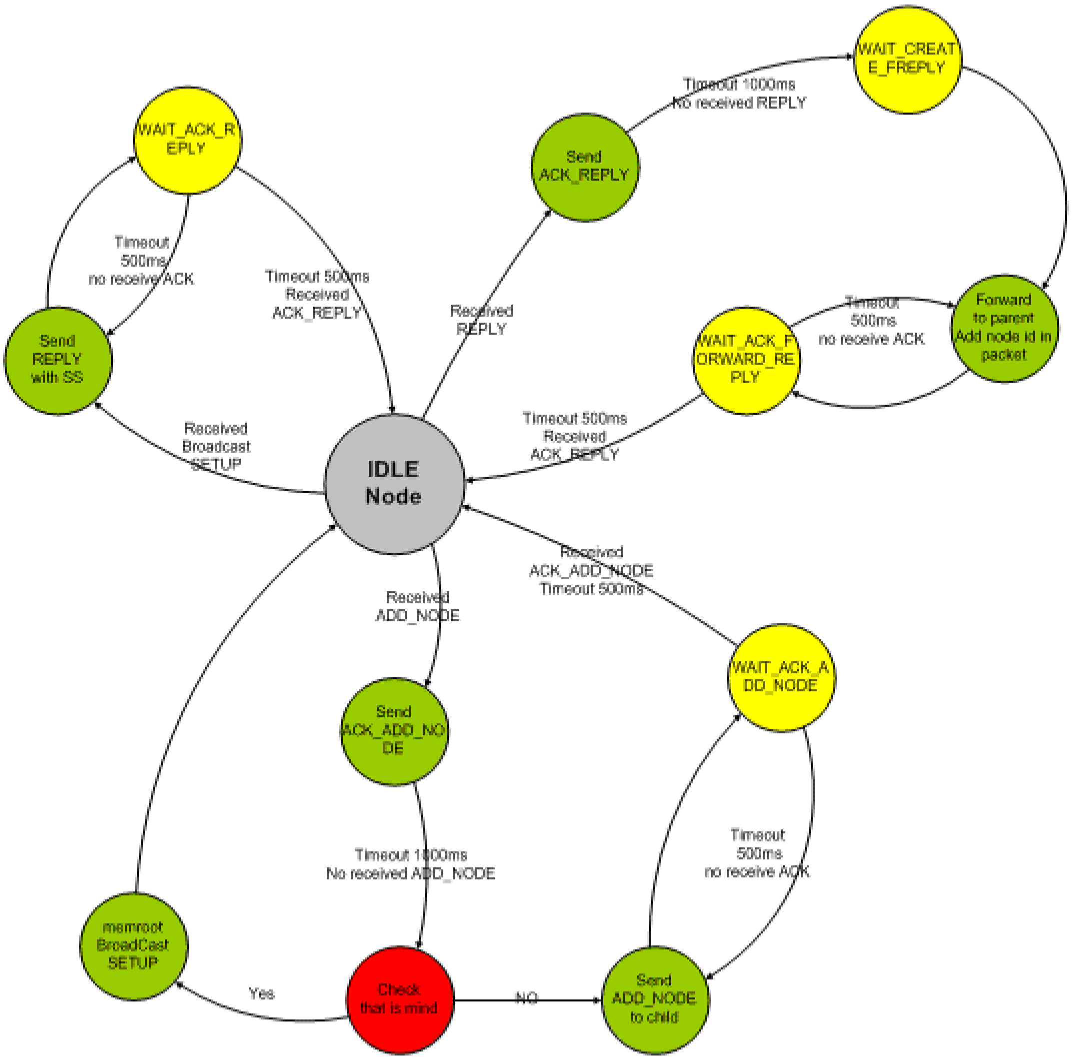
ภาพผนวกที่ 4 สถานะการทำงานของ node ในการสร้างเส้นทางใน EFMST



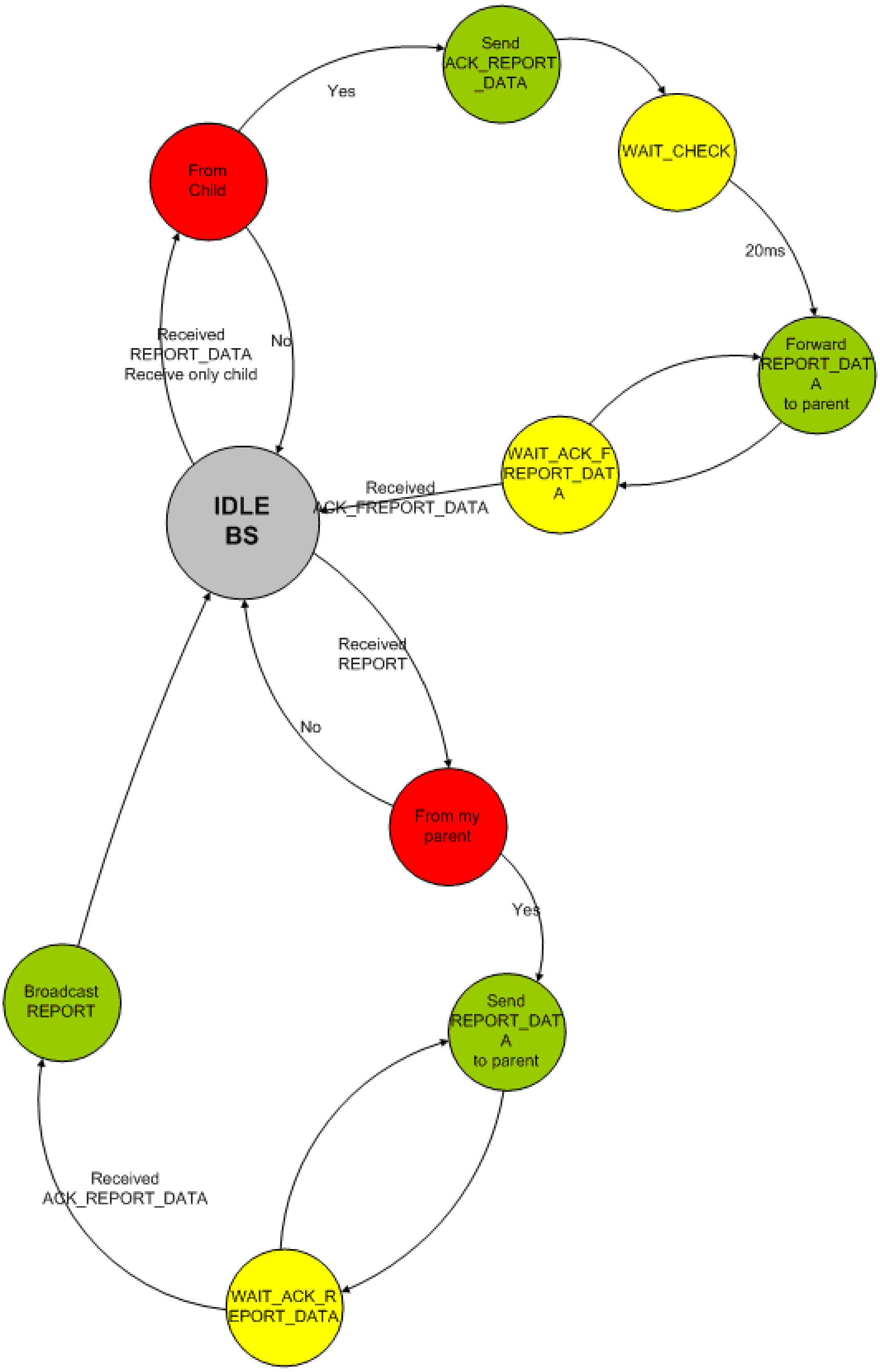
ภาพผนวกที่ 5 สถานะการทำงานของ node ในการเก็บข้อมูลและรายงานเส้นทางใน EFMST



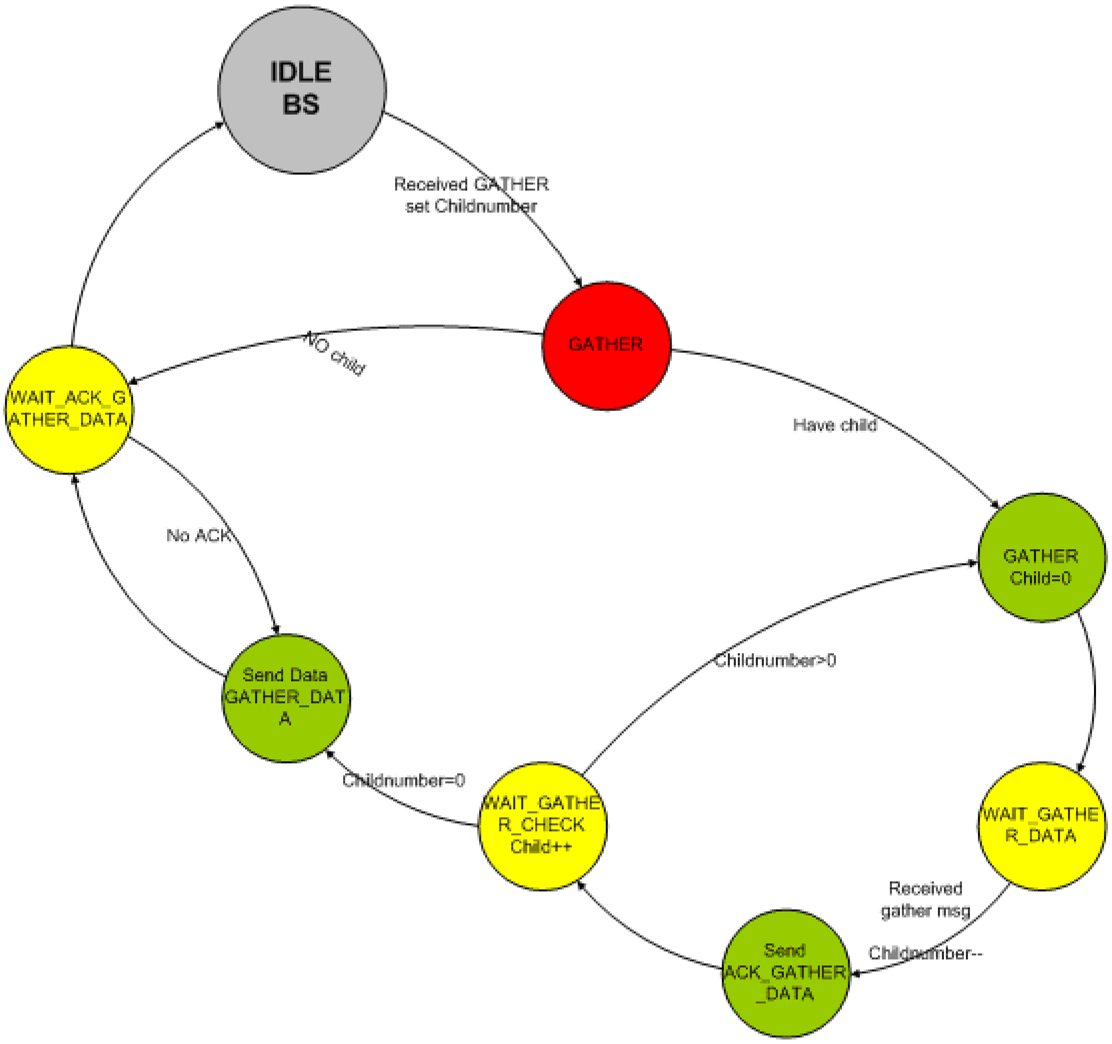
ภาพผนวกที่ 6 สถานะการทำงานของ base station ใน EFMST



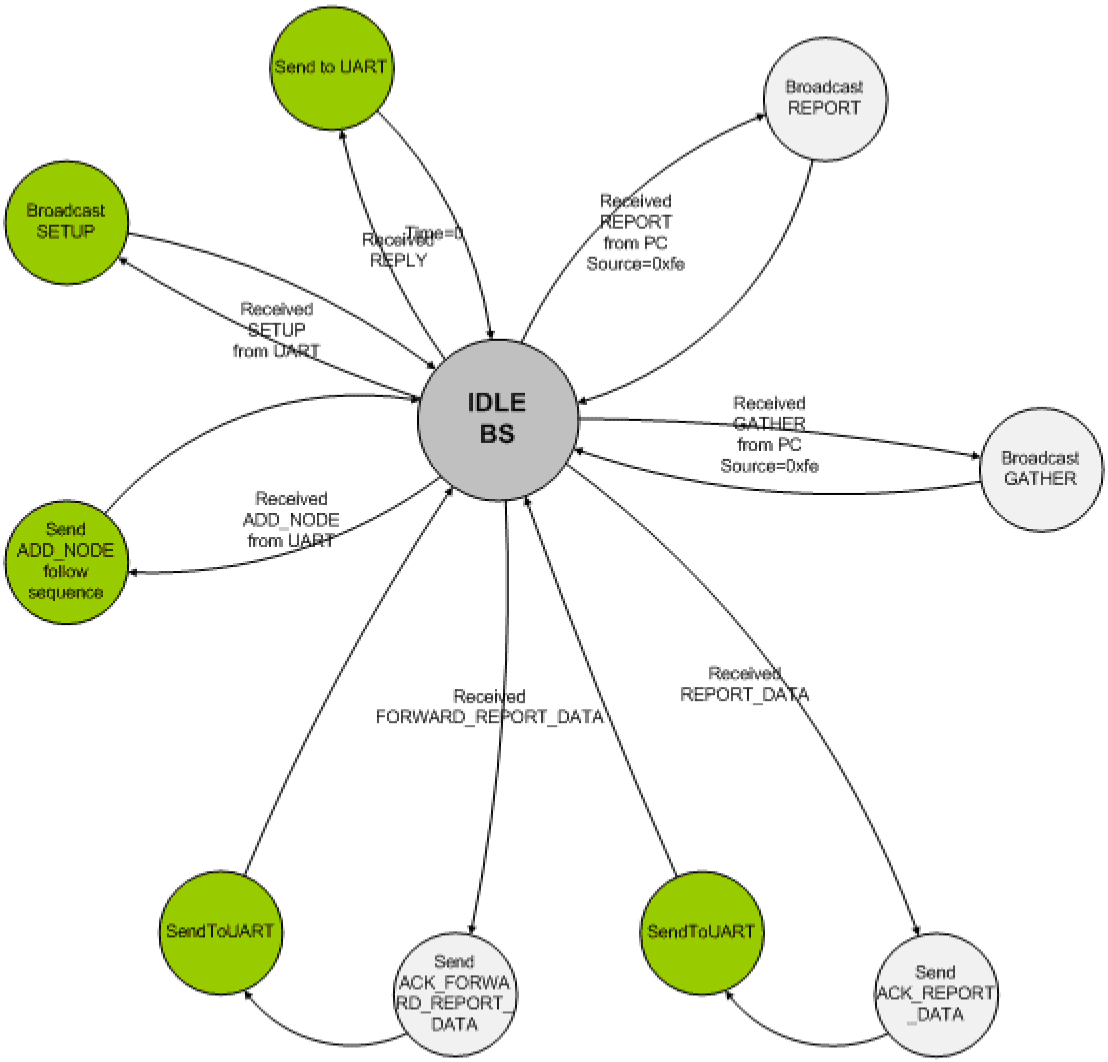
ภาพผนวกที่ 7 สถานะการทำงานของ node ในการสร้างเส้นทางใน MST



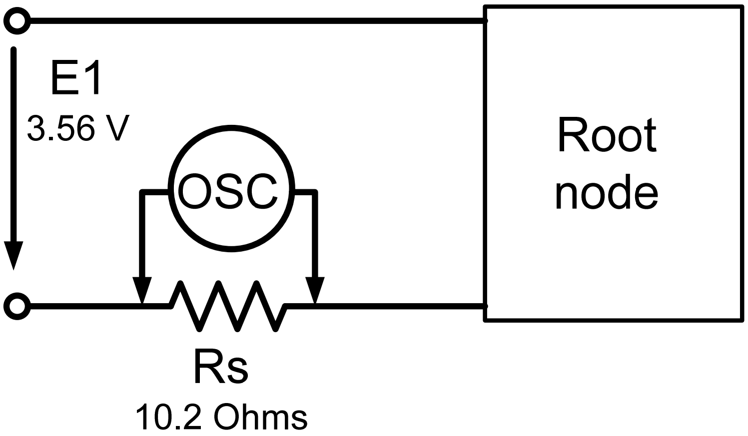
ภาพผนวกที่ 8 สถานะการทำงานของ node ในการรายงานเส้นทางใน MST



ภาพผนวกที่ 9 สถานะการทำงานของ node ในการเก็บข้อมูลใน MST



ภาพผนวกที่ 10 สถานะการทำงานของ base station ใน MST



ภาพผนวกที่ 11 วงจรวัดกระแสของ root node ที่ใช้ในการทดลองกับบอร์ดจริง

ประวัติการศึกษา และการทำงาน

ชื่อ –นามสกุล	นายเอกลักษณ์ เวศนารัตน์
วัน เดือน ปี ที่เกิด	วันที่ 9 ธันวาคม 2525
สถานที่เกิด	โรงพยาบาลเชียงใหม่
ประวัติการศึกษา	ระดับปริญญาตรี คณวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ตำแหน่งหน้าที่การงานปัจจุบัน	นักศึกษา
สถานที่ทำงานปัจจุบัน	Thai-asahi alloy
ผลงานดีเด่นและรางวัลทางวิชาการ	
ทุนการศึกษาที่ได้รับ	