



โพรโทคอลค้นหาเส้นทางสำรองสำหรับเครือข่ายตรวจสอบไร้สาย
Redundant Path Routing Protocol for Wireless Sensor Networks

วันประชา นวนสร้อย
Wanpracha Nuansoi

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

**A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Computer Engineering
Prince of Songkla University**

2553

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

(1)

ชื่อวิทยานิพนธ์ โพรโทคอลค้นหาเส้นทางสำรองสำหรับเครือข่ายตรวจจับไร้สาย
ผู้เขียน นายวันประชา นวนสร้อย
สาขาวิชา วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	คณะกรรมการสอบ
..... (ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต)ประธานกรรมการ (ดร.ศกุนา เจริญปัญญาศักดิ์)
กรรมการ (ผู้ช่วยศาสตราจารย์ ดร.ศักดิ์ชัย ทิพย์จักรมูร์ตน์)
กรรมการ (ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้บัณฑิตวิทยาลัยนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

.....
(รองศาสตราจารย์ ดร.เกริกชัย ทองหนู)
คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์	โพรโทคอลค้นหาเส้นทางสำรองสำหรับเครือข่ายตรวจจับไร้สาย
ผู้เขียน	นายวันประชา นวนสร้อย
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2552

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอโพรโทคอลค้นหาเส้นทางสำรองสำหรับเครือข่ายตรวจจับไร้สายที่สามารถใช้เพิ่มความน่าเชื่อถือในการสื่อสารข้อมูลในเครือข่ายตรวจจับไร้สาย โพรโทคอลค้นหาเส้นทางที่ประหยัดพลังงานที่มีใช้ในเครือข่ายตรวจจับไร้สายส่วนมากมีเพียงเส้นทางเดียว ส่งผลให้ต้องรอเวลาในการสร้างเส้นทางใหม่เมื่อเส้นทางหลักล้มเหลว ซึ่งเป็นปัญหาให้เกิดการขาดความต่อเนื่องในการส่งข้อมูล วิทยานิพนธ์นี้นำเสนอโพรโทคอลค้นหาเส้นทางสำรองสำหรับเครือข่ายตรวจจับไร้สายเพื่อแก้ปัญหาดังกล่าว ซึ่งจะนำเสนอในส่วนอัลกอริทึมในการค้นหาเส้นทางสำรองและการกำหนดพลังงานขั้นต่ำก่อนการเชื่อมต่อ ในกรณีเส้นทางหลักที่เชื่อมต่อล้มเหลวสามารถใช้งานเส้นทางสำรองได้ทันที โหนดต้นทางสามารถเลือกเชื่อมต่อเส้นทางที่ใช้พลังงานต่ำไปยังโหนดปลายทางได้ จากการทดลองโดยใช้โปรแกรมจำลองเครือข่าย NS-2 ได้แสดงผลของการมีเส้นทางสำรองที่สามารถเพิ่มความน่าเชื่อถือให้กับเครือข่ายตรวจจับไร้สายได้

คำสำคัญ: ความน่าเชื่อถือ, เส้นทางสำรอง, ประหยัดพลังงาน

Thesis Title	Redundant Path Routing Protocol for Wireless Sensor Networks
Author	Mr. Wanpracha Nuansoi
Major Program	Computer Engineering
Academic Year	2009

ABSTRACT

This research presents Redundant Path Routing Protocol which can be used to increase reliability in communicating information in wireless sensor networks. Most Energy Aware Routing Protocols available for wireless sensor networks recently depend on one working route, resulting in long waiting time to reconstruct the route when the first try fails. This also presents the problem of information discontinuity. This research proposes Redundant Path Routing Protocol as a solution to those problems. This protocol is presented with algorithm for redundant path and the pre-set minimum energy for connection. In the case that the main path of connection fails, the connection can work instantly on the provided redundant path of connection, without making request for route reconstruction. The connection of source node and destination node can be designed for less energy-consumed route. The NS-2 network simulation shows that the provision of redundant path can increase reliability in connection for wireless sensor networks.

Keywords: reliability, redundant path, energy aware

กิตติกรรมประกาศ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ที่ได้ให้คำปรึกษา ชี้แนะแนวทาง และให้ความรู้ในด้านต่างๆ รวมถึงการให้การสนับสนุนในเรื่องการทำวิจัย ตลอดจนช่วยตรวจและแก้ไขวิทยานิพนธ์ให้เป็นไปอย่างสมบูรณ์

ขอขอบพระคุณ ดร. สกุนา เจริญปัญญาศักดิ์ ที่กรุณาให้เกียรติเป็นประธานกรรมการสอบวิทยานิพนธ์ ให้คำแนะนำที่มีประโยชน์ต่อการวิจัยตลอดจนตรวจแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. ศักดิ์ชัย ทิพย์จักษ์รัตน์ ที่กรุณาให้เกียรติเป็นกรรมการสอบวิทยานิพนธ์ ให้คำแนะนำที่มีประโยชน์ต่อการวิจัยตลอดจนตรวจแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณ ดร.นิคม สุวรรณวร คณาจารย์ และบุคลากรในภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ให้คำปรึกษาและความช่วยเหลือในระหว่างการทำวิทยานิพนธ์

ขอขอบคุณนักศึกษาปริญญาโทสาขาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ได้ให้คำแนะนำและเป็นกำลังใจมาโดยตลอด

และสุดท้ายนี้ ขอน้อมรำลึกถึงพระคุณบิดา มารดา และครอบครัว ที่ส่งเสริมให้กำลังใจ และให้การสนับสนุนในเรื่องต่างๆ จนกระทั่งข้าพเจ้าประสบความสำเร็จในการศึกษา

วันประชา นวนสร้อย

สารบัญ

	หน้า
สารบัญ	(6)
รายการตาราง.....	(8)
รายการภาพประกอบ	(9)
สัญลักษณ์คำย่อและตัวย่อ.....	(11)
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของหัวข้อวิจัย.....	1
1.2 การตรวจเอกสาร	2
1.3 วัตถุประสงค์	3
1.4 ขอบเขตของการวิจัย	4
1.5 ขั้นตอนและวิธีการวิจัย	4
1.6 ประโยชน์ที่คาดว่าจะได้รับ	4
1.7 ทรัพยากรที่ใช้ในระบบ.....	5
1.8 ภาพรวมของขั้นตอนการทำงานและพัฒนาโปรโตคอล	5
บทที่ 2 ทฤษฎีและหลักการ	7
2.1 ความรู้เบื้องต้นเกี่ยวกับเครือข่ายตรวจจับไร้สาย.....	7
2.1.1 ตัวอย่างโปรโตคอลประหยัดพลังงานในเครือข่ายตรวจจับไร้สาย	8
2.2 โปรแกรมจำลองเครือข่าย	9
2.3 โปรโตคอล DSR	10
2.3.1 การค้นหาเส้นทางของโปรโตคอล DSR.....	11
2.3.2 การบำรุงรักษาเส้นทางของโปรโตคอล DSR	14
2.4 โปรโตคอล AODV.....	17
2.4.1 การค้นหาเส้นทาง	17
2.4.2 การบำรุงรักษาเส้นทาง	19
2.5 การประมาณค่าพลังงานในการค้นหาเส้นทาง	20
2.5.1 พลังงานที่ใช้ในการกระจายข้อความร้องขอเส้นทาง	21
2.5.2 พลังงานที่ใช้ในการส่งข้อความตอบรับเส้นทาง	23

สารบัญ (ต่อ)

	หน้า
บทที่ 3 การออกแบบระบบ.....	25
3.1 ขั้นตอนในการออกแบบระบบ.....	25
3.1.1 การพัฒนาอัลกอริทึมการค้นหาเส้นทาง และการรักษาเส้นทาง.....	25
3.1.2 ตัวอย่างการค้นหาเส้นทาง.....	29
3.1.3 ตัวอย่างการใช้เส้นทางสำรอง.....	35
3.2 ลำดับขั้นตอนการทดลอง.....	37
3.3 รูปแบบการทดลอง.....	38
3.4 สมการประมาณค่าพลังงานในการค้นหาเส้นทางของโปรโตคอล RPRP.....	40
3.4.1 โหนดต้นทาง.....	40
3.4.2 โหนดระหว่างกลาง.....	41
3.4.3 โหนดปลายทาง.....	43
บทที่ 4 พารามิเตอร์และการวิเคราะห์ผลการทดลอง.....	46
4.1 ค่า Throughput โดยเฉลี่ย (T_a).....	46
4.2 ค่าเฉลี่ยการใช้พลังงานโดยรวม (E_a).....	48
4.3 อัตราส่วนเฉลี่ยการรับส่งข้อมูล (R_a).....	50
4.4 เวลาในการค้นหาเส้นทางเฉลี่ย (RD_a).....	51
4.5 เวลาในการสลับใช้เส้นทางสำรองเฉลี่ย (ST_a).....	53
บทที่ 5 บทสรุปและข้อเสนอแนะ.....	55
5.1 บทสรุป.....	55
5.2 ข้อเสนอแนะ.....	56
เอกสารอ้างอิง.....	57
ภาคผนวก.....	59
ประวัติผู้เขียน.....	78

รายการตาราง

ตาราง	หน้า
ตาราง 3-1 สถานะโหนด.....	25
ตาราง 3-2 อัลกอริทึมการหาเส้นทางของโพรโทคอล RPRP	27
ตาราง 3-3 อัลกอริทึมการรักษาเส้นทาง.....	28
ตาราง 3-4 การคำนวณผลต่างในการคัดเลือกโหนด	30
ตาราง 3-5 สภาพแวดล้อมในการทดลอง.....	39
ตาราง 4-1 การเปรียบเทียบค่า Throughput	47
ตาราง 4-2 การเปรียบเทียบพลังงานที่ใช้ไปในการทดลอง	49
ตาราง 4-3 การเปรียบเทียบอัตราส่วนการรับส่งข้อมูล.....	51
ตาราง 4-4 การเปรียบเทียบเวลาในการค้นหาเส้นทาง	52
ตาราง 4-5 เวลาในการสลับใช้เส้นทางสำรอง.....	53

รายการภาพประกอบ

ภาพประกอบ	หน้า
ภาพประกอบ 1-1 ขั้นตอนการพัฒนาโปรโตคอล	6
ภาพประกอบ 2-1 การทำงานของเครือข่ายตรวจจับไร้สาย.....	7
ภาพประกอบ 2-2 แสดงการหาเส้นทางใน Energy Aware Routing	8
ภาพประกอบ 2-3 โครงสร้างพื้นฐานโดยรวมของ NS-2	9
ภาพประกอบ 2-4 ตัวอย่างของ NAM ใน NS-2.....	10
ภาพประกอบ 2-5 การค้นหาเส้นทางใน DSR.....	11
ภาพประกอบ 2-6 การตอบกลับข้อความร้องขอเส้นทางโดยใช้เส้นทางใน Cache.....	12
ภาพประกอบ 2-7 ข้อความตอบกลับเส้นทางท่วมท้น	13
ภาพประกอบ 2-8 ตัวอย่างการบำรุงรักษาเส้นทาง.....	14
ภาพประกอบ 2-9 การรับฟังข้อมูลเส้นทางเพื่อเก็บไว้ใน Cache	15
ภาพประกอบ 2-10 ตัวอย่างการปรับปรุงทางให้สั้นขึ้น	16
ภาพประกอบ 2-11 การค้นหาเส้นทางของโปรโตคอล AODV.....	18
ภาพประกอบ 2-12 การบำรุงรักษาเส้นทาง	20
ภาพประกอบ 2-13 การใช้พลังงานในการกระจายข้อความ ROUTE_REQ	21
ภาพประกอบ 2-14 การใช้พลังงานในการส่ง ROUTE_REP.....	23
ภาพประกอบ 3-1 การค้นหาเส้นทางในโปรโตคอล RPRP.....	29
ภาพประกอบ 3-2 การกระจายข้อความ ROUTE_REQ.....	29
ภาพประกอบ 3-3 การส่งข้อความ RESULT REQ	30
ภาพประกอบ 3-4 การส่งข้อความ RESULT REP.....	31
ภาพประกอบ 3-5 โหนดที่ได้รับการคัดเลือกให้เชื่อมต่อ	31
ภาพประกอบ 3-6 โหนดปลายทางส่งข้อความ ROUTE_REP	32
ภาพประกอบ 3-7 การส่งข้อความ ROUTE_REP กลับไปยังต้นทาง.....	32
ภาพประกอบ 3-8 การส่งข้อความ ROUTE_REP กลับไปยังต้นทาง.....	33
ภาพประกอบ 3-9 การค้นหาเส้นทางที่ 2	33
ภาพประกอบ 3-10 การตอบกลับเส้นทางที่ 2 และสิ้นสุดการคัดเลือกโหนด	34
ภาพประกอบ 3-11 การส่งข้อความ ROUTE_REP กลับไปยังต้นทางของเส้นทางที่ 2	34
ภาพประกอบ 3-12 การเลือกเส้นทางหลักและเส้นทางสำรอง	35

รายการภาพประกอบ (ต่อ)

ภาพประกอบ	หน้า
ภาพประกอบ 3-13 เส้นทางหลักล้มเหลว.....	36
ภาพประกอบ 3-14 การใช้เส้นทางสำรอง.....	36
ภาพประกอบ 3-15 ขั้นตอนการทดลอง.....	37
ภาพประกอบ 3-16 พลังงานที่โหนดต้นทางกระจาย ROUTE_REQ.....	40
ภาพประกอบ 3-17 พลังงานที่โหนดต้นทางรับ RESULT_REQ.....	41
ภาพประกอบ 3-18 พลังงานที่โหนดต้นทางส่ง RESULT_REP.....	41
ภาพประกอบ 3-19 พลังงานที่โหนดระหว่างกลางที่ได้รับการคัดเลือกส่ง ROUTE_REQ.....	42
ภาพประกอบ 3-20 พลังงานที่โหนดระหว่างกลางที่ได้รับการคัดเลือกรับ RESULT_REQ.....	42
ภาพประกอบ 3-21 พลังงานที่โหนดระหว่างกลางที่ได้รับการคัดเลือกส่ง RESULT_REP.....	43
ภาพประกอบ 3-22 พลังงานที่โหนดปลายทางรับ ROUTE_REQ.....	43
ภาพประกอบ 4-1 การเปรียบเทียบค่า Throughput โดยเฉลี่ย.....	48
ภาพประกอบ 4-2 พลังงานที่ใช้ไปในการทดลองโดยเฉลี่ย.....	50
ภาพประกอบ 4-3 การเปรียบเทียบอัตราส่วนการรับส่งข้อมูล โดยเฉลี่ย.....	51
ภาพประกอบ 4-4 การเปรียบเทียบเวลาในการค้นหาเส้นทางโดยเฉลี่ย.....	53

สัญลักษณ์คำย่อและตัวย่อ

RPRP	Redundant Path Routing Protocol.
NS-2	Network Simulator 2.
DERP	Delay-Minimum Energy-Aware Routing Protocol.
EARP	Energy Aware Routing Protocol.
DSR	Dynamic Source Routing.
AODV	Ad-hoc On-Demand Distance Vector.
SPIN	Sensor Protocols for Information via Negotiation.
LEACH	Low-Energy Adaptive Clustering Hierarchy.
TEEN	Threshold sensitive Energy Efficient sensor Network.
MECN	Minimum Energy Communication Network.
GAF	Geographic Adaptive Fidelity.
ROUTE_REQ	Route Request.
ROUTE_REP	Route Reply.
RESULT_REQ	Result Request.
RESULT_REP	Result Reply.
ROUTE_FAIL	Route Fail.
RMET	Route Maintenance Base on Energy Threshold.
RRDT	Route Request Delay Transmit.
ISI	Information Sciences Institute.
UDP	User Datagram Protocol.
TCP	Transmission Control Protocol.
RTP	Real-time Transport Protocol.
NAM	Network Animator.
TTL	Time To Live.
RERR	Route Error.
CBR	Constant Bit Rate.
ms	Millisecond.

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของหัวข้อวิจัย

เทคโนโลยีเครือข่ายตรวจจับไร้สายประกอบด้วยอุปกรณ์ขนาดเล็กที่เรียกว่าโหนดที่สามารถสื่อสารแบบไร้สายได้ เหมาะที่จะใช้กระจายการตรวจจับลงในพื้นที่ที่สนใจ และสามารถติดต่อสื่อสารเป็นเครือข่ายแบบไร้สายได้ จึงเหมาะในงานประยุกต์หลายด้าน อาทิเช่น การตรวจจับสภาพแวดล้อมและแหล่งที่อยู่ การควบคุมจราจร เหตุการณ์ฉุกเฉิน และการดูแลสุขภาพ [1, 2]

โหนดตรวจจับไร้สายใช้พลังงานจากแบตเตอรี่ ทำให้เกิดข้อจำกัดขึ้นมากมายและเป็นปัญหาวิจัยทางด้านเทคโนโลยีเครือข่ายตรวจจับไร้สาย แต่ปัญหาของการลดการใช้พลังงานในโหนดยังไม่มีวิธีการแก้ปัญหาที่ดีที่สุด ดังนั้นผู้ออกแบบ และพัฒนาจะต้องคำนึงถึงเรื่องของการใช้พลังงานในทุกระดับชั้นของระบบ เช่น เน้นการประหยัดพลังงานในการทำงานของไมโครคอนโทรลเลอร์ที่มีโหมด Sleep หรือ idle ไว้ในช่วงที่ยังไม่มีการประมวลผล หรือการสั่งโหมดการทำงาน Sleep หรือ idle ในภาครับส่งวิทยุเพื่อประหยัดพลังงานในส่วนของการรับส่งข้อมูลเป็นต้น

การส่งข้อมูลโดยทั่วไปแต่ละโหนดจะส่งข้อมูลกลับไปยังสถานีแม่ทั้งเป็นแบบ Single-hop หรือ Multi-hop ซึ่งจะทำให้โหนดที่อยู่บริเวณใกล้กับสถานีแม่หมดพลังงานก่อน ดังนั้นหัวข้อวิจัยที่น่าสนใจคือ การยืดอายุการใช้งานของเครือข่าย นอกจากนี้เครือข่ายตรวจจับไร้สายที่ต้องการเส้นทางสำรองเพื่อเพิ่มความน่าเชื่อถือให้กับเครือข่าย แต่ก็จะไปกระทบต่อการใช้พลังงานของโหนด ซึ่งความน่าเชื่อถือในการสื่อสารข้อมูลในเครือข่ายตรวจจับไร้สายยังมีน้อย เมื่อเทียบกับเครือข่ายประเภทอื่นๆ ดังนั้นการมีเส้นทางสำรองจึงจำเป็นที่จะต้องมีในเครือข่ายตรวจจับไร้สาย

โพรโทคอลการหาเส้นทางที่ประหยัดพลังงานส่วนใหญ่จะสร้างเส้นทางเดียวในการสื่อสารเพื่อเป็นการประหยัดพลังงาน แต่จะเกิดปัญหาเมื่อเส้นทางนั้นล้มเหลวเพราะจะต้องสร้างเส้นทางใหม่ ทำให้ขาดความน่าเชื่อถือในการสื่อสารข้อมูล ดังนั้นหัวข้อวิจัยในวิทยานิพนธ์จึงต้องการเน้นการสร้างเส้นทางสำรองเป็นหลัก

1.2 การตรวจเอกสาร

1. A Survey on Routing Protocols for Wireless Sensor Networks [3]

ในบทความนี้แบ่งประเภทของโพรโทคอลในเครือข่ายตรวจจับไร้สายออกเป็น 3 ประเภท คือ Data-centric, Hierarchical และ Location-based โพรโทคอลในประเภท Data-centric เช่น Directed Diffusion [4], Sensor Protocols for Information via Negotiation: (SPIN) [5] เป็นต้น โพรโทคอลในประเภท Hierarchical ตัวอย่าง เช่น Low-Energy Adaptive Clustering Hierarchy (LEACH) [6], Threshold sensitive Energy Efficient sensor Network protocol (TEEN) [7] และ โพรโทคอลประเภท Location-based เช่น Minimum Energy Communication Network (MECN) [8] และ Geographic Adaptive Fidelity (GAF) [9]

2. Energy Aware Routing for Wireless Sensor Networks [10]

บทความนี้ได้เสนอโพรโทคอลที่คำนึงถึงพลังงาน โดยมีเพียงเส้นทางเดียวจากต้นทางถึงปลายทางเพื่อประหยัดพลังงาน โดยเลือกโหนดที่มีพลังงานสูง และมีระดับสัญญาณที่ดีในการเชื่อมต่อไปยังปลายทาง ประกอบด้วยสามกลไกคือ Route Discovery, Route Reply และ Reliable Transmission ซึ่งได้อธิบายไว้ในบทที่ 2

3. A New Energy-Aware Routing Protocol for Wireless Sensor Networks [11]

บทความนี้ได้นำเสนอสองกลไกทำงานหลักคือ Route Maintenance Based on Energy Threshold (RMET) โดยนำเสนอวิธีการที่ทำให้พลังงานแบตเตอรี่ที่มีในโหนดใช้ให้นานที่สุดเพื่อที่จะประยุกต์ใช้ในการประมวลผล Route Maintenance ของ Dynamic Source Routing (DSR) ส่วนกลไกที่สองคือ Route Request Delay Transmit (RRDT) คือ โหนดระหว่างกลางที่มีระดับพลังงานสูงกว่าระดับพลังงานขั้นต่ำ จะรอเวลาที่ถึงที่ในการส่งข้อความ Route Request (ROUTE_REQ)

4. Energy Aware Routing Protocol (EARP) for Ad-Hoc Wireless Sensor Networks [12]

บทความนี้ได้เสนอโพรโทคอลที่คำนึงพลังงานสำหรับเครือข่ายตรวจจับไร้สาย และสมการประมาณค่าพลังงานที่ใช้ในการค้นหาเส้นทาง โดยสมการจะแบ่งเป็นสองส่วนคือ สมการประมาณค่าพลังงานในการกระจายข้อความร้องขอเส้นทาง และสมการประมาณค่าพลังงานในการส่งข้อความตอบกลับเส้นทาง ซึ่งในส่วนของสมการประมาณค่าพลังงานในการค้นหาเส้นทางจะได้ อธิบายไว้ในบทที่ 2

5. Energy Aware Routing for Low Energy Ad Hoc Sensor Networks [13]

บทความนี้ได้เสนอให้ใช้กลุ่มของเส้นทางที่ดีที่สุดตามโอกาสที่มี เพื่อเพิ่มช่วงเวลาการใช้เครือข่าย โดยใช้ฟังก์ชันความเป็นไปได้ในการเลือกเส้นทาง ซึ่งขึ้นอยู่กับการใช้พลังงานของแต่ละเส้นทาง มีเพียงหนึ่งในหลายเส้นทางที่ได้จากความน่าจะเป็นที่จะเพิ่มช่วงเวลาการใช้เครือข่ายในโพรโทคอลประกอบด้วย 3 ส่วนคือ การติดตั้งเส้นทาง การติดต่อสื่อสาร การบำรุงรักษาเส้นทาง

6. Delay-Minimum Energy-Aware Routing Protocol (DERP) for Wireless Sensor Networks [14]

โพรโทคอลค้นหาเส้นทางนี้ประกอบด้วยสามส่วน คือ Delay-Minimum Route Tree, Data Delivery และ Route Update ส่วนแรกของ DERP คือ Delay-Minimum Route Tree ซึ่งมีประสิทธิภาพสูง ในส่วนนี้จะพยายามหาเส้นทางที่ดีที่สุดจากโหนดต้นทางไปยังโหนดปลายทาง โดยใช้เส้นทางที่มีค่าหน่วยเวลา (Delay) น้อยสุดไปยังโหนดปลายทาง ส่วน Data Delivery เป็นการส่งข้อมูลจากโหนดต้นทางผ่านทาง Delay-Minimum Route Tree เพื่อไปยังโหนดปลายทาง และ Route Update จะเป็นการปรับปรุงเส้นทางเมื่อระดับพลังงานของโหนดในเส้นทางต่ำกว่าพลังงานที่กำหนด

7. An Energy Aware Routing Algorithm For Ad hoc And Sensor Networks [15]

บทความนี้ได้เสนอการวิเคราะห์วิธีการเลือกหาเส้นทางจากการแบ่งโซนที่คำนึงถึงการใช้พลังงาน เพื่อใช้ในเครือข่ายตรวจจับไร้สาย เป็นอัลกอริทึมที่จัดการประสิทธิภาพทรัพยากรในการรับ และส่งข้อความในเครือข่ายระหว่างที่โหนดกำลังค้นหาเส้นทาง โดยจะปรับปรุงอัตราการใช้พลังงานโดยเฉลี่ยให้ดีขึ้น

1.3 วัตถุประสงค์

1. เพื่อเพิ่มความน่าเชื่อถือในการสื่อสารข้อมูลในเครือข่ายตรวจจับไร้สาย
2. เพื่อนำเสนอโพรโทคอลค้นหาเส้นทางสำรองสำหรับเครือข่ายตรวจจับไร้สาย
3. เพื่อนำเสนอโพรโทคอลที่โหนดต้นทางสามารถเลือกใช้เส้นทางที่มีการใช้พลังงานต่ำได้
4. เพื่อนำเสนอโพรโทคอลที่สามารถเพิ่มค่า Throughput ได้จากที่มีเส้นทางสำรอง

1.4 ขอบเขตของการวิจัย

1. ออกแบบและพัฒนาวิธีการหาเส้นทางสำรองของโพรโทคอล และการสลับเส้นทางระหว่างเส้นทางหลักและเส้นทางสำรอง
2. ทดสอบและวิเคราะห์ผลการทำงานของโพรโทคอลในโปรแกรมจำลองเครือข่าย Network Simulator 2 (NS-2)
3. วิเคราะห์และเปรียบเทียบประสิทธิภาพของโพรโทคอลที่นำเสนอ

1.5 ขั้นตอนและวิธีการวิจัย

1. ศึกษาโพรโทคอลค้นหาเส้นทางในเครือข่ายตรวจจับไร้สาย
2. ศึกษาโปรแกรมจำลองเครือข่าย NS-2 โพรโทคอล Ad-hoc On-Demand Distance Vector (AODV) และโพรโทคอล DSR
3. พัฒนาโพรโทคอลในโปรแกรมจำลองเครือข่าย NS-2 บนพื้นฐานของโพรโทคอล DSR
4. ทดสอบ และแก้ไขเพื่อให้สามารถทำงานได้ถูกต้อง
5. ทำการทดสอบเพื่อวิเคราะห์ และเก็บผล
6. ทำการเปรียบเทียบประสิทธิภาพกับโพรโทคอล AODV, Redundant Path Routing Protocol (RPRP) Single Path และ DSR
7. จัดทำวิทยานิพนธ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้โพรโทคอลที่มีความน่าเชื่อถือในการสื่อสารข้อมูลในเครือข่ายตรวจจับไร้สาย
2. ได้โพรโทคอลที่สามารถนำไปประยุกต์ใช้ในการตรวจจับเหตุการณ์ที่ต้องการความน่าเชื่อถือในการสื่อสารข้อมูลสูง
3. ได้โพรโทคอลที่มีค่า Throughput สูงขึ้นสำหรับเครือข่ายขนาด 100 โหนด
4. ได้โพรโทคอลที่สามารถเลือกใช้เส้นทางที่มีการใช้พลังงานต่ำไปยังปลายทางได้จากการมีเส้นทางสำรอง

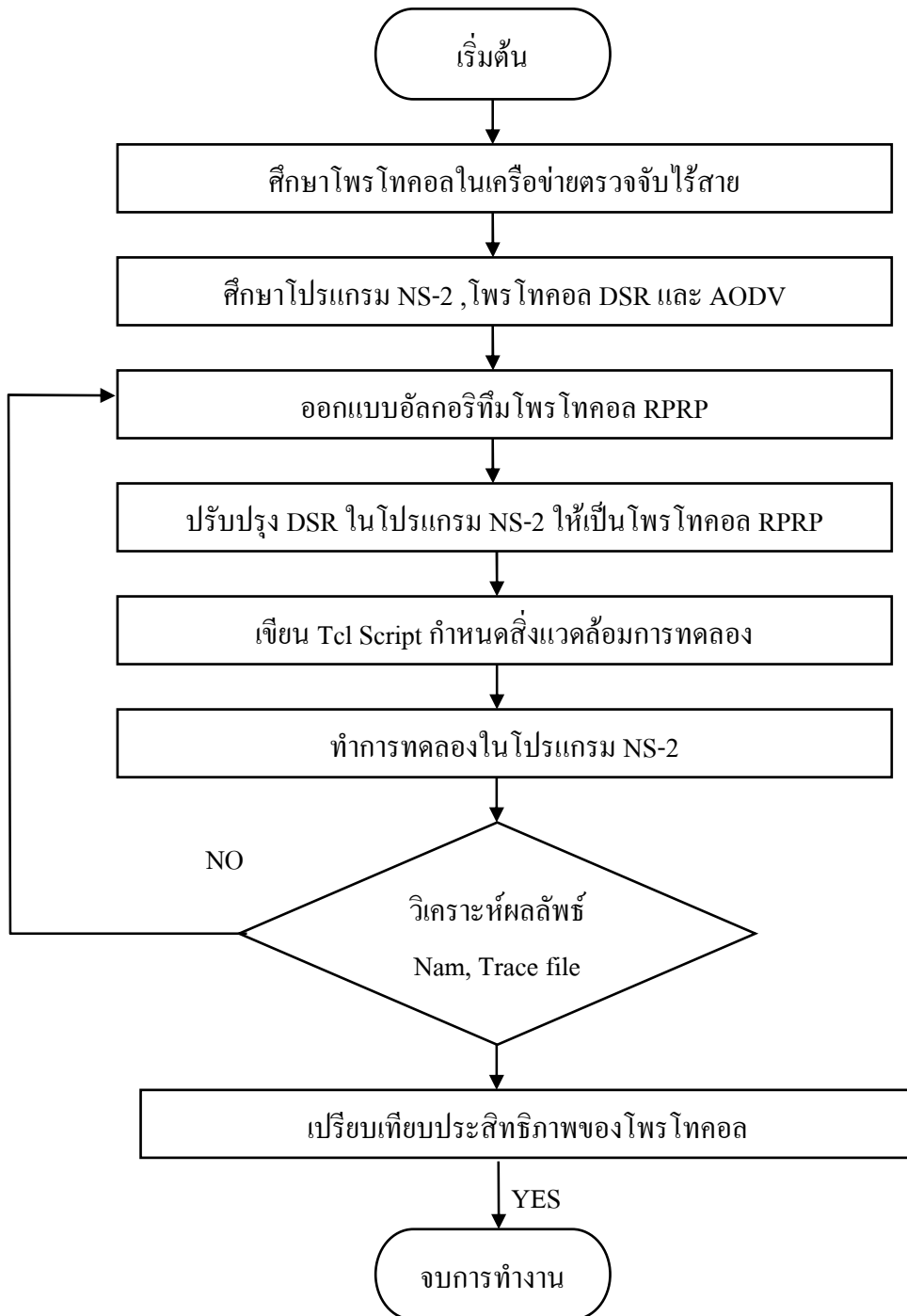
1.7 ทรัพยากรที่ใช้ในระบบ

ในการวิจัยนี้ใช้โปรแกรมจำลองเครือข่าย NS-2 มาใช้ในการทดสอบการทำงานของโพรโทคอล ซึ่งมีทรัพยากรที่ใช้ในการออกแบบและทดสอบดังนี้

1. คอมพิวเตอร์แบบพกพา มีหน่วยประมวลกลาง Intel Core 2 Duo T5200 อัตราเร็ว 1.6 GHz และหน่วยความจำ RAM ขนาด 1 GB
2. โปรแกรม NS-2 รุ่น 2.33 เป็นโปรแกรม Open-Source ที่ใช้ในการจำลองการทำงานของเครือข่ายในแบบ Discrete Event Simulator ซึ่งรองรับการจำลองเครือข่ายตรวจจับไร้สาย
3. โปรแกรม VMware Workstation รุ่น 6.0 เป็นโปรแกรมที่ใช้จำลองเครื่องคอมพิวเตอร์เป็นโปรแกรมทางการค้า
4. โปรแกรม Perl รุ่น 5.0 เป็นโปรแกรมที่ใช้ในการกรองไฟล์ข้อมูล

1.8 ภาพรวมของขั้นตอนการทำงานและพัฒนาโพรโทคอล

โดยเริ่มจากการศึกษาโพรโทคอลในเครือข่ายตรวจจับไร้สาย เพื่อหาปัญหาของโพรโทคอล จากนั้นก็ศึกษาเครื่องมือที่ใช้ในการทดลองคือโปรแกรม NS-2 ศึกษาโพรโทคอล DSR เพื่อให้เป็นโพรโทคอลพื้นฐานในการพัฒนาต่อ และศึกษาโพรโทคอล AODV เพื่อที่จะสามารถนำมาเปรียบเทียบประสิทธิภาพและวิเคราะห์ผลการทดลอง ซึ่งเมื่อได้ประเด็นปัญหาเกี่ยวกับการขาดนำเชื่อถือในการสื่อสารข้อมูลในเครือข่ายตรวจจับไร้สาย จึงได้ออกแบบอัลกอริทึมในการค้นหาเส้นทางสำรองของโพรโทคอล RPRP และหาค่าสมการประมาณค่าพลังงานที่ใช้ในการค้นหาเส้นทางของโพรโทคอล ถัดจากนั้นทำการพัฒนาโพรโทคอล DSR ให้เป็นโพรโทคอล RPRP ด้วยการเขียนโค้ดภาษาซีเพิ่มเติมปรับปรุงให้ได้ตามที่ออกแบบไว้ แล้วเขียน TCL Script เพื่อกำหนดสภาพแวดล้อมในการจำลองต่างๆ จากนั้นทำการทดลอง ซึ่งผลที่ได้จากการทดลองคือ NS Trace File และ NAM Trace File ซึ่งจะนำไปวิเคราะห์ผลการทดลองโดยการเขียน Perl Script และ Shell Script เพื่อกรองข้อมูลที่ต้องการ และนำมาเปรียบเทียบกับโพรโทคอล RPRP (Single Path), AODV และ DSR



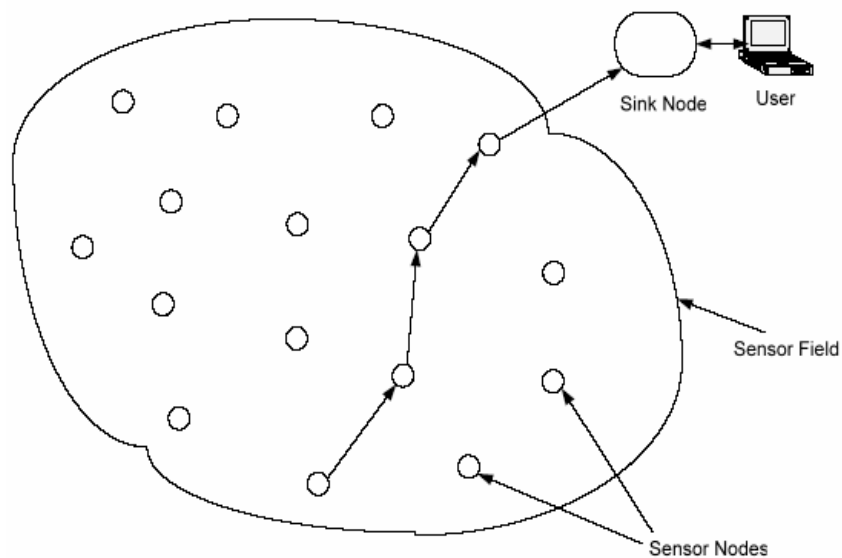
ภาพประกอบ 1-1 ขั้นตอนการทำงานและพัฒนาโพรโทคอล

บทที่ 2

ทฤษฎีและหลักการ

2.1 ความรู้เบื้องต้นเกี่ยวกับเครือข่ายตรวจจับไร้สาย

เครือข่ายตรวจจับไร้สาย เป็นเทคโนโลยีที่กระจาย ตัวตรวจจับจำนวนมากซึ่งสามารถใช้ในการตรวจสอบปรากฏการณ์ทางกายภาพ และสามารถที่จะกระจายได้ง่าย นอกจากนี้ยังสามารถประยุกต์ใช้ในงานอื่นได้หลากหลาย



ภาพประกอบ 2-1 การทำงานของเครือข่ายตรวจจับไร้สาย

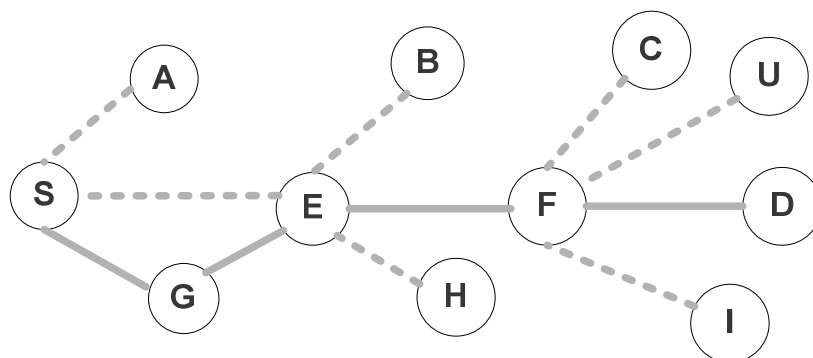
ตามภาพประกอบ 2-1 เครือข่ายตรวจจับไร้สายจะประกอบด้วย [16] โหนดตรวจจับ (Sensor Node) ซึ่งทำหน้าที่ตรวจจับและส่งข้อมูลต่อเป็นช่วงๆ ไปยังโหนดรวบรวมข้อมูล (Sink Node) เพื่อนำไปประมวลผลต่อไป ซึ่งโหนดตรวจจับจะอยู่ในพื้นที่ตรวจจับ (Sensor Field)

ซึ่งตามแนวความคิดของเครือข่ายตรวจจับไร้สาย จะมีโหนดในระบบเป็นจำนวนมาก ทำให้การเปลี่ยนแบตเตอรี่ทำได้ยาก ดังนั้นการประหยัดพลังงานจึงจำเป็นสิ่งที่สำคัญสำหรับระบบนี้

2.1.1 ตัวอย่างโพรโทคอลประหยัดพลังงานในเครือข่ายตรวจจับไร้สาย

โพรโทคอลรูปแบบนี้มีจุดมุ่งหมายที่จะประหยัดพลังงานที่มีในโหนดตรวจจับให้ได้มากที่สุด เพื่อเพิ่มเวลาในการใช้เครือข่ายตรวจจับให้นานที่สุด จึงออกแบบให้มีการหาเส้นทางเดียวไปยังปลายทาง ซึ่งต่างจากโพรโทคอลปกติที่จะกระจายการหาเส้นทาง ซึ่งสามารถมีหลายเส้นทางไปยังปลายทาง ตัวอย่างโพรโทคอลรูปแบบนี้ เช่น Energy Aware Routing for Wireless Sensor Networks [10]

ซึ่งโพรโทคอลนี้ประกอบด้วยอัลกอริทึมสามส่วนคือ Route Discovery, Route Reply และ Reliable Transmission โดยใช้สองข้อความคือ ข้อความร้องขอเส้นทาง และ ข้อความตอบรับเส้นทาง Route Discovery เป็นกระบวนการค้นหาเส้นทาง โดยโหนดต้นทางจะกระจายข้อความร้องขอเส้นทางโดยโหนดระหว่างกลางจะต้องมีพลังงาน และคุณภาพการเชื่อมต่อสูงกว่าค่าที่กำหนดไว้ จึงจะมีสิทธิ์ได้รับเชื่อมต่อ หากน้อยกว่าก็ต้องทิ้งข้อความร้องขอเส้นทางไป ซึ่งโหนดที่มีพลังงานสูงสุด จะได้รับการคัดเลือกให้กระจายข้อความร้องขอเส้นทางต่อไป หากมีระดับพลังงานเท่ากันก็จะใช้กระบวนการนับถอยหลัง ซึ่งจะกระทำตามกระบวนการนี้จนกว่าไปถึงปลายทาง ส่วน Route Reply เป็นการตอบกลับเส้นทางด้วยข้อความตอบรับเส้นทาง เมื่อค้นหาเจอโหนดปลายทาง และ Reliable Transmission เป็นกลไกในการเพิ่มความน่าเชื่อถือของข้อมูลมากขึ้น โดยข้อมูลจะอยู่ใน Cache ของโหนดที่ส่ง จนกระทั่งได้ข้อความตอบรับจากโหนดที่รับข้อมูล หากไม่ได้ข้อความตอบรับข้อมูลภายในเวลาที่กำหนด ก็จะมีข้อความรายงานความผิดพลาดกลับไปยังโหนดต้นทางของข้อมูลเพื่อให้โหนดต้นทางส่งข้อมูลใหม่อีกครั้ง

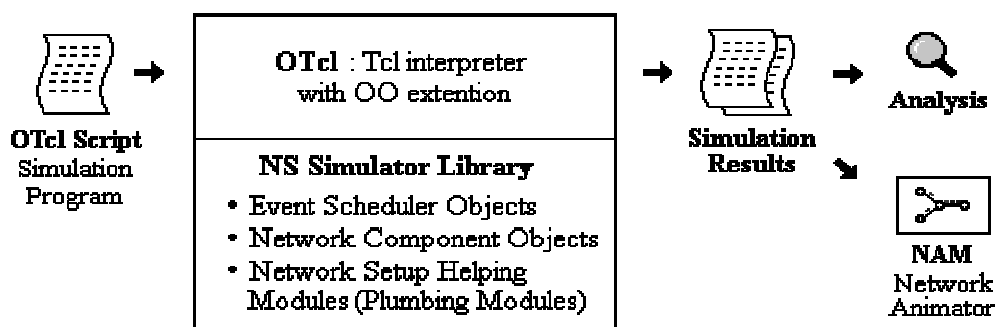


ภาพประกอบ 2-2 แสดงการหาเส้นทางใน Energy Aware Routing

ตามภาพประกอบ 2-2 เริ่มต้นโหนดต้นทางคือโหนด S กระจายข้อความร้องขอเส้นทางออกไป โหนด A, E และ G ได้รับข้อความ สมมติให้โหนด G มีพลังงานมากกว่าโหนด E และ A โหนด G ก็จะได้รับบริการเชื่อมต่อให้ส่งข้อความร้องขอเส้นทางต่อไป จากนั้นโหนด G ก็กระจายข้อความร้องขอเส้นทางออกไป โหนด E ได้รับข้อความ โหนด E มีระดับพลังงานสูงกว่าพลังงานขั้นต่ำก็ได้รับการคัดเลือก โหนด E ส่งข้อความร้องขอเส้นทางออกไป โหนดที่ได้รับข้อความคือ B, F และ H โหนด B และ โหนด F มีระดับพลังงาน และคุณภาพการเชื่อมต่อเท่ากัน ก็จะใช้กระบวนการนับถอยหลัง โดยการสุ่มตัวเลขเพื่อนับถอยหลัง ซึ่งโหนด F ใต้นับถอยหลังเสร็จก่อน ก็จะเป็นโหนดที่ได้รับคัดเลือก จากนั้นโหนด F ก็กระจายข้อความร้องขอเส้นทางออกไป ซึ่งโหนดปลายทางคือโหนด D ได้รับข้อความร้องขอเส้นทางก็จะตอบกลับเส้นทาง โดยใช้ข้อความตอบกลับเส้นทาง ไปตามเส้นทางเดิม

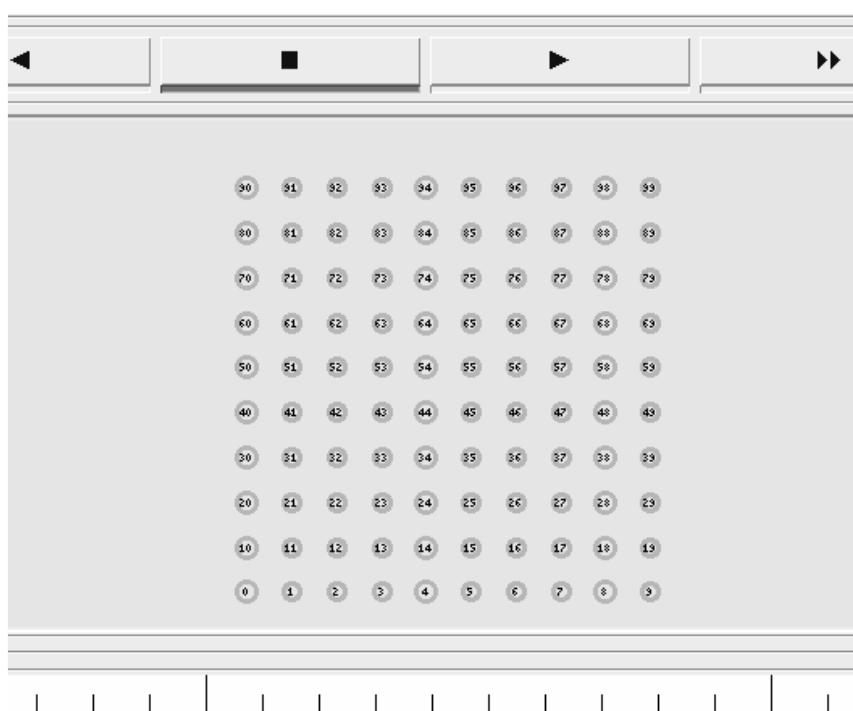
2.2 โปรแกรมจำลองเครือข่าย

NS-2 [17] เป็นโปรแกรม Open-Source ที่สามารถทำงานได้ทั้งบน Linux, FreeBSD, SunOS, Solaris และ Window ถูกพัฒนาขึ้นโดย Information Sciences Institute (ISI) NS-2 เป็นโปรแกรมที่ใช้ในการจำลองการทำงานของเครือข่ายในแบบที่เป็น Discrete Event Simulator ซึ่งสนับสนุนการจำลองเครือข่ายตรวจจับไร้สาย การจำลองการเลือกเส้นทางในการส่งแพ็กเก็ต, จำลองการทำงานของโพรโทคอลแบบ Multicast และ โพรโทคอล IP เช่น User Datagram Protocol (UDP), Transmission Control Protocol (TCP) และ Real-time Transport Protocol (RTP) ที่อยู่บนเครือข่ายประเภทที่มีสาย และเครือข่ายไม่มีสาย ซึ่ง NS-2 เป็นซอฟต์แวร์ที่มีประโยชน์มาก และยังมีความสามารถในการแสดงรายละเอียดของ Network Traffic ออกมาในรูปแบบของกราฟฟิก รวมทั้งยังสนับสนุนอัลกอริทึมในการค้นหาเส้นทาง และคิว เช่น FIFO และ Round-Robin เป็นต้น



ภาพประกอบ 2-3 โครงสร้างพื้นฐานโดยรวมของ NS-2

ตามภาพประกอบ 2-3 ผู้ใช้จะต้องเขียน OTcl Script ซึ่งเป็นการกำหนดค่าเริ่มต้นของสภาพแวดล้อมในการจำลอง และติดตั้งโครงร่างของเครือข่ายให้ Network Object ใช้งาน เมื่อโปรแกรมทำงาน OTcl Interpreter จะรับ OTcl Script และจะทำการตั้งค่าพารามิเตอร์สภาพแวดล้อมตาม Script ที่รับมา เมื่อทำการจำลองจนจบแล้ว ผลลัพธ์จากการจำลองจะมีหนึ่งไฟล์ หรือมากกว่า ซึ่งเป็นแบบ Text-Base และจะประกอบด้วยรายละเอียดของข้อมูลการจำลองคือ NS Trace File ซึ่งจะต้องเขียน Script เพื่อกรองข้อมูลให้ได้ตามที่ต้องการ หรือเป็นไฟล์กราฟฟิกที่เรียกว่า Network Animator (NAM) ซึ่งแสดงในภาพประกอบ 2-4

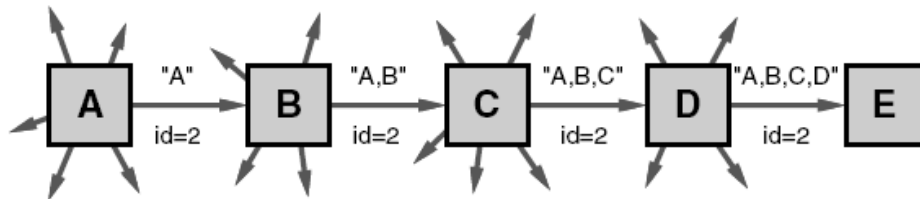


ภาพประกอบ 2-4 ตัวอย่างของ NAM ใน NS-2

2.3 โพรโทคอล DSR

เป็นโพรโทคอลพื้นฐานที่ง่าย และมีประสิทธิภาพสูง โดยออกแบบให้ใช้งานเฉพาะการสื่อสารแบบหลายๆ ช่วง ในเครือข่ายไร้สาย มีกลไกการทำงานที่ง่ายโดยประกอบด้วยสองกลไกหลักคือ กลไกการค้นหาเส้นทาง และกลไกการบำรุงรักษาเส้นทาง มีกล่าวใน The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks [18]

2.3.1 การค้นหาเส้นทางของโปรโตคอล DSR



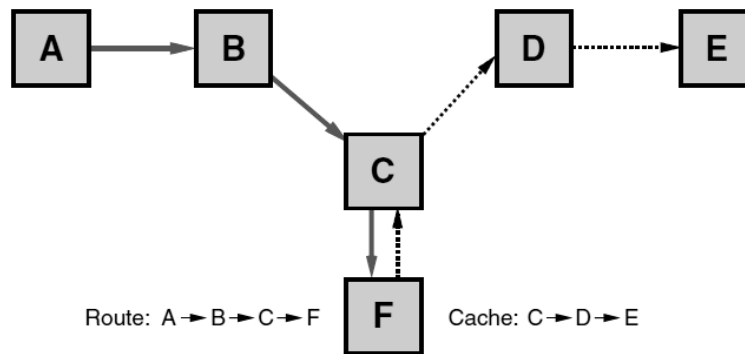
ภาพประกอบ 2-5 การค้นหาเส้นทางใน DSR

ในภาพประกอบ 2-5 เป็นการค้นหาเส้นทาง ซึ่งเมื่อโหนด A ต้องการค้นหาเส้นทางไปยังโหนด E โดยในขณะที่เริ่มต้นของการค้นหาเส้นทาง โหนด A จะส่งข้อความร้องขอเส้นทางในลักษณะการกระจายแพ็กเก็ตออกไป ซึ่งโหนดที่อยู่ภายในรัศมีการส่งของโหนด A จะได้รับแพ็กเก็ต แต่ละข้อความร้องขอเส้นทางจะระบุ โหนดต้นทาง, โหนดปลายทางของการค้นหา และหมายเลขของการร้องขอเส้นทางซึ่งจะไม่ซ้ำกัน แต่ละข้อความร้องขอเส้นทางจะประกอบด้วยรายชื่อโหนดระหว่างทางที่ผ่านมา ซึ่งในทางปฏิบัติโหนดจะคัดลอกข้อความร้องขอเส้นทางแล้วส่งต่อ เมื่อไปถึงโหนดปลายทางก็จะส่งข้อความตอบรับเส้นทางไปตามเส้นทางเดิม โดยเส้นทางจะถูกเก็บไว้ใน Route Cache ในกรณีที่โหนดที่ได้รับข้อความร้องขอเส้นทาง แล้วพบว่าไม่มีหมายเลขโหนดของตนเองอยู่ก็จะทำการทิ้งข้อความนี้ แต่หากไม่พบว่ามีหมายเลขโหนดของตนเองอยู่ก็จะทำการเพิ่มหมายเลขตัวเองลงในรายชื่อเส้นทางและกระจายข้อความการร้องขอนี้ออกไป

การส่งกลับของข้อความตอบกลับเส้นทางกลับไปยังโหนดต้นทางของการค้นหาเส้นทาง เช่น โหนด E ส่งข้อความตอบรับเส้นทางกลับไปยังโหนด A โหนด E ก็จะตรวจสอบ Route Cache ของตัวเองเพื่อหาเส้นทางกลับไปยังโหนด A และถ้าพบก็จะใช้เป็นเส้นทางส่งแพ็กเก็ตข้อความตอบรับเส้นทางกลับ

2.3.1.1 การตอบกลับข้อความร้องขอเส้นทางโดยใช้เส้นทางใน Cache

โหนดที่รับข้อความร้องขอเส้นทาง ถ้าตัวโหนดไม่ใช่โหนดปลายทาง ก็จะค้นหาเส้นทางใน Cache ของตัวเอง เพื่อหาเส้นทางไปยังปลายทาง หากพบเส้นทาง โหนดก็จะส่งข้อความตอบรับเส้นทางกลับไปยังโหนดต้นทางได้เช่นกัน

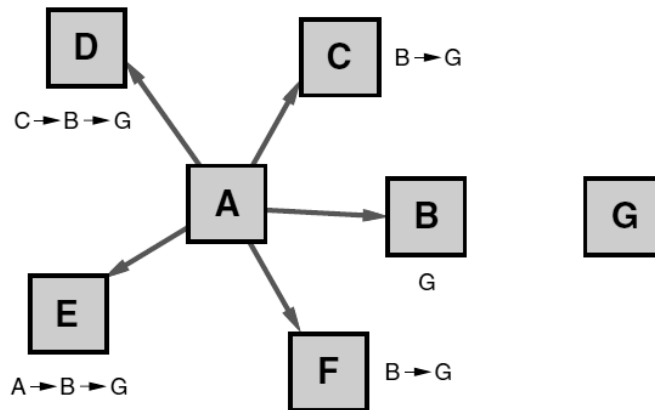


ภาพประกอบ 2-6 การตอบกลับข้อความร้องขอเส้นทางโดยใช้เส้นทางใน Cache

ตามภาพประกอบ 2-6 โหนด F ได้รับข้อความร้องขอเส้นทางซึ่งเป้าหมายของการค้นหาเส้นทางคือ โหนด E ซึ่งโหนด F มีเส้นทางไปยังโหนด E ใน Cache ของตัวเอง โหนด E ก็จะทำการต่อเส้นทางจากข้อความร้องขอเส้นทางกับเส้นทางใน Cache ของตัวเอง และจะใส่รายชื่อโหนดจากโหนด C ไปยังโหนด E แล้วส่งกลับไปยังโหนด C ซึ่งในการตอบกลับเส้นทาง โหนด F ก็จะไม่อยู่ในเส้นทางกลับ

2.3.1.2 การป้องกันข้อความตอบกลับเส้นทางชนกัน

ความสามารถของโหนดในการตอบกลับข้อความร้องขอเส้นทาง ขึ้นกับข้อมูลเส้นทางใน Cache การเกิดกรณีข้อความตอบกลับเส้นทางชนกันเกิดขึ้นในบางกรณี ซึ่งในทางปฏิบัติถ้าโหนด กระจายข้อความร้องขอเส้นทางเพื่อหาโหนดปลายทาง โหนดข้างเคียงโหนดปลายทางที่มีเส้นทางไปยังโหนดปลายทางอยู่ใน Cache ของตัวเอง แต่ละโหนดที่เป็นโหนดข้างเคียงก็อาจจะส่งข้อความตอบกลับเส้นทาง ซึ่งก็เป็นไปได้ที่จะเกิดการชนกันเกิดขึ้นของข้อความในพื้นที่นั้น



ภาพประกอบ 2-7 ข้อความตอบกลับเส้นทางท่วมทับ

ตามภาพประกอบ 2-7 โหนด B, C, D, E และ F ได้รับข้อความร้องขอเส้นทางจาก โหนด A ซึ่งโหนดปลายทางคือ โหนด G แต่ละโหนดที่ได้รับข้อความต่างก็มีเส้นทางไปยังโหนดปลายทางใน Cache ของตัวเอง ปกติโหนดทั้งหมดก็จะส่งข้อความตอบรับเส้นทางกลับไปยังโหนด A พร้อมกัน เนื่องจากได้รับข้อความร้องเส้นทางจากโหนด A พร้อมกัน ทำให้เกิดการชนกันของข้อความตอบรับเส้นทาง สถานการณ์เช่นนี้อาจเกิดขึ้นบ่อยในกรณีที่มีความแตกต่างของความยาวของเส้นทางของข้อความตอบกลับเส้นทาง ดังแสดงในภาพประกอบ 2-7

$$d = H(h-1+r) \quad (2-1)$$

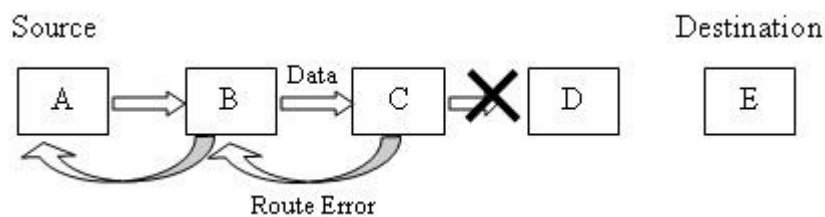
- h คือ จำนวนช่วงของเส้นทางในข้อความตอบกลับเส้นทาง
- r คือ เลขที่สุ่ม มีค่า 0, 1
- H คือ ค่าคงที่ในการหน่วง (อย่างน้อยสองเท่าของ Wireless Link Propagation Delay)

ถ้าโหนดได้รับแพ็คเกจที่มากในระยะเวลาสั้นๆ โหนดจะหน่วงเวลาการส่งข้อความตอบรับเส้นทางของตัวเอง โดยการสุ่มระยะเวลาการส่งใหม่ตามสมการ (2-1) [18]

2.3.1.3 จำนวนช่วงที่จำกัดของข้อความร้องขอเส้นทาง

แต่ละข้อความร้องขอเส้นทางประกอบด้วย จำนวนช่วงที่จำกัด ซึ่งจะใช้ในการจำกัดจำนวนโหนดระหว่างกลางในการส่งต่อ ซึ่งแพ็กเก็ตข้อความร้องขอเส้นทางจะถูกทิ้งหากจำนวนช่วงเหลือศูนย์ก่อนไปถึงปลายทาง ปัจจุบันใช้กลไกนี้ในการส่งข้อความร้องขอเส้นทางที่ไม่ต้องการกระจายเป็นกลไกที่ไม่ต้องลงทุนมากนัก คือถ้าเป้าหมายเป็น โหนดข้างเคียง หรือถ้าโหนดข้างเคียงมีเส้นทางอยู่ใน Cache ถ้าไม่ได้รับข้อความตอบกลับเส้นทางในเวลาที่กำหนดก็จะใช้กลไกนี้ ส่งข้อความร้องเส้นทางแบบจำกัดจำนวนโหนดระหว่างกลางอีกครั้ง

2.3.2 การบำรุงรักษาเส้นทางของโปรโตคอล DSR



ภาพประกอบ 2-8 ตัวอย่างการบำรุงรักษาเส้นทาง

เมื่อโหนดตั้งต้น หรือโหนดที่กำลังส่งต่อแพ็กเก็ตในเส้นทาง ในขณะที่ส่งแพ็กเก็ต ข้อมูลก็ต้องรอข้อความยืนยันการได้รับข้อมูลก่อน แล้วจึงส่งข้อมูลต่อไปได้ หากไม่ได้รับแพ็กเก็ตยืนยันในเวลาที่กำหนด โหนดก็จะส่งข้อความเส้นทางล้มเหลวกลับไปยังโหนดต้นทางเพื่อหาเส้นทางใหม่ใน Cache หรือเริ่มกระบวนการค้นหาเส้นทางใหม่อีกครั้ง

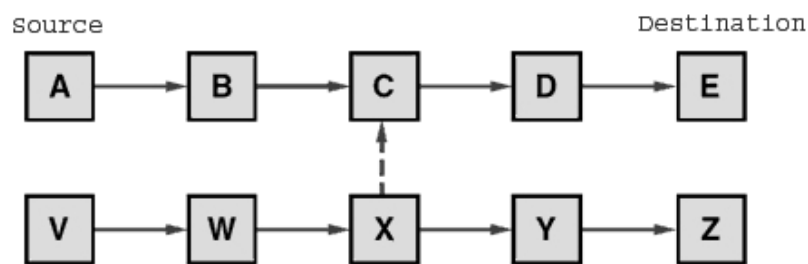
ในภาพประกอบ 2-8 โหนด A เป็นโหนดต้นทางในการส่งแพ็กเก็ตไปยังโหนด E โดยใช้เส้นทางผ่านโหนดระหว่างกลางคือ B, C, D ในกรณีนี้โหนด A จะได้รับแพ็กเก็ตยืนยันการได้รับข้อมูลที่โหนด B, โหนด B จะได้รับแพ็กเก็ตยืนยันการได้รับข้อมูลที่โหนด C, โหนด C จะได้รับแพ็กเก็ตยืนยันการได้รับข้อมูลที่โหนด D, โหนด D ได้รับแพ็กเก็ตยืนยันการได้รับข้อมูลที่โหนด E

หากในบางช่วงมีการส่งแพ็กเก็ตเข้าไปยังโหนด D มากที่สุดในหนึ่งหน่วยเวลา และไม่ได้รับแพ็กเก็ตตอบรับ โหนด C ก็จะส่งข้อความเส้นทางล้มเหลวไปยังผู้ที่ส่งแพ็กเก็ตเริ่มต้นคือโหนด A โดยโหนด A ก็จะลบเส้นทางที่ขาดนี้ออกจาก Cache ของโหนด ถ้าโหนด A มีเส้นทางอื่น

ใน Cache ไปยังโหนด E ก็สามารถใช้เส้นทางใหม่ได้ทันที หากไม่มีก็ทำการค้นหาเส้นทางใหม่ได้เช่นกัน

2.3.2.1 การรับฟังข้อมูลเส้นทางเพื่อปรับปรุงข้อมูลเส้นทางใน Cache

โหนดที่กำลังส่งแพ็กเก็ตต่อ หรือโหนดที่กำลังรอรับแพ็กเก็ตต่างๆ สามารถเพิ่มข้อมูลเส้นทางจากแพ็กเก็ตที่ได้รับข้อมูลมาเข้าไปใน Cache ของตัวเองได้



ภาพประกอบ 2-9 การรับฟังข้อมูลเส้นทางเพื่อเก็บไว้ใน Cache

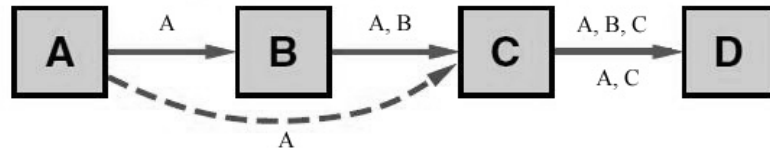
ภาพประกอบ 2-9 โหนด A กำลังใช้เส้นทางติดต่อกับโหนด E ขณะที่โหนด C กำลังส่งแพ็กเก็ตข้อมูลนั้น โหนด C สามารถเพิ่มเส้นทางได้จากการเรียนรู้แพ็กเก็ตที่ส่งต่อ เช่น การรับแพ็กเก็ตข้อมูลการส่งต่อข้อมูลจากโหนด X ในขณะที่โหนด X ส่งแพ็กเก็ตไปยังโหนดปลายทาง โหนด C ก็จะเรียนรู้เส้นทางเพิ่มเติมคือ โหนด V ติดต่อกับโหนด W โหนด W ติดต่อกับโหนด X ก็สามารถเพิ่มเข้าไปใน Cache ของตัวเองได้ และสามารถเรียนรู้ได้อีกเมื่อโหนด X ส่งแพ็กเก็ตกลับมาจากโหนด Z ก็รู้ว่า โหนด X ติดต่อกับโหนด Y และโหนด Y ก็ติดต่อกับโหนด Z

ฉะนั้นสามารถสรุปได้ว่า โหนดสามารถเรียนรู้จากการส่งข้อมูลต่อ และจากการฟังข้อความร้องขอเส้นทาง และข้อความตอบรับเส้นทางได้

2.3.2.2 การทำให้เส้นทางสั้นโดยอัตโนมัติ

การทำให้เส้นทางสั้นโดยอัตโนมัตินั้น จะเป็นการนำโหนดระหว่างกลางที่ไม่จำเป็นออกไป โหนดระหว่างกลางจะฟังแพ็กเก็ตข้อมูลซึ่งเส้นทางอยู่ในหัวของแพ็กเก็ต โดยโหนดจะ

ตรวจสอบส่วนของเส้นทางที่ไม่จำเป็น หากพบโหนดดังกล่าวก็จะส่งข้อความ Route Reply กลับไปยังโหนดต้นทางพร้อมกับเส้นทางใหม่ที่ปรับปรุงแล้ว โดยจะกำจัดโหนดที่ไม่จำเป็นออกไป



ภาพประกอบ 2-10 ตัวอย่างการปรับปรุงทางให้สั้นขึ้น

ตามภาพประกอบ 2-10 โหนด C ฟังแฟ็กข้อมูลที่ส่งจาก A ไป B แล้วส่งต่อมายัง C ซึ่ง C รู้ว่า B ไม่จำเป็นในเส้นทาง เพราะไม่ต้องผ่าน B ก็ไปถึงปลายทางได้ โหนด C จึงได้ส่งข้อความ Route Reply กลับไปยังโหนดต้นทางคือ A ซึ่งในข้อความก็จะมีเส้นทางใหม่ที่โหนด C ได้ต่อเส้นทางให้แล้วโดยการตัดโหนด B ทิ้งไปก็จะเป็น A-C-D

2.3.2.3 การเพิ่มการกระจายข้อความ Route Error

เมื่อโหนดต้นทางรับข้อความ Route Error หากโหนดต้นทางไม่มีเส้นทางอื่นที่ไปยังปลายทาง ก็จะทำการค้นหาเส้นทางใหม่ ซึ่งในการค้นหาเส้นทางใหม่นี้จะนำเอาเส้นทางที่ขาด กระจายไปกับข้อความร้องขอเส้นทางเพื่อให้โหนดอื่นๆ ได้ทราบว่าเส้นทางนี้ขาดแล้ว และยิ่งมากไปกว่านั้นข้อความตอบกลับเส้นทางที่มายังโหนดต้นทาง ถ้ามีเส้นทางที่ขาดนี้มาด้วยก็จะไม่รับเช่นกัน

สำหรับตัวอย่างในภาพประกอบ 2-8 โหนด A เรียนรู้จากข้อความ Route Error ที่มาจากโหนด C ก็จะทราบว่าเส้นทางจากโหนด C ไป D นั้นขาดแล้ว โหนด A ก็เอาเส้นทางดังกล่าวออกจาก Route Cache ของตัวเอง แล้วจะกระทำการค้นหาเส้นทางใหม่ และจะนำเส้นทางที่ขาดนี้กระจายไปกับการค้นหาเส้นทางใหม่ด้วย และจะไม่รับข้อความตอบรับเส้นทางที่มีโหนด D ติดมาด้วย

2.3.2.4 การกู้แพ็คเก็ตข้อมูลในขณะเส้นทางขาด

เมื่อเส้นทางขาดในขณะที่โหนดจะส่งข้อความ Route Error ก็จะหาเส้นทางอื่นๆ ใน Cache ของตัวเองที่จะสามารถไปยังปลายทางได้ ก่อนที่จะทิ้งข้อมูลนั้น หากมีเส้นทางไปยังโหนดปลายทางได้ก็จะกู้ข้อมูลดังกล่าว โดยการส่งข้อมูลไปยังปลายทางพร้อมทั้งส่งเส้นทางเดิมที่ขาดกลับไปยังโหนดต้นทาง เพื่อให้ทราบว่าเส้นทางนี้ขาดที่ใด ดังตัวอย่างในภาพประกอบ 2-8 หากโหนด C มีเส้นทางอื่นไปยังโหนด E โหนด C ก็สามารถกู้ข้อมูลดังกล่าวโดยการส่งไปยังเส้นทางใหม่ได้

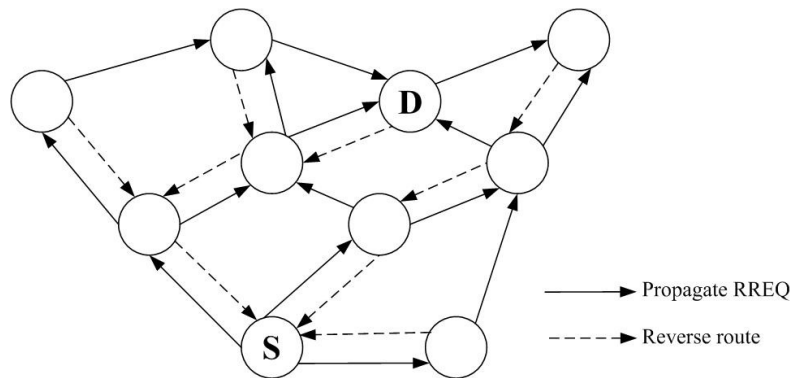
2.4 โพรโทคอล AODV

เป็นโพรโทคอลแบบ Source Initiated On-Demand Driven/Reactive คือจะทำการหาเส้นทางก็ต่อเมื่อจุดเริ่มต้นต้องการหาเส้นทาง ซึ่งจะเป็นการหาเส้นทางตามเส้นทางที่เป็นไปได้จนถึงปลายทาง โพรโทคอล AODV [19] ได้ปรับปรุงมาจากโพรโทคอล DSDV เนื่องจากมีความคล้ายคลึงกันมาก แต่ AODV จะลดจำนวนครั้งในการที่จะค้นหาเส้นทางลง โดยจะทำการค้นหาเส้นทางเมื่อมีโหนดที่ต้องการที่จะส่งข้อมูล หรือมีการเคลื่อนที่ของโหนดเท่านั้น ทำให้ไม่สิ้นเปลือง Bandwidth ในช่องสัญญาณ แต่จะต้องเสียเวลาในการที่หาเส้นทางใหม่เมื่อต้องการที่จะส่งข้อมูล

2.4.1 การค้นหาเส้นทาง

จากภาพประกอบ 2-11 โหนดต้นทางคือ โหนด S และโหนดปลายทางคือ โหนด D เมื่อโหนดต้นทางต้องการจะส่งข้อมูล โหนดจะตรวจในตารางเส้นทาง หากเจอเส้นทางก็จะทำการส่งข้อมูลนั้นเลย แต่หากไม่เจอเส้นทางไปยังโหนดปลายทางก็จะเริ่มกระบวนการค้นหาเส้นทาง โดยจะทำการสร้างแพ็คเก็ตร้องขอเส้นทางโดยจะประกอบด้วย Source IP, Current Sequence Number, Destination IP, Last Known Sequence Number และ Broadcast ID หลังจากนั้นโหนดต้นทางจะ กระจายข้อความร้องขอเส้นทางออกไปโดยจะตั้งเวลาไว้สำหรับการตอบกลับเส้นทาง

เมื่อโหนดระหว่างกลางได้รับข้อความร้องขอเส้นทาง โหนดก็จะตรวจสอบว่าเคยได้รับข้อความร้องขอเส้นทางที่มี Source IP, Broadcast ID คู่นี้หรือไม่ ถ้าไม่ ก็จะเก็บข้อมูลไว้ตามระยะเวลาที่กำหนด แต่ถ้าเคยได้รับแล้วก็จะไม่สนใจ



ภาพประกอบ 2-11 การค้นหาเส้นทางของโพรโทคอล AODV

ต่อมาโหนดจะทำการสร้าง Reverse Route ซึ่งประกอบด้วย Source IP, Sequence Number, Hop Count, IP Neighbor จากโหนดที่ได้รับข้อความร้องขอเส้นทาง โดยวิธีนี้จะทำให้โหนดรู้ว่าจะส่งต่อข้อความตอบกลับเส้นทางไปยังโหนดต้นทางได้ เมื่อข้อความร้องขอเส้นทางมาถึงโหนดปลายทาง จากนั้นโหนดปลายทางจะส่งข้อความตอบกลับเส้นทางผ่านทาง Reverse Route

หากส่งข้อความร้องขอเส้นทางแล้ว ข้อความตอบรับเส้นทางที่ตอบกลับมาไม่เจอโหนดปลายทาง หรือข้อความตอบกลับเส้นทางมาไม่ถึงในเวลาที่กำหนด โหนดต้นทางจะทำการกระจายข้อความร้องขอเส้นทางใหม่ ซึ่งจะเพิ่มค่าให้แก่ Route Retrie ซึ่งเก็บจำนวนครั้งในการส่งซ้ำของข้อความร้องขอเส้นทาง

2.4.1.1 การค้นหาเส้นทางที่เพิ่มส่วนของ Ring Search

ทุกครั้งที่โหนดต้นทางหาเส้นทาง โหนดจะทำการกระจายข้อความร้องขอเส้นทางไปทั้งระบบเครือข่าย สำหรับเครือข่ายขนาดเล็กการชนกันของแพ็กเก็ตจะมีน้อย แต่หากเป็นเครือข่ายขนาดใหญ่แล้วการชนกันก็จะเพิ่มมากขึ้น เพื่อที่จะควบคุมในส่วนนี้จึงต้องเพิ่มคุณสมบัติของ Ring Search

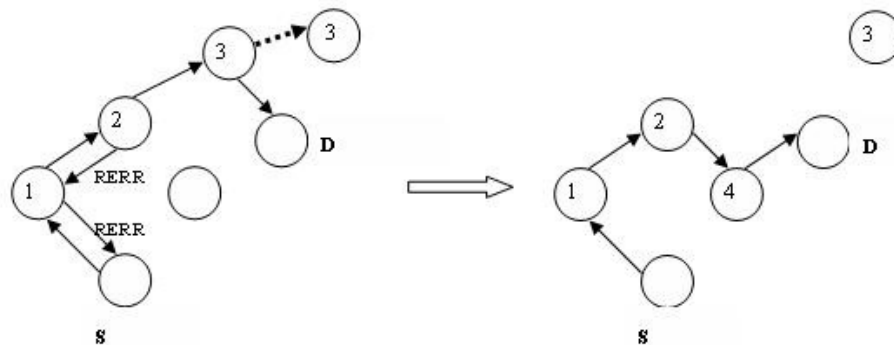
หลักการของ Ring Search คือโหนดต้นทางจะตั้งค่า Time To Live (TTL) ไว้ในข้อความร้องขอเส้นทางเรียกว่า TTL Start หากไม่มีการตอบกลับเส้นทางมาในเวลาที่กำหนด โหนดต้นทางก็จะเพิ่มค่า TTL แล้วทำการกระจายข้อความร้องขอเส้นทางใหม่ จนกว่าจะเจอปลายทางที่ค้นหา ในการกระจายข้อความร้องขอเส้นทางใหม่ในแต่ละครั้งจะทำให้ค่า Route Retrie เพิ่มขึ้นด้วย

เมื่อเส้นทางใหม่ได้ถูกสร้างขึ้น เส้นทางใหม่จะถูกบันทึกไว้ในตารางเส้นทาง แต่ถ้ามีเส้นทางที่ได้ค้นพบขึ้นมาใหม่ที่มีโหนดปลายทางเดียวกัน แต่จำนวนช่วงของเส้นทางใหม่มากกว่าเส้นทางเก่า ค่า TTL สำหรับการส่งข้อความร้องขอเส้นทางครั้งใหม่ จะตั้งโดยนำระยะทางบวกด้วยค่า TTL ที่เพิ่มขึ้น วิธีนี้ทำโหนดต้นทางสามารถรู้ว่าพื้นที่ครั้งสุดท้ายที่เจอโหนดปลายทาง หรือรู้ว่าโหนดน่าจะเคลื่อนที่ไปบริเวณใด

2.4.2 การบำรุงรักษาเส้นทาง

โดยใน AODV โหนดจะมีช่วงเวลาในการส่งข้อความ Hello เพื่อจะปรับปรุงข้อมูลในตารางเส้นทางของโหนดเอง เมื่อโหนดต้นทางได้เส้นทางไปถึงโหนดปลายทางแล้ว โหนดจะรักษาเส้นทางไว้นานเท่าที่โหนดต้นทางต้องการ การเคลื่อนที่ของโหนดจะมีผลต่อเส้นทางที่มีในโหนดเรียกเส้นทางที่มีโหนดเคลื่อนที่นี้ว่า Active Path ถ้ามีการเคลื่อนที่ของโหนดใดโหนดหนึ่ง โดยการที่โหนดที่อยู่ในเส้นทางก่อนที่จะถึงโหนดที่เคลื่อนที่จะส่งข้อความ Route Error (RERR) กลับมายังโหนดต้นทาง โดยผ่านทาง Precursor List ของโหนดเอง เมื่อโหนดข้างเคียงรู้ว่าโหนดใดได้มีการเคลื่อนที่ แล้วโหนดจะรู้ว่าไม่สามารถที่จะใช้งานโหนดดังกล่าวได้ โดยจะตั้งค่าระยะทางถึงโหนดปลายทางมีค่าเท่ากับอนันต์ และก็จะส่งข้อความ RERR ไปยังโหนดข้างเคียงในเส้นทางต่อไป เมื่อโหนดต้นทางได้รับข้อความ RERR โหนดก็จะกระทำการค้นหาเส้นทางใหม่อีกครั้ง ถ้าหากยังคงมีความต้องการที่จะใช้เส้นทางในส่งข้อมูลต่อ

จากภาพประกอบที่ 2-12 เป็นตัวอย่าง ของการบำรุงรักษาเส้นทางโดยโหนดต้นทางคือโหนด S โหนดปลายทางคือ โหนด D ซึ่งโหนด S มีเส้นทางเพื่อไปยังโหนด D คือ 1-2-3 ต่อมาโหนด 3 เคลื่อนที่ออกไปทำให้ไม่สามารถส่งข้อมูลได้ ทำให้โหนด 2 ส่งข้อความ RERR กลับมายังโหนด S จากนั้นโหนด S ก็ค้นหาเส้นทางใหม่ซึ่งได้เส้นทางเป็น 1-2-4 ไปยังโหนด D



ภาพประกอบ 2-12 การบำรุงรักษาเส้นทาง

2.4.2.1 การจัดการ Local Connectivity

เมื่อใดก็ตามที่โหนดได้รับข้อความร้องเส้นทางจากโหนดข้างเคียง โหนดก็จะทำการปรับปรุงค่าใน Lifetime Field ถ้ามีโหนดดังกล่าวอยู่ในตารางเส้นทาง แต่หากไม่มีโหนดดังกล่าวอยู่ในตารางเส้นทาง โหนดก็จะสร้างเส้นทางดังกล่าวขึ้นมาใหม่ และถ้าโหนดไม่ได้รับข้อความร้องเส้นทางจากโหนดข้างเคียงเลย ในช่วงที่เป็น Hello Interval โหนดจะกระจายข้อความ Hello ไปยังโหนดข้างเคียงเพื่อจะทราบว่าโหนดข้างเคียงดังกล่าวยังอยู่หรือไม่ นอกจากนี้เพื่อป้องกันการกระจายข้อความ Hello โหนดก็จะมีการใส่ค่า TTL เท่ากับ 1 โดยถ้ามีการเปลี่ยนแปลงโหนดก็จะมาปรับปรุงค่าในตารางเส้นทาง

2.5 การประมาณค่าพลังงานในการค้นหาเส้นทาง

EARP [12] เป็น โพรโทคอลสำหรับประมาณค่าพลังงานในการค้นหาเส้นทาง เริ่มจากพลังงานในการกระจายข้อความร้องขอเส้นทาง (Route Request (ROUTE_REQ)) และพลังงานที่ใช้ในการส่งข้อความตอบกลับเส้นทาง (Route Reply (ROUTE_REP))

โดยพลังงานในการส่งกระจายข้อความ (P_1)

$$P_1 = [\alpha_{r1} + \alpha_{r2}d^4]Bt \quad (2-2)$$

พลังงานที่ใช้รับข้อความ (P_2)

$$P_2 = \alpha_r B t_c \quad (2-3)$$

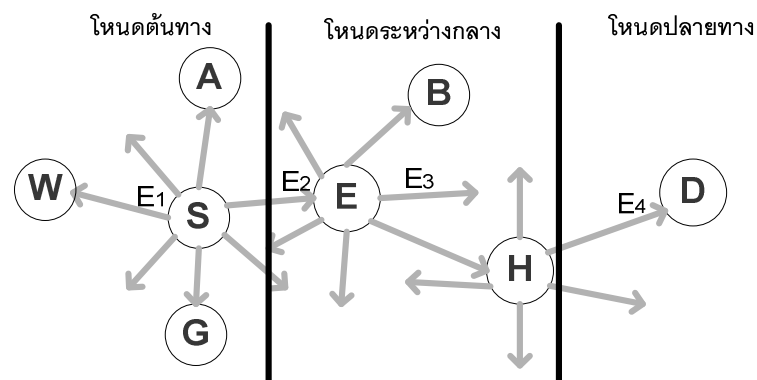
พลังงานในการส่งข้อความตามเส้นทางในลักษณะที่ไม่กระจาย (P_3)

$$P_3 = [\alpha_{t1} + \alpha_{t2} d^4] B t_c \quad (2-4)$$

โดยที่

- α_{t1}, α_{t2} เป็นค่าคงที่ในการสื่อสาร
- d^4 ค่าเฉลี่ยระหว่างโหนดข้างเคียง 2 โหนด
- B เป็นขนาดช่องทางการเชื่อมต่อโดยรวมมีหน่วยเป็นกิโลบิตต่อวินาที
- t เป็นขนาดของเฟรมโดยรวมในแต่ละวินาที
- α_r เป็นค่าคงที่เท่ากับ 135 นาโนจูลต่อบิต
- t_c เป็นเวลาของหนึ่ง Time Slot ในแต่ละวินาที

2.5.1 พลังงานที่ใช้ในการกระจายข้อความร้องขอเส้นทาง



ภาพประกอบ 2-13 การใช้พลังงานในการกระจายข้อความ ROUTE_REQ

โหนดต้นทางพยายามหาเส้นทางไปยังโหนดปลายทางโดยการกระจายข้อความร้องขอเส้นทางไปทั่วทั้งเครือข่าย

พลังงานที่โหนดต้นทาง ใช้ส่ง ROUTE_REQ (E_1)

$$E_1 = [\alpha_{i1} + \alpha_{i2}d^4]Bt \quad (2-5)$$

พลังงานที่โหนดระหว่างกลางใช้รับ ROUTE_REQ (E_2)

$$E_2 = \alpha_r Bt_c \quad (2-6)$$

พลังงานที่โหนดระหว่างกลางใช้ส่ง ROUTE_REQ (E_3)

$$E_3 = E_1 \quad (2-7)$$

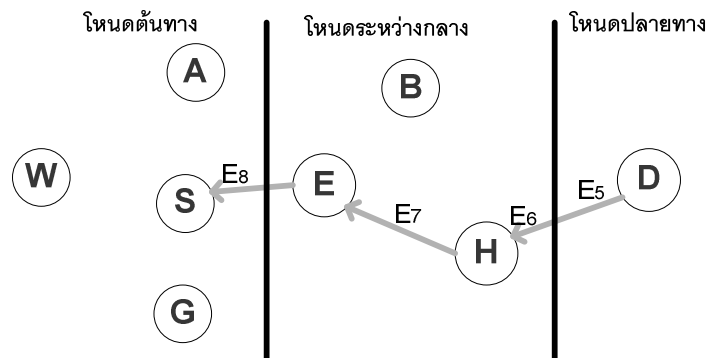
พลังงานที่โหนดปลายทาง ใช้รับ ROUTE_REQ (E_4)

$$E_4 = E_2 \quad (2-8)$$

พลังงานทั้งหมดที่ใช้ในการส่ง ROUTE_REQ ตามภาพประกอบ 2-13 จากโหนดต้นทางไปยังโหนดปลายทาง (E_q) หาได้จากพลังงานที่ใช้ส่ง ROUTE_REQ ของโหนดต้นทางรวมกับพลังงานที่ใช้รับ และส่ง ROUTE_REQ ของโหนดระหว่างกลาง คูณกับ N-2 คือโหนดระหว่างกลางที่มีทั้งหมด โดยที่ N จะเป็นจำนวนโหนดทั้งหมดในเครือข่าย และลบด้วยสองคือโหนดต้นทางและโหนดปลายทาง สุดท้ายรวม กับพลังงานที่โหนดปลายทางใช้รับ ROUTE_REQ ตามสมการ 2-9

$$\begin{aligned}
 E_q &= E_1 + (E_2 + E_3) * (N - 2) + E_4 \\
 &= P_1 + (P_2 + P_1) * (N - 2) + P_2 \\
 &= (P_1 + P_2) * (N - 1)
 \end{aligned}
 \tag{2-9}$$

2.5.2 พลังงานที่ใช้ในการส่งข้อความตอบรับเส้นทาง



ภาพประกอบ 2-14 การใช้พลังงานในการส่ง ROUTE_REP

พลังงานทั้งหมดที่โหนดปลายทางใช้ส่ง ROUTE_REP (E_5)

$$E_5 = [\alpha_{t_1} + \alpha_{t_2} d^4] * B * t_c \tag{2-10}$$

พลังงานที่โหนดระหว่างกลางใช้รับ ROUTE_REP (E_6)

$$E_6 = \alpha_r B t_c \tag{2-11}$$

พลังงานที่โหนดระหว่างกลางใช้ส่ง ROUTE_REP (E_7)

$$E_7 = E_5 \tag{2-12}$$

พลังงานที่โหนดต้นทางใช้รับ ROUTE_REP (E_8)

$$E_8 = E_6 \quad (2-13)$$

พลังงานทั้งหมดที่ใช้ในการส่ง ROUTE_REP จากโหนดปลายทางไปยังโหนดต้นทาง (E_p) ตามภาพประกอบ 2-14 โดยหาได้จากพลังงานที่โหนดปลายทางใช้ส่ง ROUTE_REP รวมกับพลังงานที่ใช้รับ และส่งข้อความ ROUTE_REP ของโหนดระหว่างกลาง คูณกับ h ซึ่งเป็นจำนวนโหนดระหว่างกลางที่อยู่ในเส้นทาง ซึ่งไม่ใช่โหนดระหว่างทั้งหมดในเครือข่ายรวมกับพลังงานที่ใช้รับ ROUTE_REP ของโหนดต้นทางตามสมการ 2-14

$$\begin{aligned} E_p &= E_5 + (E_6 + E_7) * (h) + E_8 \\ &= P_3 + (P_2 + P_3) * h + P_2 \\ &= (P_2 + P_3) * (h + 1) \end{aligned} \quad (2-14)$$

พลังงานทั้งหมดที่ใช้ในการค้นหาเส้นทางหาได้จากนำสมการ 2-9 รวมสมการ 2-14 จะได้ตามสมการ 2-15

$$\begin{aligned} &= E_q + E_p \\ &= (P_1 + P_2) * (N - 1) + (P_2 + P_3) * (h + 1) \\ &= (N - 1)P_1 + (N + h)P_2 + (h + 1)P_3 \end{aligned} \quad (2-15)$$

ในบทนี้ได้กล่าวถึง ความรู้เบื้องต้นเกี่ยวกับความรู้เบื้องต้นในเครือข่ายตรวจจับไร้สาย โพรโทคอลที่ประหยัดพลังงานในเครือข่ายตรวจจับไร้สาย โปรแกรมจำลองเครือข่าย โพรโทคอล DSR และ AODV สุดท้ายเป็นสมการประมาณค่าพลังงานในการค้นหาเส้นทาง ของโพรโทคอลที่มีลักษณะกระจายข้อความร้องขอเส้นทางไปทั่วทั้งเครือข่าย ซึ่งความรู้เหล่านี้นำไปใช้ในการออกแบบบอัลกอริทึม และทำการทดลองต่อไป

บทที่ 3

การออกแบบระบบ

งานวิจัยนี้มีจุดประสงค์เพื่อนำเสนออัลกอริทึมค้นหาเส้นทางที่มีเส้นทางสำรอง ในบทนี้จะกล่าวถึงส่วนของการออกแบบและพัฒนาระบบ รูปแบบวิธีการทดลอง สภาพแวดล้อมการทดลอง และสมการประมาณค่าพลังงานในการค้นหาเส้นทางของโปรโทคอล

3.1 ขั้นตอนในการออกแบบระบบ

โปรโทคอลค้นหาเส้นทางสำรองสำหรับเครือข่ายตรวจจับไร้สายนั้น มีการออกแบบให้มีความน่าเชื่อถือในการสื่อสารข้อมูลเพิ่มขึ้น โดยการเพิ่มเส้นทางสำรองอีกหนึ่งเส้นทางเพื่อใช้ในกรณีเส้นทางหลักล้มเหลว โดยโหนดต้นทางสามารถเลือกใช้เส้นทางที่มีการใช้พลังงานต่ำไปยังปลายทางได้เนื่องจากมีเส้นทางให้เลือก ซึ่งทั้งสองเส้นทางจะใช้โหนดที่ไม่ซ้ำกันเพื่อจะให้ทั้งสองเส้นทางมีความเสถียรภาพสูง เนื่องจากหากมีโหนดที่ซ้ำกันในทั้งสองเส้นทาง หากโหนดที่ซ้ำกันนั้นล้มเหลวนั้นก็หมายถึง ทั้งสองเส้นทางล้มเหลวด้วย

3.1.1 การพัฒนาอัลกอริทึมการค้นหาเส้นทาง และการรักษาเส้นทาง

ตามที่ได้กล่าวข้างต้น การที่ทั้งสองเส้นทางมีโหนดไม่ซ้ำกันก็ต้องมีการใช้สถานะโหนดตามข้างล่างนี้

สถานะโหนด	ความหมาย
F	ว่าง
W	ทำงาน
C	เชื่อมต่อ

ตาราง 3-1 สถานะโหนด

ตามตาราง 3-1 ในขณะที่เริ่มต้นโหนดในเครือข่ายจะมีสถานะ F ยกเว้นโหนดต้นทางที่ต้องการติดต่อ กับโหนดปลายทาง ก็จะมีสถานะ C ซึ่งโหนดต้นทางจะเป็นโหนดในเครือข่ายที่ต้องการส่งข้อมูลตรวจจับ ส่วนโหนดปลายทางก็จะเป็นโหนดหมายเลข 0 หรือโหนดที่รวบรวมข้อมูล เมื่อโหนดต้นทางกระจายข้อความร้องขอเส้นทาง โหนดที่ได้รับข้อความร้องขอเส้นทางจะเปลี่ยนสถานะเป็น W จนกว่าจะได้รับข้อความแจ้งผลการคัดเลือกโหนด หากได้รับการคัดเลือกก็จะเปลี่ยนสถานะเป็น C แต่หากไม่ได้รับการคัดเลือกก็จะเปลี่ยนสถานะกลับเป็น F เช่นเดิม โหนดที่มีสถานะ W และ C จะไม่เข้าร่วมการคัดเลือกโหนด

ข้อกำหนดของโพรโทคอลที่พัฒนา

- โหนดจะต้องวางเรียงกันตามโหนด ID หรือเป็นกลุ่มที่เรียงกัน
- พลังงานขั้นต่ำ ET (Energy Threshold) เป็นค่าคงที่ มีค่าเป็น 0.3 จูล
- ข้อความร้องขอเส้นทาง (ROUTE_REQ)
- ข้อความตอบรับการหาเส้นทาง (ROUTE_REP)
- ข้อความร้องขอการคัดเลือกการเชื่อมต่อ (RESULT_REQ (Result Request))
- ข้อความแจ้งผลการคัดเลือกการเชื่อมต่อ (RESULT_REP (Result Reply))
- ข้อความแจ้งโหนดล้มเหลว (ROUTE_FAIL (Route Fail))

ตามตาราง 3-2 เป็นอัลกอริทึมการค้นหาเส้นทาง โดยเริ่มต้นสถานะโหนดจะเป็น F นอกจากโหนดต้นทาง ซึ่งเป็นโหนดที่ต้องการส่งข้อมูลมีสถานะโหนดเป็น C ส่วนโหนดปลายทางเป็นโหนดที่รวบรวมข้อมูล โหนดต้นทางจะกระจายข้อความ ROUTE_REQ ออกไปโหนดระหว่างกลางที่ได้รับข้อความจะตรวจสอบระดับพลังงานของตัวเอง กับพลังงานขั้นต่ำ หากมีค่าพลังงานต่ำกว่าพลังงานขั้นต่ำ ก็จะทิ้งข้อความ ROUTE_REQ แต่หากมีค่าพลังงานสูงกว่าพลังงานขั้นต่ำก็จะเปลี่ยนสถานะไปเป็น W และจะส่งข้อความ RESULT_REQ เพื่อเข้าร่วมการคัดเลือกโหนด จากนั้นโหนดต้นทางจะเข้าสู่กระบวนการคัดเลือกโหนด โดยใช้วิธีการคำนวณผลต่างระหว่างโหนด ID ของโหนดปลายทาง และ ID ของโหนดที่ส่งข้อความ RESULT_REQ เข้ามา สาเหตุที่ต้องใช้วิธีการนี้เนื่องจากโพรโทคอล RPRP นั้นจะค้นเส้นทางโดยไม่กระจายข้อความ ROUTE_REQ ไปทั้งเครือข่าย แต่จะทำการคัดเลือกโหนดเพื่อกระจายข้อความแทน ทำให้หากไม่ใช้วิธีนี้ก็จะทำให้การค้นหาเส้นทางนั้น ใช้เวลานานในการค้นหา เพราะไม่มีทิศทางในการค้นหาทำให้สิ้นเปลืองพลังงานทั้งเครือข่าย แต่หากใช้วิธีการคำนวณผลต่าง ID ของโหนดซึ่ง โหนดไม่เคลื่อนที่และเรียงตาม ID ของโหนดอยู่แล้วทำให้การค้นหาที่เร็วขึ้น และไม่สิ้นเปลืองพลังงานมากเกินไป

โหนด	ขั้นตอนการทำงาน
โหนดต้นทาง	<ul style="list-style-type: none"> • กระจายข้อความ ROUTE_REQ และมีสถานะเป็น C • คำนวณผลต่างระหว่างโหนด ID ของโหนดปลายทาง และโหนดที่จะส่งข้อความ RESULT_REQ • เลือกโหนดเชื่อมต่อหนึ่งโหนด โดยเลือกโหนดที่มีผลต่างน้อยที่สุด • เมื่อได้รับข้อความ ROUTE_REQ แรก ก็จะทำการกระจายข้อความ ROUTE_REQ เพื่อหาเส้นทางที่สองโดยโหนดที่มีสถานะ C จะไม่เข้าร่วมคัดเลือกโหนด ซึ่งทำให้เส้นทางแรกและเส้นทางที่สองไม่มีโหนดที่ซ้ำกัน • ส่งข้อความ RESULT REP แจ้งผลการคัดเลือกโหนด • เมื่อได้รับข้อความ ROUTE_FAIL ให้เลือกใช้เส้นทางสำรองทันที • เมื่อได้ทั้งสองเส้นทางแล้วก็ให้เลือกเส้นทางหลัก โดยเลือกจากจำนวนช่วงที่น้อยที่สุด (Hop Count) ไปยังปลายทาง หากเท่ากันก็ให้เลือกเส้นทางที่ 2 เนื่องจากเป็นเส้นทางที่ใหม่กว่า
โหนดระหว่างกลาง	<ul style="list-style-type: none"> • โหนดที่มีพลังงานน้อยกว่า ET ทิ้งข้อความ ROUTE_REQ • โหนดที่มีพลังงานมากกว่า ET เปลี่ยนสถานะเป็น W และส่ง RESULT_REQ ไปยังโหนดที่ส่ง ROUTE_REQ • โหนดที่ได้รับการเชื่อมต่อที่ให้คำนวณผลต่างระหว่างโหนด ID ของโหนดปลายทางและโหนดที่ส่งข้อความ RESULT_REQ • โหนดที่ได้รับการเชื่อมต่อให้เลือกโหนดเชื่อมต่อหนึ่งโหนด โดยเลือกโหนดที่มีผลต่างน้อยที่สุด และประกาศผลการคัดเลือกผ่านข้อความ RESULT REP • เมื่อได้รับข้อความ RESULT_REQ หากไม่ได้รับการคัดเลือกให้ทำการทิ้งข้อความ ROUTE_REQ และเปลี่ยนสถานะเป็น F โหนดที่ได้รับการคัดเลือกก็จะเปลี่ยนสถานะเป็น C และกระจายข้อความ ROUTE_REQ ต่อไป
โหนดปลายทาง	<ul style="list-style-type: none"> • เมื่อรับข้อความ ROUTE_REQ จึงส่งข้อความ ROUTE_REQ ไปยังโหนดต้นทางตามเส้นทางเดิม

ตาราง 3-2 อัลกอริทึมการหาเส้นทางของโปรโตคอล RPRP

หลังจากผ่านกระบวนการคำนวณผลต่าง ID ของโหนดปลายทาง และ ID ของโหนดที่เข้าร่วมคัดเลือกโหนดแล้ว จากนั้นโหนดก็จะแจ้งผลการคัดเลือกโหนดผ่านข้อความ RESULT_REP เมื่อโหนดระหว่างกลางได้รับข้อความดังกล่าว ก็จะตรวจสอบผลการคัดเลือกโหนด หากไม่ได้รับการคัดเลือกก็จะเปลี่ยนสถานะเป็น F แต่หากได้รับการคัดเลือกก็จะเปลี่ยนสถานะเป็น C และจะกระจายข้อความร้องเส้นทางต่อไป และจะกระทำตามขั้นตอนการคัดเลือกโหนดต่อไป จนกว่าจะถึงโหนดปลายทาง เมื่อข้อความ ROUTE_REQ ถึงโหนดปลายทาง จากนั้นโหนดปลายทางจะส่งข้อความ ROUTE_REP กลับไปยังโหนดที่ส่งข้อความ ROUTE_REQ และกระบวนการคัดเลือกโหนดจะสิ้นสุดลง หลังจากได้รับข้อความ ROUTE_REP และข้อความดังกล่าวก็จะส่งกลับไปยังเส้นทางเดิม จนถึงโหนดต้นทางก็จะเริ่มส่งข้อมูลต่อไป

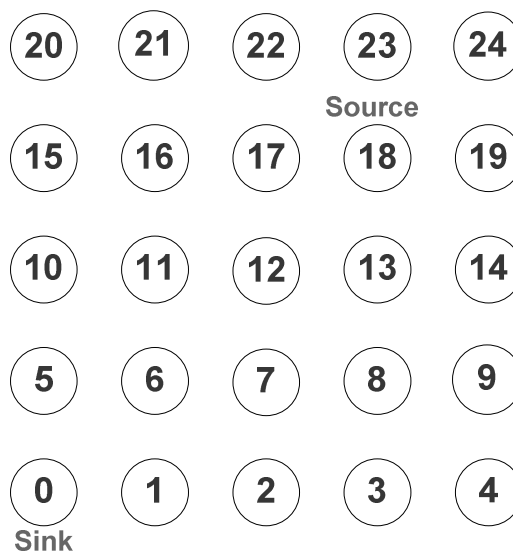
โหนด	ขั้นตอนการทำงาน
โหนดระหว่างกลาง	<ul style="list-style-type: none"> • เมื่อโหนดรับข้อความ ROUTE_FAIL ให้ส่งต่อไปตามเส้นทาง • หากส่งข้อมูลแล้วไม่ได้รับข้อความยืนยันการได้รับข้อมูลเกินเวลาที่กำหนดให้ส่งข้อความ ROUTE_FAIL กลับไปยังต้นทาง
โหนดต้นทาง	<ul style="list-style-type: none"> • เมื่อได้รับข้อความ ROUTE_FAIL ก็เลือกใช้เส้นทางสำรองทันที • ถ้าเส้นทางสำรองล้มเหลวอีก ก็ค้นหาเส้นทางใหม่ โหนดต้นทางจะไม่สนใจว่าจะหมดพลังงานหรือไม่

ตาราง 3-3 อัลกอริทึมการรักษาเส้นทาง

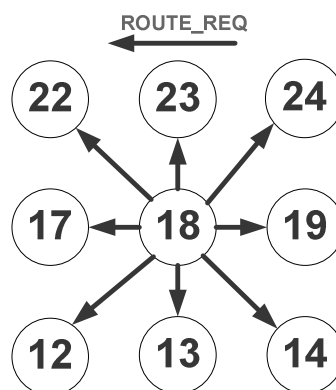
ตามตารางที่ 3-3 เป็นตารางอัลกอริทึมการรักษาเส้นทาง โดยโหนดระหว่างกลางเมื่อส่งข้อมูลแล้ว หากยังไม่ได้รับข้อความยืนยันการได้รับข้อความข้อมูลตามเวลาที่กำหนด ก็จะส่งข้อความ ROUTE_FAIL กลับไปตามเส้นทางไปยังโหนดต้นทางของข้อมูล เมื่อโหนดต้นทางได้รับข้อความ ROUTE_FAIL ก็จะทำการใช้เส้นทางสำรองทันที

3.1.2 ตัวอย่างการค้นหาเส้นทาง

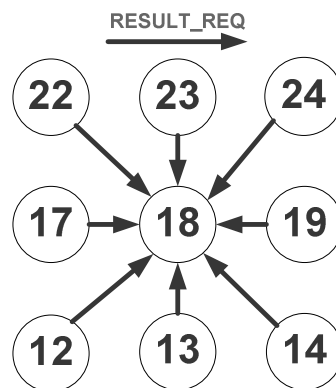
จากภาพประกอบ 3-1 โหนด 18 ซึ่งเป็นโหนดต้นทาง ต้องการติดต่อกับโหนด 0 ที่เป็นโหนดปลายทาง หรือเป็นโหนดที่รวบรวมข้อมูล ในขณะที่เริ่มต้น โหนดต่างๆ ในเครือข่ายจะมีสถานะเป็น F นอกจากโหนดต้นทางจะมีสถานะ C โหนด 18 ก็จะกระจายข้อความ ROUTE_REQ ออกไปตามภาพประกอบ 3-2 โดยโหนดที่ได้รับข้อความ คือ โหนด 24, 23, 22, 19, 17, 12, 13 และ 14



ภาพประกอบ 3-1 การค้นหาเส้นทางในโปรโตคอล RPRP



ภาพประกอบ 3-2 การกระจายข้อความ ROUTE_REQ



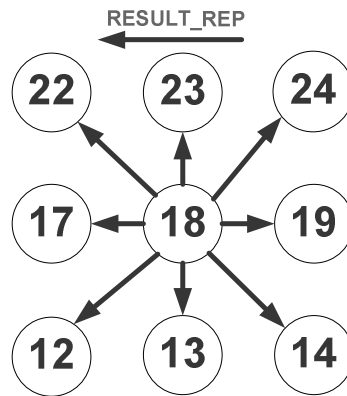
ภาพประกอบ 3-3 การส่งข้อความ RESULT REQ

ต่อไปก็จะเข้าสู่กระบวนการคัดเลือกโหนด ทุกโหนดที่ได้รับข้อความ ROUTE_REQ ก็จะตรวจสอบระดับพลังงานของตัวเอง หากมีพลังงานน้อยกว่าพลังงานขั้นต่ำก็จะทิ้งข้อความ แต่หากมีพลังงานมากกว่าพลังงานขั้นต่ำ ก็จะเข้าร่วมกระบวนการคัดเลือกโหนด โดยจะเปลี่ยนสถานะเป็น W แล้วจึงส่งข้อความ RESULT_REQ ไปยังโหนด 18 เพื่อขอเข้าร่วมกระบวนการคัดเลือกโหนด ตามภาพประกอบ 3-3 ซึ่งโหนด 18 ก็จะคำนวณหาผลต่างตามตาราง 3-4 โดยจะนำเอาโหนด ID ของโหนดที่ส่งข้อความ RESULT_REQ เข้ามา ลบด้วยโหนด ID ของโหนดปลายทาง ซึ่งโหนดที่ได้รับการคัดเลือกก็มีเพียงโหนดเดียว ที่มีผลต่างน้อยสุดตามตารางที่ 3-4 จะเป็นโหนด 12

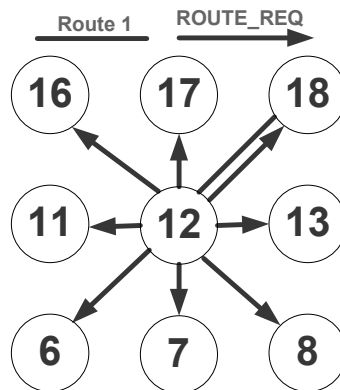
โหนดต้นทาง	โหนดปลายทาง	โหนดที่ส่ง RESULT REQ	คำนวณผลต่าง
18	0	24	$(24-0) = 24$
		23	$(23-0) = 23$
		22	$(22-0) = 22$
		19	$(19-0) = 19$
		17	$(17-0) = 17$
		14	$(14-0) = 14$
		13	$(13-0) = 13$
		12	$(12-0) = 12$

ตาราง 3-4 การคำนวณผลต่างในการคัดเลือกโหนด

หลังจากนั้น โหนดที่ 18 ก็จะประกาศผลการคัดเลือก โหนด โดยทำการส่งข้อความ RESULT_REP ตามภาพประกอบ 3-4 โหนดที่จะได้รับการคัดเลือกก็จะเปลี่ยนสถานะเป็น C ส่วน โหนดที่ไม่ได้รับการคัดเลือก ก็จะเปลี่ยนสถานะเป็น F เช่นเดิม

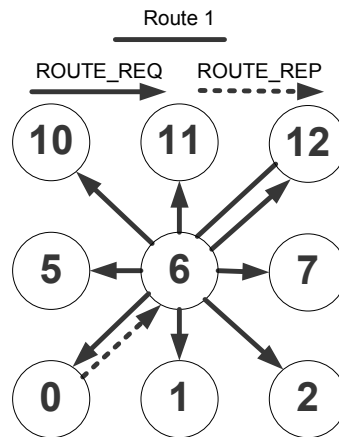


ภาพประกอบ 3-4 การส่งข้อความ RESULT REP

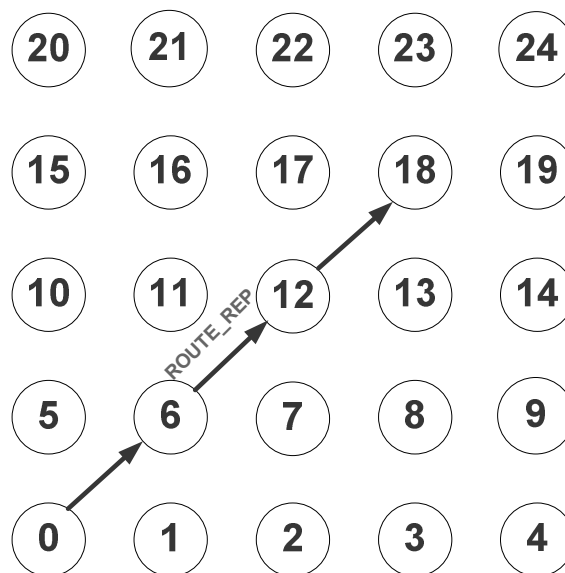


ภาพประกอบ 3-5 โหนดที่ได้รับการคัดเลือกให้เชื่อมต่อ

จากนั้น โหนด 12 เมื่อได้รับการคัดเลือก ก็เปลี่ยนสถานะไปเป็น C และจะกระจายข้อความ ROUTE_REQ ต่อไปตามภาพประกอบ 3-5 แล้วจะกระทำตามขั้นตอนการคัดเลือก โหนดตามข้างต้น จนกระทั่งถึงโหนดปลายทาง

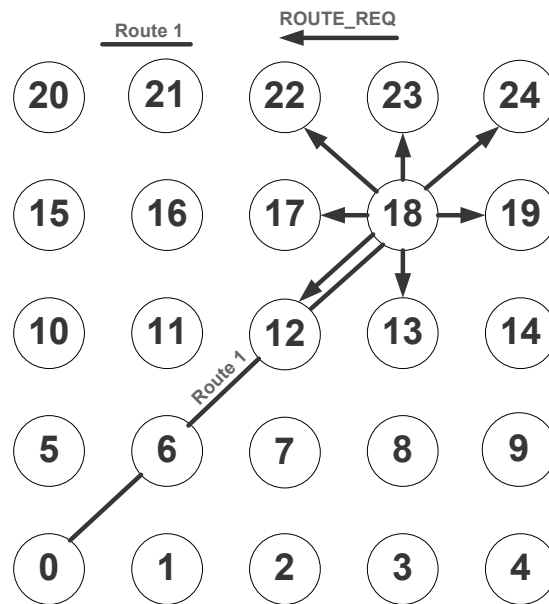


ภาพประกอบ 3-6 โหนดปลายทางส่งข้อความ ROUTE_REP



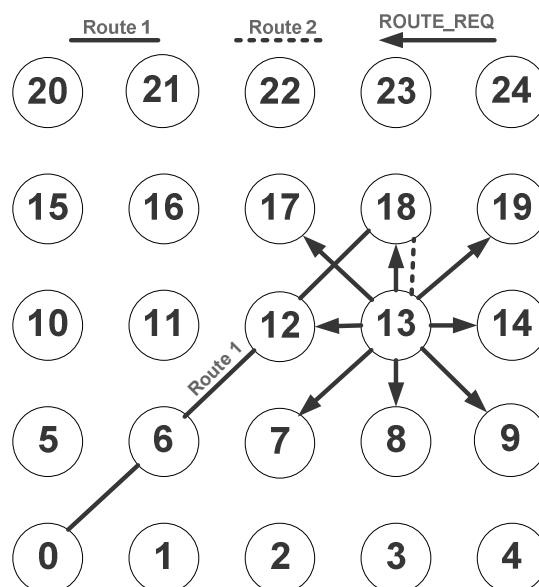
ภาพประกอบ 3-7 การส่งข้อความ ROUTE_REP กลับไปยังต้นทาง

เมื่อโหนดปลายทางได้รับข้อความ ROUTE_REQ ก็จะตอบกลับด้วยการส่งข้อความ ROUTE_REP ออกไปตามภาพประกอบ 3-6 โหนด 6 เมื่อได้รับข้อความ ROUTE_REP ก็ยุติการคัดเลือกโหนดทันที พร้อมส่งข้อความ ROUTE_REP ต่อไปตามเส้นทางไปยังโหนดต้นทาง ตามภาพประกอบ 3-7 ก็จะได้เส้นทางที่ 1



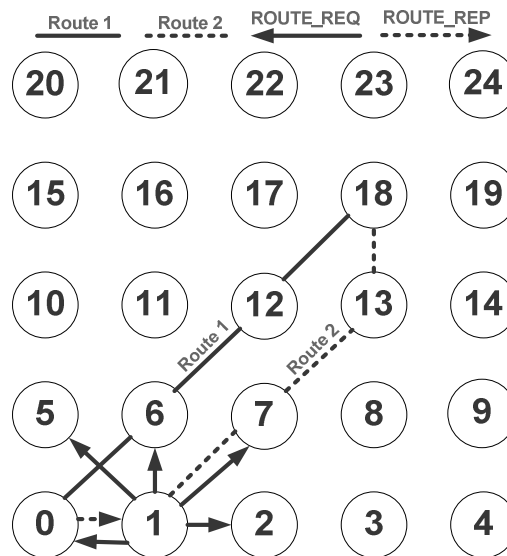
ภาพประกอบ 3-8 การส่งข้อความ ROUTE_REP กลับไปยังต้นทาง

จากนั้นเมื่อข้อความ ROUTE_REP ส่งถึงโหนด 18 จากนั้นโหนด 18 ก็จะกระจายข้อความ ROUTE_REQ ออกไปเพื่อหาเส้นทางที่ 2 ตามภาพประกอบ 3-8

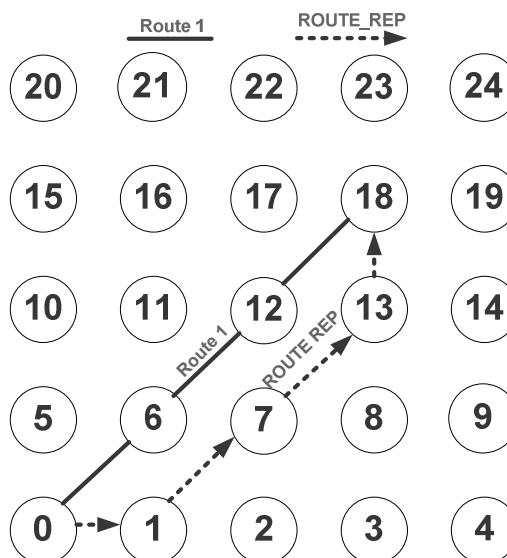


ภาพประกอบ 3-9 การค้นหาเส้นทางที่ 2

และจะกระทำตามกระบวนการคัดเลือกโหนดต่อไป ก็จะได้โหนด 13 ที่ได้รับการคัดเลือกให้ส่งข้อความ ROUTE_REQ ต่อไป ตามภาพประกอบ 3-9 แต่ในเส้นทางที่ 2 นี้โหนดที่มีสถานะ C ก็จะไม่เข้าร่วมกระบวนการคัดเลือกโหนด ซึ่งจะทำให้เส้นทางทั้งสองเส้นทางนั้นไม่มีโหนดที่ซ้ำกัน

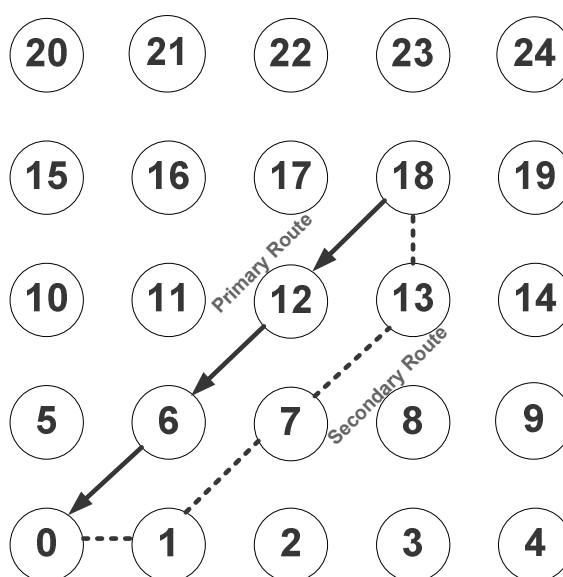


ภาพประกอบ 3-10 การตอบกลับเส้นทางที่ 2 และสิ้นสุดการคัดเลือกโหนด



ภาพประกอบ 3-11 การส่งข้อความ ROUTE_REP กลับไปยังต้นทางของเส้นทางที่ 2

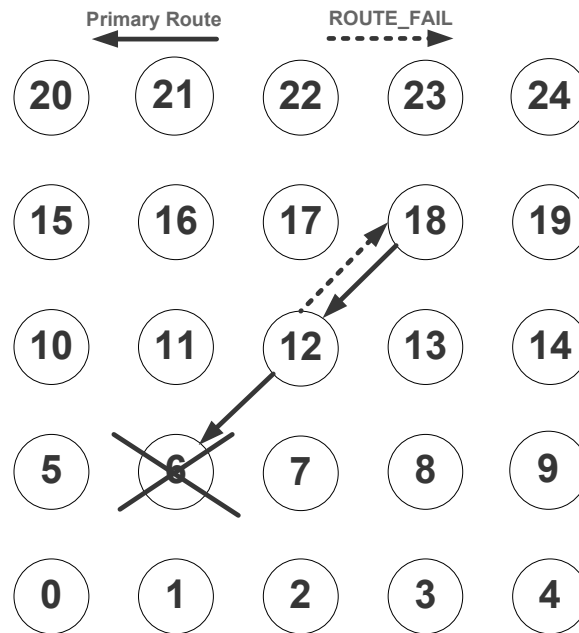
และเมื่อกระทำตามกระบวนการคัดเลือกโหนด จะกระทั่งถึงโหนดปลายทาง โดยเมื่อโหนดปลายทางได้รับข้อความ ROUTE_REQ ก็จะส่งข้อความ ROUTE_REP ไปยังโหนด 1 ก็จะสิ้นสุดกระบวนการคัดเลือกโหนด ตามภาพประกอบที่ 3-10 และหลังจากที่ข้อความ ROUTE_REP ของเส้นทางที่ 2 มาถึงโหนด 18 ตามภาพประกอบที่ 3-11 จากนั้นโหนด 18 จะทำการคัดเลือกเส้นทางหลัก โดยพิจารณาจากจำนวนช่วงของแต่ละเส้นทาง หากเส้นทางใดมีจำนวนช่วงน้อยกว่าก็จะเป็นเส้นทางหลัก อีกหนึ่งเส้นทางก็จะเป็นเส้นทางสำรอง หากเส้นทางมีจำนวนช่วงเท่ากันก็จะใช้เส้นทางที่สองเนื่องเป็นเส้นทางที่ใหม่กว่า ตามภาพประกอบ 3-12



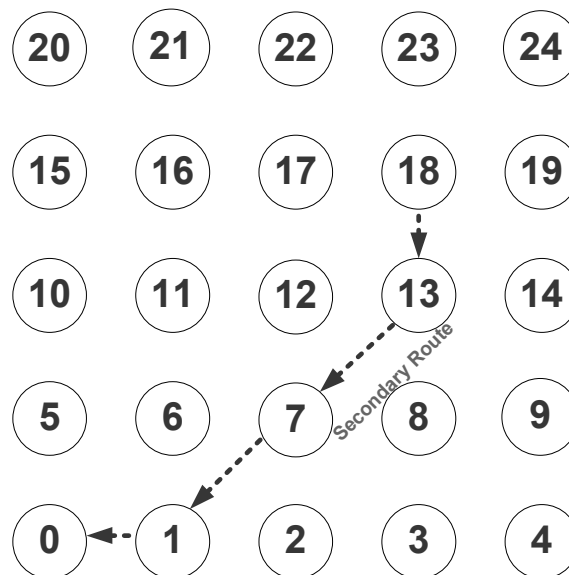
ภาพประกอบ 3-12 การเลือกเส้นทางหลักและเส้นทางสำรอง

3.1.3 ตัวอย่างการใช้เส้นทางสำรอง

ในกรณีของการใช้เส้นทางสำรองนั้น เมื่อโหนดในเส้นทางล้มเหลว โหนดที่อยู่ก่อนหน้า ก็จะส่งข้อมูลแต่ไม่ได้รับข้อความยืนยันการได้รับข้อมูลตามเวลาที่กำหนด ก็จะส่งข้อความ ROUTE_FAIL กลับไปยังโหนดต้นทางตามเส้นทางเดิม เมื่อข้อความมาถึงโหนดต้นทางก็รู้ว่าเส้นทางดังกล่าวล้มเหลว ก็จะใช้เส้นทางสำรองทันที



ภาพประกอบ 3-13 เส้นทางหลักล้มเหลว

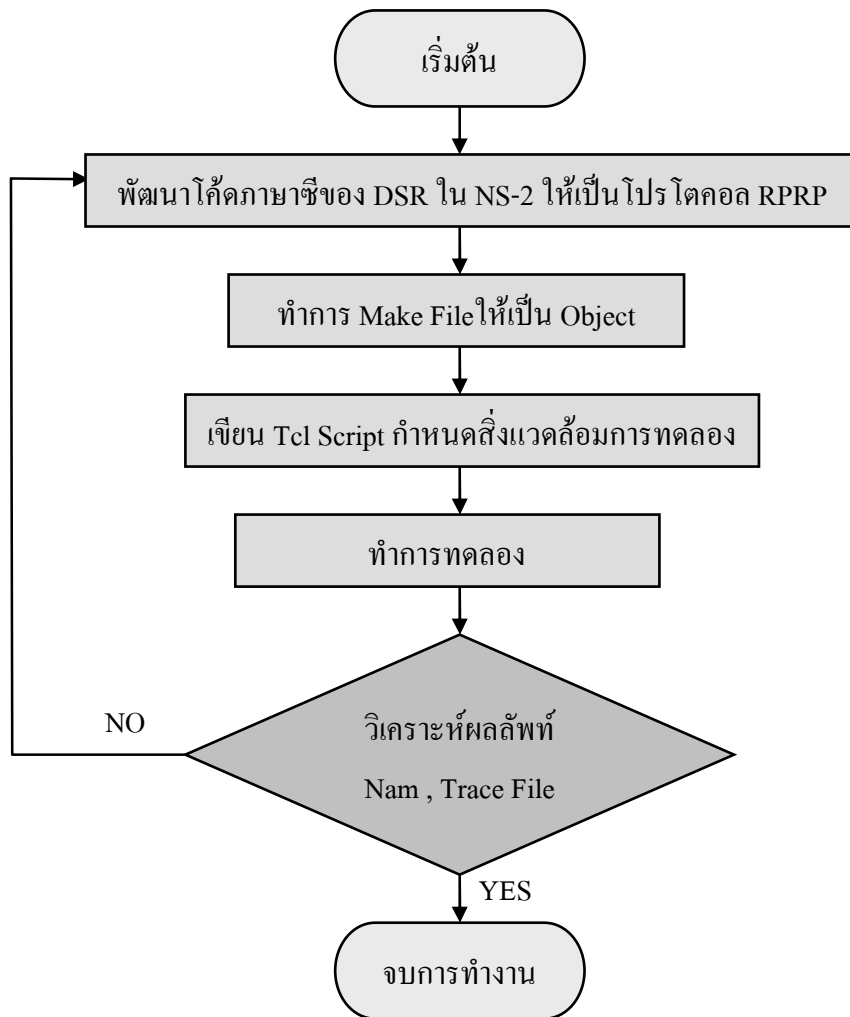


ภาพประกอบ 3-14 การใช้เส้นทางสำรอง

จากภาพประกอบ 3-13 เส้นทางหลักล้มเหลว โดยที่เส้นทางหลักคือ 18-12-6-0 และเส้นทางสำรองคือ 18-13-7-1-0 ซึ่งโหนด 6 ล้มเหลว โหนดก่อนหน้าก็จะส่งข้อมูลมายังโหนด 6 แต่ไม่ได้รับข้อความยืนยันการรับข้อมูลตามเวลาที่กำหนด ก็จะส่งข้อความ ROUTE_FAIL กลับไปยัง

โหนด 18 เมื่อโหนด 18 ได้รับข้อความก็จะรู้ว่าเส้นทางหลักนั้นล้มเหลว จึงใช้เส้นทางสำรองทันที ตามภาพประกอบ 3-14

3.2 ลำดับขั้นตอนการทดลอง



ภาพประกอบ 3-15 ขั้นตอนการทดลอง

จากภาพประกอบ 3-15 โปรโตคอลค้นหาเส้นทางสำรองสำหรับเครือข่ายตรวจจับไร้สาย พัฒนาจาก โปรโตคอล DSR ซึ่งมีขั้นตอนในการทดลองโดยเริ่มจากพัฒนาโค้ดภาษาซีของโปรโตคอล DSR ใน NS-2 ให้เป็นไปตามอัลกอริทึมของโปรโตคอล RPRP จากนั้นทำการ Make File ให้เป็น Object เพื่อให้สามารถนำไปใช้ได้ หลังจากนั้นก็เขียน Tcl Script เพื่อกำหนดสภาพแวดล้อมต่างๆ ในการทดลอง แล้วก็ทำการทดลอง ซึ่งผลที่ได้จากการทดลอง ที่นำมาวิเคราะห์ผลคือ Nam

Trace File และ NS Trace File การวิเคราะห์ผล NS Trace File จะต้องเขียน Perl Script และ Shell Script เพื่อกรองข้อมูลที่ต้องการ หากผลไม่ตรงตามอัลกอริทึมก็จะต้องไปทำการแก้ไขปรับปรุงโค้ดใหม่ แต่หากตรงตามความต้องการก็เป็นการสิ้นสุดการทดลอง

3.3 รูปแบบการทดลอง

การทดลองจะกำหนดสภาพแวดล้อมการทดลอง ตามตารางที่ 3-5 โดยจำนวนโหนดที่ใช้ในการทดลองคือ 100 โหนด โดยการทดลองจะให้โหนดปลายทางเป็นโหนด 0 ซึ่งโหนดปลายทางนี้ เปรียบเหมือนโหนดที่รวบรวมข้อมูล และส่งข้อมูลไปยังภายนอก เช่น เครื่องคอมพิวเตอร์ที่รวบรวมข้อมูลการตรวจจับ ส่วนโหนดต้นทางจะทำการสุ่มเลือกขึ้นมา โดยการทดลองจะให้โหนดตรวจจับอยู่กับที่ไม่ได้เคลื่อนไหว และวางโหนดห่างกัน 9 เมตร เป็นพื้นที่สี่เหลี่ยมจัตุรัส รัศมีคลื่นวิทยุ 20 เมตร และในการทดลองใช้ Energy Model [20] ที่มีใน NS 2.33 จะมีสีของโหนด 3 สี ที่บ่งบอกถึงระดับพลังงานคือ

สีเขียว	หมายถึง	พลังงานมีมากกว่าพลังงานขั้นต่ำ
สีเหลือง	หมายถึง	พลังงานน้อยกว่าพลังงานขั้นต่ำ
สีแดง	หมายถึง	ใกล้หมดพลังงาน

ทุกโหนดมีพลังงานเริ่มต้นที่ 3 จูล ส่วนพลังงานขั้นต่ำมีค่า 0.3 จูล จะทำการทดลอง 15 ครั้งเพื่อหาประสิทธิภาพโดยเฉลี่ยของโพรโทคอล และได้ทำการเปรียบเทียบกับโพรโทคอล DSR, AODV และ RPRP (Single Path) ซึ่งเป็นโพรโทคอล RPRP ที่มีเส้นทางเดียว

รายการ	สิ่งที่ใช้
Simulation Program	NS-2 (v2.33)
Channel Type	Wireless Channel
Network Interface Type	Phy/WirelessPhy/802.15.4
Radio-Propagation Model	TwoRayGround
MAC Type	Mac/802.15.4
Link Layer Type	LL
Antenna Model	Omni-directional Antenna
Interface Queue Type	DropTail/PriQueue
Max Packet in Interface Queue	50 ไบต์
Number of Mobile Node	100 โหนด
Routing Protocol	DSR, RPRP, RPRP(Single Path), AODV
Grid Width	90 เมตร
Grid Height	90 เมตร
Node Movement	ไม่เคลื่อนที่
Initial Energy	3 จูล
Transmitting Power	0.3 มิลลิวัตต์
Receiving Power	0.3 มิลลิวัตต์
Traffic	Constant Bit Rate (CBR)/UDP
Packet Size	500 ไบต์
Interval Time	0.2 วินาที
Source Node	สุ่ม
Sink Node	โหนดหมายเลข 0
Node Distance	9 เมตร
Radio Range	20 เมตร
Simulation Time	1000 วินาที

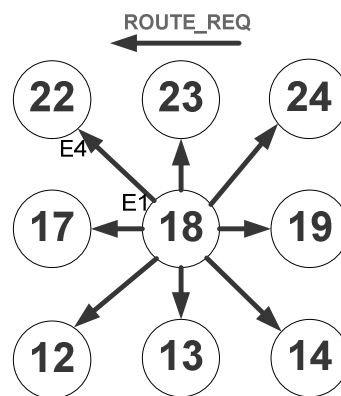
ตาราง 3-5 สภาพแวดล้อมในการทดลอง

3.4 สมการประมาณค่าพลังงานในการค้นหาเส้นทางของโปรโตคอล RPRP

พื้นฐานสมการประมาณค่าพลังงานได้กล่าวไปแล้ว ในบทที่ 2 ใน RPRP จะแบ่งออกเป็นสองส่วนหลักคือ พลังงานที่ใช้ในการกระจายการร้องขอเส้นทาง และพลังงานที่ใช้ในการตอบกลับเส้นทาง ซึ่งในส่วนของพลังงานที่ใช้ในการกระจายการร้องขอเส้นทางจะแบ่งออกเป็นสามส่วนด้วยกันคือ โหนดต้นทาง โหนดระหว่างกลาง และโหนดปลายทาง ดังนี้

3.4.1 โหนดต้นทาง

พลังงานที่โหนดต้นทางใช้กระจาย ROUTE_REQ (E_1)



ภาพประกอบ 3-16 พลังงานที่โหนดต้นทางกระจาย ROUTE_REQ

$$E_1 = [\alpha_{r1} + \alpha_{r2}d^4] * B * t \quad (3-1)$$

พลังงานที่โหนดต้นทางใช้รับ RESULT_REQ (E_2)

$$E_2 = \alpha_r * B * t_c \quad (3-2)$$

พลังงานที่โหนดต้นทางใช้ส่ง RESULT_REP (E_3)

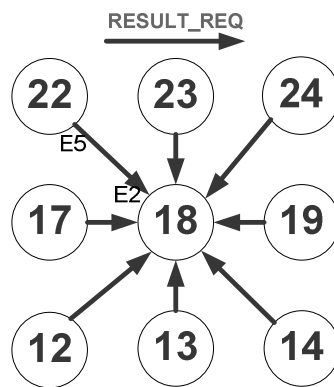
$$E_3 = E_1 \quad (3-3)$$

3.4.2 โหนดระหว่างกลาง

พลังงานที่โหนดระหว่างกลางใช้รับ ROUTE_REQ (E_4)

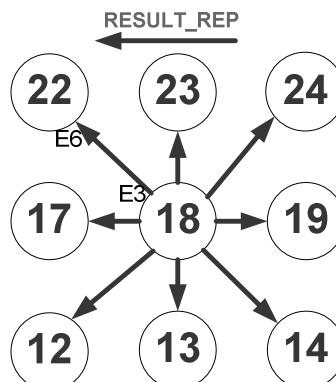
$$E_4 = E_2 \quad (3-4)$$

พลังงานที่โหนดระหว่างกลางใช้ส่ง RESULT_REQ (E_5)



ภาพประกอบ 3-17 พลังงานที่โหนดต้นทางรับ RESULT_REQ

$$E_5 = [\alpha_{t_1} + \alpha_{t_2} d^4] * B * t_c \quad (3-5)$$

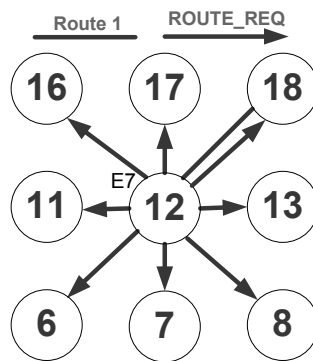


ภาพประกอบ 3-18 พลังงานที่โหนดต้นทางส่ง RESULT_REQ

พลังงานที่โหนดระหว่างกลางใช้รับ RESULT_REP (E_6)

$$E_6 = E_4 \quad (3-6)$$

พลังงานที่โหนดระหว่างกลางที่ได้รับการคัดเลือกใช้ส่ง ROUTE_REQ (E_7)

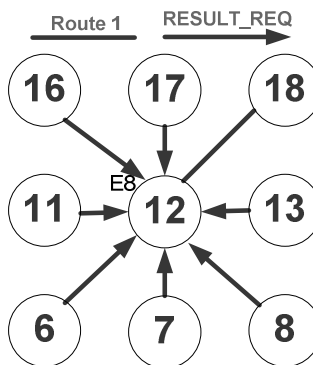


ภาพประกอบ 3-19 พลังงานที่โหนดระหว่างกลางที่ได้รับการคัดเลือกส่ง ROUTE_REQ

$$E_7 = E_1 \quad (3-7)$$

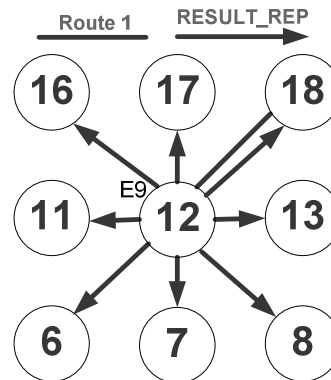
พลังงานที่โหนดระหว่างกลางที่ได้รับการคัดเลือกใช้รับ RESULT_REP (E_8)

$$E_8 = E_4 \quad (3-8)$$



ภาพประกอบ 3-20 พลังงานที่โหนดระหว่างกลางที่ได้รับการคัดเลือกรับ RESULT_REQ

พลังงานที่โหนดระหว่างกลางที่ได้รับการคัดเลือกใช้ ส่ง RESULT_REP (E_9)

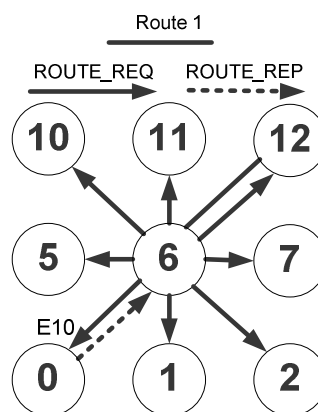


ภาพประกอบ 3-21 พลังงานที่โหนดระหว่างกลางที่ได้รับการคัดเลือกส่ง RESULT_REP

$$E_9 = E_3 \quad (3-9)$$

3.4.3 โหนดปลายทาง

พลังงานที่โหนดปลายทางใช้รับ ROUTE_REQ (E_{10})



ภาพประกอบ 3-22 พลังงานที่โหนดปลายทางรับ ROUTE_REQ

$$E_{10} = E_2 \quad (3-10)$$

พลังงานทั้งหมดที่ใช้ในการส่ง ROUTE_REQ จากโหนดต้นทางไปยังโหนดปลายทาง (E_q) หาได้มาจาก พลังงานที่โหนดต้นทางส่ง ROUTE_REQ (E_1) รวมกับพลังงานที่โหนดต้นทางรับข้อความ RESULT_REQ (E_2) รวมกับพลังงานที่โหนดต้นทางส่ง RESULT_REP (E_3) จากนั้นก็รวมกับพลังงานที่โหนดระหว่างกลางใช้รับข้อความ ROUTE_REQ (E_4) คูณกับจำนวนโหนดระหว่างกลางที่อยู่ในเส้นทางการค้นหา (M) จากนั้นก็รวมกับพลังงานที่โหนดระหว่างกลางใช้ส่งข้อความ RESULT_REQ (E_5) และรับข้อความ RESULT_REP (E_6) คูณด้วยผลลัพธ์ที่ได้จาก จำนวนโหนดระหว่างกลางที่อยู่ในเส้นทางลบด้วยจำนวนโหนดระหว่างกลางที่ได้รับคัดเลือกให้เชื่อมต่อ (m) จากนั้นก็รวมกับพลังงานที่โหนดระหว่างกลางที่ได้รับการคัดเลือกใช้ส่งข้อความ ROUTE_REQ (E_7) รวมกับพลังงานที่โหนดระหว่างกลางที่จะได้รับการคัดเลือกใช้รับข้อความ RESULT_REQ (E_8) รวมกับพลังงานที่โหนดระหว่างกลางที่จะได้รับการคัดเลือกใช้ส่งข้อความ RESULT_REP (E_9) และสุดท้ายรวมกับพลังงานที่โหนดปลายทางใช้รับข้อความ ROUTE_REQ (E_{10}) จากนั้นก็ทำการแปลงให้เป็นรูปแบบของพลังงานตามบทที่ 2 ก็จะได้สมการพลังงานในการค้นเส้นทางของการกระจายข้อความร้องขอเส้นทางตามสมการที่ 3-11

$$\begin{aligned}
 E_q &= (E_1 + E_2 + E_3) + ME_4 + (M - m)(E_5 + E_6) \\
 &+ hE_7 + hE_8 + hE_9 + E_{10} \\
 &= (P_1 + P_2 + P_1) + MP_2 + (M - m)(P_3 + P_2) \\
 &+ hP_1 + hP_2 + hP_1 + P_2 \\
 &= (2 + 2h)P_1 + (h + 2M + 2 - m)P_2 + (M - m)P_3
 \end{aligned} \tag{3-11}$$

ส่วนพลังงานในการตอบกลับเส้นทาง (E_p) เป็นการส่ง ROUTE_REP กลับไปตามเส้นทางเดิมในการค้นหาสามารถใช้สมการใน (2-14) ได้

$$E_p = (P_2 + P_3) * (h + 1) \tag{3-12}$$

ดังนั้นพลังงานโดยรวมที่ใช้ในการค้นหาเส้นทางทั้งหมดหาได้จาก การนำเอาสมการในการค้นหาเส้นทางในการกระจายข้อความร้องขอเส้นทางในสมการที่ 3-11 รวมกับสมการการตอบรับเส้นทางตามสมการที่ 3-12 ก็จะได้สมการประมาณค่าพลังงานในการค้นหาเส้นทางทั้งหมดตามสมการที่ 3-13

$$\begin{aligned}
&= E_q + E_p \\
&= (2 + 2h)P_1 + (h + 2M + 2 - m)P_2 + (M - m)P_3 + P_3 \\
&\quad + (P_2 + P_3) * (h + 1) \\
&= (2 + 2h)P_1 + (2h + 2M - m + 3)P_2 + (M - m + h + 1)P_3
\end{aligned} \tag{3-13}$$

เมื่อทำการเปรียบเทียบกับสมการค้นหาเส้นทางใน Energy Aware Routing Protocol (EARP) for Ad-Hoc Wireless Sensor Networks [12] ซึ่งเป็นสมการเดียวกับโพรโทคอลหาเส้นทางในเครือข่ายไร้สาย ที่มีการกระจายข้อความร้องเส้นทางไปทั่วทั้งเครือข่าย รวมทั้งโพรโทคอล DSR ด้วย ดังนี้

$$\begin{aligned}
&= E_q + E_p \\
&= (P_1 + P_2) * (N - 1) + (P_2 + P_3) * (h + 1) \\
&= (N - 1)P_1 + (N + h)P_2 + (h + 1)P_3
\end{aligned} \tag{3-14}$$

เห็นได้ว่าสัมประสิทธิ์ของ P_1 ในสมการที่ (3-13) ซึ่งเป็นของโพรโทคอล RPRP น้อยกว่าในสมการที่ (3-14) เนื่องจาก h มีค่าน้อยกว่า N ส่วน P_2 นั้นในสมการที่ (3-13) จะมีค่ามากกว่าหรือน้อยกว่าในสมการที่ (3-14) ได้ เนื่องจากขึ้นอยู่กับ M ซึ่งถ้า M มีค่ามากนั้นก็หมายถึง ระยะทางระหว่างโหนดต้นทางกับปลายทางมีระยะทางที่ห่างกัน ทำให้ต้องมีโหนดระหว่างกลางมาก แต่ถ้าระยะทางระหว่างโหนดระหว่างกลางน้อย โหนดระหว่างกลางก็มีน้อยเช่นกัน ทำให้มีค่าน้อยกว่าได้เช่นกัน ส่วน P_3 ในสมการที่ (3-13) มีค่ามากกว่าเนื่องจากมีค่าของ M, m เพิ่มขึ้นมาจากการที่ต้องส่งผลการคัดเลือกโหนด และการส่งข้อความร้องขอการเชื่อมต่อ

ในบทนี้ได้แสดงถึงขั้นตอนการออกแบบโพรโทคอล RPRP อัลกอริทึมการค้นหาเส้นทาง อัลกอริทึมการบำรุงรักษาเส้นทาง ตัวอย่างการค้นหาเส้นทางและใช้เส้นทางสำรอง กล่าวถึงรูปแบบการทดลอง สภาพแวดล้อมการทดลอง

สุดท้ายเป็นการนำเสนอสมการประมาณค่าพลังงานในการค้นหาเส้นทางของโพรโทคอล RPRP ซึ่งผลที่ได้จะมีระดับการใช้พลังงานขึ้นอยู่กับจำนวนโหนดระหว่างกลาง หากมีโหนดระหว่างกลางน้อยก็หมายถึงเจอปลายทางได้เร็ว ก็จะทำให้เสียพลังงานในการค้นหาน้อย แต่หากโหนดระหว่างกลางมาก หรือเจอปลายทางช้าจะทำให้เสียพลังงานมาก

บทที่ 4

พารามิเตอร์และการวิเคราะห์ผลการทดลอง

ในบทนี้จะกล่าวถึงผลการทดลองโดยวัดจากพารามิเตอร์ดังต่อไปนี้ อัตราการใช้พลังงานเฉลี่ย อัตรารับส่งข้อมูลเฉลี่ย อัตราการค้นหาเส้นทางเฉลี่ย และการใช้เวลาในการสลับเส้นทางสำรองโดยเฉลี่ย รวมถึงการวิเคราะห์ผลการทดลองในแต่ละกรณี เพื่อที่จะแสดงให้เห็นถึงความสามารถของโพรโทคอล RPRP จากพารามิเตอร์ข้างต้น

จากการทดลองจำนวนโหนด 100 โหนด เวลาในการทดลอง 1000 วินาที โดยส่งข้อมูลแบบ CBR ซึ่งส่งข้อมูล 500 ไบต์ทุกๆ 0.2 วินาที ผลที่ได้จากการทดลองแบ่งออกเป็นสองส่วนที่นำมาวิเคราะห์คือ NAM Trace File และ NS Trace File โดยที่ NAM Trace File จะเป็นภาพกราฟฟิคในการแสดงการจำลองเครือข่าย ส่วน NS Trace File จะเป็นเหตุการณ์ทั้งหมดในการทดลอง ซึ่ง NS Trace File นั้นจะต้องเขียน Perl Script และ Shell Script เพื่อจะกรองข้อมูลให้ได้ตามที่ต้องการ จากนั้นนำข้อมูลดังกล่าวมาสร้างกราฟเพื่อหาประสิทธิภาพของโพรโทคอลต่างๆ ดังนี้

4.1 ค่า Throughput โดยเฉลี่ย (T_a)

หาได้จากจำนวนแพ็กเก็ตข้อมูลที่ได้รับโดยโหนดปลายทางคูณด้วย 8 (เพื่อทำให้เป็นบิต)หารด้วยเวลาในการทดลอง

$$T_a = (\text{จำนวนแพ็กเก็ตข้อมูลที่ได้รับโดยโหนดปลายทาง} * 8) / \text{เวลาในการทดลอง}$$

ตามตารางที่ 4-1 แสดงการเปรียบเทียบค่า Throughput กับโพรโทคอลต่างๆ โดยได้พิจารณาในส่วนแรกคือ ค่า Throughput ที่ดีที่สุด (Best Case) โพรโทคอล AODV มีค่า Throughput ที่ดีที่สุดคือ 320 Bit Per Second (bps) เป็นค่าที่ได้จากโหนดต้นทางที่อยู่ใกล้โหนดปลายทาง หรือโหนดปลายทางเป็นโหนดข้างเคียงของโหนดต้นทาง ทำให้โหนดต้นทางส่งข้อมูลได้โดยตรงไปยังโหนดปลายทาง ทำให้ค่า Throughput มีค่าสูงที่สุด ส่วนของโพรโทคอล DSR มีค่า Throughput คือ 345 bps เป็นค่าที่ได้จากกรณีที่โหนดต้นทางที่อยู่ใกล้กับโหนดปลายทางเช่นกัน แต่มีค่ามากกว่าโพรโทคอล AODV เนื่องจากโพรโทคอล DSR นั้น ในการค้นหาเส้นทางครั้งแรกจะให้ค่า TTL มี

ค่าเท่ากับศูนย์ ทำให้เจอบรรยากาศโดยที่ไม่ต้องกระจายข้อความร้องขอเส้นทางไปทั้งเครือข่าย ก็จะมีเวลาในการส่งข้อมูลได้มากขึ้นค่า Throughput ก็จะมีค่ามากกว่าโพรโทคอล AODV ส่วนโพรโทคอล RPRP และ RPRP (Single Path) จะค้นหาเส้นทางเป็นช่วงๆ ซึ่งเมื่อโหนดปลายทางเป็นโหนดข้างเคียง ก็จะเจอในการค้นหาเส้นทางรอบแรกจึงทำให้ค่า Throughput มีค่าเท่ากับโพรโทคอล DSR เช่นกัน

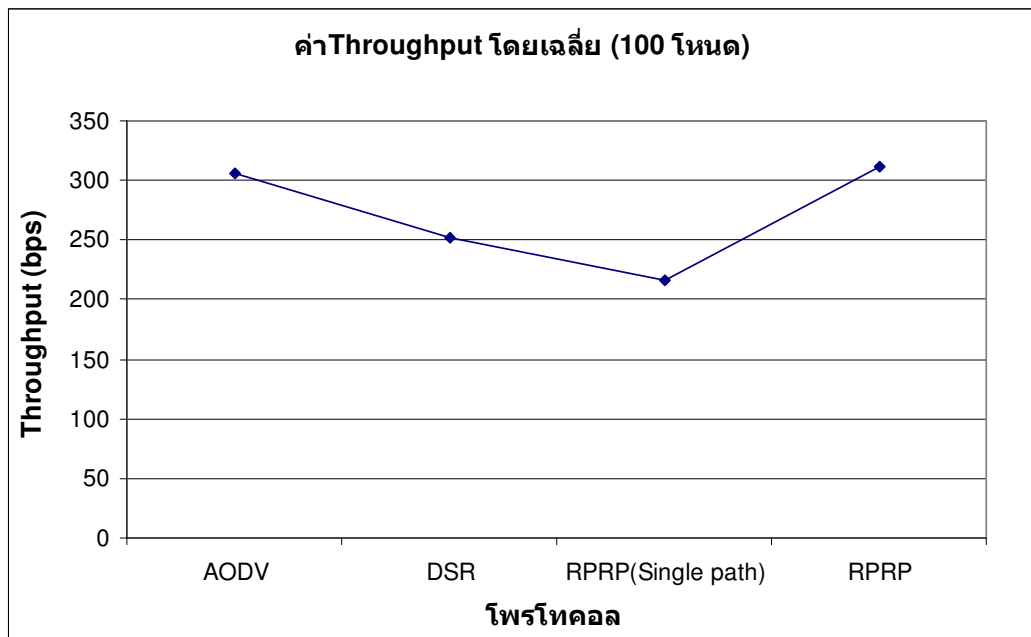
พิจารณาในส่วนที่สองคือค่า Throughput ที่มีค่าน้อยที่สุด (Worst Case) จะเป็นค่าที่ได้จากโหนดต้นทางที่อยู่ห่างไกลจากโหนดปลายทาง ซึ่งจะทำให้เสียเวลาในการค้นหาเส้นทางและเสียเวลาในการส่งข้อมูลเพิ่มขึ้น ทำให้ค่า Throughput ของแต่ละโพรโทคอลมีค่าน้อย

การเปรียบเทียบค่า Throughput (bps) โหนดในการทดลอง 100 โหนด			
โพรโทคอล	Best Case (bps)	Worst Case (bps)	ค่าเฉลี่ย (bps)
AODV	320	269	306
DSR	345	206	252
RPRP (Single Path)	345	134	216
RPRP	345	254	311

ตาราง 4-1 การเปรียบเทียบค่า Throughput

พิจารณาในส่วนของค่า Throughput โดยเฉลี่ย ตามภาพประกอบที่ 4-1 โพรโทคอล RPRP มี Throughput โดยเฉลี่ยสูงสุด เนื่องจากเมื่อส่งข้อมูลจนกระทั่งเส้นทางขาด ก็จะใช้เส้นทางสำรองทำให้ส่งข้อมูลได้ต่อเนื่อง ทำให้มีค่า Throughput ที่สูง ส่วนโพรโทคอล AODV นั้นได้มี Throughput ที่สูงเป็นลำดับถัดมา เนื่องจากเป็นโพรโทคอลที่มีการค้นหาเส้นทางที่ค่อยๆ เพิ่มค่า TTL ในข้อความร้องขอเส้นทางจนกว่าจะเจอบรรยากาศ ทำให้เมื่อเส้นทางขาด ก็จะใช้เส้นทางที่มีในโหนด หรือหากค้นหาเส้นทางใหม่ ก็ารู้ค่า TTL ที่จะส่งข้อความร้องขอเส้นทางทำให้ไม่เสียเวลาในการค้นหาเส้นทางมากนัก ซึ่งจะทำให้ค่า Throughput มีค่าที่สูง ส่วนในโพรโทคอล DSR มีค่า Throughput รองลงมาจากโพรโทคอล AODV เนื่องจากโพรโทคอล DSR นั้นการค้นหาเส้นทางจะกระจายข้อความร้องขอเส้นทางไปทั้งเครือข่าย เมื่อเส้นทางขาดโหนดต้นทางก็จะหาเส้นทางอื่นไปยังปลายทางที่มีในโหนด หากไม่มีก็จะค้นหาเส้นทางใหม่ โดยการกระจายข้อความร้องขอเส้นทางไปทั้งเครือข่าย ซึ่งทำให้เสียเวลามากกว่าโพรโทคอล AODV ที่กำหนด TTL ของข้อความร้องขอเส้นทางสุดท้าย โพรโทคอล RPRP (Single Path) ซึ่งเป็นโพรโทคอล RPRP ที่มีเพียงเส้นทางเดียวเท่านั้นมี

ค่า Throughput ที่น้อยที่สุด เนื่องจากมีเส้นทางเดียวไปยังปลายทาง เมื่อโหนดล้มเหลวก็ต้องค้นหาเส้นทางใหม่ ซึ่งเวลาในการค้นหาเส้นทางก็มีค่ามากกว่าโปรโตคอล AODV และ DSR ทำให้การส่งข้อมูลไม่ต่อเนื่องค่า Throughput จึงมีค่าน้อยที่สุด



ภาพประกอบ 4-1 การเปรียบเทียบค่า Throughput โดยเฉลี่ย

4.2 ค่าเฉลี่ยการใช้พลังงานโดยรวม (E_a)

หาได้จาก การนำผลรวมของค่าผลต่างของพลังงานเริ่มต้น (E_i) ลบด้วยพลังงานสิ้นสุด (E_f) หารด้วยจำนวนโหนดที่สูญเสียพลังงานงานทั้งหมด ซึ่งมีการกล่าวใน [10]

$$E_a = \frac{\sum_{k=1}^{k=n} (E_{ik} - E_{fk})}{N} \quad (4-1)$$

การเปรียบเทียบพลังงานที่ใช้ไปในการทดลอง 100 โหนด (จุด)			
โพรโทคอล	Best Case (จุด)	Worst Case (จุด)	ค่าเฉลี่ย (จุด)
AODV	12	117	86
DSR	11	105	80
RPRP (Single Path)	11	94	72
RPRP	11	118	93

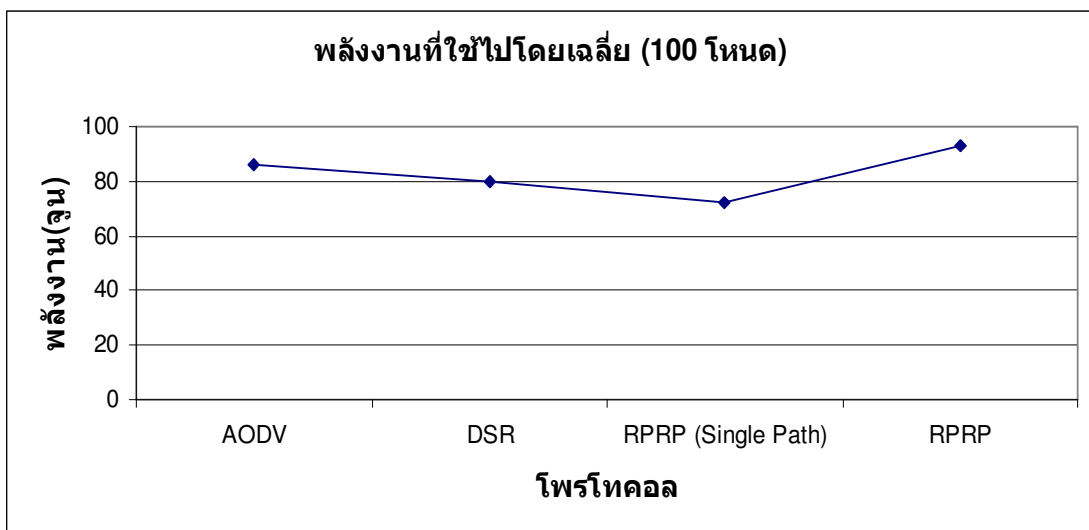
ตาราง 4-2 การเปรียบเทียบพลังงานที่ใช้ไปในการทดลอง

ตามตารางที่ 4-2 เป็นตารางการเปรียบเทียบพลังงานที่ใช้ไปในการทดลอง โดยจะพิจารณาในส่วนแรก คือพลังงานที่ใช้ไปน้อยที่สุด (Best Case) จะเกิดในกรณีของโหนดต้นทางที่อยู่ใกล้โหนดปลายทาง จะทำให้การค้นพบโหนดปลายทางได้เร็วกว่าโหนดต้นทางที่อยู่ห่างจากโหนดปลายทาง ในส่วนของโพรโทคอล AODV จะมีพลังงานที่ใช้ไปมากกว่าโพรโทคอลอื่น เนื่องจากถึงแม้โหนดต้นทางจะอยู่ใกล้โหนดปลายทาง แต่ก็จะต้องกระจายข้อความร้องขอเส้นทางตามค่า TTL ที่กำหนด แต่หากเป็นโพรโทคอล DSR ในครั้งแรกจะให้ค่า TTL มีค่าเท่ากับศูนย์ จึงใช้พลังงานน้อยกว่า และในส่วนของโพรโทคอล RPRP และ RPRP (Single Path) ดังที่ได้กล่าวไปแล้ว ในส่วนของค่า Throughput คือจะค้นหาเส้นทางเป็นช่วงๆ ในการค้นหาในช่วงแรกก็จะเจอโหนดปลายทางทันที โดยที่ไม่ต้องกระจายข้อความร้องขอเส้นทาง ทำให้ใช้พลังงานน้อยเช่นกัน

พิจารณาในส่วนที่สองคือ ส่วนของ Worst Case เป็นกรณีที่ใช้พลังงานมากที่สุดจะเกิดขึ้นในกรณีที่ โหนดต้นทางอยู่ห่างจากโหนดปลายทาง เนื่องจากต้องเสียเวลาในการค้นหาเส้นทาง ทำให้โหนดในเครือข่ายใช้พลังงานมาก ทำให้ภาพรวมการใช้พลังงานทั้งเครือข่ายมากด้วยเช่นกัน เป็นผลให้แต่ละโพรโทคอลใช้พลังงานมาก

ในส่วนที่สามคือ พลังงานที่ใช้ไปโดยเฉลี่ยตามภาพประกอบที่ 4-2 โพรโทคอล RPRP ใช้พลังงานมากที่สุด เนื่องจากใช้พลังงานในการส่งข้อมูลมากกว่าโพรโทคอลอื่น จะเห็นได้จากค่า Throughput ที่มีค่ามากที่สุด และพลังงานที่ใช้ในการค้นหาเส้นทางก็มีค่ามาก จะเห็นได้จากเวลาในการค้นหาเส้นทางโดยเฉลี่ยจาก ตารางที่ 4-4 ส่วนโพรโทคอล AODV จะใช้พลังงานสูงรองจากโพรโทคอล RPRP เนื่องจากใช้พลังงานในการส่งข้อมูลเหมือนกัน โดยจะเห็นได้จากค่า Throughput ที่มีค่ามากเป็นอันดับที่สอง ส่วนโพรโทคอล DSR มีการใช้พลังงานมากในลำดับถัดมา เนื่องจากใช้พลังงานในการส่งข้อมูลน้อยกว่าโพรโทคอล RPRP และ AODV ส่วนสุดท้าย โพรโทคอล RPRP (Single Path) จะพลังงานน้อยที่สุด เนื่องจากเหตุผลแรกคือ ใช้พลังงานในการส่งข้อมูลน้อยที่สุด

เมื่อดูจากค่า Throughput ที่มีค่าน้อยที่สุด และเนื่องจากโหนดต้นทางเสียพลังงานในการค้นหาเส้นทางสูงกว่าโพรโทคอลอื่นๆ เมื่อดูได้จากสมการประมาณค่าพลังงานของโพรโทคอล RPRP ฉะนั้นเมื่อเส้นทางขาดก็ต้องค้นหาเส้นทางใหม่ ทำให้โหนดต้นทางเสียพลังงานมากจนหมดพลังงาน ต่างกับโพรโทคอล RPRP ที่มีเส้นทางสำรองก็ไม่ต้องค้นหาเส้นทางหลายๆ ครั้ง และในทั้งสองเส้นทางก็มีโหนดที่ไม่ซ้ำกัน



ภาพประกอบ 4-2 พลังงานที่ใช้ไปในการทดลองโดยเฉลี่ย

4.3 อัตราส่วนเฉลี่ยการรับส่งข้อมูล (R_a)

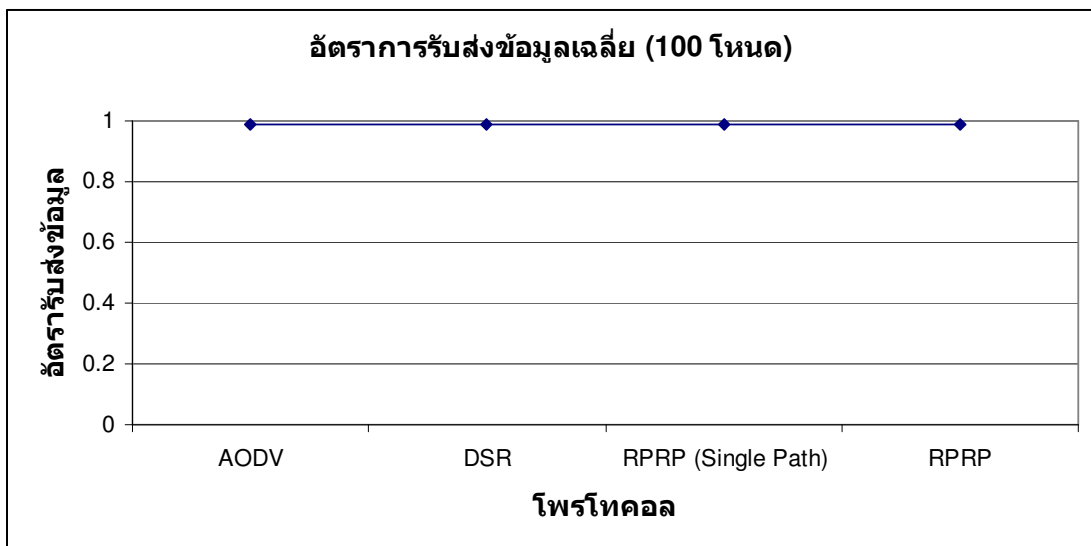
หาได้จากผลรวมจำนวนข้อมูลที่รับโดยปลายทางหารด้วยผลรวมจำนวนข้อมูลที่ส่งโดยต้นทาง ซึ่งมีกล่าวใน [10]

$$R_a = \text{ผลรวมจำนวนข้อมูลที่รับโดยปลายทาง} / \text{ผลรวมจำนวนข้อมูลที่ส่งโดยต้นทาง}$$

ตามตารางที่ 4-3 อัตราส่วนการรับส่งข้อมูลนั้นมีค่าใกล้เคียง 1.0 เนื่องจากการเชื่อมต่อเพียงคู่การสื่อสารเดียวในเครือข่าย ก็จะมีการทิ้งข้อมูลน้อยมาก หรือไม่มีการทิ้งข้อมูล ตามภาพประกอบที่ 4-3

การเปรียบเทียบอัตราส่วนการรับส่งข้อมูลในการทดลอง 100 โหนด			
โปรโตคอล	Best Case	Worst Case	ค่าเฉลี่ย
AODV	1.0	0.99	0.99
DSR	1.0	0.99	0.99
RPRP (Single Path)	1.0	0.99	0.99
RPRP	1.0	0.99	0.99

ตาราง 4-3 การเปรียบเทียบอัตราส่วนการรับส่งข้อมูล



ภาพประกอบ 4-3 การเปรียบเทียบอัตราส่วนการรับส่งข้อมูลโดยเฉลี่ย

4.4 เวลาในการค้นหาเส้นทางเฉลี่ย (RD_a)

หาได้จากผลรวมของเวลาที่สิ้นสุดการค้นหาเส้นทาง ลบด้วยผลรวมของเวลาที่เริ่มค้นหาเส้นทาง ซึ่งเวลาที่ได้อาจมีหน่วยเป็น Millisecond (ms)

$$RD_a = \text{ผลรวมของเวลาที่สิ้นสุดการค้นหาเส้นทาง} - \text{ผลรวมของเวลาที่เริ่มค้นหาเส้นทาง}$$

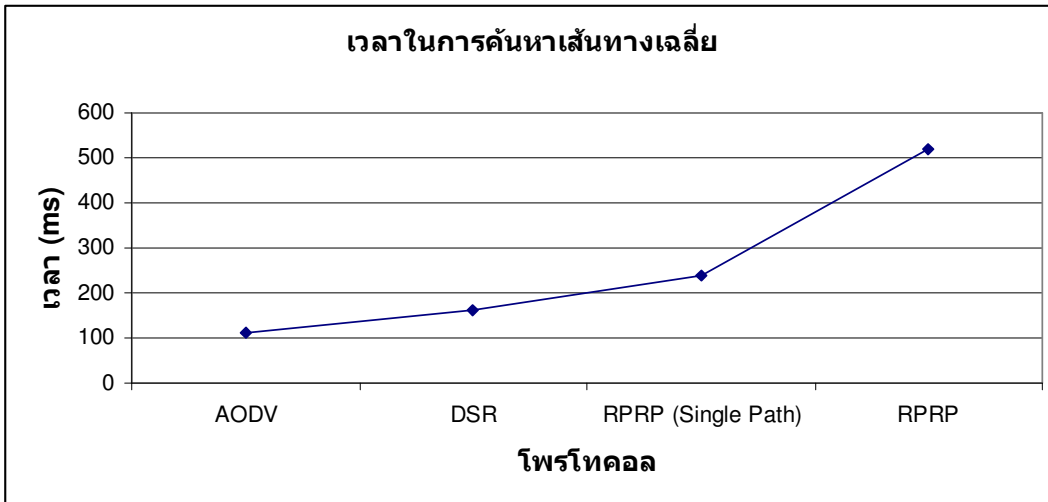
การเปรียบเทียบเวลาในการค้นหาเส้นทางในการทดลอง 100 โหนด (ms)			
โพรโทคอล	Best Case (ms)	Worst Case (ms)	ค่าเฉลี่ย (ms)
AODV	70	130	110
DSR	30	190	160
RPRP (Single Path)	30	420	240
RPRP	30	140	520

ตาราง 4-4 การเปรียบเทียบเวลาในการค้นหาเส้นทาง

ตามตาราง 4-4 เป็นการแสดงการเปรียบเทียบเวลาในการค้นหาเส้นทาง โดยพิจารณาในส่วนแรกคือ กรณี Best Case หรือเวลาในการค้นหาที่ดีที่สุด โดยมีสามโพรโทคอลที่ใช้เวลาในการค้นหาเส้นทางดีที่สุดคือ โพรโทคอล DSR, RPRP, RPRP (Single Path) ในส่วนของโพรโทคอล DSR ดังที่ได้กล่าวไปแล้ว ในการค้นหาเส้นทางคือ ในครั้งแรกจะกำหนดค่า TTL ให้เท่ากับศูนย์ทำให้โหนดปลายทางอยู่ใกล้ก็จะเจอทันที ส่วนโพรโทคอล RPRP และ RPRP (Single Path) ดังที่ได้กล่าวไปแล้วในการค้นหาเส้นทางคือ จะค้นหาเส้นทางเป็นช่วงๆ ไม่ได้กระจายไปทั้งเครือข่ายเมื่อโหนดปลายทางอยู่ใกล้ก็จะเจอทันทีเช่นกัน ส่วนโพรโทคอล AODV ถึงแม้โหนดปลายทางอยู่ใกล้แต่จะต้องกระจายข้อความร้องขอเส้นทางตามค่า TTL ที่กำหนดไว้ ทำให้เสียเวลามากกว่าโพรโทคอลอื่นๆ

ในส่วนของ Worst Case เป็นกรณีที่ใช้เวลาในการค้นหาเส้นทางมากที่สุด ก็จะเกิดในส่วนของโหนดต้นทางที่อยู่ห่างไกลจากโหนดปลายทาง ทำให้เสียเวลาในการค้นหาเส้นทาง

ตามภาพประกอบที่ 4-4 เป็นเปรียบเทียบเวลาในการค้นหาเส้นทางโดยเฉลี่ยโพรโทคอล RPRP มีเวลาในการค้นหาเส้นทางมากที่สุด เนื่องจากต้องค้นหาเส้นทางทีละช่วง และต้องหาเส้นทาง 2 เส้นทางด้วยกันทำให้เสียเวลามาก ส่วนโพรโทคอล RPRP (Single Path) นั้นใช้เวลาค้นหาเส้นทางโดยเฉลี่ยมากเป็นลำดับถัดมา ด้วยเหตุผลเดียวกันเพียงแต่ค้นหาเส้นทางเดียว ส่วนโพรโทคอล DSR ใช้เวลาในการค้นหาเส้นทางมากเป็นลำดับที่สาม เนื่องจากการค้นหาเส้นทางที่ส่งกระจายข้อความร้องขอเส้นทางไปทั่วทั้งเครือข่าย และมีเวลาค้นหาเส้นทางมากกว่าโพรโทคอล AODV เนื่องจากโพรโทคอล AODV นั้นค่อยๆ เพิ่มค่า TTL ในการกระจายข้อความร้องขอเส้นทางไม่กระจายไปทั้งเครือข่ายในครั้งเดียว ทำให้เวลาในการค้นหาเส้นทางดีที่สุด



ภาพประกอบ 4-4 การเปรียบเทียบเวลาในการค้นหาเส้นทางโดยเฉลี่ย

4.5 เวลาในการสลับใช้เส้นทางสำรองเฉลี่ย (ST_a)

หาได้จากผลรวมของผลต่างระหว่างเวลาในขณะเส้นทางขาด t_f ลบด้วยเวลาที่ใช้เส้นทางสำรอง t_s หารด้วยจำนวนครั้งที่สลับใช้เส้นทางสำรอง (N)

$ST_a = \text{เวลาในขณะเส้นทางขาด} - \text{เวลาที่ใช้เส้นทางสำรอง} / \text{จำนวนครั้งที่สลับใช้เส้นทางสำรอง}$

เวลาในการสลับใช้เส้นทางสำรองในการทดลอง 100 โหนด (ms)			
โพรโทคอล	Best Case (ms)	Worst Case (ms)	ค่าเฉลี่ย (ms)
RPRP	8	190	102

ตาราง 4-5 เวลาในการสลับใช้เส้นทางสำรอง

ตามตาราง 4-5 แสดงเวลาในการสลับใช้เส้นทางสำรองของโพรโทคอล RPRP โดยเป็นหาช่วงเวลาที่สลับใช้เส้นทางสำรองโดยนับตั้งแต่เวลาที่เส้นทางขาด และรองจนกว่าจะได้ใช้เส้นทางสำรอง กรณีของ Best Case คือเวลาในการสลับใช้เส้นทางสำรองที่ดีที่สุด จะเกิดขึ้นในกรณีที่เส้นทางขาดนั้นใกล้กับโหนดต้นทางที่ส่งข้อมูล ก็จะสลับใช้เส้นทางสำรองได้ทันที ในกรณีของ

Worst Case จะเกิดขึ้นในกรณีของโหนดที่อยู่ในเส้นทางที่ล้มเหลวอยู่ห่างจากโหนดต้นทาง ทำให้จะต้องเสียเวลาในการส่งข้อความ ROUTE_FAIL กลับไปยังโหนดต้นทาง ส่วนค่าเฉลี่ยเวลาในการสลับใช้เส้นทางรองคือ 102 ms

ในบทนี้แสดงถึง ผลการทดสอบโพรโทคอล โดยได้แสดงประสิทธิภาพของโพรโทคอลในด้านต่างๆ ดังนี้คือ เปรียบเทียบค่า Throughput, การใช้พลังงานทั้งหมดในเครือข่าย, อัตราการรับส่งข้อมูล, เวลาในการค้นหาเส้นทาง และเวลาในการสลับใช้เส้นทางสำรอง

โพรโทคอล RPRP เป็นโพรโทคอลที่เพิ่มความน่าเชื่อถือในการสื่อสารข้อมูลโดยเพิ่มเส้นทางสำรอง ซึ่งผลจากทดลองค่า Throughput ของโพรโทคอลมีค่ามากที่สุด และเส้นทางสำรองจะเป็นเส้นทางที่ไม่มีโหนดที่ซ้ำกับเส้นทางหลัก ดังนั้นหากเส้นทางหลักล้มเหลวก็ไม่กระทบกับเส้นทางสำรอง อีกทั้งโพรโทคอลยังออกแบบให้โหนดต้นทางสามารถเลือกใช้เส้นทางที่มีต้นทุนต่ำไปยังปลายทางเนื่องจากมีสองเส้นทางให้เลือก

บทที่ 5

บทสรุปและข้อเสนอแนะ

ในบทนี้จะกล่าวสรุปผลการวิจัยที่ได้ดำเนินการสำหรับวิทยานิพนธ์นี้ รวมทั้งข้อเสนอแนะต่างๆ ที่จะป็นประโยชน์ต่อการทำวิจัยด้าน โพรโทคอลในเครือข่ายตรวจจับไร้สายต่อไป

5.1 บทสรุป

วิทยานิพนธ์ฉบับนี้ได้นำเสนอโพรโทคอลค้นหาเส้นทางสำรองสำหรับเครือข่ายตรวจจับไร้สาย ซึ่งได้เสนออัลกอริทึมในการหาเส้นทางสำรอง เพื่อเพิ่มความน่าเชื่อถือในการสื่อสารข้อมูลในเครือข่ายตรวจจับไร้สาย อีกทั้งโหนดค้นหาสามารถเลือกใช้เส้นทางที่มีต้นทุนต่ำไปยังปลายทางได้จากการมีเส้นทางสำรอง

โดยเริ่มจากการออกแบบอัลกอริทึมในการหาเส้นทางสำรอง จากนั้นศึกษาโพรโทคอล DSR เพื่อเป็นข้อมูลในการพัฒนาสู่โพรโทคอลค้นหาเส้นทางสำรองสำหรับเครือข่ายตรวจจับไร้สาย และศึกษาโปรแกรมจำลองเครือข่าย NS-2 เพื่อนำมาทำการทดลอง

ซึ่งลำดับขั้นในการทดลองเริ่มจากการปรับแต่งโค้ดภาษาซีในโปรแกรม NS-2 หลังจากนั้นทำการ Make File ให้เป็น Object เพื่อให้สามารถนำไปใช้งานได้ ถัดไปก็เป็นการเขียน Tcl Script เพื่อกำหนดสภาพแวดล้อมในการทดลองต่างๆ แล้วทำการทดลอง ผลการทดลองออกมาสองแบบที่จะนำมาวิเคราะห์ คือ Nam Trace File และ NS Trace File ในการวิเคราะห์ข้อมูลจะต้องเขียน Perl Script และ Shell Script กรองข้อมูลตามที่ต้องการ สุดท้ายคือการเปรียบเทียบประสิทธิภาพกับโพรโทคอล DSR, AODV และ RPRP (Single Path)

สุดท้ายในส่วนผลการทดลองที่ได้ทดสอบประสิทธิภาพของโพรโทคอลมีดังนี้คือการเปรียบเทียบค่า Throughput, การใช้พลังงานทั้งหมดในเครือข่าย, อัตราการรับส่งข้อมูล, เวลาในการค้นหาเส้นทาง และเวลาในการสลับใช้เส้นทางสำรอง ซึ่งโพรโทคอล RPRP มีค่า Throughput เฉลี่ยสูงสุด และโพรโทคอล RPRP (Single Path) มีค่าน้อยที่สุด ส่วนอัตราการรับส่งข้อมูลเฉลี่ยมีค่าเท่ากันทุกโพรโทคอล เวลาในการค้นหาเส้นทางโดยเฉลี่ย โพรโทคอล RPRP มีเวลาในการค้นหาเส้นทางสูงสุด และโพรโทคอล AODV มีเวลาในการค้นหาเส้นทางโดยเฉลี่ยน้อยที่สุด ส่วนเวลา

ในการสลับใช้เส้นทางสำรองของโพรโทคอล RPRP นั้นเวลาขึ้นกับโหนดระหว่างกลางที่อยู่ในเส้นทางที่ล้มเหลว หากโหนดตัวกล่าวอยู่ใกล้โหนดต้นทาง ก็จะใช้เวลาน้อยในการสลับใช้เส้นทางสำรอง เพราะไม่ต้องเสียเวลาในการส่งข้อความ ROUTE_FAIL กลับไปยังโหนดต้นทาง แต่ถ้าหากโหนดที่ล้มเหลวอยู่ไกลจากโหนดต้นทางที่ส่งข้อมูล ก็เสียเวลาในการส่งข้อความ ROUTE_FAIL มากขึ้นเช่นกัน

5.2 ข้อเสนอแนะ

1. ในงานวิจัยนี้ยังมีส่วนที่ปรับปรุงหรือพัฒนาต่อคือ อัลกอริทึมที่จะต้องรองรับการทำงานได้กับหลายๆ รูปแบบของเครือข่าย และในส่วนของพลังงานที่ใช้ไปในเครือข่าย
2. นักวิจัยที่จะพัฒนาปรับปรุงโพรโทคอลจะต้องมีพื้นฐานความรู้คือ โปรแกรมจำลองเครือข่าย โพรโทคอลในเครือข่ายตรวจจับไร้สาย สมการประมาณค่าพลังงาน
3. อัลกอริทึม RPRP สามารถนำไปประยุกต์ใช้กับเครือข่ายตรวจจับไร้สายทางด้านการแพทย์ การตรวจจับแผ่นดินไหว เช่น กรณีที่มีเซนเซอร์โหนดจำนวนมาก อัลกอริทึมนี้สามารถสร้างเส้นทางสำรองได้ทันทีซึ่งจะเหมาะสมกับเหตุการณ์ฉุกเฉิน

เอกสารอ้างอิง

- [1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", In Proc. of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA-02), 2002.
- [2] Y. Gsottberger, X. Shi, G. Stromberg, T. F. Sturm, and W. Weber, "Embedding Low-Cost Wireless Sensors into Universal Plug and Play Environments". In Proc. Of EWSN, 2004.
- [3] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks", Ad Hoc Networks, vol. 3, no. 3, 2005.
- [4] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", In the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, August 2000.
- [5] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," In the Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, WA, August 1999.
- [6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks," In the Proceeding of the Hawaii International Conference System Sciences, Hawaii, January 2000.
- [7] A. Manjeshwar and D. P. Agrawal, "TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks," In the Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.
- [8] V. Rodoplu and T.H. Ming, "Minimum energy mobile wireless networks," IEEE Journal of Selected Areas in Communications, Vol. 17, No. 8, 1999.
- [9] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," In the Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01), Rome, Italy, July 2001.

- [10] R. Vidhyapriya and P.T. Vanathi, "Energy Aware Routing for Wireless Sensor Networks," In Proc. Of the IEEE-ICSCN 2007, MIT Campus, Anna University, Chennai, India. Feb.22-24, 2007.
- [11] L. Hu, Y. Li, Q. Chen, J. Liu and K. Long, "A New Energy-Aware Routing Protocol for Wireless Sensor Networks", Research Centre for Optical Internet & Wireless Information Networks, Chongqing University of Posts & Telecommunications, Chongqing, 2007.
- [12] R. P. Mann, K. Namuduri and R. Pendse, "Energy Aware Routing Protocol (EARP) for Ad-Hoc Wireless Sensor Networks", Wichita State University, Wichita, KS 67260, USA, 2004.
- [13] R. Shah and J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", In the Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Orlando, FL, March 2002.
- [14] L. He, "Delay-Minimum Energy-Aware Routing Protocol (DERP) for Wireless Sensor Networks", Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007.
- [15] A. M. Bernardos, P. Tarrío and J. R. Casar, "An Energy Aware Routing Algorithm For Ad hoc And Sensor Networks", The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2006.
- [16] T. Sriporamanont and G. Liming, "Wireless Sensor Network Simulator", School of Information Science, Computer and Electrical Engineering Halmstad University, Technical report, IDE0602, January 2006.
- [17] The Network Simulator –ns-2, Information Science Institute, USC Viterbi School of Engineering, <http://www.isi.edu/nsnam/ns>.
- [18] D. B. Johnson, D. A. Maltz and J. Broch "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.
- [19] C. Perkins et al, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561 AODV Routing, July 2003.
- [20] Energy Model update in ns-2, http://www.isi.edu/ilense/software/smac/ns2_energy.html.

ภาคผนวก

ภาคผนวก ก.

ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์



PEEC-7

การประชุมวิชาการทางวิศวกรรมศาสตร์ ครั้งที่ 7
The 7th PSU-Engineering Conference
21-22 พฤษภาคม 2552
ณ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

โปรโตคอลหาเส้นทางประหยัดพลังงานที่มีเส้นทางสำรอง สำหรับเครือข่ายตรวจจับไร้สาย Redundant path Energy Aware Routing Protocol (RERP) For Wireless Sensor Networks

วรรณรัช สันติอมรทัต¹ นิคม สุวรรณวร² วันประชา นวนสร้อย³

^{1,2,3}ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ อ.หาดใหญ่ จ.สงขลา 90112
E-mail: manyou66@hotmail.com

บทคัดย่อ

การประหยัดพลังงานในเครือข่ายตรวจจับไร้สายนั้นสำคัญมาก ทำให้มีผลต่อเวลาในการใช้เครือข่ายที่เพิ่มขึ้น ซึ่งโพรโทคอลหาเส้นทางที่ประหยัดพลังงานส่วนใหญ่มีเพียงเส้นทางเดียว ทำให้เมื่อเส้นทางล้มเหลวก็ต้องทำการสร้างเส้นทางใหม่ทำให้ต้องเสียเวลาในการสร้างเส้นทางใหม่รวมทั้งขาดความต่อเนื่องในการส่งข้อมูล บทความนี้ได้เสนอโพรโทคอลหาเส้นทางที่ประหยัดพลังงานที่มีเส้นทางสำรองเพื่อแก้ปัญหาดังกล่าว อัลกอริทึมในการหาเส้นทางสำรองที่คำนึงถึงพลังงานที่มีในโหนด โดยกำหนดพลังงานขั้นต่ำที่ทำให้มีการเชื่อมต่อ คุณภาพสัญญาณในการเชื่อม เมื่อเส้นทางหลักล้มเหลวสามารถใช้เส้นทางสำรองได้ ทำให้ไม่ต้องร้องขอเส้นทางใหม่ ซึ่งได้ทำการทดลองในโปรแกรมจำลองเครือข่าย ns-2 และมีการคิดสมการประมาณค่าพลังงานเมื่อเปรียบเทียบโพรโทคอล DSR มีอายุการใช้งานเครือข่ายมากกว่า

คำหลัก ประหยัดพลังงาน, เวลาใช้งานเครือข่าย, เส้นทางสำรอง

Abstract

It's a significant idea to use less energy in Wireless Sensor Networks (WSNs). Most of energy aware routing protocols are used with limitations using lesser energy directly increases working time and those route protocols are often working with a single main route without the minor route. In the second case, when the failure occurs the protocols start the whole process

again. This causes wasted time and inconsistency of information transfer. This paper presents an option of redundant path energy aware routing protocol (RERP)

for WSNs with the availability of minor route. The algorithm process of minor route creation is based on the available energy. With the preset minimum required energy and quality of connection, the protocol is able to continue with the minor route when the main route starts to fail. The simulation is made with network simulation 2. Compared against DSR protocol, the protocol uses less energy and this can lead to longer the lifetime of network.

Keywords: Less energy, Lifetime of network, Minor route

1. บทนำ

เครือข่ายตรวจจับไร้สาย เป็นเทคโนโลยีที่กระจายการตรวจจับซึ่งสามารถใช้ในการตรวจสอบปรากฏการณ์ทางกายภาพและสามารถที่จะกระจายได้ง่ายและยิ่งไปกว่านั้นสามารถประยุกต์ใช้งานได้หลากหลาย

เครือข่ายตัวตรวจจับไร้สายนี้ทำได้โดยการกระจายโหนดตรวจจับหลายๆโหนดซึ่งต้องการการหาเส้นทางแบบการส่งต่อเป็นช่วงๆ การจัดอันดับการประยุกต์ใช้เครือข่ายตรวจจับตามความสำคัญของปัญหาเป็นดังนี้ การตรวจจับสภาพแวดล้อมและแหล่งที่อยู่ การควบคุมจรรยาจรเหตุการณ์ฉุกเฉิน และการดูแลสุขภาพ [1, 2, 3]

โหนดตรวจจับใช้พลังงานจากแบตเตอรี่ ส่วนใหญ่

การประยุกต์ใช้เครือข่ายตรวจจับต้องการโหนดตรวจจับจำนวนมาก ซึ่งจะกระจายครอบคลุมพื้นที่ที่ต้องการตรวจจับจากการที่มีโหนดจำนวนมากทำให้การเปลี่ยนแปลงเตอริทำได้ยาก

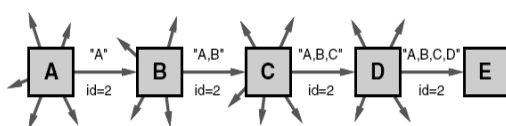
การประหยัดการใช้พลังงานเป็นสิ่งสำคัญมากสำหรับเครือข่ายตัวตรวจจับ แต่ปัญหาใหญ่ของการลดใช้พลังงานในโหนด คือ ยังไม่มีวิธีการแก้ปัญหาที่ดีพอ ถ้าทุกๆ โหนดส่งข้อมูลโดยตรงไปยังสถานีแม่ โหนดที่อยู่ไกลจากสถานีแม่ พลังงานจะหมดก่อน อีกวิธีหนึ่งคือ การส่งข้อมูลผ่านกันมาเป็นช่วงๆ ก็จะทำให้โหนดที่อยู่ใกล้สถานีแม่หมดพลังงานก่อน ทำให้เครือข่ายไม่สามารถส่งข้อมูลไปยังสถานีแม่ได้ การทำให้เวลาของเครือข่ายตรวจจับไร้สายมีอายุการใช้งานมากที่สุด ที่สำคัญคือการยืดอายุการใช้งานของแต่ละโหนดตรวจจับโดยการลดการใช้พลังงานในการส่งข้อมูล และการส่งข้อมูลผ่านเส้นทางที่สามารถหลีกเลี่ยงโหนดตรวจจับที่มีพลังงานต่ำ

โพรโทคอลการหาเส้นทางที่ประหยัดพลังงานนั้นส่วนใหญ่จะสร้างเส้นทางเดียวในการสื่อสารเพื่อเป็นการประหยัดพลังงานแต่จะเกิดปัญหาเมื่อเส้นทางนั้นล้มเหลว ก็จะต้องสร้างเส้นทางใหม่ไม่สามารถใช้เส้นทางอื่นได้

ในบทความนี้ ในส่วนที่สอง จะเป็นการเสนอโพรโทคอลการหาเส้นทางที่ประหยัดพลังงานที่มีเส้นทางสำรอง โดยจะกล่าวถึงอัลกอริทึมการเลือกเส้นทางสำรอง ส่วนที่สามเป็นส่วนของการหาสมการพลังงานในการค้นหาเส้นทางของโพรโทคอลที่เสนอ และสุดท้ายก็จะเป็นบทสรุป และงานที่จะทำในอนาคต

1.1 โพรโทคอล DSR (Dynamic Source Routing Protocol)

เป็นโพรโทคอลหาเส้นทางในเครือข่ายไร้สายทั่วไปมีกล่าวใน [4] ที่ต้นทางต้องมีส่วนของเส้นทางที่สมบูรณ์อยู่ใน packet header ก่อนที่จะส่งข้อมูล โพรโทคอล DSR เป็นโพรโทคอลที่ประกอบด้วยสองกลไกหลักคือ Route Discovery และ Route Maintenance ส่วนของ Route Discovery เป็นกลไกในการหาเส้นทางไปยังปลายทาง ส่วนของ Route Maintenance เป็นกลไกไว้สำหรับสร้างเส้นทางใหม่อีกครั้ง



รูปที่ 1 แสดงการค้นหาเส้นทางใน DSR

ในรูปที่ 1 เป็นการค้นหาเส้นทางในโพรโทคอล DSR

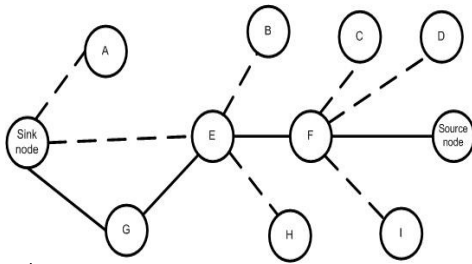
โหนด A ต้องการติดต่อกับโหนด E ก็จะกระจายข้อความร้องขอเส้นทางออกไปทั่วเครือข่ายโดยมีหมายเลขการค้นหา คือ 2 ซึ่งโหนดที่รับข้อความร้องขอเส้นทางจะตรวจสอบว่าเคยได้รับข้อความหมายเลขนี้หรือไม่ และชื่อตัวเองอยู่ในหัวของข้อความหรือไม่ ถ้าเป็นทั้งสองกรณีข้างต้นจะทิ้งข้อความนี้ แต่หากไม่เกิดทั้งสองกรณีข้างต้นก็ใส่ชื่อโหนดของตัวเองในหัวของข้อความแล้วส่งต่อไปจนกว่าจะถึงปลายทาง เมื่อถึงโหนดปลายทางก็จะส่งข้อความตอบรับเส้นทางกลับไปยังโหนดปลายทางตามรายชื่อโหนดที่อยู่ในหัวของข้อความตามเส้นทางที่มา

1.2 โพรโทคอลหาเส้นทางที่ประหยัดพลังงานในเครือข่ายตรวจจับไร้สาย

โพรโทคอลรูปแบบนี้มีมุ่งหมายที่จะประหยัดพลังงานที่มีในโหนดตรวจจับให้ได้มากที่สุดเพื่อเพิ่มเวลาในการใช้เครือข่ายตรวจจับให้นานที่สุด จึงออกแบบให้มีการหาเส้นทางเดียวไปยังปลายทางซึ่งต่างจากโพรโทคอลปกติที่จะกระจายการหาเส้นทางโดยจะมีหลายเส้นทางไปยังปลายทาง ซึ่งเมื่อลดการหาเส้นทางให้เหลือเพียงเส้นทางเดียวก็จะทำให้ประหยัดพลังงานได้มากกว่าโพรโทคอลปกติ ตัวอย่างของโพรโทคอลนี้ เช่น Energy Aware Routing for Wireless Sensor Networks [5]

ซึ่ง Algorithm ประกอบด้วยสามส่วนคือ Route Discovery, Route Reply และ Reliable Transmission โดยใช้สองข้อความคือ ข้อความร้องขอเส้นทาง และ ข้อความตอบรับของเส้นทาง เมื่อโหนดต้องการติดต่อสื่อสาร โหนดก็จะมีกลไกในการส่งข้อความการค้นหาเส้นทาง ไปยังปลายทาง ซึ่งจะมีเพียงเส้นทางเดียวไปยังปลายทาง แต่ยังมีปัญหาในเรื่องของเวลาในการหาเส้นทางเนื่องจากจะเลือกเชื่อมต่อกับโหนดที่มีพลังงานสูงและคุณภาพการเชื่อมต่อที่ดีทำให้เสียเวลาเพิ่มขึ้นในการเป็นถึงโหนดปลายทาง และการมีเพียงเส้นทางเดียวก็ทำให้ต้องสร้างเส้นทางใหม่หาเส้นทางล้มเหลว ตัวอย่างการหาเส้นทางดังรูปที่ 1

รูปที่ 1 เริ่มต้นจะมีการกระจายข้อความออกไป โหนด A, E และ รับข้อความ สมมุติว่าโหนด G มีพลังงานมากกว่าโหนด E และ A และ โหนด G ก็จะได้รับ การเชื่อมต่อ และโหนด G ก็จะกระจายข้อความร้องขอเส้นทางไปยังโหนดข้างเคียงต่อไป โหนด E ถูกเลือกหลังจากโหนด G โหนด E ส่งข้อความร้องขอเส้นทางออกมา ซึ่งโหนดที่รับมี โหนด B, F และ H โหนด H ตรวจสอบ



รูปที่ 1 แสดงการหาเส้นทางใน Energy Aware Routing

ลำดับหมายเลขแล้วก็ปล่อยแพ็คก็เกิดขึ้นไป เพราะพบทั้ง โหนด B และ F มีระดับพลังงานเท่ากัน และมีคุณภาพการเชื่อมต่อที่เท่ากันด้วย ดังนั้นทั้ง B และ F จะเริ่มตั้งเวลานับถอยหลังและ ถ้าเวลาที่นับถอยหลังของโหนด F จบก่อนโหนด B ก็ถือว่า ACK จะส่งโดยโหนด F ไปยังโหนด E โหนด E ส่ง ACK ไปยังโหนด B เพื่อบอกว่าการเลือกโหนดที่จะส่งเสร็จสิ้นแล้ว ดังนั้นโหนด B ก็จะหยุดการนับเวลาถอยหลัง นี่เป็นขั้นตอนการทำงานและเมื่อข้อความการร้องขอเส้นทางมาถึงเป้าหมายปลายทาง ปลายทางก็จะส่งแพ็คก็เกิดตอบรับผ่านเส้นทางเดิมที่มาซึ่งก็คือ F-E-G

2. โพรโทคอลหาเส้นทางสำรองที่ประหยัดพลังงานสำหรับเครือข่ายตรวจจับไร้สาย

โพรโทคอลการหาเส้นทางที่ประหยัดพลังงานนั้นส่วนใหญ่จะสร้างเส้นทางเดียวในการสื่อสารเพื่อเป็นการประหยัดพลังงานแต่จะเกิดปัญหาเมื่อเส้นทางนั้นล้นแล้ว ก็จะต้องสร้างเส้นทางใหม่ไม่สามารถใช้เส้นทางอื่นได้ ซึ่งโพรโทคอลนี้ได้มีส่วนของเส้นทางสำรองเพื่อให้สามารถใช้เส้นทางได้ทันทีหลังจากเส้นทางหลักล้นแล้ว และจากการที่กำหนดพลังงานขั้นต่ำในการเชื่อมต่อ สามารถกระจายการใช้งานพลังงานในเครือข่ายได้ โดยมีสถานะโหนดดังนี้

- F สถานะว่าง
- W สถานะทำงาน
- C สถานะเชื่อมต่อ

ในตอนเริ่มต้นโหนดในเครือข่ายจะมีสถานะว่าง ยกเว้นโหนดต้นทางที่ต้องการติดต่อกับโหนดปลายทางจะมีสถานะเชื่อมต่อ เมื่อโหนดต้นทางกระจายข้อความร้องขอเส้นทางโหนดที่ได้รับข้อความร้องขอเส้นทางจะเปลี่ยนสถานะเป็น W จนกว่าจะได้รับข้อความผลการคัดเลือกโหนด หากได้รับการคัดเลือกจะเปลี่ยนสถานะเป็น C แต่หากไม่ได้รับการคัดเลือกก็จะเปลี่ยนสถานะเป็น F เช่นเดิมกำหนดให้

- พลังงานขั้นต่ำ ET (Energy Threshold) เป็นค่าคงที่

- ข้อความร้องขอเส้นทาง (ROUTE_REQ)
- ข้อความตอบรับการหาเส้นทาง (ROUTE_REP)
- ข้อความร้องขอการคัดเลือกการเชื่อมต่อ (RESULT REQ)
- ข้อความผลการคัดเลือกการเชื่อมต่อ (RESULT REP)
- ข้อความแจ้งการกำลังถูกใช้งานอยู่ (USE_REP)
- ข้อความยกเลิกระดับพลังงานขั้นต่ำ (CANCEL_ET)

ตารางที่ 1 อัลกอริทึมการหาเส้นทาง

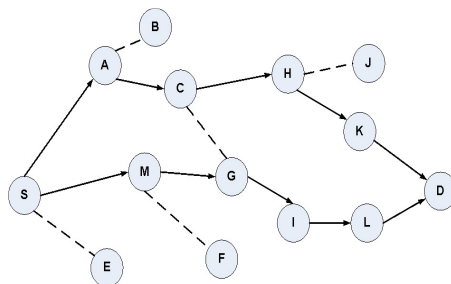
โหนด	ขั้นตอนการทำงาน
Source node	<ul style="list-style-type: none"> • กระจายข้อความ ROUTE_REQ • รอจนกว่าข้อความ ROUTE_REP แรก มาถึง • เลือกโหนดที่ส่งข้อความ RESULT REQ มาถึงก่อนสองโหนด ถ้ามีโหนดที่ส่งข้อความ RESULT REQ มาถึงพร้อมกันมากกว่า 2 โหนดขึ้นไป ก็สุ่มเลือกโหนดใดโหนดหนึ่งถ้ามีโหนดที่ส่งข้อความ RESULT REQ กลับมาเพียงหนึ่งโหนด ก็เลือกเพียงแค่นั้นเดียว • ส่งข้อความ RESULT REP แจ้งผลการคัดเลือกโหนด • เลือกเส้นทางที่บรรจุในข้อความ ROUTE_REP
Intermediate node	<ul style="list-style-type: none"> • โหนดที่มีพลังงานน้อยกว่า ET หักข้อความ ROUTE_REQ • เมื่อได้รับข้อความ RESULT REP ถ้าไม่ได้รับการคัดเลือกให้ทั้งข้อความ ROUTE_REQ • เลือกโหนดที่ส่งข้อความ RESULT REQ มาถึงก่อนหนึ่งโหนด ถ้ามีโหนดที่ส่งข้อความ RESULT REQ มาถึงพร้อมกันมากกว่า 1 โหนดขึ้นไป ก็สุ่มเลือกโหนดใดโหนดหนึ่ง • ถ้าไม่มีโหนดใดตอบ RESULT REQ กลับมาก็ให้โหนดส่งข้อความ CANCEL_ET เพื่อยกเลิกระดับพลังงานขั้นต่ำ • หลีกเลี่ยงที่จะเลือกโหนดที่ใช้ในเส้นทางอื่น โดยโหนดที่ใช้ในเส้นทาง

	อื่นๆจะส่งข้อความ USE_REP เพื่อเป็นการบอกว่าโหนดดังกล่าวถูกใช้อยู่ในเส้นทางอื่น แต่ถ้าไม่มีโหนดอื่นให้เลือกจึงจะเลือกโหนดที่ถูกใช้ในเส้นทางอื่น
Destination node	<ul style="list-style-type: none"> เมื่อรับข้อความ ROUTE_REQ ข้อความแรก และหลังจากนั้นส่งข้อความ ROUTE_REP ไปยังโหนดต้นทางตามเส้นทางเดิม

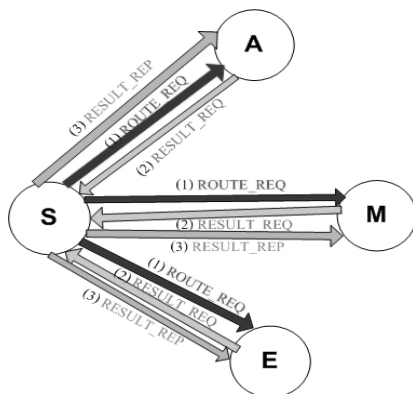
ตารางที่ 2 อัลกอริทึมการรักษาเส้นทาง

โหนด	ขั้นตอนการทำงาน
Source node	<ul style="list-style-type: none"> ถ้าเส้นทางหลักล้มเหลวก็เลือกเส้นทางสำรองทันที ถ้าเส้นทางสำรองล้มเหลวอีก ก็เลือกเส้นทางใหม่ที่เก็บไว้ในโหนด หรือค้นหาเส้นทางใหม่ โหนดต้นทางจะไม่สนใจว่าจะหมดพลังงานหรือไม่

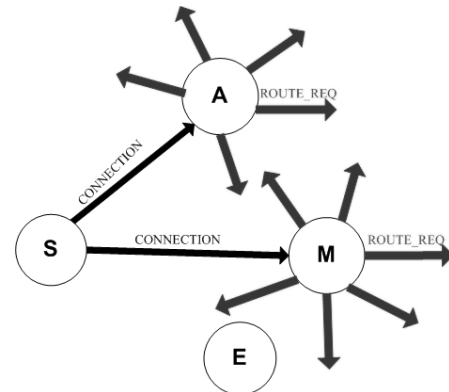
โพรโทคอลหาเส้นทางที่ประหยัดพลังงานที่มีเส้นทางสำรอง พัฒนาจากโพรโทคอล DSR รายละเอียดดังที่ได้กล่าวในข้างต้น เหตุผลที่เลือกโพรโทคอล DSR มาพัฒนานั้น เพราะ โพรโทคอล DSR นั้นง่าย และสามารถเปรียบเทียบประสิทธิภาพกับโพรโทคอลอื่นได้ง่าย



รูปที่ 2 แสดงการค้นหาเส้นทางใน RERP



รูปที่ 3 แสดงการกระบวนการคัดเลือกโหนดในการเชื่อมต่อ



รูปที่ 4 แสดงโหนดที่ได้รับการคัดเลือกให้เชื่อมต่อ

จากรูปที่ 2 โหนด S ซึ่งเป็นโหนดต้นทาง ต้องการติดต่อกับโหนด D ในรูปที่ 3 โหนด S มีสถานะเป็น C ก็จะกระจายข้อความ ROUTE_REQ ออกไป โหนด A, M และ E มีพลังงานสูงกว่าพลังงานขั้นต่ำเมื่อได้รับข้อความ โหนด A, M และ E จะเปลี่ยนสถานะเป็น W และส่งข้อความ RESULT REQ กลับไปยังโหนด S โดยที่โหนด A ส่งมาถึงเร็วที่สุด ตามด้วยโหนด M และ E โหนด S ก็ประกาศข้อความ RESULT REP ผลการคัดเลือกสองเส้นทางโหนดคือ โหนด A และ M ทั้งสองโหนดก็จะเปลี่ยนสถานะเป็น C ส่วนโหนด E ก็จะเปลี่ยนสถานะเป็น F เช่นเดิม

ในรูปที่ 4 โหนด A และ M ก็ได้รับการเชื่อมต่อ โหนด A และ M ก็กระจายข้อความ ROUTE_REQ ต่อไป แล้วจะทำตามขั้นตอนการคัดเลือกโหนดตามข้างต้น แต่โหนดระหว่างทางนั้นจะเลือกโหนดเชื่อมต่อเพียงเส้นทางเดียว ทำอย่างนั้นจนกระทั่งถึงปลายทาง โดยเส้นทางใดถึงปลายทางก่อนเส้นทางนั้นจะเป็นเส้นทางหลัก เมื่อถึงปลายทางก็จะตอบกลับด้วยข้อความ ROUTE_REP

3. ผลการหาสมการพลังงานในการค้นหาเส้นทาง และการทดลอง

การประมาณค่าพลังงานที่ใช้ในการค้นหาเส้นทางมีกรกล่าวใน Energy Aware Routing Protocol (EARP) for Ad-Hoc Wireless Sensor Networks [6] พลังงานที่โหนดใช้การกระจายการร้องขอเส้นทาง (RREQ)

$$E_1 = [\alpha_{i1} + \alpha_{i2}d^4] * B * t = P_1 \tag{1}$$

พลังงานที่โหนดใช้รับข้อความร้องขอเส้นทาง (RREQ)

$$E_2 = \alpha_r * B * t_c = P_2 \tag{2}$$

พลังงานที่โหนดส่งข้อความตอบรับเส้นทาง (RREP)

$$E_5 = [\alpha_{i1} + \alpha_{i2}d^4] * B * t_c = P_3 \tag{3}$$

ซึ่งพลังงานทั้งหมดในการค้นหาเส้นทางนั้น ได้มาจากผลรวมของพลังงานที่ใช้ในการส่งข้อความร้องขอเส้นทางและการตอบรับเส้นทาง

$$E_t = E_q + E_p \quad (4)$$

โดยที่

α_{i1}, α_{i2} เป็นค่าคงที่ในการสื่อสารโดยที่ $\alpha_{i1} = 45n$ J/b,

$$\alpha_{i2} = 10p$$
 J/b

d^4 ค่าเฉลี่ยระหว่างโหนดข้างเคียง 2 โหนด และเป็น 4 สำหรับการคำนวณทั้งหมด

B เป็นขนาดช่องทางการเชื่อมโดยรวมมีหน่วยเป็น Kbps

t เป็นขนาดของเฟรมโดยรวมในแต่ละวินาที

α_r เป็นค่าคงที่เท่ากับ 135n J/b

t_c เป็นเวลาของหนึ่ง time slot ในแต่ละวินาที

h เป็นจำนวนช่วงในการส่งต่อ

N เป็นจำนวนโหนดทั้งหมดในเครือข่าย

3.1 การประมาณค่าพลังงานที่ใช้ในการหาเส้นทางใน Redundant path Energy Aware Routing Protocol (RERP) for Wireless Sensor Networks

โดยจะแบ่งออกเป็นสองส่วนหลักคือ พลังงานที่ใช้ในการกระจายการร้องขอเส้นทาง(RREQ) และพลังงานที่ใช้ในการตอบกลับเส้นทาง ซึ่งในส่วนของ พลังงานที่ใช้ในการกระจายการร้องขอเส้นทาง(RREQ) จะคิดแยกออกเป็นสามส่วนด้วยกันคือ โหนดต้นทาง(S) โหนดระหว่างกลาง และ โหนดปลายทาง(D) ดังนี้

ก. โหนดต้นทาง (S)

- พลังงานที่โหนด S ใช้กระจาย ROUTE_REQ

$$E_1 = [\alpha_{i1} + \alpha_{i2}d^4] * B * t = P_1 \quad (5)$$

- พลังงานที่โหนด S ใ้รับ RESULT_REQ

$$E_2 = \alpha_r * B * t_c = P_2 \quad (6)$$

- พลังงานที่โหนด S ใช้ส่ง RESULT_REQ

$$E_3 = P_1 \quad (7)$$

ข. โหนดระหว่างกลาง

- พลังงานที่โหนดข้างเคียง ใ้รับ ROUTE_REQ

$$E_4 = P_2 = E_{10} \quad (8)$$

- พลังงานที่โหนดข้างเคียง ใ้ส่ง RESULT_REQ

$$E_5 = [\alpha_{i1} + \alpha_{i2}d^4] * B * t_c = P_3 = E_{11} \quad (9)$$

- พลังงานที่โหนดข้างเคียง ใ้รับ RESULT_REQ

$$E_6 = P_2 = E_{12} \quad (10)$$

- พลังงานที่โหนดข้างเคียงที่ใ้รับการคัดเลือก ใ้ส่ง ROUTE_REQ

$$E_7 = P_1 \quad (11)$$

- พลังงานที่โหนดข้างเคียงที่ใ้รับการคัดเลือก ใ้รับ RESULT_REQ

$$E_8 = P_2 \quad (12)$$

- พลังงานที่โหนดข้างเคียงที่ใ้รับการคัดเลือก ใ้ส่ง RESULT_REQ

$$E_9 = P_1 \quad (13)$$

ค. โหนดปลายทาง(D)

- พลังงานที่โหนด D ใ้รับ ROUTE_REQ

$$E_{10} = P_2 \quad (14)$$

พลังงานทั้งหมดที่ใช้ในการส่ง RREQ จากโหนดต้นทาง(S) ไปยังโหนดปลายทาง (D)

$$\begin{aligned} E_q &= (E_1 + E_2 + E_3) + ME_4 + (M-m)(E_5 + E_6) + 2hE_7 + 2hE_8 + 2hE_9 + E_{10} \\ &= (P_1 + P_2 + P_1) + MP_2 + (M-m)(P_3 + P_2) + 2hP_1 + 2hP_2 + 2hP_1 + P_2 \\ &= (2+4h)P_1 + (2h+2M+2-m)P_2 + (M-m)P_3 \end{aligned} \quad (15)$$

ส่วนพลังงานในการตอบกลับเส้นทาง จะเป็นการใช้เส้นทางเดิมในการค้นหาสามารถใช้สมการใน [2] ได้

$$\begin{aligned} E_p &= E_5 + (E_6 + E_7) * (h) + E_8 \\ &= P_3 + (P_2 + P_3) * h + P_2 \end{aligned} \quad (16)$$

$$= (P_2 + P_3) * (h + 1)$$

ดังนั้นพลังงานโดยรวมที่ใช้ในการค้นหาเส้นทาง

$$\begin{aligned} &= E_q + E_p \\ &= (2+4h)P_1 + (2h+2M+2-m)P_2 + (M-m)P_3 + P_3 + (P_2 + P_3) * h + P_2 \\ &= (2+4h)P_1 + (3h+2M-m+3)P_2 + (M-m+h+1)P_3 \end{aligned} \quad (17)$$

โดยที่

M เป็นจำนวนโหนดระหว่างกลางซึ่งมีค่า $\leq N$ ขึ้นอยู่กับ

หาพบโหนดปลายทางเร็วหรือช้า

m เป็นจำนวนโหนดระหว่างกลางที่ใ้รับการคัดเลือก

ซึ่งเมื่อเปรียบเทียบกับสมการค้นหาเส้นทางใน Energy Aware Routing Protocol (EARP) for Ad-Hoc Wireless Sensor Networks ซึ่งเป็นสมการเดียวกับโปรโตคอลหาเส้นทางในเครือข่ายไร้สายที่ต้องการหาเส้นทางแบบ ad-hoc routing ทั่วๆ ไปรวมทั้งโปรโตคอล DSRด้วย ดังนี้

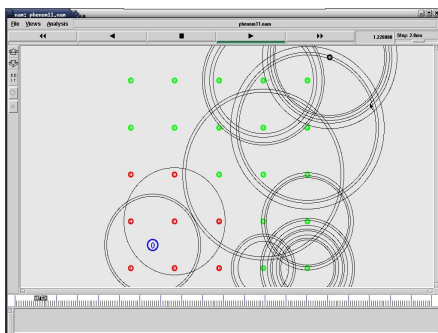
$$\begin{aligned} &= E_q + E_p \\ &= (P_1 + P_2) * (N - 1) + (P_2 + P_3) * (h + 1) \quad (18) \\ &= (N - 1)P_1 + (N + h)P_2 + (h + 1)P_3 \end{aligned}$$

จะเห็นได้ว่าสัมประสิทธิ์ของ P_1 และ P_2 ในสมการที่ 17 น้อยกว่าในสมการที่ 18 เนื่องจาก h, M, m มีค่าน้อยกว่า N ส่วนสัมประสิทธิ์ของ P_3 มีค่ามากกว่าเล็กน้อยเนื่องจากมีค่าของ $M-m$ เพิ่มขึ้น

จากสมการจึงสรุปได้ว่าความน่าเชื่อถือเพิ่มขึ้นจากการที่มีเส้นทางสำรองเมื่อเทียบกับโพรโทคอลประหยัดพลังงานที่มีเพียงเส้นทาง ส่วนในเรื่องของพลังงานก็สามารถประหยัดพลังงานกว่าโพรโทคอล DSR

3.2 การทดลอง

ในการทดลองนั้นจะพัฒนาอยู่บนพื้นฐานของโพรโทคอล DSR และทำการออกแบบและพัฒนาเพิ่มในส่วนของการหาเส้นทางสำรอง กำหนดพลังงานขั้นต่ำ สถานะโหนด และการสลับเส้นทางระหว่างเส้นทางหลัก และเส้นทางสำรอง กำหนดให้มีโหนดในการทดลองแบ่งเป็น 25, 50 และ 100 โหนด ตามลำดับ ทำการจำลองในโปรแกรม NS2 [7] ดังในรูปที่ 5 เพื่อหาระยะเวลาใช้งานของเครือข่ายระยะเวลาในการค้นหาเส้นทาง พลังงานที่มีเหลือในโหนดโดยเฉลี่ย จากนั้นเปรียบเทียบกับ DSR, AODV, Energy Aware Routing Protocol



รูปที่ 5 แสดงการจำลองเครือข่ายตรวจจับไร้สายใน ns-2

3. สรุป

บทความนี้ได้เสนอโพรโทคอลหาเส้นทางที่ประหยัดพลังงานในเชิงที่จะเพิ่มคุณภาพในการส่งข้อมูลกว่าโพรโทคอลประหยัดพลังงานเดิม จากการที่มีเส้นทางสำรองขึ้นมาในกรณีของเส้นทางล้มเหลวก็สามารถใช้เส้นทางสำรองได้ เพื่อลดการสร้างเส้นทางใหม่ และจากการที่ระดับพลังงานขั้นต่ำในการทำงานทำให้สามารถกระจายการใช้พลังงานให้เพิ่มมากขึ้น และจากสมการประมาณค่าพลังงานแสดงให้เห็นว่าสามารถประหยัดพลังงานกว่าโพรโทคอลหาเส้นทาง DSR

ในอนาคตของโพรโทคอลหาเส้นทางในเครือข่ายตรวจจับไร้สายคงเน้นในเรื่องของการจัดการพลังงานที่มีใน

โหนดเพื่อเพิ่มระยะเวลาการใช้เครือข่ายให้สูงขึ้น และการเพิ่มคุณภาพการให้บริการ

เอกสารอ้างอิง

- [1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", In Proc. of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA-02), 2002
- [2] Y. Gsofftberger, X. Shi, G. Stromberg, T. F. Sturm, and W. Weber, "Embedding Low-Cost Wireless Sensors into Universal Plug and Play Environments". In Proc. Of EWSN, 2004
- [3] K.A. Delin, "Sensor Web in Antarctica - Developing an Intelligent, Autonomous Platform for Locating Biological Flourishes in Cryogenic Environments". In Proc. of the 34th Lunar and Planetary Science Conf (LPSC 03). 2003
- [4] David B. Johnson, David A. Maltz and Josh Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15313-3891, <http://www.monarch.cs.cmu.edu/>
- [5] R. Vidhyapriya and P.T. Vanathi, "Energy Aware Routing for Wireless Sensor Networks," In Proc. Of the IEEE-ICSCN 2007, MIT Campus, Anna University, Chennai, India. Feb.22-24, 2007. pp.545-550.
- [6] Raminder P. Mann, Kamesh Namuduri and Ravi Pendse, "Energy Aware Routing Protocol (EARP) for Ad-Hoc Wireless Sensor Networks", Wichita State University, KS 67260, USA.
- [7] The Network Simulator –ns-2, Information Science Institute, USC Viterbi School of Engineering, <http://www.isi.edu/nsnam/ns>



International Conference on Embedded Systems
and Intelligent Technology

Redundant Path Routing Protocol (RPRP) For Wireless Sensor Networks

Wanpracha Nuansoi¹, Wannarat Santiamomtut²
^{1,2}Department of Computer Engineering, Faculty of Engineering,
 Prince of Songkla University, Hat Yai,
 Songkhla 90112 Thailand

Abstract-Wireless networks are not as reliable in communicating information compared to other types of networks. Consumed energy is considered very important in the field of wireless sensor network. The more energy consumed the longer the working time of the network. Most energy-aware routing protocols available currently depend on one working route, resulting in long waiting time to reconstruct the route when the first try fails. This also presents the problem of information discontinuity. This article introduces Energy aware Routing Protocol with Redundant path for Wireless Sensor Networks. The Algorithm for redundant path is presented. The possible minimum energy for connection is pre-set. In the case that the main path of connection fails, the connection can work instantly on the provided redundant path of connection. The ns-2 network simulation shows that connection reliability can be increased by the provision of redundant path.

I. INTRODUCTION

A wireless sensor network is a distributed sensing technology that can be used to monitor physical phenomenon and can be easily deployed and it's useful for many applications. A sensor network is made of the distribution of lots of sensor nodes which require a hop routing paradigm. Sensor networks application ranges from important issues such as environmental and habitat monitoring, traffic control, emergency scenarios, and health care [1, 2, 3].

Sensor nodes are battery-operated. Most of sensor network applications need a lot of sensor nodes which are distributed over the sensor areas. A requirement of lot of sensor nodes leads to a difficulty in battery change.

There is not much reliability in of information communication through wireless sensor network, compared to other types of networks. The energy conservation is also important in sensor network, but the main problem is fact that there is no perfect solution for energy conservation.

Most of energy aware routing protocols are working on a single connective route in order to save the energy. But a problem occurs when the route fails. Time is needed to reconstruct the new connective route. This also presents a problem of communicative reliability.

The remainder of the paper is organized as follows. Section II provides an overview of the energy aware routing protocol with redundant path to increase communicative reliability in wireless sensor network. The redundant path selecting algorithm is also described in this section. Section III explains the selection of equation to identify route and the simulation results. Section IV provides conclusion of the work and discusses the future directions.

II. RELATED WORK

A. Dynamic Source Routing Protocol

DSR is one of routing protocols used worldwide for wireless network [4] which needs perfect part of source node in the form of packet header before starting sending the data. DSR protocol work efficiently with 2 main mechanisms: Route discovery and Route maintenance. Route Discovery mechanism searches for the route connected the destination. On the other hand, Route Maintenance is for reconstruct the route.

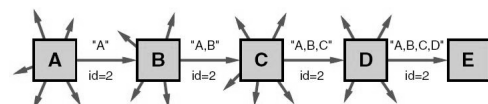


Figure 1 Route Discovery in DSR

Figure 1 shows the process of route discovery in DSR protocol. In the diagram, node A, which is required to connect to node E, will distribute the route request

message with the predetermined target number 2 throughout the network. Any nodes receiving the routing request check whether or not it has ever received the message and whether its name is included in the message title. If so, it will delete the entire message. If the answer is no for the request, the node will add its name into the message title and send it afterwards until the destination. When the message reaches the target node, the target will send a reply message back to the source node relying on the node names attached in the received message's title.

B. Energy aware routing protocol in wireless sensor network

This kind of protocol is aimed to save as much as energy in sensor node for the purpose of increase working time. It is designed to construct only one path to the destination. It's different from the general protocols which distribute their several searching routes to the destination. Because of its only one working route, DSR protocol uses less energy than other types. A sample of this type of protocol is Energy Aware Routing for Wireless Sensor Networks [5]

Algorithm consists of 3 main parts; Route Discovery, Route Reply, and Reliable Transmission. Two types of messages are used. They are route-requested message and route-responded message. When any nodes

want to communicate, they will send mechanism route discovery message to the destination, in which only one route can reach the destination. A problem about time used occurs with the process. It's due to the route chooses to connect with high energy nodes and good quality of connection increases the working time, and the reconstruction of route is needed when the connection fails. The illustration is shown in Fig. 2.

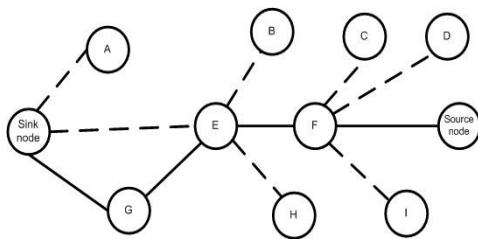


Figure 2 Route discovery in Energy aware routing.

Figure 2 shows the whole process of route discovery. A route-requested message is sent out, node A, E and G receive the message. Node G is the highest in energy, so it's chosen to be connected. Then, node G distributes the message to the nodes nearby. Node E is chosen afterwards. It passes out the message to node B, F and H. Node H examines the message's order number and leaves the packet because it finds that node B and F contain the same level of energy and their quality of connection is equal. So

Both B and F start counting backwards. If node F finishes first, it means ACK will be sent by node F to node E, which will send ACK to node B to inform about the completion of node selection. Afterwards, node B ends its backward count. When route request message reach the destination node, the destination will send back reply package on the same route of F-E-G.

C. The estimation of energy used in route discovery is given in Energy Aware Routing Protocol (EARP) for Ad-Hoc Wireless Sensor Networks

Energy used by each node to distribute route request message (RREQ) is described as follows [6].

$$E_1 = [\alpha_{t_1} + \alpha_{t_2} d^4] * B * t = P_1 \quad (1)$$

Energy used to receive route request message (RREQ) is described as follows.

$$E_2 = \alpha_r * B * t_c = P_2 \quad (2)$$

Energy used to send route reply message (RREP) is as follows.

$$E_3 = [\alpha_{t_1} + \alpha_{t_2} d^4] * B * t_c = P_3 \quad (3)$$

Total energy used in route discovery is from the addition of energy used in sending and replying the message.

$$E_t = E_q + E_p \quad (4)$$

$\alpha_{t_1}, \alpha_{t_2}$ = Constant value in communication with $\alpha_{t_1} = 45\text{nJ/b}$, $\alpha_{t_2} = 10\text{p J/b}$

d^4 = the mean of 2 neighbor nodes and n is 4 for all calculation

B = the size of general connection channel measured in Kbps

t = total frame size in each second
 α_r = constant value at 135n J/b
 t_c = the duration of a time slot in each second
 H = hop
 N = Number of all nodes in the network

III. REDUNDANT PATH ROUTING PROTOCOL (RPRP) IN WIRELESS NETWORK

Redundant Path Routing Protocol for wireless network is designed to increase communication reliability by providing a redundant path. The additional path is useful in case the first connection fails. Moreover, the source node can choose to connect with the route with low energy required. The two routes of connection will use the different nodes to work on in order to stabilize the connection. If they use the same node, it will lead to the failure of the connection. Nodes will show their status as follows to avoid the repetition.

Nodes in the network are Free at the beginning, except the source node and destination node. They are in connected status. The source node distributes the routing request and nodes which receive the message will turn their status into Working and when it receives the result of chosen node, it changes its status into C. If not, it stays in the same status, F.

Table 1 Node status

Node status	Meaning
F	Free
W	Working
C	Connected

Define

- Nodes are placed orderly on their id
- ET (Energy Threshold) is constant
- (ROUTE_REQ) is for route request message
- (ROUTE_REP) is for route response message
- (RESULT_REQ) is for result request message
- (RESULT_REP) is for result response message
- (ROUTE_FAIL) is for route failure message

Table 2
Routing Algorithm

Nodes	Procedure
Source node	<ul style="list-style-type: none"> • Distributes ROUTE_REQ message and shows its status as C. • The calculation for the distance between destination node ID and source node ID is made. • Choose to connect to a node with the minimum distance. • When the first ROUTE_REP received, it continues to distribute ROUTE_REQ for the second route with the exception of C nodes, resulting in no repetition of the first and second route. • Send RESULT REP message to inform the result of node choosing • When the NODE_FAIL message is received, it works on redundant route instead. • The main route with the lowest Hop Counts is chosen from two routes. If it comes up with the same number of Hop Counts, the second is chosen because it's newer.
Intermediate node	<ul style="list-style-type: none"> • Nodes with lower energy than ET drop out ROUTE_REQ message • Nodes with higher level of energy than ET change their status into W and send out RESULT_REQ message to source node which sends the c • The connected nodes calculate for the ID difference between destination node and the node sending ROUTE_REQ. • The connected node choose to connect to a node with the minimum difference and inform the result node choosing though RESULT REP. • When the RESULT REP message received, drop out ROUTE_REQ message and turns to F status, if the node is not chosen. • The connected nodes turn to C status and pass on RESULT_REQ message.
Destination node	<ul style="list-style-type: none"> • When ROUTE_REQ message is received, ROUTE_REP message is sent out to the source node on using the same route.

Table 3
Route Maintenance Algorithm

Nodes	Procedure
Intermediate node	<ul style="list-style-type: none"> When a node receives ROUTE_FAIL message, it passes the message on the connected route. If the result of distribution is not shown within the allocated time, ROUTE_FAIL message is sent back to the Source node.
Source node	<ul style="list-style-type: none"> When the main route fails, choose to work on the redundant route immediately. If the redundant route also fails, discover the new route. In this case, the source node doesn't work under energy consideration.

Route Discovery of RPRP

Figure 3 shows that node 18, the source node, needs to connect to node 0, the destination node. It collects data of connective nodes. From the beginning, all nodes in the network are in F status, except the source node whose status is C. Node 18 will distribute ROUTE_REQ message as shown in Figure 4a. The nodes which receive the message are node 24, 23, 22, 19, 17, 12, 13 and 14.

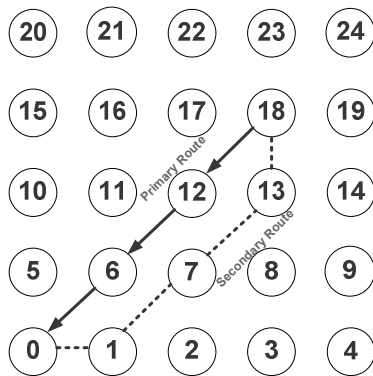


Figure 3: Route Discovery in RPRP

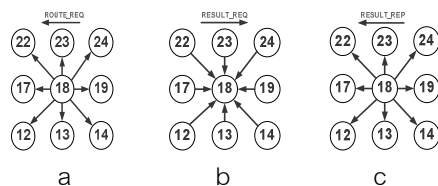


Figure 4: Procedure of node selecting for connection

The procedure of node selecting is presented. Every node receiving ROUTE_REQ message will examine their energy. If their energy is lower than the required amount, it will drop out the message. If they have enough energy, they will turn their status into W and send RESULT_REQ message to S node to join the selecting procedure. The illustration is shown in Figure 4b. Node 18 will calculate for the difference as in Table 4. ID of nodes sending REQUEST_REQ message will be minused by ID of the destination node. The node with the minimum difference will be chosen. As it's shown in Table below, node 12 is chosen.

Table 4
The calculation for difference for node selection

Source node	Destination node	Node sending RESULT_REQ	Difference calculation
18	0	24	(24-0) = 24
		23	(23-0) = 23
		22	(22-0) = 22
		19	(19-0) = 19
		17	(17-0) = 17
		14	(14-0) = 14
		13	(13-0) = 13
		12	(12-0) = 12

After node 18 sends RESULT_REP message to show the result of node selection as illustrated in Figure 4c, the selected nodes will turn to C status. The nodes which are not selected will turn back to F status. Node 12 is selected. It will turn to C and distribute ROUTE_REQ message as in Figure 5a. Then the same procedure is made until it connects the destination node.

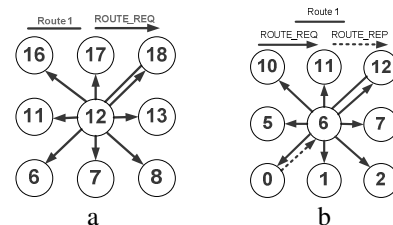


Figure 5 Route Request and Route Response

The destination node will send ROUTE_REP message when it receives ROUTE_REQ message. The illustration is shown in Figure 5b. Node 6 receives ROUTE_REP message, stops selecting nodes

and sends out ROUTE_REP to the source node.

When node 18 receives ROUTE_REP message, it will distribute ROUTE_REQ message for the second route. Then, the selecting of nodes is proceeded. Nodes which are in C status will not be included in this procedure. As a result, the 2 routes work on the different groups of nodes.

After that, node 18 will choose the main route, based on the number of hop counts in each route. The route with fewer hop counts is considered the main route. The other route will be the redundant route. If the 2 routes have the same number of hop counts, the second route will be chosen because it's newer. As in Figure 3, the main route is on node 18, 12, 6, 0, while the redundant route is on node 18, 13, 7, 1,0. Then, the communicating of messages is started.

Sample Use of Redundant Route

When a node in the main route fails to work, node before it will send ROUTE_FAIL message on the same route to the source node because it doesn't receive the confirmation of message in the given time. The source node will notice the route failure and turn to work on the redundant route instead.

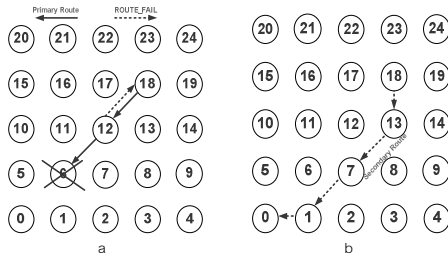


Figure 6: The Use of redundant Route

Figure 6a shows that the nodes on the main route are 18, 12, 6, 0 and the main route fails. Node 6 fails to work, and node 12 sends a message to node 6. When the confirmation of message is not received, node 12 will send ROUTE_FAIL message back to node 18. Node 18 is informed the route failure and turns to use the redundant route, after the ROUTE_FAIL message is received, as shown in Figure 6b.

IV. RESULTS OF ENERGY EQUATION IN ROUTE DISCOVERY AND SIMULATION RESULT

A. The estimation of energy used in route discovery for Redundant Path Routing Protocol (RPRP) for Wireless Sensor Networks

Energy is divided into 2 main parts; Energy for route request distribution and Energy for route request reply.

Energy used for route request distribution is differentiated into 3 different types; source node (S), intermediate node (M) and destination node (D).

Source node (S)

- Energy used by node S to distribute ROUTE_REQ

$$E_1 = [\alpha_{t1} + \alpha_{t2}d^4] * B * t = P_1 \quad (5)$$

- Energy used by node S to receive RESULT_REQ

$$E_2 = \alpha_r * B * t_c = P_2 \quad (6)$$

- Energy used by node S to send RESULT_REP

$$E_3 = P_1 \quad (7)$$

Intermediate node

- Energy used by intermediate node to receive ROUTE_REQ

$$E_4 = P_2 = E_{10} \quad (8)$$

- Energy used by intermediate node to send RESULT_REQ

$$E_5 = [\alpha_{t1} + \alpha_{t2}d^4] * B * t_c = P_3 = E_{11} \quad (9)$$

- Energy used by intermediate node to receive RESULT_REP

$$E_6 = P_2 = E_{12} \quad (10)$$

- Energy used by the chosen intermediate node to send ROUTE_REQ

$$E_7 = P_1 \quad (11)$$

- Energy used by the chosen intermediate node to receive RESULT_REQ

$$E_8 = P_2 \quad (12)$$

- Energy used by the chosen intermediate node to send RESULT_REP

$$E_9 = P_1 \quad (13)$$

Destination node

- Energy used by node D to receive ROUTE_REQ

$$E_{10} = P_2 \quad (14)$$

Energy used to send RREQ from node S to node D is as follows.

$$\begin{aligned} E_q &= (E_1 + E_2 + E_3) + ME_4 + (M - m)(E_5 + E_6) \\ &+ hE_7 + hE_8 + hE_9 + E_{10} \\ &= (P_1 + P_2 + P_1) + MP_2 + (M - m)(P_3 + P_2) \\ &+ hP_1 + hP_2 + hP_1 + P_2 \\ &= (2 + 2h)P_1 + (h + 2M + 2 - m)P_2 + (M - m)P_3 \end{aligned} \quad (15)$$

To reply to the RREQ, node uses the same route. It can be described by equation presented in (6).

$$\begin{aligned} E_p &= E_5 + (E_6 + E_7) * (h) + E_8 \\ &= P_3 + (P_2 + P_3) * h + P_2 \\ &= (P_2 + P_3) * (h + 1) \end{aligned} \quad (16)$$

Total energy used to discover route is as follows.

$$\begin{aligned} &= E_q + E_p \\ &= (P_1 + P_2) * (N - 1) + (P_2 + P_3) * (h + 1) \\ &= (N - 1)P_1 + (N + h)P_2 + (h + 1)P_3 \end{aligned} \quad (17)$$

M = number of intermediate node with $\leq N$ depending on how fast it can connect to the destination node

m = number of chosen intermediate nodes.

When route discovery equations are compared, the same equation is used in Energy Aware Routing Protocol (EARP) for Ad-Hoc Wireless Sensor Networks, and general ad-hoc routing protocol, including DSR protocol.

$$\begin{aligned} &= E_q + E_p \\ &= (P_1 + P_2) * (N - 1) + (P_2 + P_3) * (h + 1) \\ &= (N - 1)P_1 + (N + h)P_2 + (h + 1)P_3 \end{aligned} \quad (18)$$

It can be seen that coefficient of P_1 in the equation 17 is lower than in equation 18. It's because h is lower than N. In contrast, P_2 in equation 17 might be lower or higher than in equation 18 depending on M. It means longer end-to-end time if M is high. It needs a lot of intermediate nodes. M can be lower if the intermediate nodes are denser. P_3 is higher in equation 17 than in equation 18. The increased m can be caused by the need to send the result of node selection and connection request message.

It can be concluded that how much energy is needed by this kind of protocol depends a lot on the distance between source node to destination node. It uses less energy when the distance is short, even with a lot of nodes in the network. But it will use more energy than the general protocol if the distance is longer, even with fewer nodes.

Simulation and the Results

There are 100 nodes in the simulation as shown in Figure 7. The given time for the simulation is 1,000 seconds. The information is sent in the form of CBR in which the information of 500 bites is sent every 0.2 second. The environments of the simulation are set as follows.

Table 5
Simulation environment

List	Description
Simulation Program	NS2.23
Number of nodes	100 Node
Radio-propagation model	TwoRayGround
Interface queue type	DropTail/PriQueue
Antenna model	OmniAntenna

Max packet in interface queue	50 Byte
Routing protocol	AODV, DSR, RPRP, RPRP(Single path)
Traffic	Constant Bit Rate (CBR)/UDP
CBR Interval Time	0.2 Second
MAC Type	Mac/802.15.4
Node Movement	Fix
Area	90*90 Meter
Initial Energy	3 Joule
Packet Size	500 Byte
Source Node	Random
Destination Node	Node number 0
Node Distance	9 Meter
Radio Range	20 Meter
Simulation Time	1000 Second

The efficient of the protocol is calculated and the comparison to other protocols is made. *The Average of Throughput (T_a)*

T_a is calculated from the number of information packages received by the destination node multiplied by 8 and divided by the time of simulation.

$T_a = (\text{No. of information packages received by the destination node} \times 8 / \text{time of the simulation})$

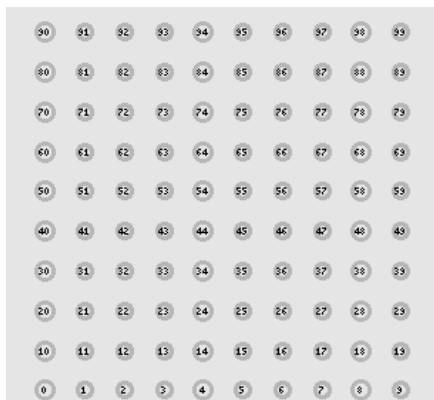


Figure 7 the simulation of 100- node wireless sensor network

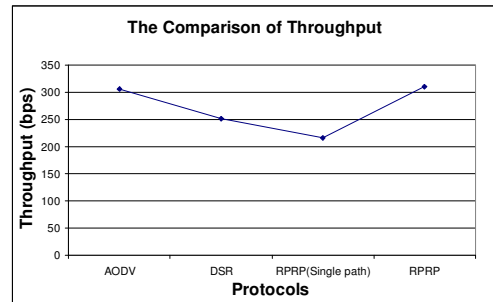


Figure 8 The Comparison of Throughput

Figure 8 shows that the average of throughput of RPRP protocol is the highest compared to others. It's because it can operate on the redundant route. As a result, it gives continuous communication. So it has highest throughput. AODV protocol has the next highest throughput. It's a protocol whose TTL will gradually increase in sending ROUTE_REQ message until it reaches the destination node. When the route is disconnected, it will make use of nodes in the same route, or if it discovers the new route, TTL is kept and it doesn't have to waste time sending new ROUTE_REQ message. Its throughput is also high. Throughput of DSR comes the third. DSR protocol distributes ROUTE_REQ message to the entire network. When the connection fails, the source node will find the other route on the available nodes to the destination node. If there are no nodes available, it will distribute the message to the network again. This wastes more time than AODV protocol. Lastly, RPRP (single path) protocol has the lowest throughput because it takes longer time to discover new route. The discontinuous connection causes the lowest score of throughput.

The Average of Energy Used for the Entire Network (E_a)

E_a is calculated from the sum of difference between starting energy (E_i) and final energy (E_f), divided by the number of nodes losing their energy in the connection. This is mentioned in [6].

$$E_a = \frac{\sum_{k=1}^{k=n} (E_{ik} - E_{fk})}{N} \quad (19)$$

Figure 9 shows the average of used energy.

RPRP consumes the highest energy, resulted from the fact that it spends lots of energy for sending messages. Its throughput is the highest and the energy for route discovery is also high. It can be proved by the average time used for route discovery presented in Figure 10. AODV uses the second highest energy. It also needs lots of energy for sending messages. It has the second highest throughput. DSR consumes the third highest energy. It consumes less energy for sending message than RPRP and AODV. RPRP (single path) consumes the least energy. It consumes the least energy to send message, as it's proved by the lowest throughput. When its connection fails, all energy may be needed to reconstruct the new route,

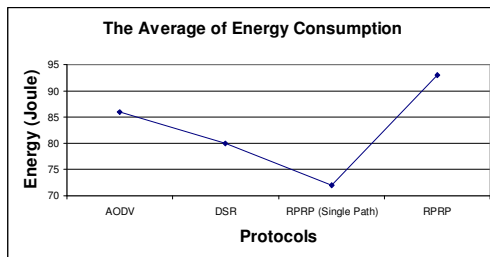


Figure 9 The Average of Energy Consumption

contrasting to RPRP with Redundant path in which there is no need to reconstruct the new route.

The Average of Time for Route Discovery (RD_a)

RD_a is calculated from the sum of ending time of route discovery, minused by the sum of starting time. It can be measured in the form of Millionsecond (ms).

$RD_a = \text{Sum of ending time of route discovery} - \text{sum of starting time}$

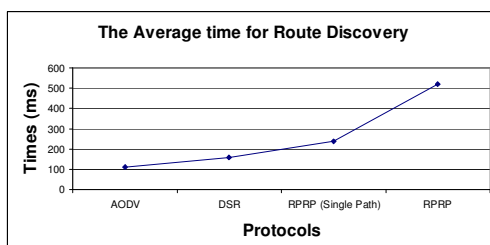


Figure 10 The Average time for Route Discovery

Figure 10 compares the average time for route discovery from different protocols.

RPRP uses the longest time because it needs to discover a connective node at a time for the route. Besides, it needs to find 2 routes. RPRP (single path) uses the second longest time. It also loses time for each connective node for the route. DSR uses the third longest time. It needs to distribute the messages to the entire network and use longer time to find route than AODV. AODV uses time best. It distributes messages one time the entire network.

The Average Time for Route Swaps (ST_a)

ST_a is calculated from the sum of difference between time of disconnection and time spent on the redundant route, divided by the number of time switching to redundant route (N).

$ST_a = \text{time of disconnection} - \text{time spent on redundant route} / \text{No. of times witching to redundant route.}$

Time for RPRP to switch to redundant route is counted from the moment the route starts to disconnect until it works successfully on the redundant route. The average switching time is 102 ms.

V. CONCLUSION

This paper presents Redundant Path Routing Protocol for wireless networks. The simulation shows that the throughput of RPRP is the highest. As a result, it can increase the reliability of connection. From the energy estimation for this protocol, it can help save energy if the source node closes to the destination node. The source node has choice to use the less energy-consumed route. However, it still has some limitations. That is, nodes on the connective route need to be placed according to ID order and it spends quite long time for route discovery.

REFERENCES

- [1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", In Proc. of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA-02), 2002.
- [2] Y. Gsottberger, X. Shi, G. Stromberg, T. F. Sturm, and W. Weber, "Embedding Low-Cost Wireless Sensors into Universal Plug and Play Environments". In Proc. Of EWSN, 2004.
- [3] K.A. Delin, "Sensor Web in Antarctica - Developing an Intelligent, Autonomous Platform for Locating Biological Flourishes in Cryogenic Environments". In Proc. of the 34th Lunar and Planetary Science Conf (LPSC 03), 2003.

- [4] David B. Johnson, David A. Maltz and Josh Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15313-3891, <http://www.monarch.cs.cmu.edu/>.
- [5] R. Vidhyapriya and P.T. Vanathi, "Energy Aware Routing for Wireless Sensor Networks," In Proc. Of the IEEE-ICSCN 2007, MIT Campus, Anna University, Chennai, India. Feb.22-24, 2007. pp.545-550.
- [6] Raminder P. Mann, Kamesh Namuduri and Ravi Pendse, "Energy Aware Routing Protocol (EARP) for Ad-Hoc Wireless Sensor Networks", Wichita State University, KS 67260, USA.

ประวัติผู้เขียน

ชื่อ สกุล	นายวันประชา นวนสร้อย		
รหัสประจำตัวนักศึกษา	4910120103		
วุฒิการศึกษา			
วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา	
วิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์)	สถาบันเทคโนโลยีพระจอม เกล้าเจ้าคุณทหารลาดกระบัง	2549	

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนยกเว้นค่าหน่วยกิต คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

การตีพิมพ์เผยแพร่ผลงาน

วรรณรัช สันตอมรทัต, นิคม สุวรรณวร, วันประชา นวนสร้อย, “โพรโทคอลหาเส้นทาง
ประหยัดพลังงานที่มีเส้นทางสำรองสำหรับเครือข่ายตรวจจับไร้สาย”, การประชุมวิชาการทาง
วิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ครั้งที่ 7, 21-22 พฤษภาคม 2552, 132-137.

Wanpracha Nuansoi, Wannarat Santiamomtut, “Redundant Path Routing Protocol (RPRP) For
Wireless Sensor Networks”, The 2010 International Conference on Embedded Systems and
Intelligent Technology (ICESIT2010), 5-7 February 2010.