

การตั้งเส้นทางสำรองแบบหลายจุดสำหรับระบบเครือข่าย AODV-2T
MULTI-HOP BACKUP ROUTE ESTABLISHMENT FOR AODV-2T



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของวารสารศึกษาค้นคว้าตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

KMUTT. 2009-EN-M-230-085

การสร้างเส้นทางสำรองแบบหลายฮอปสำหรับระบบเครือข่าย AODV-2T

MULTI-HOP BACKUP ROUTE ESTABLISHMENT FOR AODV-2T



T105161



หฤยวรรณ สมนึก

HARISAVAN SOMNUK

กพ.
๒๕๒๙๗๗
๒๕๕๒

เลขหมู่.....
เลขทะเบียน.....105161
วัน,เดือน,ปี.....1.6.๗๗.๒๕๕๒

.b.....121661๙๐
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2552

KMITL 2009-EN-M-230-085

MULTI-HOP BACKUP ROUTE ESTABLISHMENT FOR AODV-2T



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2009

KMITL 2009-EN-M-230-085

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2009

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การสร้างเส้นทางสำรองแบบหลายฮอปสำหรับระบบเครือข่าย AODV-2T
Thesis Title Multi-Hop Backup Route Establishment for AODV-2T
นักศึกษา นางสาวหฤชวรรณ สมนึก
รหัสประจำตัว 47061104
ปริญญา วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา วิศวกรรมสารสนเทศ
อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ.มยุรี เลิศเวชกุล
หมายเลขวิทยานิพนธ์ KMITL-2009-EN-M-230-085

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.พิทักษ์	ธรรมาวริน	
รศ.ดร.ปิติเขต	สุรักษา	
รศ.ดร.ชวลิต	เบญจางคประเสริฐ	
ดร.วีระพล	โมนยะกุล	
ผศ.มยุรี	เลิศเวชกุล	

วัน / เดือน / ปี ที่สอบ วันพฤหัสบดีที่ 28 พฤษภาคม พ.ศ. 2552 เวลา 16.30-18.30 น.

สถานที่สอบ ณ อาคาร A ชั้น 3 ห้องประชุม 2

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว

(รองศาสตราจารย์ ดร.กอบชัย เดชหาญ)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 28 พฤษภาคม พ.ศ. 2552

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การสร้างเส้นทางสำรองแบบหลายฮอปสำหรับระบบเครือข่าย AODV-2T
นักศึกษา	นางสาว หฤชวรรณ สมนึก
รหัสประจำตัว	47061104
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2552
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.มยุรี เลิศเวชกุล

บทคัดย่อ

การสื่อสารในระบบเครือข่ายแอ็ดฮอคซึ่งเป็นระบบเครือข่ายที่มีโครงสร้างเปลี่ยนแปลงตลอดเวลา เพราะอุปกรณ์ไร้สายต่าง ๆ สามารถเคลื่อนที่ได้โดยอิสระ จึงเป็นปัจจัยหลักที่ทำให้เส้นทางสื่อสารมีโอกาสที่จะถูกตัดขาดได้ การขาดการเชื่อมต่อของเส้นทางสื่อสารนั้นอาจเกิดจากปัญหาการขาดพลังงานแบตเตอรี่ของโหนดสื่อสารได้เช่นกัน โพรโทคอล AODV-2T[1] ได้ถูกพัฒนาขึ้นมาด้วยความพยายามที่จะปรับปรุงเสถียรภาพของการสื่อสารแบบแอ็ดฮอค[2] ด้วยการตรวจสอบควบคุมระดับสัญญาณการเชื่อมต่อและระดับพลังงานแบตเตอรี่ของโหนดสื่อสารในเส้นทาง เพื่อให้มีการสร้างเส้นทางสำรองและการควบคุมให้สลับไปใช้เส้นทางสำรอง ก่อนที่เส้นทางหลักจะถูกตัดขาด อย่างไรก็ตาม โพรโทคอล AODV-2T ไม่สามารถรองรับรูปแบบการสร้างเส้นทางสำรองที่มีระยะทางมากกว่า 1 ฮอปได้ ผู้วิจัยจึงได้ปรับปรุงขั้นตอนการสร้างเส้นทางสำรองของ AODV-2T ให้สามารถรองรับการสร้างเส้นทางย่อยสำรองที่มีระยะทางมากกว่า 1 ฮอปได้ (Multi-hop AODV-2T) และได้จำลองผลการทำงานของระบบเครือข่ายไร้สายแบบแอ็ดฮอคแบบหลายฮอปเพื่อทดสอบประสิทธิภาพของโพรโทคอลที่ได้พัฒนาขึ้น ซึ่งก็พบว่าโพรโทคอล Multi-hop AODV-2T สามารถช่วยเพิ่มประสิทธิภาพในการสร้างเส้นทางสำรอง เพราะการสร้างเส้นทางสำรองที่รองรับเส้นทางย่อยหลายฮอปจะช่วยให้ลดโอกาสที่จะทำให้แพ็คเกจสูญหายลงได้

Thesis	Multi-Hop Backup Route Establishment For AODV-2T
Student	Ms.Harisavan Somnuk
Student ID.	47061104
Degree	Master of Engineering
Program	Information Engineering
Year	2009
Thesis Advisor	Asst.Prof.Mayuree Lertwatechakul

ABSTRACT

Communication in Ad Hoc wireless network may generally be uncertain because the established route could be broken anytime. The major cause of link failures in Ad hoc wireless network is the dynamic movement of the mobile nodes themselves. The other is mobile node could be turning off or running out of battery. Many research works have proposed to solve the problem.

Some of those protocols generate a lot of overhead and overload mobile nodes with routing functions. And many protocols are too slow to response to the link failure event. AODV-2T [1] was proposed to compromise these problems by a conservative and circumspect concept. AODV-2T prepares a backup route just before the route might be broken and then suddenly switches packet transmission to the backup route with no delay when the main route is breaking down. Although AODV-2T performs rather well compared to the early proposed protocols, AODV-2T does not support multihop backup route. On the assumption that Multihop backup route establishment might be more flexible and gives more chance of success in Ad hoc route maintenance, so we have extended AODV-2T capability to support Multihop backup route. The improved protocol called Multi-hop AODV-2T was implemented into NS-2 model. We have simulated Multihop AODV-2T function and compared its performance to the route maintenance mechanism of AODV and AODV-2T. The results show that Multihop AODV-2T can reduce number of route break and packet loss obviously.

กิตติกรรมประกาศ

ผู้เขียนขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.เลิศเวชกุล เป็นอย่างสูง ที่ได้ให้ความกรุณา
สละเวลารับเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์เรื่องนี้ ซึ่งได้ให้คำแนะนำถึงประเด็นต่าง ๆ ใน
การศึกษาและชี้แนวทางในการแก้ปัญหา การค้นคว้าหาข้อมูลเพิ่มเติม อันเป็นประโยชน์ในการ
วิเคราะห์และสรุปผลการวิจัย รวมทั้งการแก้ไขงานวิจัยให้สมบูรณ์เป็นอย่างยิ่ง

ขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.ปิติเขต ผู้รักษา และคณะกรรมการสอบ
วิทยานิพนธ์ทุกท่านที่ให้ความกรุณาให้คำแนะนำในการแก้ไขข้อบกพร่องต่าง ๆ ของงานวิจัย

ขอกราบขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบัน
เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ ให้
คำแนะนำตลอดการศึกษาที่ผ่านมา

ขอขอบพระคุณการไฟฟ้าฝ่ายผลิตแห่งประเทศไทยที่ให้ทุนสนับสนุนในการลาศึกษาต่อ
ในระดับปริญญาวิศวกรรมศาสตรมหาบัณฑิต

ขอขอบคุณเจ้าหน้าที่บัณฑิตศึกษาและบัณฑิตวิทยาลัย คณะวิศวกรรมศาสตร์ทุกท่านที่
ความดูแล ช่วยเหลือมาตลอดหลักสูตรการศึกษา

ขอขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ในภาควิชาวิศวกรรมสารสนเทศ ระดับบัณฑิตศึกษา
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่คอยช่วยเหลือและให้กำลังใจ
เสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นแรง
สนับสนุน และกำลังใจทุกเรื่อง ๆ จนทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้
ด้วยดี

คุณความดีและประโยชน์ทั้งปวงที่พึงมีจากงานวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบแต่ผู้มี
พระคุณทุกท่าน

หฤชวรรณ สมนึก

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....	2
1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน.....	2
1.6 ขอบเขตการวิจัย.....	3
1.7 ขั้นตอนการศึกษา.....	3
บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในการวิจัย.....	4
2.1 เครือข่ายเคลื่อนที่ไร้สายชนิดแอดฮอค.....	4
2.1.1 เปรียบเทียบความแตกต่างของระบบเครือข่ายไร้สายแบบแอดฮอกกับเครือข่ายไร้สายแบบอาศัยโครงสร้างพื้นฐาน.....	4
2.1.2 โพรโทคอลที่ใช้ในการค้นหาเส้นทาง (Routing Protocols).....	6
2.2 โพรโทคอล AODV.....	7
2.2.1 การค้นหาเส้นทางของโพรโทคอล AODV.....	8
2.2.2 การซ่อมแซมเส้นทาง.....	11
2.3 Ad Hoc On-demand Distance Vector Routing Protocol (AODV).....	13
2.3.1 โครงสร้างแพ็คเกจของโพรโทคอล AODV.....	13
2.3.1.1 โครงสร้างของแพ็คเกจร้องขอเส้นทาง.....	13
2.3.1.2 โครงสร้างของแพ็คเกจตอบกลับ.....	13
2.3.1.3 โครงสร้างของแพ็คเกจควบคุมความผิดพลาด.....	15

สารบัญ (ต่อ)

หน้า

2.3.1.4	โครงสร้างของแพ็คเก็ตรับทราบการตอบกลับ.....	16
2.3.2	การทำงานของโปรโตคอล AODV.....	16
2.3.2.1	การจัดการเลขลำดับ (Sequence Number).....	16
2.3.2.2	ตารางเส้นทางของโปรโตคอล AODV.....	17
2.3.2.3	การส่งแพ็คเก็ตร้องขอเส้นทาง.....	18
2.3.2.4	การรับแพ็คเก็ตร้องขอเส้นทาง.....	19
2.3.2.5	การส่งแพ็คเก็ตตอบกลับการขอเส้นทาง.....	19
2.3.2.6	การรับแพ็คเก็ตตอบกลับการขอเส้นทาง.....	20
2.3.2.7	แพ็คเก็ตทักทาย.....	21
2.3.2.8	การส่งแพ็คเก็ตควบคุมความผิดพลาด.....	21
2.3.2.9	การซ่อมแซมเส้นทาง.....	21
2.4	โปรโตคอล AODV Backup Route Discovery with 2 Levels of Threshold (AODV-2T).....	22
2.4.1	การตรวจวัดระดับพลังงานของโหนดและระดับสัญญาณที่ได้รับของโหนด.....	16
2.4.1.1	การตรวจวัดระดับพลังงาน.....	22
2.4.1.1	การตรวจวัดระดับสัญญาณ.....	24
2.4.2	การค้นหาเส้นทางสำรอง.....	24
บทที่ 3	การสร้างเส้นทางสำรองแบบหลายฮอปสำหรับระบบเครือข่าย AODV-2T.....	27
3.1	การตรวจวัดระดับพลังงานของโหนดและระดับสัญญาณที่ได้รับของโหนด.....	27
3.1.1	หลักการตรวจวัดระดับพลังงานของโหนด.....	28
3.1.2	การตรวจวัดระดับสัญญาณ.....	29
3.1.3	การกำหนดระดับเรชโฮลด์.....	30
3.1.3.1	การกำหนดระดับเรชโฮลด์ของแบตเตอรี่.....	30
3.1.3.2	การกำหนดระดับเรชโฮลด์ของสัญญาณที่ได้รับ.....	31
3.2	กระบวนการค้นหาเส้นทางสำรอง.....	33
3.2.1	การส่งแพ็คเก็ตร้องขอเส้นทางสำรอง.....	36
3.2.2	การรับแพ็คเก็ตร้องขอเส้นทางสำรอง.....	37
3.2.3	การส่งแพ็คเก็ตตอบกลับการร้องขอเส้นทางสำรอง.....	40

สารบัญ (ต่อ)

	หน้า
3.2.4 การรับแจ้งเกิดตอบกลับการร้องขอเส้นทางสำรอง	41
3.2.5 การรับแจ้งเกิดแจ้งเตือน.....	45
บทที่ 4 การจำลองการทำงานของระบบ.....	44
4.1 การประมวลผลประสิทธิภาพของเครือข่ายจากการจำลองการทำงาน.....	44
4.2 การจำลองการทำงานในสภาพแวดล้อมต่าง ๆ	45
4.3 ผลการจำลอง.....	48
4.3.1 ผลการจำลองในสภาพแวดล้อมที่ 1	48
4.3.1.1 อัตราการส่งข้อมูลสำเร็จในสภาพแวดล้อมที่ 1.....	49
4.3.1.2 จำนวนแพ็คเกจที่ถูกทิ้งในสภาพแวดล้อมที่ 1.....	50
4.3.1.3 จำนวนแพ็คเกจควบคุมในสภาพแวดล้อมที่ 1.....	50
4.3.1.4 ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลในสภาพแวดล้อมที่ 1.....	51
4.3.1.5 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในสภาพแวดล้อมที่ 1.....	52
4.3.2 ผลการจำลองในสภาพแวดล้อมที่ 2	52
4.3.2.1 อัตราการส่งข้อมูลสำเร็จในสภาพแวดล้อมที่ 2.....	53
4.3.2.2 จำนวนแพ็คเกจที่ถูกทิ้งในสภาพแวดล้อมที่ 2.....	53
4.3.2.3 จำนวนแพ็คเกจที่ถูกควบคุมในสภาพแวดล้อมที่ 2.....	54
4.3.2.4 ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลในสภาพแวดล้อมที่ 2.....	55
4.3.2.5 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในสภาพแวดล้อมที่ 2.....	55
4.3.3 ผลการจำลองในสภาพแวดล้อมที่ 3	56
4.3.3.1 อัตราการส่งข้อมูลสำเร็จในสภาพแวดล้อมที่ 3.....	56
4.3.3.2 จำนวนแพ็คเกจที่ถูกทิ้งในสภาพแวดล้อมที่ 3.....	57
4.3.3.3 จำนวนแพ็คเกจที่ถูกควบคุมในสภาพแวดล้อมที่ 3.....	57
4.3.3.4 ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลในสภาพแวดล้อมที่ 3.....	58
4.3.3.5 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในสภาพแวดล้อมที่ 3.....	59
4.4 การจำลองโดยการปรับเปลี่ยนค่าเรชโฮลด์.....	59
4.4.1 การจำลองโดยการปรับเปลี่ยนเรชโฮลด์ของแบตเตอรี่.....	59
4.4.2 การจำลอง โดยการปรับเปลี่ยนเรชโฮลด์ของระดับสัญญาณที่ได้รับ.....	59

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	64
บรรณานุกรม.....	65
ภาคผนวก.....	67
ภาคผนวก ก. การจำลองการทำงานของเครือข่ายโดยใช้โปรแกรม NS-2.....	67
ภาคผนวก ข. ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่.....	115
ประวัติผู้เขียน.....	122



สารบัญตาราง

ตารางที่	หน้า
2.1 คำอธิบายรายละเอียดของฟิล์มต่าง ๆ ในแพ็คเกจร้องขอเส้นทาง.....	14
2.2 คำอธิบายรายละเอียดของฟิล์มต่าง ๆ ในแพ็คเกจตอบกลับ.....	15
2.3 คำอธิบายรายละเอียดของฟิล์มต่าง ๆ ในแพ็คเกจควบคุมความผิดพลาด.....	15
2.4 ตารางเส้นทางของโปรโตคอล AODV.....	17
3.1 คำอธิบายฟิล์มต่าง ๆ ในแพ็คเกจแจ้งเตือน.....	33
3.2 คำอธิบายฟิล์มต่าง ๆ ในแพ็คเกจร้องขอเส้นทางสำรอง.....	36
3.3 คำอธิบายฟิล์มต่าง ๆ ในแพ็คเกจตอบกลับการร้องขอเส้นทางสำรอง.....	40
4.1 ข้อกำหนดในการจำลองการทำงานในสภาพแวดล้อมที่ 1	46
4.2 ข้อกำหนดในการจำลองการทำงานในสภาพแวดล้อมที่ 2	46
4.3 ข้อกำหนดในการจำลองการทำงานในสภาพแวดล้อมที่ 3	47
4.4 ข้อกำหนดในการจำลองการทำงานในสภาพแวดล้อมที่ 4	60
4.5 ข้อกำหนดในการจำลองการทำงานในสภาพแวดล้อมที่ 5	62

สารบัญรูป

รูปที่	หน้า
2.1 แสดงเครือข่ายไร้สายแบบอาศัยโครงสร้างพื้นฐาน.....	5
2.2 แสดงเครือข่ายไร้สายแบบแอดฮอค.....	5
2.3 แสดงการแบ่งประเภทของระบบเครือข่ายไร้สายชนิดแอดฮอค.....	6
2.4 ก. แสดงการค้นหาเส้นทางในโปรโตคอล AODV.....	8
2.4 ข. แสดงการส่งแพ็คเก็ต RREQ เพื่อร้องขอเส้นทาง.....	9
2.4 ค. แสดงการส่งแพ็คเก็ต RREQ ต่อไปเป็นทอด ๆ เพื่อร้องขอเส้นทาง.....	9
2.4 ง. แสดงการส่งแพ็คเก็ต RREP ตอบกลับ.....	9
2.4 จ. แสดงการส่งแพ็คเก็ต RREQ ไปจนถึงโหนดปลายทาง.....	10
2.4 ฉ. แสดงการส่งแพ็คเก็ต RREP ตอบกลับไปยังโหนดต้นทาง.....	10
2.4 ช. แสดงเส้นทางที่ถูกสร้างขึ้น.....	10
2.5 ก. แสดงการตรวจพบเส้นทางที่ถูกตัดขาด.....	11
2.5 ข. แสดงการซ่อมแซมเส้นทางของโปรโตคอล AODV.....	11
2.5 ค. แสดงเส้นทางใหม่ที่ทำให้การซ่อมแซมเส้นทางสำเร็จ.....	11
2.5 ง. แสดงการส่งแพ็คเก็ต RREP ตอบกลับ กรณีซ่อมแซมเส้นทางไม่สำเร็จ.....	11
2.6 แสดงโครงสร้างของแพ็คเก็ต RREQ.....	13
2.7 แสดงโครงสร้างของแพ็คเก็ต RREP.....	14
2.8 แสดงโครงสร้างของแพ็คเก็ต RERR.....	15
2.9 แสดงโครงสร้างของแพ็คเก็ต RREP-ACK.....	16
2.10 แสดงการส่งแพ็คเก็ตแจ้งเตือนแก่โหนดในกรณีที่พลังงานเริ่มอ่อนลง.....	23
3.1 แสดงการส่งแพ็คเก็ตแจ้งเตือนแก่โหนดก่อนหน้าในกรณีที่พลังงานแบตเตอรี่เริ่มอ่อนลง.....	29
3.2 การส่งแพ็คเก็ตแจ้งเตือนในกรณีที่โหนดเคลื่อนที่ห่างออกไปจากเส้นทาง.....	32
3.3 แสดงโครงสร้างของแพ็คเก็ตแจ้งเตือน.....	32
3.4 แสดงการส่งแพ็คเก็ตแจ้งเตือนแก่โหนดในกรณีที่สัญญาณเริ่มอ่อนลง.....	33
3.5 แสดงการส่งแพ็คเก็ต Sub_RREQ ไปยังโหนดข้างเคียง พร้อมทั้งลดจำนวน TTL ลงไปเรื่อย ๆ จนกว่าจะส่งถึงโหนดปลายทางที่ต้องการ.....	34
3.6 แสดงการส่งแพ็คเก็ต RREP กลับไปยังโหนดที่ได้รับการแจ้งเตือน เพื่อยืนยันเส้นทางสำรอง.....	34
3.7 แสดงโหนดที่ได้รับการแจ้งเตือน Sub_RREQ (6) ไม่อยู่ในระยะที่สามารถติดต่อกับโหนดถัดไป (3) (4) หรือ (D) ได้.....	35

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.8 แสดงการส่งแพ็คเกจ RERR ไปยังโหนดค้นหา.....	35
3.9 แสดงโครงสร้างของแพ็คเกจ Sub_RREQ	36
3.10 แสดงการส่งแพ็คเกจ Sub_RREQ	38
3.11 แสดงการรับแพ็คเกจ Sub_RREQ	39
3.12 แสดงโครงสร้างของแพ็คเกจ Sub_RREP	40
3.13 แสดงการรับแพ็คเกจ Sub_RREP	41
3.14 แสดงการรับแพ็คเกจแจ้งเตือน.....	43
4.1 อัตราการส่งข้อมูลสำเร็จในสภาพแวดล้อมที่ 1.....	49
4.2 จำนวนแพ็คเกจที่ถูกทิ้งในการจำลองสภาพแวดล้อมที่ 1.....	50
4.3 จำนวนแพ็คเกจควบคุมในการจำลองสภาพแวดล้อมที่ 1.....	50
4.4 ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลในการจำลองสภาพแวดล้อมที่ 1.....	51
4.5 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในการจำลองสภาพแวดล้อมที่ 1.....	52
4.6 อัตราการส่งข้อมูลสำเร็จในสภาพแวดล้อมที่ 2.....	53
4.7 จำนวนแพ็คเกจที่ถูกทิ้งในการจำลองสภาพแวดล้อมที่ 2.....	53
4.8 จำนวนแพ็คเกจควบคุมที่ใช้ในการค้นหาเส้นทางในสภาพแวดล้อมที่ 2.....	54
4.9 ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลในสภาพแวดล้อมที่ 2.....	55
4.10 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในการจำลองสภาพแวดล้อมที่ 2.....	55
4.11 อัตราการส่งข้อมูลสำเร็จในสภาพแวดล้อมที่ 3.....	56
4.12 จำนวนแพ็คเกจที่ถูกทิ้งในการจำลองสภาพแวดล้อมที่ 3.....	57
4.13 จำนวนแพ็คเกจควบคุมในการจำลองสภาพแวดล้อมที่ 3.....	57
4.14 ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลในการจำลองสภาพแวดล้อมที่ 3.....	58
4.15 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในการจำลองสภาพแวดล้อมที่ 3.....	59
4.16 จำนวนของเส้นทางที่ถูกตัดขาดในการทดลองปรับเปลี่ยนค่าเรชโฮลด์ของแบดเตอร์.....	61
4.17 จำนวนแพ็คเกจควบคุมในการจำลองโดยการปรับเปลี่ยนค่าเรชโฮลด์ของแบดเตอร์.....	61
4.18 จำนวนของเส้นทางที่ถูกตัดขาดในการจำลองโดยการปรับเปลี่ยนค่าเรชโฮลด์ของระดับ สัญญา.....	63
4.19 จำนวนแพ็คเกจควบคุมในการจำลองโดยการปรับเปลี่ยนค่าเรชโฮลด์ของระดับสัญญา.....	63

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันระบบเครือข่ายไร้สายได้ช่วยเอื้ออำนวยให้เกิดความสะดวกสบายในการสื่อสารและการทำงานต่อผู้ใช้เป็นอย่างมาก ผู้ใช้สามารถย้ายตำแหน่งการเชื่อมต่อ หรือเคลื่อนที่ในขณะที่สื่อสารได้อย่างเป็นอิสระ โดยระบบเครือข่ายไร้สายมีรูปแบบที่แบ่งตามลักษณะทางตรรกะออกเป็น 2 รูปแบบ คือแบบที่มีโครงสร้างพื้นฐาน (Infrastructure-based) เป็นรูปแบบที่เครื่องลูกข่าย (End Node) จะเชื่อมโยงกันผ่านตัวกลาง (Access Point) ซึ่งเป็นตัวควบคุมการสื่อสาร ส่วนอีกรูปแบบคือ ระบบเครือข่ายไร้สายแบบแอดฮอค (Ad Hoc Network)[2] ซึ่งรูปแบบนี้จะไม่มีความควบคุมกลาง และอาศัยโหนดสื่อสารทำงานเป็นทั้งเครื่องลูกข่ายและตัวกลางในการสื่อสาร (Intermediate Node) ในคราวเดียวกัน ซึ่งระบบเครือข่ายแบบแอดฮอคนี้เหมาะสมสำหรับการสื่อสารข้อมูลในสถานะแวดล้อมที่ไม่เอื้ออำนวยต่อการจัดตั้งโครงสร้างพื้นฐานแบบถาวร เช่น ในพื้นที่ทุรกันดารห่างไกล หรือพื้นที่ที่มีการจัดให้มีการสัมมนาหรือจัดกิจกรรมแบบชั่วคราว หรือเมื่อมีเหตุจำเป็นเมื่อโครงสร้างพื้นฐานไม่สามารถทำงานให้บริการการสื่อสารได้ เช่น เมื่อเกิดภัยพิบัติต่าง ๆ เป็นต้น

แต่ปัญหาหลักในการสื่อสารในระบบเครือข่ายไร้สายแบบแอดฮอครู้ก็คือ การรักษาเสถียรภาพของการสื่อสารให้ผู้ใช้สามารถใช้งานระบบเครือข่ายได้อย่างต่อเนื่องตลอดช่วงเวลาสื่อสารอยู่โดยช่องทางการสื่อสารไม่เสียหายไป เพราะโหนดสื่อสารต่างก็มีอิสระในการเคลื่อนที่และบางโหนดอาจจะเคลื่อนที่ตลอดเวลาด้วยความเร็วในการเคลื่อนที่ที่แตกต่างกันออกไปทำให้เส้นทางการสื่อสารมีโอกาสได้รับความเสียหายได้ตลอดเวลา ดังนั้นระบบเครือข่ายไร้สายแบบแอดฮอครู้ควรมีวิธีในการค้นหาเส้นทางสำรองที่สามารถหาเส้นทางได้รวดเร็ว และมีความคงทนเมื่อเส้นทางรับส่งข้อมูลต้องเผชิญกับสถานะที่เสี่ยงต่อการถูกตัดขาดการเชื่อมต่อ เช่น อุปกรณ์ไร้สายเคลื่อนที่ห่างออกไปจากขอบเขตของสัญญาณ หรือแบตเตอรี่ของอุปกรณ์ไร้สายหมด

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วัตถุประสงค์ของวิทยานิพนธ์เรื่องกระบวนการสร้างเส้นทางสำรองแบบ Multi-hop AODV-2T นี้ก็เพื่อเพิ่มความสามารถในการสร้างเส้นทางสำรองเพื่อให้รองรับการทำงานแบบหลายฮอป โดยปรับปรุงและพัฒนาจากรูปแบบการทำงานของโปรโตคอล AODV-2T ซึ่งปรับกระบวนการหาเส้นทางของการสื่อสารแบบแอดฮอค เพื่อให้มีการสร้างเส้นทางสำรองและเปลี่ยนไปใช้เส้นทางสำรองก่อนที่จะเกิดความเสียหายต่อเส้นทางหลักจนทำให้การสื่อสารของผู้ใช้หยุดชะงัก

เอกสารนี้จัดทำขึ้นที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 สมมติฐานของการศึกษา

โปรโตคอล AODV-2T สามารถปรับปรุงเสถียรภาพของการสื่อสารแบบแอ็ดฮอคได้โดยการตรวจสอบระดับสัญญาณการเชื่อมต่อ และระดับพลังงานแบตเตอรี่ของโหนดสื่อสารในเส้นทาง เพื่อให้มีการสร้างเส้นทางสำรองและสลับไปใช้เส้นทางสำรองก่อนที่จะเกิดความเสียหายต่อเส้นทางหลักจนทำให้การสื่อสารของผู้ใช้หยุดชะงักลงได้ทันท่วงที อย่างไรก็ตามโปรโตคอล AODV-2T ไม่สามารถรองรับรูปแบบการสร้างเส้นทางสำรองที่มีระยะทางมากกว่า 1 ฮอปได้ ดังนั้นหากมีการปรับปรุงให้โปรโตคอล AODV-2T สามารถรองรับการสร้างเส้นทางย่อยสำรองที่มีระยะทางมากกว่า 1 ฮอปได้ (Multi-hop AODV-2T) ซึ่งน่าจะเป็นการเพิ่มโอกาสในการค้นหาเส้นทางย่อยใหม่ทำให้ประสิทธิภาพในการซ่อมแซมเส้นทางเพิ่มสูงขึ้นได้

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

วิทยานิพนธ์ฉบับนี้ได้นำเสนอแนวคิดในการแจ้งเตือนล่วงหน้าก่อนที่เส้นทางเชื่อมต่อจะถูกตัดขาดให้สามารถรองรับการหาเส้นทางย่อยสำรองแบบหลายฮอป โดยพัฒนามาจากการทำงานของโปรโตคอล AODV-2T ซึ่งมีการกำหนดค่าเรชโฮลด์ (Threshold) ของพลังงานแบตเตอรี่ และระดับสัญญาณที่ได้รับ โดยจะกำหนดไว้เป็นสองระดับ ระดับแรกไว้เพื่อแจ้งเตือนแก่โหนดที่อยู่ในเส้นทางรับ-ส่งข้อมูลให้ทำการเตรียมหาเส้นทางสำรองไว้ ส่วนระดับที่สองไว้สำหรับเปลี่ยนมาใช้เส้นทางสำรองที่ได้เตรียมไว้ก่อนเส้นทางเดิมจะถูกตัดขาด ซึ่งจะช่วยให้ไม่ต้องเริ่มหาเส้นทางในการรับ-ส่งข้อมูลใหม่ ซึ่งในการหาเส้นทางสำรองนี้จะใช้แนวคิดในการใช้โหนดใกล้เคียง (Neighbor Node) มาเป็นโหนดสำรองถ้าโหนดนั้นสามารถส่งข้อมูลไปยังโหนดปลายทางได้ ซึ่งในวิทยานิพนธ์เล่มนี้ได้นำหลักการดังกล่าวมาพัฒนาให้รองรับรูปแบบการสร้างเส้นทางย่อยสำรองแบบหลายฮอป และใช้การจำลองผลด้วยโปรแกรมจำลองผลการทำงาน (Computer Simulation) เพื่อเปรียบเทียบวิธีการที่นำเสนอกับโปรโตคอล AODV แบบพื้นฐานและโปรโตคอล AODV-2T

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

เพื่อตรวจวัดประสิทธิภาพของโปรโตคอล Multi-Hop AODV-2T โดยนำผลจำลองการทำงานระบบเครือข่ายมาเปรียบเทียบกับผลการทำงานของโปรโตคอล AODV แบบพื้นฐาน และโปรโตคอล AODV-2T ด้วยการพิจารณาเปรียบเทียบในส่วนของปริมาณของแพ็คเก็ตที่ถูกทิ้งว่าลดลงหรือไม่ นอกจากนั้นยังเปรียบเทียบจำนวนแพ็คเก็ตที่ได้รับ และจำนวนแพ็คเก็ตควบคุมที่ถูกส่งออกไปในระบบ เพื่อนำผลที่ได้ทั้งหมดมาเปรียบเทียบหาประสิทธิภาพของการรับ-ส่งข้อมูล

1.6 ขอบเขตการวิจัย

วิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการปรับปรุงโปรโตคอล AODV-2T ให้สามารถรองรับการสร้างเส้นทางย่อยสำรองแบบหลายฮอป เปรียบเทียบกับโปรโตคอล AODV แบบพื้นฐาน และโปรโตคอล AODV-2T โดยการสร้างแบบจำลองของโปรโตคอล Multi-hop AODV-2T และจำลองผลด้วยโปรแกรม NS-2 (Network simulation Version 2)[3] ผลที่ได้แสดงจำนวนแพ็คเกจที่ถูกส่งออกไปในระบบ จำนวนแพ็คเกจที่รับได้ในระบบ อัตราส่วนของการส่งข้อมูลสำเร็จ จำนวนแพ็คเกจที่ถูกลบทิ้ง จำนวนแพ็คเกจควมคุมในระบบ และดีเลย์ (Delay) ที่เกิดขึ้นในระบบ

1.7 ขั้นตอนของการศึกษา

- วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บท คือ
- บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ ขอบเขตของการวิจัย และขั้นตอนการศึกษา
 - บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในการวิจัย และพื้นฐานของเครือข่ายไร้สายชนิด แอ็ดฮอค ซึ่งประกอบด้วย โปรโตคอลที่ใช้ในเครือข่าย พื้นฐานของโปรโตคอล AODV และโปรโตคอล AODV-2T
 - บทที่ 3 กล่าวถึงการปรับปรุงโปรโตคอล Multi-hop AODV-2T
 - บทที่ 4 กล่าวถึงการหาค่าประสิทธิภาพของระบบ พารามิเตอร์ที่ใช้ และผลที่ได้จากการจำลองระบบ เพื่อแสดงให้เห็นว่าวิธีการที่นำเสนอสามารถช่วยให้ระบบมีประสิทธิภาพที่ดีขึ้น
 - บทที่ 5 บทสรุปผลการวิจัย

บทที่ 2

ทฤษฎีพื้นฐานที่ใช้ในการวิจัย

2.1 เครือข่ายเคลื่อนที่ไร้สายชนิดแอดฮอค

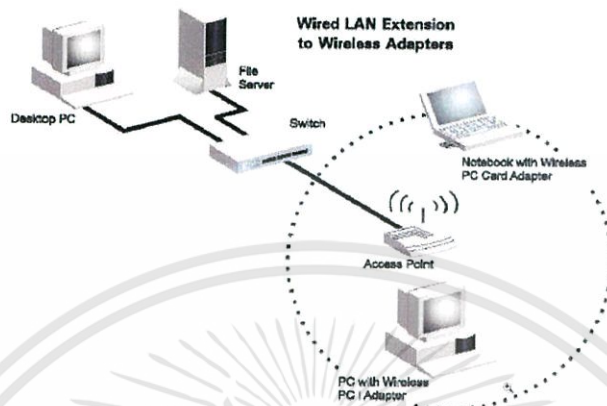
โครงข่ายเคลื่อนที่ไร้สายชนิดแอดฮอค (Mobile Ad hoc NETwork) หรือเขียนสั้น ๆ ว่า MANET เริ่มมีการศึกษาคิดค้นตั้งแต่ช่วงกลางทศวรรษที่ 1960 ซึ่งในเริ่มแรกใช้ชื่อว่า Packet Radio Network แต่ได้หยุดการพัฒนาไปในช่วงทศวรรษที่ 1980 เนื่องจากเทคโนโลยีด้านหน่วยประมวลผลและหน่วยความจำยังมีราคาแพงมาก ปัจจุบันปัญหาที่ว่านั้นได้หมดไปและแนวความคิดเรื่อง MANET กลับมาได้รับความสนใจและมีการพัฒนาอีกครั้งอย่างต่อเนื่องตั้งแต่ปี 1995 เป็นต้นมา

MANET คือ กลุ่มของโหนดเคลื่อนที่ไร้สาย (Wireless Mobile Nodes) รวมกันก่อให้เกิดเป็นโครงข่ายอิสระที่มีโครงสร้างการเชื่อมโยงแบบไร้สายทั้งหมด และมีลักษณะการทำงานแบบพลวัต (Dynamic Autonomous Network) โดยโหนดต่าง ๆ จะทำการติดต่อสื่อสารระหว่างกันได้โดยไม่ต้องผ่านการเชื่อมโยงที่สถานีฐาน (Base Stations) หรือแอคเซสพอยต์ (Access Points) ซึ่งก็หมายความว่า โหนดแต่ละโหนดจะทำหน้าที่เป็นได้ทั้งเราท์เตอร์ (Router) และโฮสต์ (Host) ในโหนดเดียวกันและเนื่องจากโหนดเคลื่อนที่ไร้สายมีข้อจำกัดในการส่งสัญญาณติดต่อกันภายในระยะทางแค่ช่วงระยะหนึ่ง ดังนั้นการติดต่อสื่อสารระหว่างโหนดต่าง ๆ ที่อยู่ในโครงข่ายจะส่งต่อกันเป็นทอด ๆ มีลักษณะที่เรียกว่า Multiple Hops

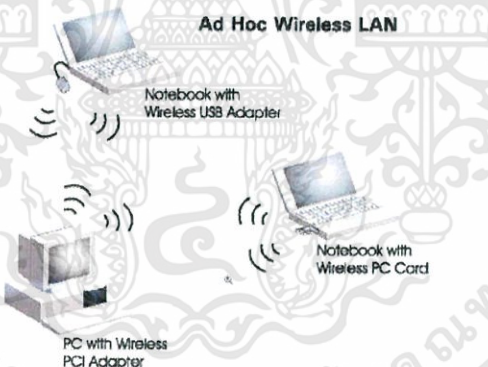
2.1.1 เปรียบเทียบความแตกต่างของระบบเครือข่ายไร้สายแบบแอดฮอคกับเครือข่ายไร้สายแบบอาศัยโครงสร้างพื้นฐาน

ระบบเครือข่ายไร้สาย (Wireless Local Area Network) คือระบบการสื่อสารข้อมูลที่มีความคล่องตัวมาก ซึ่งอาจจะนำมาใช้ทดแทนหรือเพิ่มต่อกับระบบเครือข่ายแบบใช้สายแบบดั้งเดิม โดยใช้การส่งคลื่นความถี่วิทยุในย่านวิทยุ RF และคลื่นอินฟราเรด ในการรับและส่งข้อมูลระหว่างคอมพิวเตอร์แต่ละเครื่องผ่านอากาศ ทะลุกำแพง เพดานหรือสิ่งก่อสร้างอื่น ๆ โดยปราศจากความต้องการของการเดินสาย นอกจากนี้ระบบเครือข่ายไร้สายก็ยังมีคุณสมบัติครอบคลุมทุกอย่างเหมือนกับระบบเครือข่ายแบบใช้สาย ที่สำคัญก็คือการไม่ใช้สายทำให้การเคลื่อนย้ายการใช้งานทำได้โดยสะดวก ไม่เหมือนระบบเครือข่ายแบบใช้สายที่ต้องใช้เวลาและการลงทุนในการปรับเปลี่ยนตำแหน่ง การใช้งานเครื่องคอมพิวเตอร์ในเรื่องความปลอดภัยของข้อมูลอุปกรณ์ไร้สายสามารถกำหนดให้มีการส่งข้อมูลด้วยวิธีการเข้ารหัสได้ เพื่อป้องกันการดักรับข้อมูลนอกจากนี้ระบบเครือข่ายไร้สายยังสามารถเชื่อมต่อกับระบบอินเทอร์เน็ต โดยผ่านอุปกรณ์ที่เรียกว่า แอคเซสพอยต์ ซึ่งถูกติดตั้งในบริเวณพื้นที่ให้บริการ แม้ว่าราคาของอุปกรณ์ที่

ใช้ในระบบเครือข่ายไร้สายจะมีราคาสูงกว่าของระบบเครือข่ายแบบใช้สาย แต่ระบบเครือข่ายแบบไร้สายสามารถติดตั้งและบำรุงรักษาได้ง่าย ลดค่าใช้จ่ายเรื่องสายสัญญาณ เหมาะสำหรับสถานที่ที่ติดตั้งสายสัญญาณได้ยากหรือต้องมีการเคลื่อนย้ายบ่อยๆ เช่น งานนิทรรศการ เป็นต้น



รูปที่ 2.1 แสดงเครือข่ายไร้สายแบบอาศัยโครงสร้างพื้นฐาน



รูปที่ 2.2 แสดงเครือข่ายไร้สายแบบแอ็ดฮอค

ลักษณะที่สำคัญของระบบเครือข่ายไร้สายแบบแอ็ดฮอค คือ Self-organizing และการปรับตัวได้ดีกับโครงสร้างใหม่ๆ นั่นคือ โหนดที่อยู่ในเครือข่ายแอ็ดฮอค โหนดทุกโหนดสามารถที่จะจัดการตนเองได้ในเรื่องต่าง ๆ ไม่ว่าจะเป็นการปรับปรุงข้อมูลของตนเองเกี่ยวกับโครงสร้างของเครือข่ายหรือการตัดสินใจต่าง ๆ เกี่ยวกับแพ็คเกต เป็นต้น โดยอุปกรณ์ภายในเครือข่ายไร้สายแบบแอ็ดฮอค ครอบคลุมถึงอุปกรณ์อื่น ๆ เช่น ปาล์ม พ็อกเก็ตพีซี โทรศัพท์มือถือ และอื่น ๆ

หากกลับมาพิจารณาระบบเครือข่ายแบบแอ็ดฮอคซึ่งไม่จำเป็นต้องมีอุปกรณ์แอกเซส

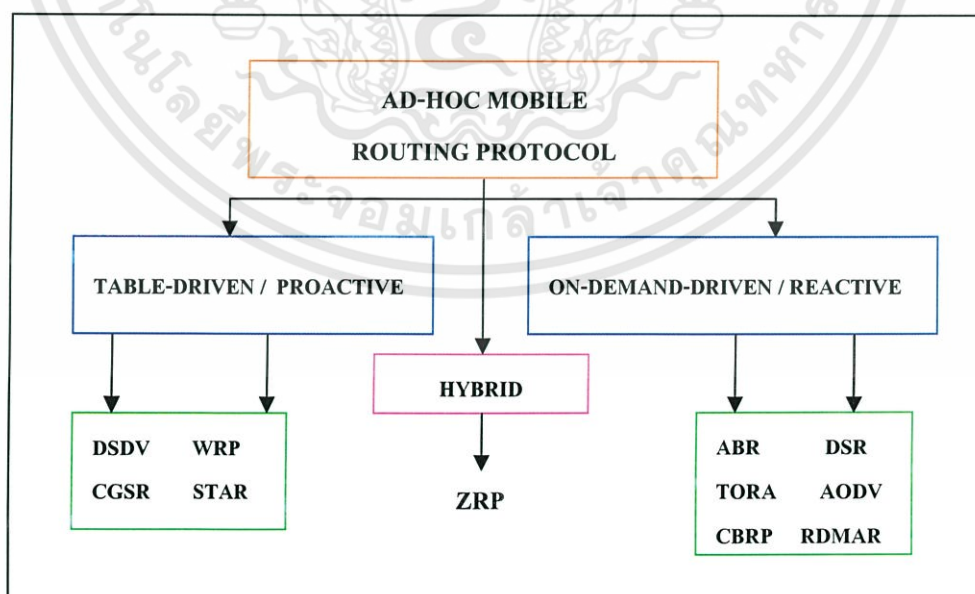
เอ็กสโพยต์ แล้ว จะสามารถใช้ประโยชน์เพิ่มขึ้นอีกมากมาย ตัวอย่างเช่น การประชุมกลางแจ้งเพียงแค่ว่า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีอุปกรณ์ตั้งแต่ 2 ตัวขึ้นไป ก็สามารถสร้างระบบเครือข่ายไร้สายได้ หรืออาจจะใช้ในเรื่องของการทหารเมื่ออยู่ในสนามรบก็สามารถใช้อุปกรณ์วิทยุสื่อสารของทหารเป็นโหนดในการติดต่อเป็นต้น

รูปแบบของระบบเครือข่ายไร้สายแบบแอ็ดฮอคมีทั้งหมด 3 รูปแบบ คือ รูปแบบที่หนึ่ง การติดต่อแบบ Peer to Peer คือมีอุปกรณ์สื่อสารจำนวน 1 คู่มาเชื่อมต่อกัน รูปแบบที่สองคือการติดต่อแบบ Remote to Remote คือมีอุปกรณ์สื่อสารตั้งแต่ 2 ตัวขึ้นไป เชื่อมต่อกันเป็นเครือข่ายในลักษณะที่แน่นอน อยู่กับที่และรูปแบบที่สามคือการติดต่อแบบลูกผสม (Hybrid) คือ อุปกรณ์สื่อสารตั้งแต่ 2 ตัวขึ้นไปมาเชื่อมต่อกันเป็นเครือข่าย แต่มีลักษณะการเคลื่อนที่ของโหนดในเครือข่ายไม่แน่นอน คาดเดาไม่ได้ โดยส่วนใหญ่แล้วเมื่อมีการใช้งานระบบเครือข่ายไร้สายแบบแอ็ดฮอค จะเป็นรูปแบบการติดต่อในแบบลูกผสมเพราะโหนดแต่ละโหนดส่วนใหญ่จะเป็นโหนดของอุปกรณ์ที่สามารถเคลื่อนที่ได้ง่าย และหากมีการเคลื่อนที่ของอุปกรณ์เหล่านั้นเกิดขึ้น ซึ่งจะทำให้เกิดโครงสร้างเปลี่ยนแปลงเสมอ เหตุการณ์เหล่านี้ทำให้เกิดองค์ประกอบสำคัญที่จะนำมาพิจารณาในการจัดตั้งระบบเครือข่ายไร้สายแบบแอ็ดฮอค ซึ่งมีดังต่อไปนี้

2.1.2 โพรโทคอลที่ใช้ในการค้นหาเส้นทาง (Routing Protocols)

การทำเส้นทางในระบบเครือข่ายไร้สายแบบแอ็ดฮอค แต่ละอุปกรณ์จะถูกมองเป็นโหนดสื่อสาร เมื่อต้องการจะติดต่อส่งข้อมูลจากโหนดหนึ่งไปยังอีกโหนดหนึ่ง จะต้องมีการส่งข้อความร้องขอเส้นทาง เพื่อทำการขอจัดตั้งเส้นทางเพื่อใช้ในการรับ-ส่งข้อมูลไปยังเครื่องปลายทางได้



รูปที่ 2.3 แสดงการแบ่งประเภทของระบบเครือข่ายไร้สายชนิดแอ็ดฮอค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอลที่ใช้ในการหาเส้นทางในระบบเครือข่ายไร้สายชนิดแอดฮอค สามารถแบ่งได้เป็น 3 ประเภทคือ

- 2.1.1 โปรโตคอลค้นหาเส้นทางแบบโปรแอกทีฟ (Proactive Protocol) เป็นโปรโตคอลที่ต้องมีการค้นหาเส้นทางอยู่ตลอดเวลา ด้วยวิธีการส่งแพ็คเก็ตควบคุมการค้นหาเส้นทางออกไปทุก ๆ คาบเวลา ไม่ว่าจะรูปร่างของโครงข่ายจะเปลี่ยนแปลงไปหรือไม่ ทั้งนี้เพื่อให้ได้เส้นทางที่จะไปยังโหนดต่าง ๆ มาเก็บไว้ในตารางเส้นทาง เมื่อโหนดต้องการส่งข้อมูลก็สามารถที่จะใช้ข้อมูลเส้นทางที่มีอยู่มาใช้ในการรับ-ส่งข้อมูลได้ทันที ถ้าในตารางไม่มีข้อมูลเส้นทางไปโหนดปลายทาง ก็จะทำการค้นหาเส้นทางใหม่ทันที เพื่อให้ได้มาซึ่งเส้นทางในการรับ-ส่งข้อมูล โดยใช้วิธีการส่งแบบกระจายแพ็คเก็ต (Broadcast) โปรโตคอลชนิดนี้ จะไม่มีค่าหน่วงเวลา (Delay) สำหรับการค้นหาเส้นทางเมื่อต้องการส่งข้อมูล เพราะว่ามีข้อมูลของเส้นทางในการรับ-ส่งข้อมูลอยู่ในตารางเส้นทางอยู่แล้ว ตัวอย่างของโปรโตคอลชนิดนี้คือ (DSDV) [7], OLSR [8], WRP [9]
- 2.1.2 โปรโตคอลที่จะค้นหาเส้นทางแบบรีแอกทีฟ (Reactive Protocol) เป็นโปรโตคอลที่จะค้นหาข้อมูลเส้นทางก็ต่อเมื่อโหนดต้นทางต้องการส่งข้อมูลไปยังโหนดปลายทาง ซึ่งจะไม่มีการเตรียมเส้นทางไว้ในตารางเส้นทางก่อนล่วงหน้า ดังนั้นในช่วงเวลาที่มีการค้นหาเส้นทางที่ใช้ส่งข้อมูลจะมีค่าหน่วงเวลาเกิดขึ้น การค้นหาเส้นทางของโปรโตคอลชนิดนี้ จะมีการใช้ข้อมูลควบคุมในการค้นหาเส้นทางได้อย่างมีประสิทธิภาพมากกว่าโปรโตคอลแบบโปรแอกทีฟที่ใช้เทคนิคการส่งทุก ๆ คาบเวลา เพราะโหนดต่าง ๆ ที่ไม่เกี่ยวข้องกับการรับส่งข้อมูลก็จะต้องเก็บข้อมูลเส้นทางที่ไม่จำเป็น และมีเพียงโหนดในเส้นทางหลักเท่านั้นที่จะเก็บข้อมูลเส้นทางเพื่อใช้ในการรับ-ส่งข้อมูล ซึ่งเป็นการใช้หน่วยความจำให้มีประสิทธิภาพมากขึ้นด้วย ตัวอย่างของโปรโตคอลชนิดนี้คือ DSR, AODV
- 2.1.3 Hybrid เป็นการใช้งานทั้ง 2 โปรโตคอลร่วมกัน

2.2. โปรโตคอล AODV

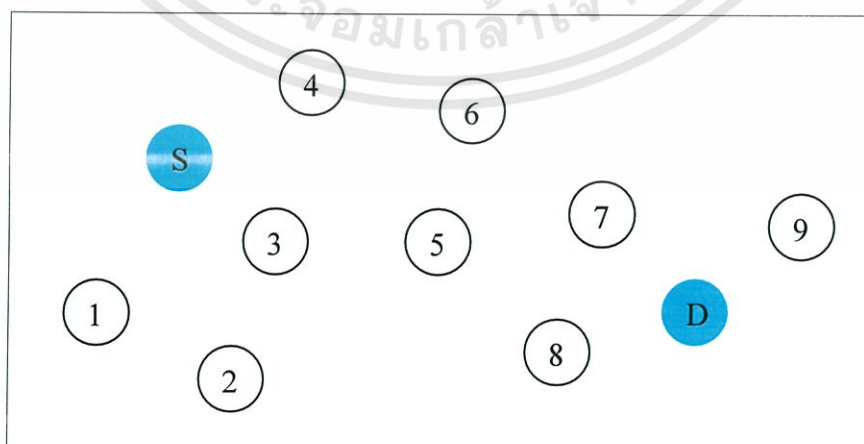
โปรโตคอล AODV (Ad Hoc On-demand Distance Vector) เป็นโปรโตคอลที่ต้องมีการสร้างเส้นทางก่อนที่จะเริ่มการส่งข้อมูล เพราะเป็นแบบรีแอกทีฟโปรโตคอล ซึ่งไม่มีการสร้างตารางเส้นทางไว้ล่วงหน้า ทำให้สามารถตอบสนองต่อการเปลี่ยนแปลงของโครงข่ายได้เป็นอย่างดี และได้มีการนำเลขลำดับปลายทาง (Destination Sequence Number) มาใช้เพื่อป้องกันการเกิดลูป และหลีกเลี่ยงการนับไม่สิ้นสุด (Count to Infinity) ที่มักจะเกิดขึ้นในโปรโตคอลชนิด

2.2.1 การค้นหาเส้นทางของโปรโตคอล AODV

ในขั้นตอนการค้นหาเส้นทางใหม่ของโหนดต้นทางที่ต้องการจะส่งข้อมูลไปยังโหนดปลายทาง โหนดต้นทางจะกระจายแพ็คเกจร้องขอเส้นทาง (Route Request Packet: RREQ) ออกไปในเครือข่าย เมื่อแต่ละโหนดในเครือข่ายได้รับแพ็คเกจร้องขอเส้นทางก็จะสร้างหรือปรับปรุงตารางเส้นทางขากลับ (Reverse Route) ไปยังโหนดต้นทาง ถ้าไม่สามารถหาเส้นทางกลับไปถึงโหนดปลายทางได้ โหนดต้นทางจะทำการกระจายแพ็คเกจร้องขอใหม่อีกครั้ง

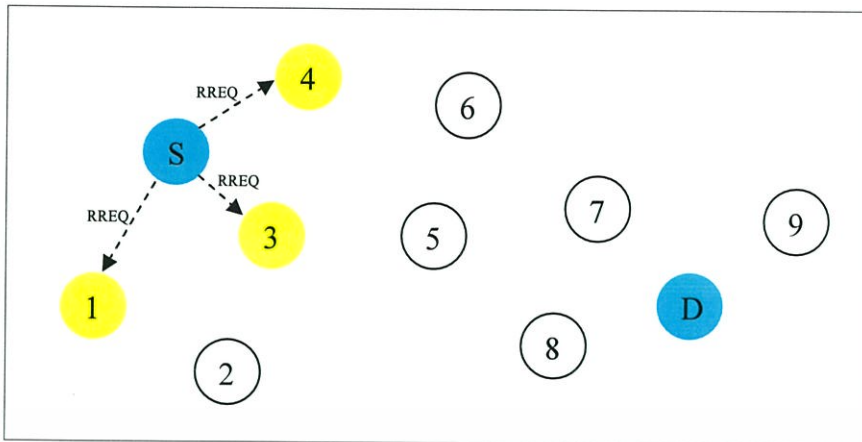
เมื่อแพ็คเกจร้องขอได้ถูกกระจายออกไปจนถึงโหนดปลายทาง ซึ่งในแต่ละโหนดระหว่างทางเมื่อได้รับแพ็คเกจร้องขอก็จะสร้างตารางเส้นทางขากลับไว้แล้ว ดังนั้นเมื่อโหนดปลายทางได้รับแพ็คเกจร้องขอแล้วก็จะส่งแพ็คเกจตอบรับ (Route Reply Packet: RREP) กลับไปให้โหนดต้นทางตามตารางเส้นทางขากลับ เมื่อแต่ละโหนดได้รับแพ็คเกจตอบรับแล้วจะทำการอัปเดตหรือสร้างตารางเส้นทางส่งข้อมูล (Forward Route) ที่จะส่งไปยังโหนดปลายทางเอาไว้ หลังจากนั้นก็ส่งแพ็คเกจตอบรับกลับไปตามเส้นทางขากลับไปเรื่อย ๆ จนกระทั่งถึงโหนดต้นทาง โหนดต้นทางก็จะทำการปรับปรุงหรือสร้างตารางเส้นทางในการรับ-ส่งข้อมูลแล้วจะจึงเริ่มทำการรับ-ส่งข้อมูลกับโหนดปลายทาง

ซึ่งในแพ็คเกจร้องขอ (RREQ) จะระบุข้อมูลของหมายเลขสำคัญของโหนดต้นทาง (Source Identifier: SrcID) หมายเลขสำคัญของโหนดปลายทาง (Destination Identifier: DestID) เลขลำดับของโหนดต้นทาง (Source Sequence Number: SrcSeqNum) หมายเลขลำดับของโหนดปลายทาง (Destination Sequence Number: DesSeqNum) หมายเลขของการส่งกระจาย (Broadcast Identifier: BcastID) อายุของแพ็คเกจ (Time to Live: TTL) หมายเลขลำดับของโหนดปลายทางจะบอกถึงความใหม่ของเส้นทาง ถ้าเส้นทางที่ได้รับมีมากกว่าหนึ่งเส้นทาง โหนดก็จะเลือกเอาค่าหมายเลขลำดับโหนดปลายทางที่มากกว่า ถ้าโหนดได้รับแพ็คเกจร้องขอหลายครั้งก็จะไม่สนใจแพ็คเกจนั้น โดยดูจาก Broadcast Identifier และ Source Identifier

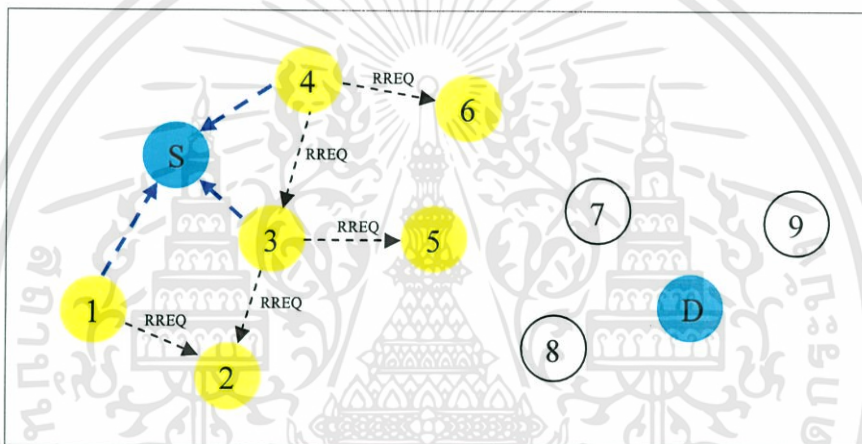


รูปที่ 2.4 ก. แสดงการค้นหาเส้นทางในโปรโตคอล AODV

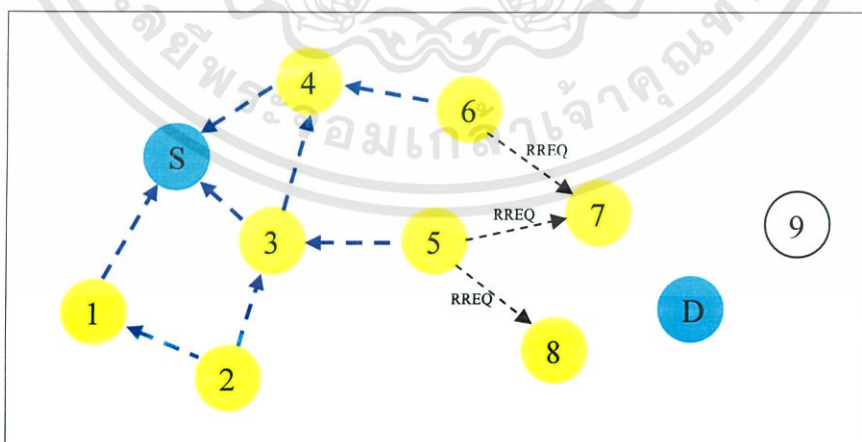
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ข. แสดงการส่งแพ็คเก็ต RREQ เพื่อร้องขอเส้นทาง

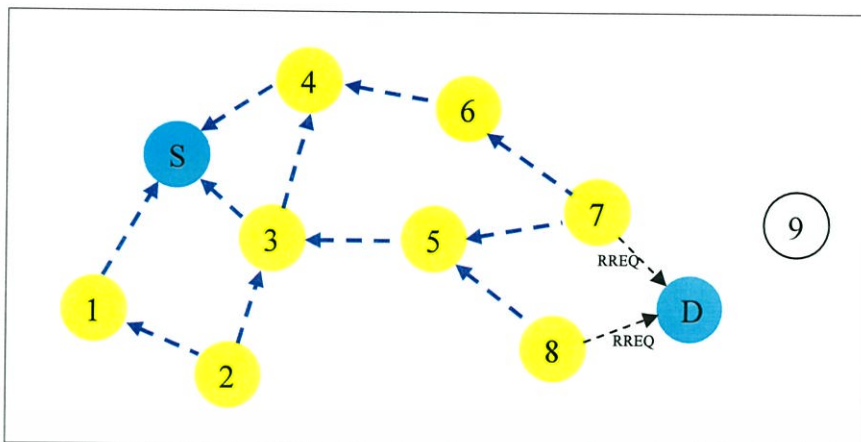


รูปที่ 2.4 ค. แสดงการส่งแพ็คเก็ต RREQ ต่อไปเป็นทอด ๆ เพื่อร้องขอเส้นทาง

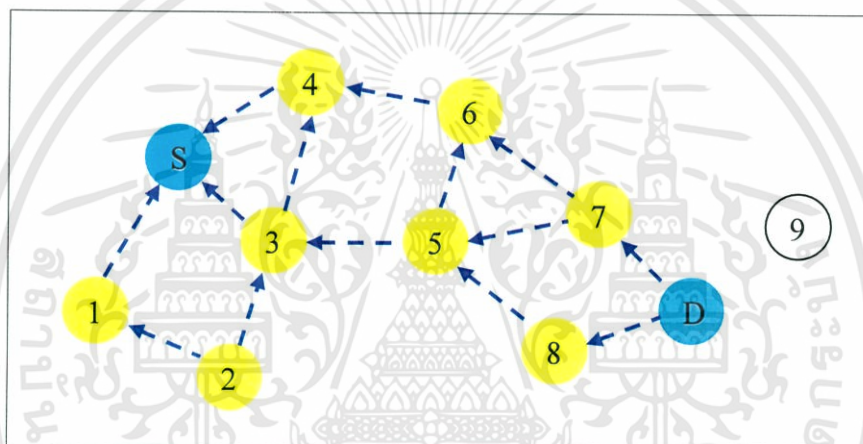


รูปที่ 2.4 ง. แสดงการส่งแพ็คเก็ต RREQ ตอบกลับ

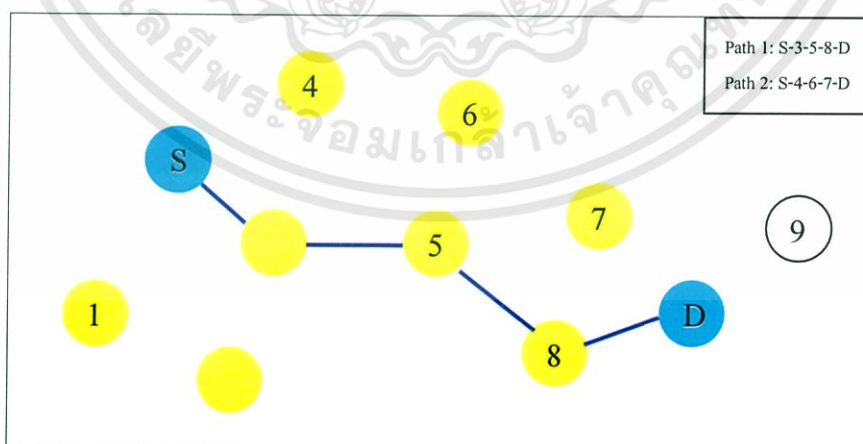
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 จ. แสดงการส่งแพ็คเกจ RREQ ไปจนถึง โหนดปลายทาง



รูปที่ 2.4 ฉ. แสดงการส่งแพ็คเกจ RREP ตอบกลับไปยังโหนดต้นทาง



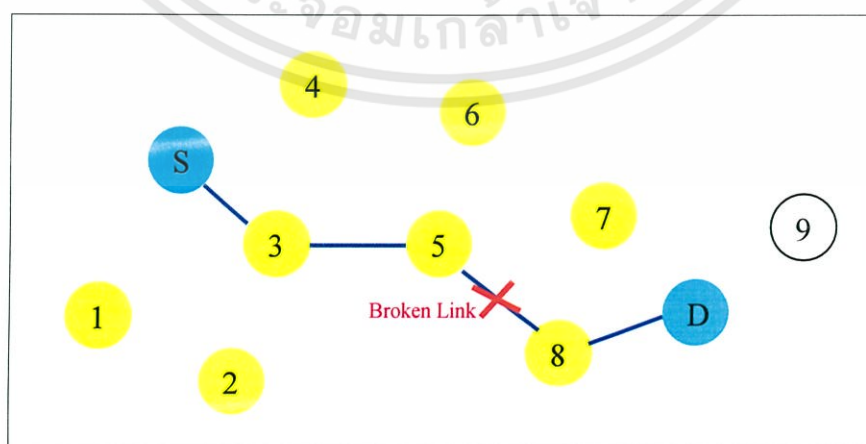
รูปที่ 2.4 ช. แสดงเส้นทางถูกสร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.4 ก เมื่อโหนด S ต้องการส่งข้อมูลไปให้โหนด D ก็จะเริ่มต้นโดยการกระจายแพ็คเกจ RREQ (Route Request Packet: RREQ) ออกไปยังโหนดเพื่อนบ้าน โดยสมมุติให้มีหมายเลขลำดับของโหนดปลายทางเป็น 3 และมีหมายเลขลำดับของโหนดต้นทางเป็น 1 เมื่อโหนด 1 โหนด 3 และโหนด 4 ได้รับแพ็คเกจ RREQ ก็จะตรวจสอบว่าตัวเองมีเส้นทางไปยังโหนดปลายทางหรือไม่ ในกรณีนี้จะเห็นว่าทั้งสามโหนดไม่มีเส้นทางไปยังโหนดปลายทาง ดังนั้นจึงต้องส่งแพ็คเกจ RREQ ไปยังโหนดข้างเคียงต่อไปคือโหนด 2 โหนด 5 และโหนด 6 สมมุติให้โหนด 5 และโหนด 6 มีเส้นทางไปยังโหนดปลายทาง โดยโหนด 5 มีเส้นทางเป็น 5-8-D และโหนด 6 มีเส้นทางเป็น 6-7-D ถ้าหมายเลขลำดับของโหนดปลายทางของโหนด 6 เป็น 2 และของโหนด 5 เป็น 4 ซึ่งแสดงให้เห็นว่าโหนด 5 เป็นเส้นทางที่ใหม่กว่า ดังนั้นโหนดต้นทางจะเลือกเส้นทางจากโหนด 5 เก็บไว้ในตารางเส้นทางหากไม่มีเส้นทางที่ใหม่กว่านี้ ถ้าแพ็คเกจ RREQ ถูกส่งไปจนถึงโหนดปลายทางตามเส้นทาง 8-D โหนดปลายทางจะส่งแพ็คเกจ RREP ตอบกลับมา ในกรณีที่มี RREP มากกว่าหนึ่งแพ็คเกจ โหนดต้นทางจะเลือกเส้นทางที่ใหม่กว่าจากเลขลำดับโหนดปลายทางที่มากกว่า และจะเลือกเส้นทางที่สั้นที่สุดโดยนับจากจำนวนฮอป

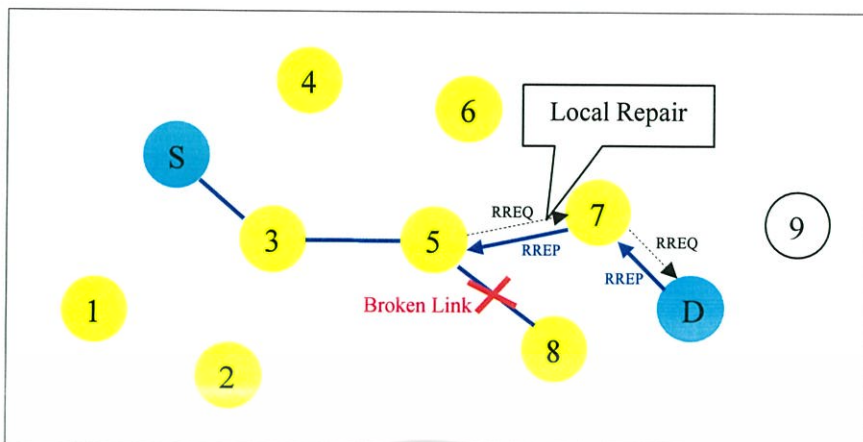
2.2.2 การซ่อมแซมเส้นทาง

เมื่อเส้นทางที่เชื่อมต่อถูกตัดขาด โหนดที่ตรวจพบจะเริ่มทำโลคอลลีเพอร์ โดยการส่งแพ็คเกจร้องขอเส้นทางออกไปใหม่และเก็บแพ็คเกจข้อมูลที่ต้องการส่งไว้ในบัฟเฟอร์ เมื่อทำโลคอลลีเพอร์สำเร็จและได้รับแพ็คเกจตอบกลับจากโหนดปลายทาง โหนดที่ตรวจพบจึงจะเริ่มทำการส่งข้อมูลที่ได้เก็บไว้ในบัฟเฟอร์ออกไป แต่ถ้าทำโลคอลลีเพอร์ไม่สำเร็จโหนดดังกล่าวจะส่งแพ็คเกจผิดพลาด (Route Error Packet: RERR) ออกไปให้โหนดต้นทาง ซึ่งแพ็คเกจ RERR จะระบุถึงโหนดปลายทางที่ไม่สามารถส่งไปถึงได้ เมื่อโหนดต้นทางได้รับแพ็คเกจ RERR แล้วจะระบุให้เส้นทางรับส่งข้อมูลเป็นเส้นทางที่ใช้ไม่ได้แล้ว และเริ่มทำการหาเส้นทางใหม่ต่อไป



รูปที่ 2.5 ก. แสดงการตรวจพบเส้นทางเส้นทางที่ถูกตัดขาด

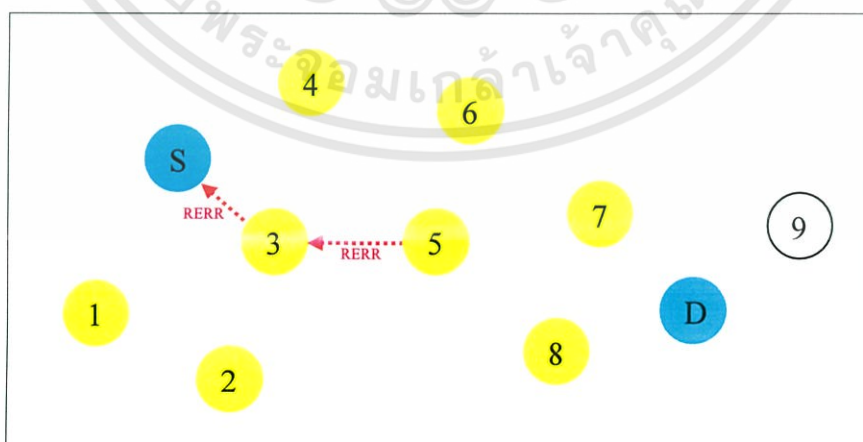
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ข. แสดงการซ่อมแซมเส้นทางของโปรโตคอล AODV



รูปที่ 2.5 ค. แสดงเส้นทางใหม่ที่ทำการซ่อมแซมเส้นทางสำเร็จ



รูปที่ 2.5 ง. แสดงการส่งแพ็คเกจ RERR ตอบกลับ กรณีซ่อมแซมเส้นทางไม่สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.12 เมื่อโหนด 5 พบเส้นทางที่ถูกตัดขาด โหนดที่ 5 จะทำโลคอลลีแพร์ก่อนลำดับแรกโดยส่งการแพ็คเก็ต RREQ ออกไป เมื่อไปถึงโหนดปลายทางโหนด D ก็จะส่งแพ็คเก็ต RREP ตอบกลับมาถือว่าทำโลคอลลีแพร์สำเร็จแล้วโหนด 5 จึงจะทำการส่งข้อมูลต่อไป แต่ถ้าไม่ได้รับ RREP ตอบกลับมาในเวลาที่กำหนดโหนด 5 จะส่งแพ็คเก็ต RERR กลับไปที่โหนดต้นทาง ดังรูปที่ 2.15 แล้วโหนดต้นทางจะทำการค้นหาเส้นทางใหม่ต่อไป

2.3.1 โครงสร้างแพ็คเก็ตของโปรโตคอล AODV

2.3.1.1. โครงสร้างของแพ็คเก็ตร้องขอเส้นทาง (Route Request Packet: RREQ)

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Type								J	R	G	D	U	Reserved										Hop Count								
RREQ ID																															
Destination IP Address																															
Destination Sequence Number																															
Originator IP Address																															
Originator Sequence Number																															

รูปที่ 2.6 แสดง โครงสร้างของแพ็คเก็ต RREQ

ตารางที่ 2.1 คำอธิบายรายละเอียดของฟิลด์ต่าง ๆ ในแพ็คเกจร้องขอเส้นทาง (RREQ)

Type	1
J (Join Flag)	ใช้สำหรับมัลติคาสต์
R (Repair Flag)	ใช้สำหรับมัลติคาสต์
G (Gratuitous RREP Flag)	เมื่อโหนดในเส้นทางได้รับแพ็คเกจนี้ ให้ส่ง RREP กลับไปยังโหนดต้นทาง
D (Destination Only Flag)	โหนดปลายทางเท่านั้นที่จะตอบกลับได้
U (Unknown sequence number)	แสดงว่าโหนดต้นทางไม่รู้ค่าเลขลำดับปลายทาง
Reserved	โหนดปลายทางจะไม่สนใจค่านี้เมื่อมีค่าเท่ากับศูนย์
Hop Count	จำนวนฮอปจากโหนดต้นทางไปยังโหนดปลายทาง
RREQ ID	ใช้ร่วมกับหมายเลขไอพีของโหนดต้นทางเพื่ออ้างอิงแพ็คเกจร้องขอ
Destination IP Address	หมายเลขไอพีของโหนดปลายทาง
Destination Sequence Number	เลขลำดับของโหนดปลายทาง
Originator IP Address	หมายเลขไอพีของโหนดต้นทาง
Originator Sequence Number	เลขลำดับของโหนดต้นทาง

2.3.1.2. โครงสร้างของแพ็คเกจตอบกลับ (Route Reply Packet: RREP)

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
Type										R	A	Reserved										Prefix Sz					Hop Count						
Destination IP Address																																	
Destination Sequence Number																																	
Originator IP Address																																	
Lifetime																																	

รูปที่ 2.7 แสดงโครงสร้างของแพ็คเกจ RREP

ตารางที่ 2.2 คำอธิบายรายละเอียดของฟิลด์ต่าง ๆ ในแพ็คเกจตอบกลับ (RREP)

Type	2
R	ใช้สำหรับมัลติคาสต์
A (Acknowledgement require)	ให้โหนดที่ได้รับส่งแพ็คเกจ RREP-ACK ตอบกลับมา
Reserved	โหนดปลายทางจะไม่สนใจค่านี้เมื่อมีค่าเท่ากับศูนย์
Prefix Sz	จะมีอยู่ 5 บิต ถ้าไม่เท่ากับศูนย์แสดงว่าฮอปถัดไปอาจจะถูกโหนดอื่นใช้เป็นเส้นทางส่งข้อมูลเส้นทางเดียวกัน
Hop Count	จำนวนฮอปจากโหนดต้นทางไปยังโหนดปลายทาง
Destination IP Address	หมายเลขไอพีของโหนดปลายทาง
Destination Sequence Number	เลขลำดับของโหนดปลายทาง
Originator IP Address	หมายเลขไอพีของโหนดต้นทาง
Lifetime	ช่วงเวลาที่เส้นทางสามารถใช้งานได้

2.3.1.3. โครงสร้างของแพ็คเกจควบคุมความผิดพลาด (Route Error Packet: RERR)

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Type								N	Reserved												Dest Count										
Unreachable Destination IP Address																															
Unreachable Destination Sequence Number																															
Additional Unreachable Destination IP Address																															
Additional Unreachable Destination Sequence Number																															

รูปที่ 2.8 แสดงโครงสร้างของแพ็คเกจ RERR

ตารางที่ 2.3 คำอธิบายรายละเอียดของฟิลด์ต่าง ๆ ในแพ็คเกจควบคุมความผิดพลาด (RERR)

Type	3
N	ห้ามลบเส้นทางรับ-ส่งข้อมูลในกรณีที่โหนดมีการซ่อมแซมเส้นทาง
Reserved	โหนดปลายทางจะไม่สนใจค่านี้เมื่อมีค่าเท่ากับศูนย์
Dest Count	จำนวนโหนดปลายทางที่ไม่สามารถส่งข้อมูลได้ ต้องมีอย่างน้อยเท่ากับหนึ่ง
Unreachable Destination IP Address	คือหมายเลขไอพีของโหนดปลายทางที่ไม่สามารถส่งข้อมูลได้
Unreachable Destination Sequence Number	คือหมายเลขลำดับของเส้นทางที่ไปยังโหนดปลายทางที่ไม่สามารถส่งข้อมูลได้

2.3.1.4. โครงสร้างของแพ็คเกจรับทราบการตอบกลับ (Route Reply Acknowledgment: RREP-ACK)

โหนดที่ได้รับแพ็คเกจ RREP ที่กำหนดเฟล็กเป็น A จะต้องส่งแพ็คเกจ RREP-ACK กลับไปเพื่อยืนยันตัวเอง ซึ่งในกรณีนี้จะใช้เมื่อโหนดไม่ทราบว่าจะส่งแพ็คเกจ RREP ไปที่โหนด (Unidirectional Link) ใช้เพื่อป้องกันการหาเส้นทางใหม่ซ้ำ ๆ (Route Discovery Cycle)

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
TYPE									Reserve						

รูปที่ 2.9 แสดงโครงสร้างของแพ็คเกจ RREP-ACK

2.3.2 การทำงานของโปรโตคอล AODV

ในหัวข้อนี้จะอธิบายถึงอัลกอริทึมต่าง ๆ ที่ใช้ในโปรโตคอล AODV ซึ่งจะอธิบายถึงการจัดการเกี่ยวกับเลขลำดับ การส่งแพ็คเกจ RREQ RREP และ RERR และจะกล่าวถึงการซ่อมแซมเส้นทางด้วย

2.3.2.1 การจัดการเลขลำดับ (Sequence Number)

ในตารางเส้นทางในทุก ๆ โหนดจะต้องมีค่าเลขลำดับล่าสุดของเส้นทางเก็บไว้ ซึ่งเลขลำดับนี้ก็คือเลขลำดับปลายทางหรือ Destination Sequence Number ซึ่งจะปรับปรุงเมื่อโหนดได้รับข้อมูลเลขลำดับที่มาจากแพ็คเกจ RREQ RREP หรือ RERR ซึ่งมีความเกี่ยวข้องกับโหนดปลายทาง โดยโปรโตคอล AODV จะปรับปรุงข้อมูลค่าเลขลำดับปลายทางให้เป็นข้อมูลที่ใหม่ที่สุดและป้องกันการเกิดลูปในเส้นทางการส่งข้อมูล ในการเพิ่มค่าเลขลำดับของโหนดต้นทางและโหนดปลายทางนั้นจะดำเนินการดังขั้นตอนต่อไปนี้

- ก่อนที่โหนดต้นทางจะทำการหาเส้นทาง ก็จะเพิ่มค่าเลขลำดับของตัวเอง (Originator Sequence Number) เพื่อป้องกันการสับสนกับกับเลขลำดับของเส้นทางก่อนหน้านี้ที่เป็นเส้นทางจากกลับมายังโหนดต้นทาง
- ก่อนที่โหนดปลายทางจะสร้างแพ็คเกจ RREP ก็จะต้องเพิ่มเลขลำดับปลายทาง (Destination Sequence Number) ก่อน

ถ้าเลขลำดับมีค่าเพิ่มขึ้นไปจนเป็นค่ามากที่สุดซึ่งเป็นเลขจำนวนเต็มแบบไม่มีเครื่องหมายจำนวน 32 บิตซึ่งจะมีค่ามากที่สุดอยู่ที่ 4294967295 เมื่อเลขลำดับถึงค่านี้อแล้วค่าถัดไปก็จะให้มีค่าเป็นศูนย์

2.3.2.2 ตารางเส้นทางของโปรโตคอล AODV

โปรโตคอล AODV มีการเก็บข้อมูลของเส้นทางไว้ในตารางเส้นทางซึ่งจะประกอบไปด้วยข้อมูลเส้นทางตามตารางที่ 2.4

ตารางที่ 2.4 ตารางเส้นทางของโปรโตคอล AODV

Destination IP Address	หมายเลขไอพีของโหนดปลายทาง
Destination Sequence Number	เลขลำดับโหนดปลายทาง
Hop Count	จำนวนฮอปจากโหนดต้นทางไปยังโหนดปลายทาง
Next Hop	ฮอปถัดไปในเส้นทาง
Precursor List	รายการไอพีแอดเดรสของโหนดก่อนหน้านี้
Route Expire	ช่วงเวลาเส้นทางสามารถใช้งานได้
Route Flag	ใช้สำหรับบอกสถานะของเส้นทาง ซึ่งได้แก่ <ul style="list-style-type: none"> • RTF_DOWN หมายถึงเป็นเส้นทางขากลับใช้สำหรับใช้สำหรับส่งแพ็คเกตกลับไปต้นทาง • RTF_UP หมายถึงเป็นเส้นทางขาไป ใช้สำหรับส่งแพ็คเกตไปยังโหนดปลายทาง RTF_IN_REPAIR หมายถึงเส้นทางอยู่ในระหว่างการซ่อมแซม ซึ่งจะใช้ในการ โลกอลรีเพร์ ถ้าเส้นทางอยู่ในสถานะนี้โหนดจะหยุดการสื่อสารชั่วคราวและเก็บแพ็คเกตข้อมูลไว้ในบัฟเฟอร์ก่อน

เมื่อโหนดระหว่างทางได้รับแพ็คเกตควบคุมจากโหนดข้างเคียง จะทำการตรวจสอบเส้นทางในตารางเส้นทางเพื่อหาเส้นทางไปยังโหนดปลายทางนั้น โดยโหนดจะทำการสร้างหรือปรับปรุงข้อมูลของเลขลำดับปลายทางในตารางเส้นทางตามค่าที่ได้รับจากแพ็คเกตควบคุม และในการปรับปรุงของข้อมูลในตารางเส้นทางนั้นต้องเป็นไปตามเงื่อนไขใด ๆ ดังต่อไปนี้

- ค่าเลขลำดับปลายทางในแพ็คเกตควบคุมมีค่ามากกว่าในตารางเส้นทาง
- ค่าเลขลำดับปลายทางในแพ็คเกตควบคุมและในตารางมีค่าเท่ากันแต่จำนวนฮอปในแพ็คเกตควบคุมถ้านำมาบวกด้วยหนึ่งแล้วมีค่าน้อยกว่าจำนวนฮอปในตารางเส้นทาง
- ไม่รู้หรือไม่มีค่าเลขลำดับปลายทาง

2.3.2.3 การส่งแพ็คเก็ตร้องขอเส้นทาง

เมื่อโหนดต้นทางต้องการหาเส้นทางไปยังโหนดปลายทาง โหนดต้นทางก็จะกระจายแพ็คเก็ต RREQ ออกไปในเครือข่าย ซึ่งแพ็คเก็ต RREQ จะบรรจุข้อมูลตามฟิลด์ต่าง ๆ ตามรูปที่ 2.5 ซึ่งจะมีรายละเอียดดังนี้

- เลขลำดับปลายทาง (Destination Sequence Number) จะปรับปรุงโดยบันทึกเลขลำดับปลายทางล่าสุดของโหนดปลายทางที่จะทำการสื่อสารนี้ลงในแพ็คเก็ต ซึ่งจะคัดลอกมาจากเลขลำดับปลายทางในตารางเส้นทาง
- เลขลำดับโหนดต้นทาง (Originator Sequence Number) จะเป็นเลขลำดับของโหนดต้นทางเอง ซึ่งจะต้องเพิ่มค่าขึ้นก่อนที่จะกระจายแพ็คเก็ตออกไป
- เลขหมายประจำตัวแพ็คเก็ต RREQ (RREQ ID) เป็นเลขหมายของแพ็คเก็ต RREQ จะบวกเพิ่มขึ้นอีกหนึ่งค่า
- จำนวนฮอป (Hop Count) จะถูกกำหนดให้มีค่าเป็นศูนย์

ก่อนที่จะกระจายแพ็คเก็ต RREQ ออกไปโหนดต้นทางจะเก็บ RREQ ID และ หมายเลขไอพีของโหนดต้นทางไว้ก่อน (เป็นหมายเลขไอพีของตัวเอง) เพื่อให้เวลาในการหาเส้นทาง ซึ่งเป็นค่า PATH_DISCOVERY_TIME เมื่อโหนดต้นทางได้รับ RREQ จากโหนดข้างเคียงก็จะไม่ทำการใด ๆ และไม่ส่งแพ็คเก็ต RREQ ออกไปซ้ำอีก ในโปรโตคอล AODV จะกำหนดอัตราการกระจายของแพ็คเก็ต RREQ โดยใช้ค่า RREQ_RATELIMIT

หลังจากที่กระจายแพ็คเก็ต RREQ ออกไปโหนดต้นทางจะรอแพ็คเก็ตตอบกลับ RREP (หรือแพ็คเก็ตควบคุมเส้นทางอื่น ๆ) ถ้าโหนดต้นทางไม่ได้รับแพ็คเก็ตใด ๆ เลยในเวลาที่กำหนด ซึ่งจะกำหนดโดยค่า NET_TRAVERSAL_TIME ซึ่งมีหน่วยเป็นมิลลิวินาที โหนดต้นทางจะกระจายแพ็คเก็ต RREQ ออกไปใหม่อีกครั้ง ซึ่งจำนวนครั้งสูงสุดของการกระจายแพ็คเก็ต RREQ ออกไปใหม่นี้จะกำหนดโดยค่า RREQ_RETIRE ซึ่งการกระจายแพ็คเก็ตใหม่ในแต่ละครั้งนี้จะต้องเพิ่มค่า RREQ ID ด้วยทุกครั้งเสมอ

แพ็คเก็ตข้อมูลจะถูกเก็บไว้ในบัฟเฟอร์ในระหว่างที่รอแพ็คเก็ต RREP การเก็บในบัฟเฟอร์จะเก็บแบบเข้าก่อนออกก่อน (First-In, First-Out: FIFO) ถ้าการหาเส้นทางมีการกระจายแพ็คเก็ตออกไปเป็นจำนวนตามค่า RREQ_RETIRE จนมีค่ามากกว่าค่า MAX_TTL (Maximum TTL) โดยปราศจากการได้รับแพ็คเก็ตใด ๆ ตอบกลับมา แพ็คเก็ตข้อมูลที่อยู่ในบัฟเฟอร์ก็จะถูกทิ้ง

เพื่อป้องกันความคับคั่งในเครือข่ายการส่งแพ็คเก็ต RREQ ซ้ำอีกครั้งจะนำหลักการของเอกซ์โปเนนเชียลแบ็คออฟ (Exponential Back off) มาใช้ เมื่อการส่ง RREQ ออกไปครั้งแรก โหนดต้นทางจะรอแพ็คเก็ตตอบกลับเป็นเวลาตาม NET_TRAVERSAL_TIME เมื่อยังไม่ได้รับแพ็คเก็ตตอบกลับมาก็จะเพิ่มเวลารอแพ็คเก็ตตอบกลับเป็นสองเท่า นั่นคือจะใช้เวลารอทั้งหมด

เอกสารนี้เป็น 2x NET_TRAVERSAL_TIME งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อป้องกันไม่ให้แพ็คเกต RREQ กระจายออกไปในเครือข่ายโดยไม่จำเป็น โหนดต้นทางจะต้องใช้เทคนิคการกระจายแพ็คเกตที่เรียกว่า เทคนิคการขยายวงค้นหา (Expanding Ring Search Technique) โดยโหนดต้นทางจะกำหนดค่าเริ่มต้นให้กับค่าอายุขัยของแพ็คเกต (Time to Live : TTL) ให้เท่ากับค่า TTL_START และกำหนดเวลารอรับแพ็คเกตตอบกลับตามค่า RING_TRAVERSTIME หากภายในเวลาที่กำหนดโหนดต้นทางยังไม่ได้รับแพ็คเกตตอบกลับมา โหนดต้นทางก็จะส่งแพ็คเกต RREQ ออกไปใหม่โดยจะเพิ่มค่า TTL ตามค่า TTL_INCREMENT หากยังไม่ได้รับแพ็คเกตตอบกลับมาก็จะทำการเพิ่มค่า TTL ไปเรื่อย ๆ แต่หากเพิ่มไปจนระดับที่กำหนดตาม TTL_TRESHOLD โหนดต้นทางก็จะกำหนดค่า TTL ให้เท่ากับค่า NET_DIAMETER ในการส่งครั้งต่อไป

2.3.2.4 การรับแพ็คเกตร้องขอเส้นทาง

เมื่อโหนดในเครือข่ายได้รับแพ็คเกต RREQ ขึ้นแรกโหนดจะสร้างหรือปรับปรุงเส้นทางไปยังโหนดฮอปที่แล้ว หลังจากนั้นจะตรวจสอบว่าเป็นแพ็คเกต RREQ ที่ส่งมาจากโหนดต้นทางที่ซ้ำกันหรือเปล่า ถ้าแพ็คเกต RREQ นั้นเคยได้รับมาแล้ว โหนดก็จะทิ้งแพ็คเกตนั้นไป

ถ้าได้รับแพ็คเกต RREQ ในครั้งแรก โหนดระหว่างทางจะหาเส้นทางกลับไปยังโหนดต้นทางเพื่อใช้สำหรับส่งแพ็คเกต RREP กลับไปยังโหนดต้นทาง ถ้าไม่มีเส้นทางขากลับก็จะสร้างตารางเส้นทางขากลับ หรือถ้ามีอยู่แล้วก็ต้องค้นพบตารางเส้นทางขากลับดังนี้

- จะนำเลขลำดับโหนดต้นทางมาเปรียบเทียบกับเลขลำดับโหนดปลายทางที่มีอยู่ในตารางเส้นทางแล้วจะบันทึกค่าที่มากที่สุดแทนที่ลงไป ในเลขลำดับปลายทางของตารางเส้นทาง
- กำหนดให้ค่าเลขลำดับให้ใช้งานได้ (Valid Sequence Number)
- ฮอปถัดไปในตารางเส้นทางขากลับจะเป็นโหนดที่ส่ง RREQ มา
- จำนวนฮอปจะบันทึกตามจำนวนฮอปที่ติดมาในแพ็คเกต RREQ

เมื่อเส้นทางขากลับถูกสร้างเสร็จแล้ว โหนดจะตรวจสอบว่ามีเส้นทางไปยังโหนดปลายทางหรือไม่ ถ้าไม่มีก็จะปรับปรุงแพ็คเกต RREQ และส่งแพ็คเกต RREQ ออกไปโดยจะลด TTL ลงหนึ่งและเพิ่มจำนวนฮอปขึ้นอีกหนึ่ง ในทางกลับกันถ้าโหนดมีเส้นทางไปยังโหนดปลายทางก็ทิ้งแพ็คเกต RREQ ไปและสร้างแพ็คเกต RREP เพื่อส่งกลับไปยังโหนดต้นทาง

2.3.2.5 การส่งแพ็คเกตตอบกลับการขอเส้นทาง

โหนดระหว่างทางจะสร้าง RREP ถ้า

- ตัวของมันเองเป็นโหนดปลายทาง
- มีเส้นทางไปยังโหนดปลายทาง และเลขลำดับปลายทางที่อยู่ในตารางเส้นทางมีค่ามากกว่าหรือเท่ากับเลขลำดับปลายทางที่อยู่ในแพ็คเกต RREQ

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อโหนดสร้างแพ็คเก็ต RREP จะบันทึกหมายเลขไอพีของโหนดปลายทางและเลขลำดับของโหนดต้นทางจากแพ็คเก็ต RREQ ในกรณีที่ตัวโหนดเองเป็นโหนดปลายทางมันจะต้องเพิ่มค่าเลขลำดับของมันขึ้นอีกหนึ่ง ถ้าเลขลำดับปลายทางในแพ็คเก็ต RREQ มีค่าเท่ากับเลขลำดับของมัน แต่ถ้าเลขลำดับของโหนดปลายทางมีค่ามากกว่าอยู่แล้ว โหนดปลายทางก็จะไม่เปลี่ยนค่าเลขลำดับของมัน หลังจากนั้นโหนดปลายทางจะใช้ค่าเลขลำดับนั้นเป็นเลขลำดับปลายทาง (Destination Sequence Number) สำหรับแพ็คเก็ต RREP จากนั้นกำหนดให้จำนวนฮอปเป็นศูนย์ และโหนดปลายทางจะแทนฟิลด์อายุขัย (Lifetime) ด้วยค่า MY_ROUTE_TIMEOUT

ในกรณีที่โหนดระหว่างทางมีเส้นทางไปยังโหนดปลายทาง โหนดระหว่างทางจะบันทึกเลขลำดับปลายทางที่มีอยู่ในตารางเส้นทางลงไปในเลขลำดับปลายทางในแพ็คเก็ต RREP ถ้าเลขลำดับปลายทางที่มีอยู่ในตารางเส้นทางมีค่ามากกว่า และแทนค่าจำนวนฮอปด้วยค่าจำนวนฮอปในตารางเส้นทาง แต่ในกรณีที่เลขลำดับปลายทางน้อยกว่าค่าเลขลำดับปลายทางในตารางโหนดระหว่างทางก็จะไม่ส่ง RREP กลับแต่จะปรับปรุงเลขลำดับปลายทางในตารางแล้วส่งต่อ RREQ ต่อไป

2.3.2.6 การรับแพ็คเก็ตตอบกลับการขอเส้นทาง

เมื่อโหนดระหว่างทางได้รับแพ็คเก็ต RREP โหนดจะหาเส้นทางกลับไปยังโหนดก่อนหน้าหลังจากนั้นจะเพิ่มจำนวนฮอป (Hop Count) ขึ้นอีกหนึ่ง แล้วจึงส่งต่อไปจนถึงโหนดต้นทาง

ในกรณีที่เคยมีเส้นทางไปยังโหนดปลายทางแล้ว เมื่อโหนดต้นทางได้รับแพ็คเก็ต RREP โหนดต้นทางจะเทียบค่าเลขลำดับปลายทาง (Destination Sequence Number) ที่อยู่ในแพ็คเก็ต RREP กับเลขลำดับปลายทางที่มีอยู่ในตารางเส้นทาง หลังจากเปรียบเทียบกันแล้วโหนดจะปรับปรุงตารางเส้นทางก็ต่อเมื่อ เงื่อนไขต่อไปนี้เป็นอย่างน้อยที่สุดหนึ่งข้อ

- เลขลำดับที่อยู่ในตารางเส้นทางถูกกำหนดให้มีสถานะใช้งาน ไม่ได้ (Invalid Route)
- เลขลำดับปลายทางในแพ็คเก็ต RREP มีค่ามากกว่าเลขลำดับปลายทางในตารางเส้นทาง
- เลขลำดับปลายทางเท่ากัน แต่เส้นทางถูกกำหนดให้มีสถานะไม่ตอบสนอง (Inactive Route)
- เลขลำดับปลายทางเท่ากัน แต่จำนวนฮอปน้อยกว่าในตารางเส้นทาง

การปรับปรุงตารางเส้นทางมีขั้นตอนต่อไปนี้

- เส้นทางจะถูกเปลี่ยนให้เป็นสถานะที่ใช้งานได้ (Active Route)
- สถานะเลขลำดับปลายทางถูกเปลี่ยนให้เป็นใช้งานได้ (Valid Sequence Number)
- โหนดถัดไปในตารางเส้นทางจะเป็นโหนดที่ส่ง RREP กลับมา
- จำนวนฮอปจะถูกเปลี่ยนให้เป็นจำนวนฮอปที่ได้รับจากแพ็คเก็ต RREP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลขลำดับปลายทางจะถูกเปลี่ยนให้เป็นเลขลำดับปลายทางที่อยู่ในแพ็คเก็ต RREP หลังจากนั้นโหนดต้นทางก็จะทำการส่งข้อมูลไปยังโหนดปลายทาง

2.3.2.7 แพ็คเก็ตทักทาย (Hello Packet)

โหนดต่าง ๆ จะส่งแพ็คเก็ตทักทายออกมาทุกช่วงเวลา HELLO_INTERVAL โดยแพ็คเก็ตนี้ จะใช้ในการตรวจสอบสถานะของเส้นทางที่มีอยู่ว่ายังสามารถใช้งานได้หรือไม่ โดยโหนดระหว่างทางจะรอรับแพ็คเก็ตทักทายจากโหนดข้างเคียง และถ้าไม่ได้รับแพ็คเก็ตจากโหนดข้างเคียงภายในระยะเวลาที่กำหนดคือ $ALLOWED_HELLO_LOSS * HELLO_INTERVAL$ โหนดจะถือว่าเส้นทางนั้นใช้งานไม่ได้ แล้วโหนดระหว่างทางจะทำการซ่อมแซมเส้นทางต่อไป

2.3.2.8 การส่งแพ็คเก็ตควบคุมความผิดพลาด

โหนดระหว่างทางจะสร้างแพ็คเก็ต RERR ก็ต่อเมื่อเกิดเหตุการณ์ใดเหตุการณ์หนึ่งดังต่อไปนี้

- ถ้าโหนดระหว่างทางตรวจพบการเชื่อมต่อถูกตัดขาดในฮอปถัดไปของตัวเองในระหว่างการส่งข้อมูล
- ถ้าโหนดระหว่างทางได้รับแพ็คเก็ตข้อมูลที่ถูกระบุมาที่โหนดเองแต่มันไม่มีเส้นทางให้ใช้ส่งข้อมูล
- เมื่อได้รับแพ็คเก็ต RERR จากโหนดข้างเคียง

2.3.2.9 โดคอรี่แพร์

เมื่อการเชื่อมต่อถูกตัดขาดโหนดระหว่างทางที่เป็นโหนดก่อนหน้าที่เส้นทางเชื่อมต่อถูกตัดขาด จะตรวจดูว่าจำนวนฮอปที่ไปถึงโหนดปลายทางมีค่าน้อยกว่าจำนวนฮอปใน MAX_REPAIR_TTL หรือไม่ ถ้ามีน้อยกว่าแสดงว่าโหนดระหว่างทางนั้นอยู่ใกล้โหนดปลายทางแล้ว ดังนั้นโหนดระหว่างทางจะซ่อมแซมเส้นทางโดยเพิ่มค่าเลขลำดับปลายทาง (Destination Sequence Number) แล้วกระจายแพ็ค RREQ ออกไปเพื่อหาเส้นทางไปยังโหนดปลายทาง โดยให้ค่า TTL เริ่มต้นเป็นไปตามสูตรคำนวณนี้

$$\max(\text{MIN_REPAIR_TTL}, 0.5 * \#hops) + \text{LOCAL_ADD_TTL}$$

ซึ่งค่า #hops จะเป็นจำนวนฮอปจากต้นทางไปยังโหนดระหว่างทางที่ไม่สามารถส่งแพ็คเก็ตต่อไปได้ หลังจากนั้นโหนดระหว่างทางจะรอแพ็คเก็ต RREP ภายในช่วงเวลาที่กำหนด ในระหว่างนั้นแพ็คเก็ตข้อมูลจะถูกเก็บลงไปในบัฟเฟอร์ ถ้าเลยเวลาที่กำหนดแล้วโหนดระหว่าง

ทางยังไม่ได้รับแพ็คเกต RREP ตอบกลับมา โหนดระหว่างทางก็จะทิ้งแพ็คเกตไปและส่งแพ็คเกต RERR กลับไปยังโหนดต้นทางเพื่อให้โหนดต้นทางทำการเริ่มหาเส้นทางใหม่

แต่ในทางกลับกันถ้าโหนดระหว่างทางได้รับแพ็คเกต RREP ตอบกลับมา และเส้นทางใหม่นั้นใช้จำนวนฮอปมากกว่าเส้นทางเดิม โหนดระหว่างทางจะส่งแพ็คเกต RERR โดยกำหนดสถานะให้เป็น N (ซึ่งจะหมายถึงห้ามลบเส้นทางไปยังโหนดปลายทาง) จากนั้นโหนดระหว่างทางเมื่อได้รับ RREP ก็จะใช้เส้นทางที่ได้มาใหม่นั้นส่งแพ็คเกตข้อมูลไปยังโหนดปลายทางต่อไป

เมื่อโหนดระหว่างทางได้รับแพ็คเกต RERR ที่กำหนดสถานะเป็น N ก็จะไม่ลบเส้นทางที่ไปยังโหนดปลายทางทิ้งแล้วส่งกลับไปยังโหนดต้นทาง ถ้าโหนดต้นทางได้รับแพ็คเกต RERR ที่เซทสถานะเป็น N จากโหนดถัดไปที่อยู่บนเส้นทางของการส่งข้อมูล โหนดต้นทางจะหาเส้นทางใหม่ ถ้าการส่งข้อมูลไปยังโหนดปลายทางยังไม่เสร็จ แต่ถ้าการส่งข้อมูลไปยังโหนดปลายทางใกล้เสร็จหรือเสร็จแล้ว โหนดต้นทางก็ไม่จำเป็นต้องหาเส้นทางใหม่ก็ได้

2.4. โพรโตคอล AODV-2T

โพรโตคอล AODV เมื่อมีเส้นทางเชื่อมต่อที่ถูกตัดขาด AODV จะทำการซ่อมแซม โดยใช้วิธีการที่เรียกว่า โคลลลิแพร์ตามที่ได้กล่าวไปแล้ว ซึ่งวิธีการนี้จะทำให้การสื่อสารหยุดลงชั่วคราวเมื่อโหนดในเส้นทางหลักดำเนินการหาเส้นทางไปยังโหนดปลายทาง และถ้าการทำ โคลลลิแพร์ไม่สำเร็จก็ต้องเริ่มค้นหาเส้นทางใหม่ ทำให้ต้องเสียเวลาและเพิ่มแพ็คเกตควบคุมเข้าไปในระบบ จึงมีผู้วิจัยพัฒนากระบวนการสร้างเส้นทางสำรองจากแนวคิดดังกล่าวเพื่อเพิ่มความสามารถให้โพรโตคอลสามารถแจ้งเตือนก่อนที่เส้นทางจะถูกตัดขาด เพื่อให้มีการสร้างเส้นทางสำรองไว้ล่วงหน้า แล้วจึงเปลี่ยนไปใช้เส้นทางสำรองก่อนที่เส้นทางจะถูกตัดขาด ด้วยวิธีการเช่นนี้ก็จะสามารถเพิ่มประสิทธิภาพของการส่งข้อมูลและลดจำนวนแพ็คเกตที่ถูกทิ้งลงได้

โพรโตคอล AODV-2T (AODV Backup Route Discovery with 2 Levels of Threshold) เป็นโพรโตคอลที่มีหลักการให้โหนดในเส้นทางหลักสามารถสร้างเส้นทางสำรองไว้ได้ล่วงหน้า ก่อนที่เส้นทางจะเสียหายจนใช้งานไม่ได้ ดังนั้นโหนดต่าง ๆ จึงจำเป็นต้องตรวจสอบความสามารถในการเชื่อมต่อสัญญาณ ซึ่งขึ้นอยู่กับระดับสัญญาณสื่อสารและระดับพลังงาน

2.4.1 การตรวจวัดระดับพลังงานของโหนดและระดับสัญญาณที่ได้รับของโหนด

2.4.1.1 การตรวจวัดระดับพลังงาน

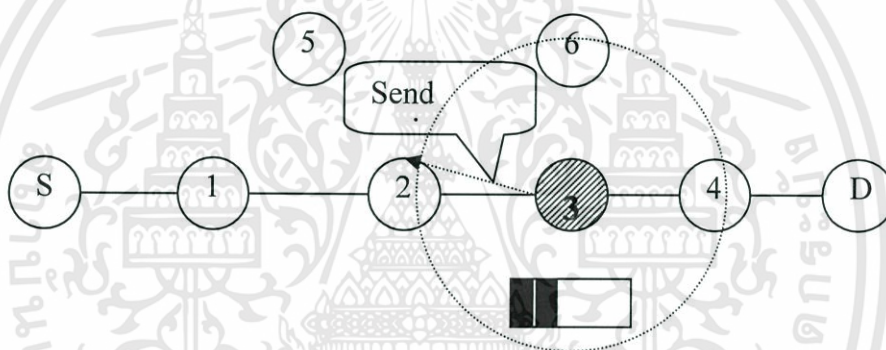
พลังงานของโหนดไร้สายใช้ จำแนกออกเป็น 2 ประเภท คือ

- พลังงานที่เกี่ยวข้องกับการสื่อสาร (Communication Related Energy)
- พลังงานที่ไม่เกี่ยวข้องกับการสื่อสาร (Non-Communication Related Energy)

ซึ่งพลังงานที่เกี่ยวข้องกับการสื่อสารยังสามารถแบ่งออกได้อีก 2 ประเภทคือ

- พลังงานที่ใช้รับ-ส่งข้อมูล (Transceiver Energy)
- พลังงานที่ใช้ดำเนินการ (Processing Energy)

ซึ่งจะนำค่าพลังงานของโหนดในขณะนั้นมาเปรียบเทียบกับเรซโซลต์ของระดับพลังงานที่กำหนด ซึ่งจะมียู่ 2 ระดับ โดยระดับที่ 1 เพื่อส่งแพ็คเกจแจ้งเตือนแก่โหนดที่อยู่ก่อนหน้า เพื่อให้โหนดที่อยู่ก่อนหน้าทำการหาเส้นทางสำรอง และในระดับที่ 2 ซึ่งหมายความว่าเส้นทางหลักที่ใช้เสี่ยงต่อการถูกตัดขาดการเชื่อมต่อ ดังนั้นโหนดที่อยู่ในเส้นทางหลักต้องเปลี่ยนไปใช้งานเส้นทางสำรองทันที โดยการตรวจวัดระดับพลังงานนี้จะกระทำทุกครั้งเมื่อโหนดได้รับแพ็คเกจข้อมูลเข้ามา



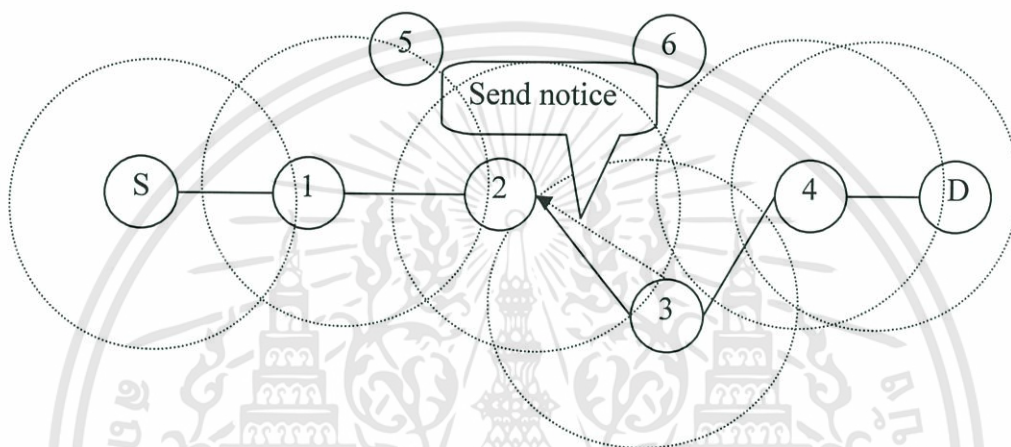
รูปที่ 2.10 แสดงการส่งแพ็คเกจแจ้งเตือนแก่โหนดในกรณีที่พลังงานเริ่มอ่อนลง

ในส่วนของอัลกอริทึมได้นำเอาการตรวจวัดระดับพลังงานไปเพิ่มเติมในส่วนของโปรแกรมรับ แพ็คเกจของ AODV ซึ่งเริ่มต้นด้วยการตรวจดูว่าแพ็คเกจที่ได้รับมานั้นเป็นแพ็คเกจควบคุมในการหาเส้นทางหรือไม่ เช่น แพ็คเกจ RREQ, RREP, RERR เป็นต้น ถ้าใช่ก็จะส่งแพ็คเกจเข้าไปในกระบวนการหาเส้นทางต่อไป ถ้าไม่ใช่ก็จะตรวจสอบว่าแพ็คเกจที่ได้รับมานั้นเป็นแพ็คเกจที่โหนดส่งมาให้ตัวเองหรือเปล่า เพื่อป้องกันการเกิดลูป แล้วจึงมาตรวจเช็คระดับพลังงานถ้าน้อยกว่าเรซโซลต์ระดับพลังงานที่ 1 โหนดที่ตรวจสอบก็จะส่งแพ็คเกจแจ้งเตือนออกไปยังโหนดก่อนหน้าที่อยู่ในเส้นทางสื่อสาร ถ้าระดับพลังงานน้อยลงไปอีกจนต่ำกว่าเรซโซลต์ระดับพลังงานที่ 2 โหนดก่อนหน้าที่ได้รับแพ็คเกจแจ้งเตือนก็จะเปลี่ยนไปใช้เส้นทางสำรอง

2.4.1.2 การตรวจวัดระดับสัญญาณ

เมื่อโหนดเคลื่อนที่ห่างออกไประดับสัญญาณที่ได้รับย่อมอ่อนลง เมื่อโหนดเคลื่อนที่ห่างออกไปจากรัศมีการสื่อสารก็จะทำให้การเชื่อมต่อถูกตัดขาด

ในส่วนของอัลกอริทึมจะเหมือนกันกับการตรวจวัดระดับพลังงานซึ่งได้กล่าวไว้แล้วในหัวข้อที่ได้กล่าวไปแล้ว จากรูปที่ 5 จะเห็นว่าโหนดสาม เริ่มเคลื่อนที่ห่างจากโหนดสอง ทำให้โหนดสาม ได้รับสัญญาณจากโหนดสองอ่อนลง แสดงว่าโหนดสามเคลื่อนที่ห่างออกไปแล้ว โหนดสามจึงแจ้งเตือนแก่โหนดสองว่าตัวมันเองเคลื่อนที่ห่างออกจากวงรัศมีการสื่อสารแล้ว



รูปที่ 2.11 แสดงการส่งแฟล็กเกตแจ้งเตือนในกรณีที่โหนดเคลื่อนที่ห่างออกไปจากเส้นทาง

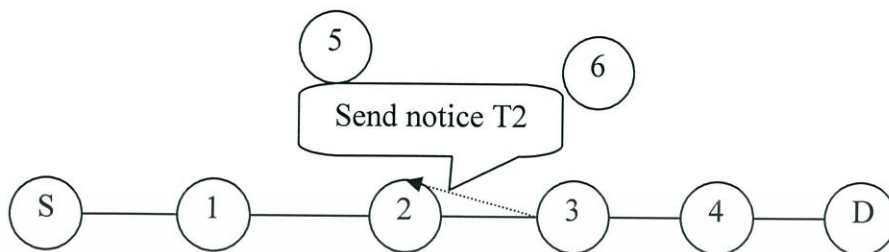
ซึ่งในแฟล็กเกตแจ้งเตือนจะประกอบไปด้วยที่อยู่ของโหนดที่แจ้งเตือน (Notify Source Address : NSA) ที่อยู่ของโหนดถัดไป (Next Hop Address : NHA) เพื่อจะใช้เป็นที่อยู่ของเป้าหมายของเส้นทางสำรอง (Backup Destination Address : BDA) และที่อยู่ของโหนดที่จะส่งแฟล็กเกตแจ้งเตือน (Notify Destination Address : NDA) ซึ่งเราจะส่งกลับไปยังโหนดก่อนหน้าแค่ 1 ฮอป จากรูปที่ 5 โหนดสามจะเป็น NSA โหนดที่สองจะเป็น NDA โหนดที่สี่จะเป็น NHA หรือ BNA

2.4.2 การค้นหาเส้นทางสำรอง

เมื่อโหนดในเส้นทางหลักได้รับการแจ้งเตือนจากโหนดถัดไปมันก็จะเตรียมหาเส้นทางสำรองไว้ในกรณีที่เส้นทางเชื่อมต่อถูกตัดขาด ซึ่งกระบวนการนี้ทำในขณะที่การสื่อสารยังคงดำเนินไปอย่างต่อเนื่อง หลังจากที่ได้รับการแจ้งเตือนจากโหนดข้างเคียงแล้ว โหนดที่ได้รับแฟล็กเกตแจ้งเตือนจะทำการส่งแฟล็กเกต ร้องขอเส้นทางสำรอง (Backup Route Request: B-RREQ) ไปยังโหนดข้างเคียง (Neighbor Node) เมื่อโหนดข้างเคียงได้รับแฟล็กเกตร้องขอเส้นทางสำรองแล้วจะทำการตรวจสอบตารางเส้นทางของตัวเองว่ามีโหนดข้างเคียงโหนดใดบ้างที่เป็น

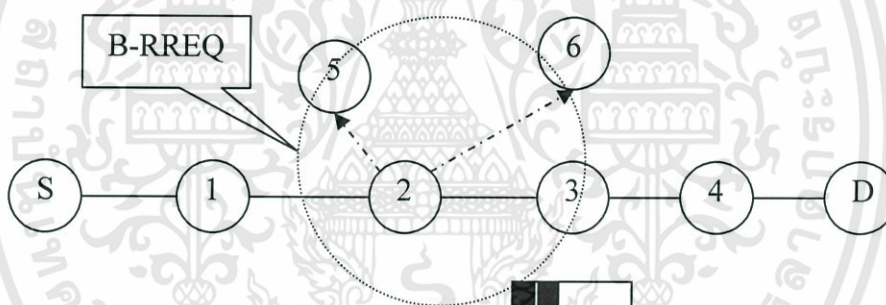
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนดปลายทางของเส้นทางสำรอง ถ้ามีก็สามารถกำหนดให้ตัวเองโหนดสำรอง แล้วส่งแพ็คเกจตอบกลับ (Backup Route Reply: B-RREP) ไปยังโหนดที่ร้องขอเส้นทางสำรอง



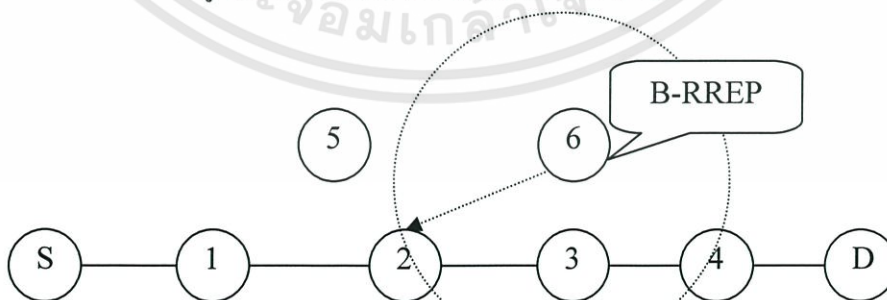
$E(u) < E(th1)$

รูปที่ 2.12 แสดงการส่งแพ็คเกจแจ้งเตือน



$E(u) < E(th1)$

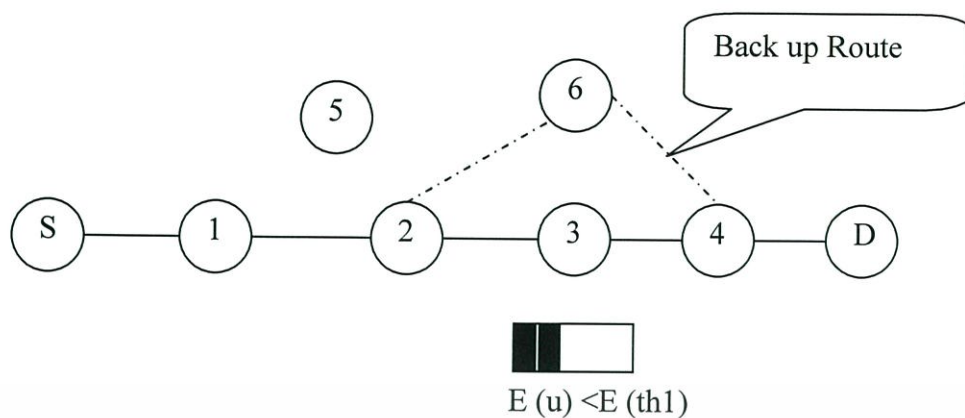
รูปที่ 2.13 แสดงการส่งแพ็คเกจ B-RREQ



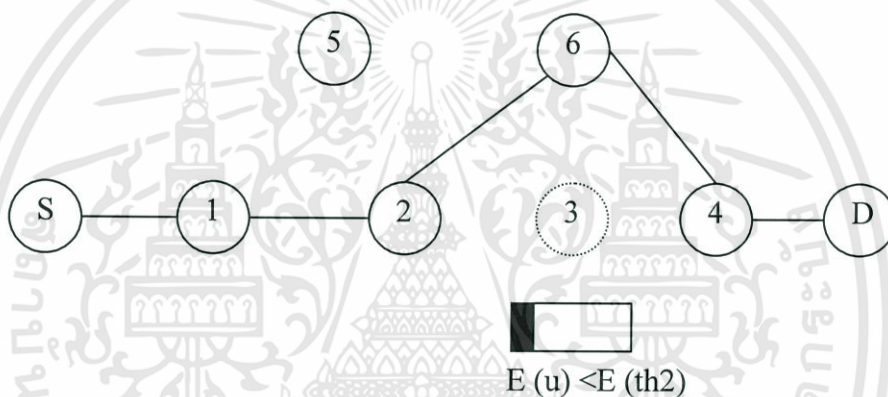
$E(u) < E(th1)$

รูปที่ 2.14 แสดงการส่งแพ็คเกจ B-RREP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงเส้นทางสำรองได้ถูกสร้างแล้ว



รูปที่ 2.16 แสดงการเปลี่ยนมาใช้เส้นทางสำรอง

ตัวอย่างจากรูปที่ 2.12 เมื่อโหนดสามมีพลังงานน้อยกว่าเซรชโหนดระดับแรก โหนดสามจะแจ้งเตือนไปยังโหนดสอง แล้วโหนดสองจะส่งแพ็คเกจ B-RREQ ออกไปเพื่อหาเส้นทางตามรูปที่ 2.13 โดยจะกำหนดโหนดปลายทางของเส้นทางสำรองเป็นโหนดสี่ เมื่อโหนดหกได้รับ B-RREQ แล้ว โหนดหกจะตรวจดูว่าตัวเองมีโหนดสี่เป็นโหนดข้างเคียงหรือไม่ ซึ่งในที่นี้ปรากฏว่ามี ดังนั้นโหนดหกจะกำหนดตัวเองให้เป็นโหนดสำรอง และจะบันทึกโหนดสี่ลงในตารางเส้นทางว่าเป็นโหนดปลายทางของเส้นทางสำรอง จากนั้นโหนดหกจะส่ง B-RREP กลับไปยังโหนดสอง ดังรูปที่ 2.14 โหนดสองก็จะบันทึกโหนดหกไว้ในตารางเส้นทางสำรอง ขั้นตอนนี้จะได้เส้นทางสำรองดังแสดงในรูปที่ 2.15 ต่อมาเมื่อพลังงานลดต่ำลงไปอีกจนต่ำกว่าระดับเซรชโหนดที่ 2 โหนดสามก็จะแจ้งเตือนมายังโหนดสองอีกครั้งเพื่อให้โหนดสองเปลี่ยนไปใช้เส้นทางสำรองดังในรูปที่ 2.16

บทที่ 3

การสร้างเส้นทางสำรองแบบหลายฮอปสำหรับระบบเครือข่าย

AODV-2T

โปรโตคอล AODV-2T (AODV Backup Route Discovery with 2 Levels of Threshold) มีหลักการให้โหนดในเส้นทางหลักสามารถสร้างเส้นทางสำรองไว้ได้ล่วงหน้าก่อนที่เส้นทางจะเสียหายจนใช้งานไม่ได้ โดยพัฒนามาจากโปรโตคอล AODV โดยการกำหนดค่าธรชโฮลด์ (Threshold) ของพลังงานแบตเตอรี่ และระดับสัญญาณที่ได้รับออกเป็นสองระดับ โดยระดับแรกกำหนดไว้เพื่อแจ้งเตือนแก่โหนดที่อยู่ในเส้นทางการรับ-ส่งข้อมูล ให้ทำการเตรียมหาเส้นทางสำรองไว้ ส่วนระดับที่สองไว้สำหรับเปลี่ยนมาใช้เส้นทางสำรองที่ได้เตรียมไว้ก่อนเส้นทางเดิมจะถูกตัดขาด ซึ่งในการหาเส้นทางสำรองนี้จะใช้แนวคิดในการใช้โหนดใกล้เคียง (Neighbor Node) มาเป็นโหนดสำรองถ้าโหนดนั้นสามารถส่งข้อมูลไปยังโหนดปลายทางได้ ซึ่งจะช่วยให้ไม่ต้องเริ่มหาเส้นทางในการรับ-ส่งข้อมูลใหม่ ดังนั้น โหนดต่าง ๆ จึงจำเป็นต้องตรวจสอบความสามารถในการเชื่อมต่อสัญญาณ ซึ่งขึ้นอยู่กับระดับสัญญาณสื่อสาร และระดับพลังงานของโหนด เนื่องจากใน MANET นั้นการสื่อสารของโหนดเคลื่อนที่มักจะใช้แบตเตอรี่มากกว่าระบบจ่ายไฟฟ้าโดยตรง ดังนั้นวิธีการแจ้งเตือนโอกาสที่เส้นทางจะเสียหายของโหนดที่อยู่ในเส้นทางหลักจึงต้องคำนึงถึงสองกรณีคือ กรณีพลังงานของโหนดลดลง และกรณีของระดับสัญญาณการสื่อสารลดลง

จากหลักการข้างต้นของโปรโตคอลดังกล่าวได้นำมาพัฒนาเป็น โปรโตคอล Multi-hop AODV-2T (Multi-hop AODV-2T) โดยเนื้อหาในบทนี้จะเป็นการนำเสนอการทำงานของโปรโตคอลที่มีหลักการให้โหนดในเส้นทางหลักสามารถสร้างเส้นทางย่อยสำรองไว้ได้ล่วงหน้าก่อนที่เส้นทางจะเสียหายจนใช้งานไม่ได้ และสามารถรองรับการสร้างเส้นทางย่อยสำรองแบบหลายฮอป ด้วยวิธีการเช่นนี้จะสามารถเพิ่มประสิทธิภาพของการส่งข้อมูลและลดจำนวนแพ็คเก็ตที่ถูกทิ้งลงได้

3.1 หลักการตรวจวัดระดับพลังงานของโหนดและระดับสัญญาณที่ได้รับของโหนด

ปัญหาหลักในการสื่อสารในระบบเครือข่ายไร้สายแบบแอ็ดฮอดก็คือการรักษาเสถียรภาพของการสื่อสารให้ผู้ใช้งานระบบเครือข่ายได้อย่างต่อเนื่อง ตลอดช่วงเวลาที่มีการสื่อสารอยู่ โดยช่องทางการสื่อสารไม่เสียหายไป เพราะโหนดสื่อสารต่างก็มีอิสระในการเคลื่อนที่และบางโหนดอาจจะเคลื่อนที่ตลอดเวลาด้วยความเร็วในการเคลื่อนที่ที่แตกต่างกันออกไป ทำให้เส้นทางสื่อสารมีโอกาสได้รับความเสียหายได้ตลอดเวลา การตรวจสอบระดับสัญญาณการสื่อสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เชื่อมต่อ และระดับพลังงานแบตเตอรี่ของโหนดสื่อสารในเส้นทาง เพื่อให้มีการสร้างเส้นทางสำรองและสลับไปใช้เส้นทางสำรองก่อนที่จะเกิดความเสียหายต่อเส้นทางหลักจนทำให้การสื่อสารของผู้ใช้หยุดชะงักลงได้ทันทั่วทั้งที่ จึงมีความจำเป็นอย่างยิ่ง

3.1.1 หลักการตรวจวัดระดับพลังงานของโหนด

ระดับพลังงานของโหนดไร้สาย สามารถจำแนกออกเป็น 2 ประเภท คือ

- พลังงานที่เกี่ยวข้องกับการสื่อสาร (Communication Related Energy)
- พลังงานที่ไม่เกี่ยวข้องกับการสื่อสาร (Non-Communication Related Energy)

ซึ่งพลังงานที่เกี่ยวข้องกับการสื่อสารยังสามารถแบ่งย่อยออกได้อีก 2 ประเภทคือ

- พลังงานที่ใช้รับ-ส่งข้อมูล (Transceiver Energy)
- พลังงานที่ใช้ดำเนินการ (Processing Energy)

ในการจำลองผลการสื่อสารในระบบเครือข่ายชนิดเอ็ดฮอคนั้น สมการที่ใช้คำนวณค่าพลังงานที่เหลืออยู่หลังจากที่โหนด u ทำการส่งข้อมูลออกไปได้ ดังสมการต่อไปนี้

$$E(u) = C(u) - (P_{Tx} * k * E_t) \quad (3.1)$$

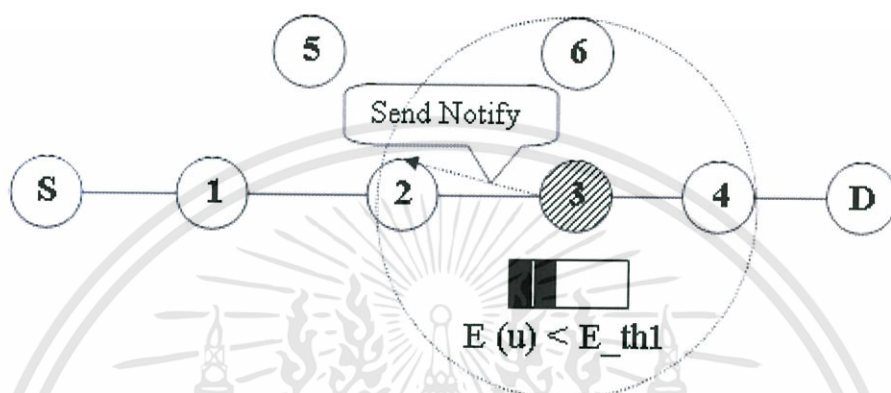
โดย $E(u)$ คือ ค่าพลังงานที่เหลือของโหนด u
 E_t คือ ค่าพลังงานที่ใช้ในการส่งแพ็คเกต
 k คือ ค่าสัมประสิทธิ์ของพลังงานในการส่งแพ็คเกต
 $C(u)$ คือ ค่าพลังงานเริ่มต้นของโหนด u
 P_{Tx} คือ ระยะเวลาที่ใช้ในการส่งแพ็คเกต

นอกจากนั้นยังสามารถหาค่าพลังงานที่เหลืออยู่ของโหนด u หลังจากที่ได้รับแพ็คเกตแล้ว โดยใช้สมการดังต่อไปนี้

$$E(u) = C(u) - (E_r * P_{Rx} + idle_time_energy) \quad (3.2)$$

โดย E_r คือ ค่าพลังงานที่ใช้ในการรับแพ็คเกต
 P_{Rx} คือ ช่วงระยะเวลาที่โหนดใช้ในการรับแพ็คเกต

โดยจะนำค่า $E(u)$ ซึ่งก็คือค่าพลังงานแบตเตอรี่ของโหนดในขณะนั้น มาเปรียบเทียบกับค่าเรซโซลต์ของระดับพลังงาน (E_{th}) ที่มีอยู่ 2 ระดับ โดยระดับที่ 1 ใช้เพื่อส่งแพ็คเก็ตแจ้งเตือนแก่โหนดที่อยู่ก่อนหน้า เพื่อให้โหนดที่อยู่ก่อนหน้าทำการค้นหาเส้นทางสำรอง และในระดับที่ 2 เพื่อให้เส้นทางหลักที่เสี่ยงต่อการถูกตัดขาดการเชื่อมต่อ เปลี่ยนไปใช้งานเส้นทางสำรองทันที โดยการตรวจวัดระดับพลังงานนี้จะกระทำทุกครั้งเมื่อโหนดได้รับแพ็คเก็ตข้อมูลเข้ามา



รูปที่ 3.1 แสดงการส่งแพ็คเก็ตแจ้งเตือนแก่โหนดก่อนหน้าในกรณีที่พลังงานแบตเตอรี่เริ่มอ่อนลง

ในส่วนของอัลกอริทึมของโปรโตคอล ได้นำเอาการตรวจวัดระดับพลังงานของแบตเตอรี่ไปเพิ่มเติมในส่วนของโปรแกรมรับแพ็คเก็ตของโปรโตคอล AODV ซึ่งเริ่มต้นด้วยการตรวจสอบว่าแพ็คเก็ตที่ได้รับมานั้นเป็นแพ็คเก็ตควบคุมในการหาเส้นทาง เช่น แพ็คเก็ต RREQ, RREP, RERR หรือไม่ ถ้าใช่ก็จะส่งแพ็คเก็ตเข้าไปในกระบวนการหาเส้นทางต่อไป แต่ถ้าไม่ใช่ก็จะตรวจสอบว่าแพ็คเก็ตที่ได้รับมานั้นเป็นแพ็คเก็ตที่โหนดส่งมาให้ตัวเองหรือเปล่า เพื่อป้องกันการเกิดลูป แล้วจึงตรวจเช็คระดับพลังงาน ถ้าน้อยกว่าเรซโซลต์ระดับพลังงานที่ 1 โหนดที่ตรวจสอบก็จะส่งแพ็คเก็ตแจ้งเตือนออกไปยังโหนดก่อนหน้าที่อยู่ในเส้นทางสื่อสาร โดยแพ็คเก็ตแจ้งเตือนที่ได้ส่งออกไปนั้นจะประกอบด้วยข้อมูลที่สำคัญที่จะใช้ในการสร้างเส้นทางสำรอง เช่น ข้อมูลที่อยู่ของโหนดต้นทาง และที่อยู่ของโหนดปลายทาง ให้โหนดก่อนหน้าสร้างเส้นทางย่อยสำรอง และถ้าระดับพลังงานน้อยลงไปอีกจนต่ำกว่าเรซโซลต์ระดับพลังงานที่ 2 โหนดก่อนหน้าที่ได้รับแพ็คเก็ตแจ้งเตือนก็จะเปลี่ยนไปใช้เส้นทางสำรองทันที

3.1.2 การตรวจวัดระดับสัญญาณ

เมื่อโหนดในเส้นทางหลักที่ใช้ในการรับ-ส่งข้อมูล เคลื่อนที่ห่างออกไป ระดับสัญญาณที่ได้รับย่อมอ่อนลงไปด้วย เมื่อโหนดเคลื่อนที่ห่างออกไปจากรศมีการสื่อสารก็จะทำให้การเชื่อมต่อถูกตัดขาด ซึ่งในงานวิจัยนี้เราได้ใช้แบบจำลองแบบ Two Ray Ground [21] ซึ่งจะมี

สมการในการคำนวณหาค่ากำลังส่วนรับ หรือ P_r เพื่อใช้ในการพิจารณาค่าของระดับสัญญาณที่ได้รับระหว่างโหนดข้างเคียงกัน

$$P_r = \frac{P_t \cdot G_t \cdot G_r \cdot h_t^2 \cdot h_r^2}{d^4 \cdot L} \quad (3.3)$$

โดย	P_r	คือ ค่ากำลังที่ได้รับ (w)
	P_t	คือ ค่ากำลังส่ง (w)
	G_t	คือ Antenna gain ฟังส่ง
	G_r	คือ Antenna gain ฟังรับ
	h_t	คือ ความสูงของเสาอากาศฟังส่ง (m)
	h_r	คือ ความสูงของเสาอากาศฟังรับ (m)
	d	คือ ระยะห่างระหว่างโหนด (m)
	L	คือ ค่า Loss Factor

ซึ่งในโปรแกรมจำลองการทำงานของระบบเครือข่าย NS-2 ได้กำหนดให้มีค่าเป็นดังนี้

$$P_t = 0.28183815 \text{ w}$$

$$L = 1$$

$$G_t, G_r = 1$$

$$h_t, h_r = 1.5 \text{ m}$$

3.1.3 การกำหนดระดับเรชโฮลด์

3.1.3.1 การกำหนดระดับเรชโฮลด์ของแบตเตอรี่

โดยจะกำหนดระดับเรชโฮลด์ที่สองของแบตเตอรี่ก่อน โดยอ้างอิงจากค่าพลังงานที่น้อยที่สุดที่โหนดไร้สายสามารถทำงานได้ ซึ่งในการจำลองเราจะใช้แบตเตอรี่ของโน้ตบุ๊กมาใช้ ส่วนการกำหนดกำหนดระดับเรชโฮลด์ที่หนึ่งของแบตเตอรี่นั้น จะนำค่าระดับเรชโฮลด์ที่สองของแบตเตอรี่มาบวกกับค่าพลังงานสูงสุดโดยประมาณที่ใช้ไปในการหาเส้นทางสำรอง ซึ่งจะเป็นไปตามสมการที่ (3.5)

$$B_{th1} = B_{th2} + Assum_rate \times \Delta d \quad (3.4)$$

โดย B_{th1} คือ ระดับเรชโฮลด์ที่หนึ่งของแบตเตอรี่ (Joules)

B_{th2} คือ ระดับเรชโฮลด์ที่สองของแบตเตอรี่ (Joules)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$Assum_rate$ คือ อัตราการใช้พลังงานในการส่งมีค่าเป็น 1 w (กำหนดจาก simulator)

Δd คือ ช่วงเวลาที่ใช้ในการหาเส้นทางสำรอง (Back Up Recovery Delay)

ซึ่งเวลาที่ใช้ในการค้นหาเส้นทางสำรองสามารถคำนวณได้จากสมการ ดังนี้

$$\Delta d = NODE_TRAVERSAL_TIME \times NET_DAIMETER \quad (3.5)$$

โดย $NODE_TRAVERSAL_TIME$ คือ เวลาที่มากที่สุดที่ใช้ในการส่งแพ็คเก็ตไป 1 ฮอป

$NET_DAIMETER$ คือ จำนวนฮอปที่มากที่สุดที่เป็นไปได้จากโหนดต้นทางไปยังโหนดปลายทาง

3.1.3.2 การกำหนดระดับเรชโฮลด์ของสัญญาณที่ได้รับ

โดยจะกำหนดระดับเรชโฮลด์ที่สองของสัญญาณที่ได้รับก่อน จากสมการที่ (3.4) จากนั้นนำค่าเรชโฮลด์ระดับที่หนึ่งของสัญญาณที่ได้รับที่ได้จากสมการมากำหนดค่าเรชโฮลด์ระดับที่สองของสัญญาณที่ได้รับจากสมการต่อไปนี้

$$p_a = \frac{\Delta Pr_{i+1,i}}{\Delta t_{i+1,i}} \quad (3.6)$$

โดย p_a คือ อัตราของสัญญาณที่ได้รับระหว่างแพ็คเก็ต i กับแพ็คเก็ต $i+1$ (w/s)

$\Delta Pr_{i+1,i}$ คือ ความต่างของสัญญาณที่ได้รับระหว่างแพ็คเก็ต i กับแพ็คเก็ต $i+1$ (w)

$\Delta t_{i+1,i}$ คือ ช่วงเวลาระหว่างแพ็คเก็ต i กับแพ็คเก็ต $i+1$ (s)

เมื่อได้ค่าของ p_a จากสมการที่ (3.7) แล้วจึงนำมาคำนวณหาค่าระดับเรชโฮลด์ที่หนึ่งของสัญญาณที่ได้รับตามสมการต่อไปนี้

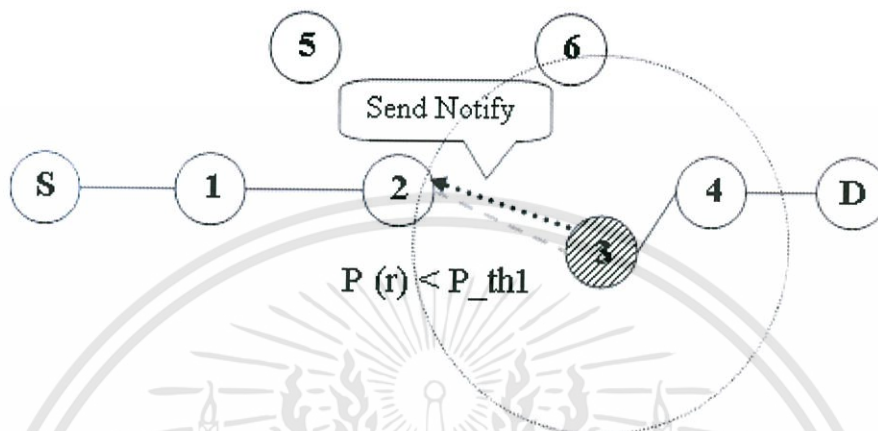
$$Pr_{ih1} = Pr_{ih2} + (Pa \times \Delta d) \quad (3.7)$$

โดย Pr_{ih1} คือ ระดับเรชโฮลด์ที่หนึ่งของสัญญาณที่ได้รับ

Pr_{ih2} คือ ระดับเรชโฮลด์ที่สองของสัญญาณที่ได้รับ

Δd คือ ช่วงเวลาที่ใช้ในการหาเส้นทางสำรองซึ่งได้จากสมการ (3.5)

ในส่วนของอัลกอริทึมจะเหมือนกันกับการตรวจวัดค่าระดับพลังงาน ซึ่งได้กล่าวไว้แล้วในหัวข้อที่ได้กล่าวไปแล้ว จากรูปที่ 3.2 จะเห็นว่าโหนดสาม เริ่มเคลื่อนที่ห่างจากโหนดสอง ทำให้โหนดสามได้รับสัญญาณจากโหนดสองอ่อนลง แสดงว่าโหนดสามเคลื่อนที่ห่างออกไปแล้ว โหนดสามจึงแจ้งเตือนแก่โหนดสองว่าตัวมันเองเคลื่อนที่ห่างออกจากวงรัศมีของการสื่อสารแล้ว



รูปที่ 3.2 แสดงการส่งแจ้งเตือนแจ้งเตือนในกรณีที่โหนดเคลื่อนที่ห่างออกไปจากเส้นทาง

ซึ่งในแพ็คเกจแจ้งเตือนจะประกอบไปด้วยรายละเอียดต่าง ๆ เช่น ที่อยู่ของโหนดที่แจ้งเตือน (Notify Source Address: NSA) ที่อยู่ของโหนดถัดไป (Next Hop Address : NHA) เพื่อจะใช้เป็นที่อยู่ของเป้าหมายของเส้นทางสำรอง (Backup Destination Address : BDA) และที่อยู่ของโหนดที่จะส่งแจ้งเตือน (Notify Destination Address : NDA) ซึ่งจะส่งกลับไปยังโหนดก่อนหน้าแค่ 1 ฮอป จากรูปที่ 3.2 โหนดสามจะเป็น NSA โหนดที่สองจะเป็น NDA โหนดที่สี่จะเป็น NHA หรือ BNA

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Type								T	T	Reserved												Hop Count									
								1	2																						
Notice Destination Address																															
Backup Destination address																															
Backup Destination Sequence Number																															
Source Notice Address																															

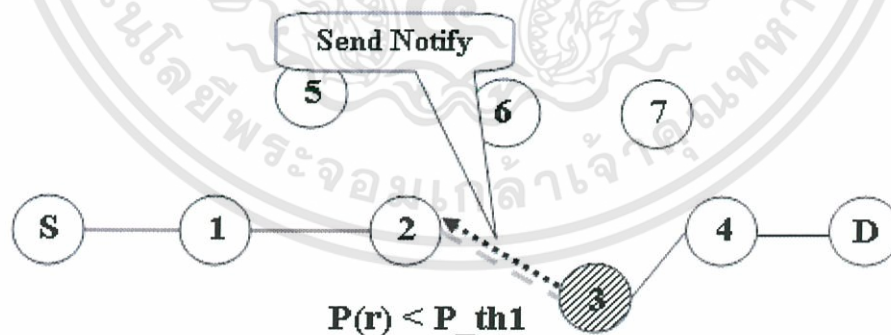
รูปที่ 3.3 แสดงโครงสร้างของแพ็คเกจแจ้งเตือน

ตารางที่ 3.1 คำอธิบายฟิลด์ต่างๆในแพ็กเก็ตแจ้งเตือน

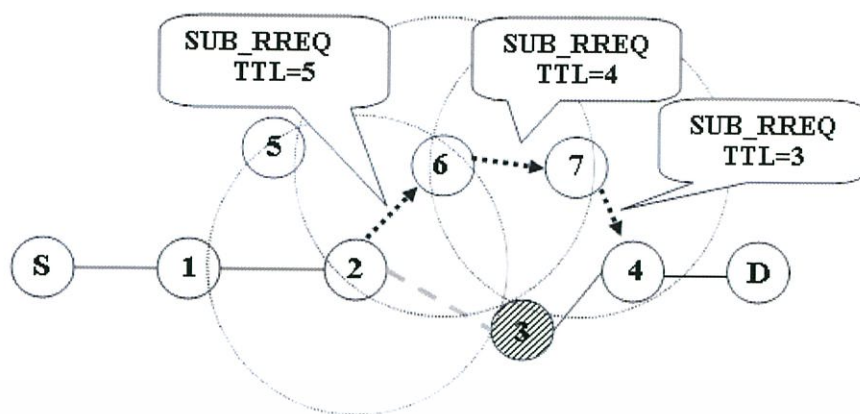
Type	5
T1	เซรชโฮลด์ต่ำกว่าระดับที่ 1
T2	เซรชโฮลด์ต่ำกว่าระดับที่ 2
Reserved	โหนดปลายทางจะไม่สนใจค่านี้เมื่อมีค่าเท่ากับ 0
Hop Count	จำนวนฮอป
Notify Destination Address	หมายเลขไอพีโหนดเป้าหมายของการแจ้งเตือน
Backup Destination address	หมายเลขไอพีของโหนดเป้าหมายที่ใช้เป็นเส้นทางสำรอง
Backup Destination Sequence Number	เลขลำดับของเส้นทางสำรอง
Source Notify Address	หมายเลขไอพีของโหนดที่ส่งแพ็กเก็ตแจ้ง

3.2 กระบวนการค้นหาเส้นทางสำรอง

เมื่อโหนดที่อยู่ในเส้นทางสื่อสารมีโอกาสที่จะถูกตัดขาด โหนดในเส้นทางหลักจะได้รับการแจ้งเตือนจากโหนดถัดไป เพื่อเตรียมหาเส้นทางสำรองไว้ในกรณีที่เส้นทางเชื่อมต่อถูกตัดขาด ซึ่งกระบวนการนี้จะกระทำในขณะที่การสื่อสารยังคงดำเนินไปอย่างต่อเนื่อง หลังจากที่ได้รับแจ้งเตือนจากโหนดถัดไปแล้ว โหนดที่ได้รับแพ็กเก็ตแจ้งเตือนจะทำการส่งแพ็กเก็ตร้องขอเส้นทางสำรอง (Sub Route Request: Sub_RREQ) ออกไปยังโหนดข้างเคียง (Neighbor Node)



รูปที่ 3.4 แสดงการส่งแพ็กเก็ตแจ้งเตือนแก่โหนดในกรณีที่สัญญาณเริ่มอ่อนลง



รูปที่ 3.5 แสดงการส่งแพ็คเกจ Sub_RREQ ไปยังโหนดข้างเคียง พร้อมทั้งลดจำนวน TTL ลงไปเรื่อย ๆ จนกว่าจะส่งถึงโหนดปลายทางที่ต้องการ

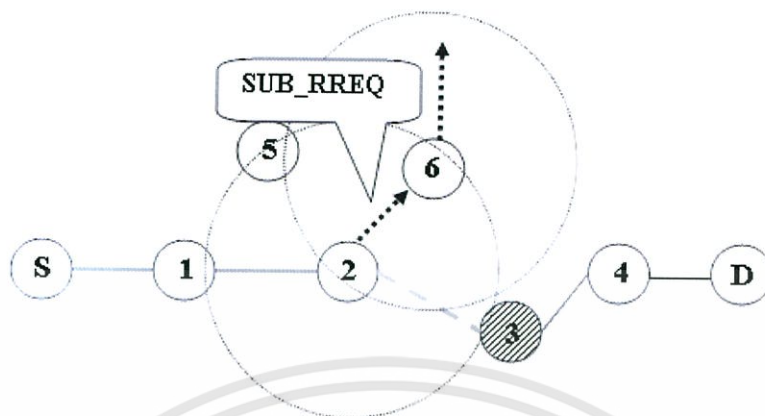
หลังจากที่โหนดสองได้รับการแพ็คเกจแจ้งเตือน โหนดสองจะทำการกระจายแพ็คเกจ Sub_RREQ (Subroute Request) ออกไปยังโหนดเพื่อนบ้าน และแพ็คเกจ Sub_RREQ จะประกอบไปด้วยที่อยู่ของโหนดที่แจ้งเตือน (Notify Source Address : NSA) ที่อยู่ของโหนดถัดไป (Next Hop Address : NHA) เพื่อจะใช้เป็นที่อยู่ของเป้าหมายของเส้นทางสำรอง (Backup Destination Address : BDA) และที่อยู่ของโหนดที่จะส่งแพ็คเกจแจ้งเตือน (Notify Destination Address : NDA) เลขลำดับแพ็คเกจ Sub_RREQ และค่า TTL (Time To Live) เมื่อโหนด 6 และ 7 ได้รับแพ็คเกจดังกล่าว จะตรวจสอบว่าตัวเองมีโหนดที่เป็นโหนดข้างเคียงหรือไม่ ถ้ายังไม่ใช้ก็จะส่งแพ็คเกจ Sub_RREQ ต่อไปเรื่อย ๆ พร้อมทั้งลดจำนวน TTL ลงครั้งละ 1 ไปจนกว่าจะถึงโหนดปลายทางที่ต้องการ



รูปที่ 3.6 แสดงการส่งแพ็คเกจ RREP กลับไปยังโหนดที่ได้รับการแจ้งเตือน เพื่อยืนยันเส้นทางสำรอง

ในทุก ๆ โหนดที่ได้รับแพ็คเกจ Sub_RREQ จะทำการตรวจสอบตารางเส้นทางของตัวเองว่ามีโหนดข้างเคียงโหนดใดบ้างที่เป็นโหนดปลายทางของเส้นทางสำรอง แล้วทำการบันทึกในตารางเส้นทางโดยกำหนดให้ตัวเองเป็นโหนดสำรอง แล้วส่งแพ็คเกจตอบกลับโดยส่ง

แพ็คเก็ต Sub_RREP (Subroute Reply) ไปยังโหนดปลายทางที่ร้องขอเส้นทางสำรองพร้อมทั้ง
 ระบุค่าของ TTL ควบคู่ไปด้วย



รูปที่ 3.7 แสดง โหนดที่ได้รับแพ็คเก็ต Sub_RREQ (6) ไม่อยู่ในระยะที่สามารถติดต่อกับโหนด
 ถัดไป (3) (4) หรือ (D) ได้



รูปที่ 3.8 แสดงการส่งแพ็คเก็ต RERR ไปยังโหนดต้นทาง

ในกรณีที่ไม่สามารถสร้างเส้นทางสำรองได้ อันอาจเกินจากโหนดที่ได้รับแพ็คเก็ต
 Sub_RREQ (6) ไม่อยู่ในระยะที่สามารถติดต่อกับโหนดถัดไป (3) (4) หรือ (D) ได้ หรือสร้าง
 เส้นทางสำรองไม่ทันเวลา (Time Out) โหนดที่ได้รับการแจ้งเตือน (2) จะทำการส่งแพ็คเก็ต
 RERR ไปยังโหนดต้นทาง เพื่อให้โหนดต้นทางทำการค้นหาเส้นทางใหม่

อัลกอริทึมในการค้นหาเส้นทางสำรอง สามารถแบ่งย่อยออกเป็น 5 ขั้นตอน ดังนี้

1. ขั้นตอนการส่งแพ็คเก็ตร้องขอเส้นทางสำรอง
2. ขั้นตอนการรับแพ็คเก็ตร้องขอเส้นทางสำรอง
3. ขั้นตอนการส่งแพ็คเก็ตตอบกลับการร้องขอเส้นทางสำรอง
4. ขั้นตอนการรับแพ็คเก็ตตอบกลับการร้องขอเส้นทางสำรอง
5. ขั้นตอนการรับแพ็คเก็ตแจ้งเตือน

เอกสารนี้ซึ่งจะกล่าวโดยละเอียดในหัวข้อถัดไปเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 การส่งแพ็กเก็ตร้องขอเส้นทางสำรอง

ในส่วนนี้จะเป็นการกำหนดค่าในแพ็กเก็ต Sub_RREQ เพื่อกระจายแพ็กเก็ตออกไปยัง โหนดข้างเคียง ในส่วนรายละเอียดของ Sub_RREQ มีรายละเอียดดังนี้

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Type						Reserved												Hop Count													
Back up Destination Address																															
Destination Address																															
Back up Destination Sequence Number																															
Originator Address																															

รูปที่ 3.9 แสดงโครงสร้างของแพ็กเก็ต Sub_RREQ

ตารางที่ 3.2 คำอธิบายฟิลด์ต่าง ๆ ในแพ็กเก็ตร้องขอเส้นทางสำรอง (Sub_RREQ)

Type	6
Reserve	โหนดปลายทางจะไม่สนใจค่านี้เมื่อมีค่าเท่ากับ 0
Hop Count	จำนวนฮอป
Back up Destination Address	หมายเลขไอพีของโหนดที่ใช้เป็นเส้นทางสำรอง
Destination Address	หมายเลขไอพีของโหนดปลายทางในเส้นทางหลัก
Back up Destination Sequence Number	เลขลำดับเส้นทางสำรอง
Originator Address	หมายเลขไอพีของโหนดต้นทางที่ส่ง Sub_RREQ

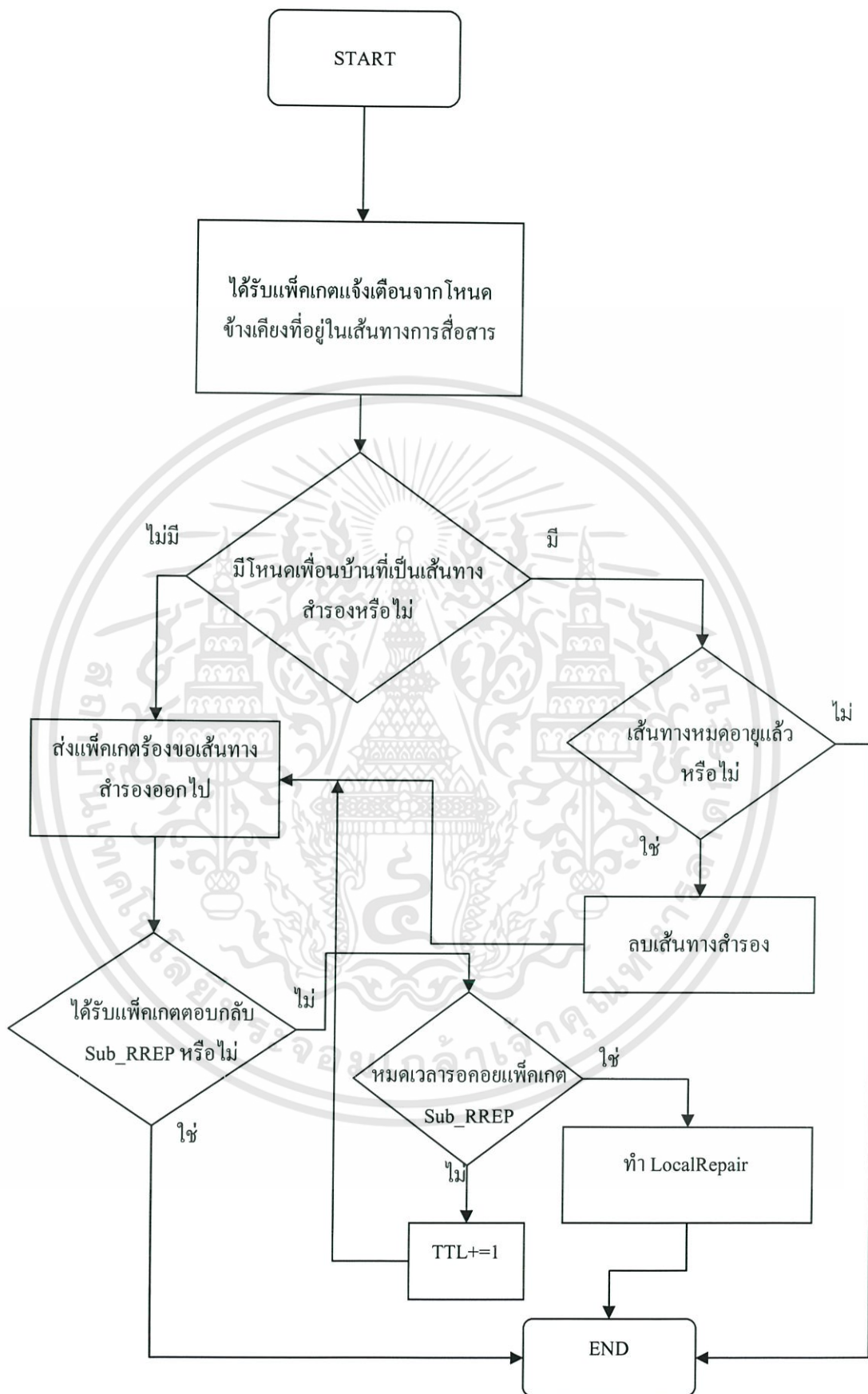
ซึ่ง Back up Destination Address คือที่อยู่ของโหนดในเส้นทางหลักที่จะใช้เป็นจุดหมายในการค้นหาเส้นทางสำรอง จากรูปที่ 3.5 ก็คือโหนดสี ส่วน Backup Destination Sequence Number คือเลขที่บอกลำดับของการร้องขอเส้นทางสำรอง ถ้าแพ็กเก็ตใดมีค่ามากกว่าแสดงว่าเป็นการขอเส้นทางที่ใหม่กว่า ซึ่งในอัลกอริทึมเราจะเลือกใช้ค่าลำดับที่มากกว่าเสมอ เพื่อป้องกันการเกิดการวนซ้ำ และ Originator Address คือที่อยู่ของโหนดที่ส่งแพ็กเก็ต Sub_RREQ ออกไป ค่านี้เอาไว้เพื่อส่งค่าแพ็กเก็ตตอบกลับ Sub_RREP ไปยังโหนดที่ทำการร้องขอเส้นทางสำรอง

เมื่อโหนดระหว่างทางได้รับแพ็กเก็ตแจ้งเตือนก็จะเริ่มหาเส้นทางสำรองดังอัลกอริทึมที่แสดงในรูปที่ 3.5 ขั้นแรกโหนดระหว่างทางจะตรวจสอบดูก่อนว่าตัวมันเองมีเส้นทางสำรองไปยังโหนดปลายทางที่ได้กำหนดไว้ในแพ็กเก็ตแจ้งเตือนหรือไม่ ถ้ามีจะทำการตรวจสอบว่าเส้นทางยังใช้อยู่หรือไม่ ถ้าใช้ได้ก็จะรอแพ็กเก็ตแจ้งเตือนให้เปลี่ยนไปใช้เส้นทางสำรองจนกว่าจะหมดอายุของเส้นทาง ถ้าหมดเวลาแล้วโหนดระหว่างทางจะลบเส้นทางนั้นทิ้งไป ถ้ายังไม่มีเส้นทางสำรอง

โหนดระหว่างทางจะส่งแพ็คเกจร้องขอเส้นทางสำรองออกไปยังโหนดข้างเคียง แล้วรอจนกว่าจะได้รับแพ็คเกจตอบกลับจากโหนดข้างเคียงภายในเวลาที่กำหนด ถ้ายังไม่ได้รับแพ็คเกจตอบกลับภายในเวลาที่กำหนด โหนดระหว่างทางจะเพิ่มค่า TTL ขึ้นอีกหนึ่ง แล้วส่งแพ็คเกจร้องขอเส้นทางสำรองออกไปใหม่ ถ้าโหนดระหว่างทางได้รับแพ็คเกจตอบกลับมากก็จะเข้าสู่ขั้นตอนการรับแพ็คเกจ Sub_RREP ต่อไป

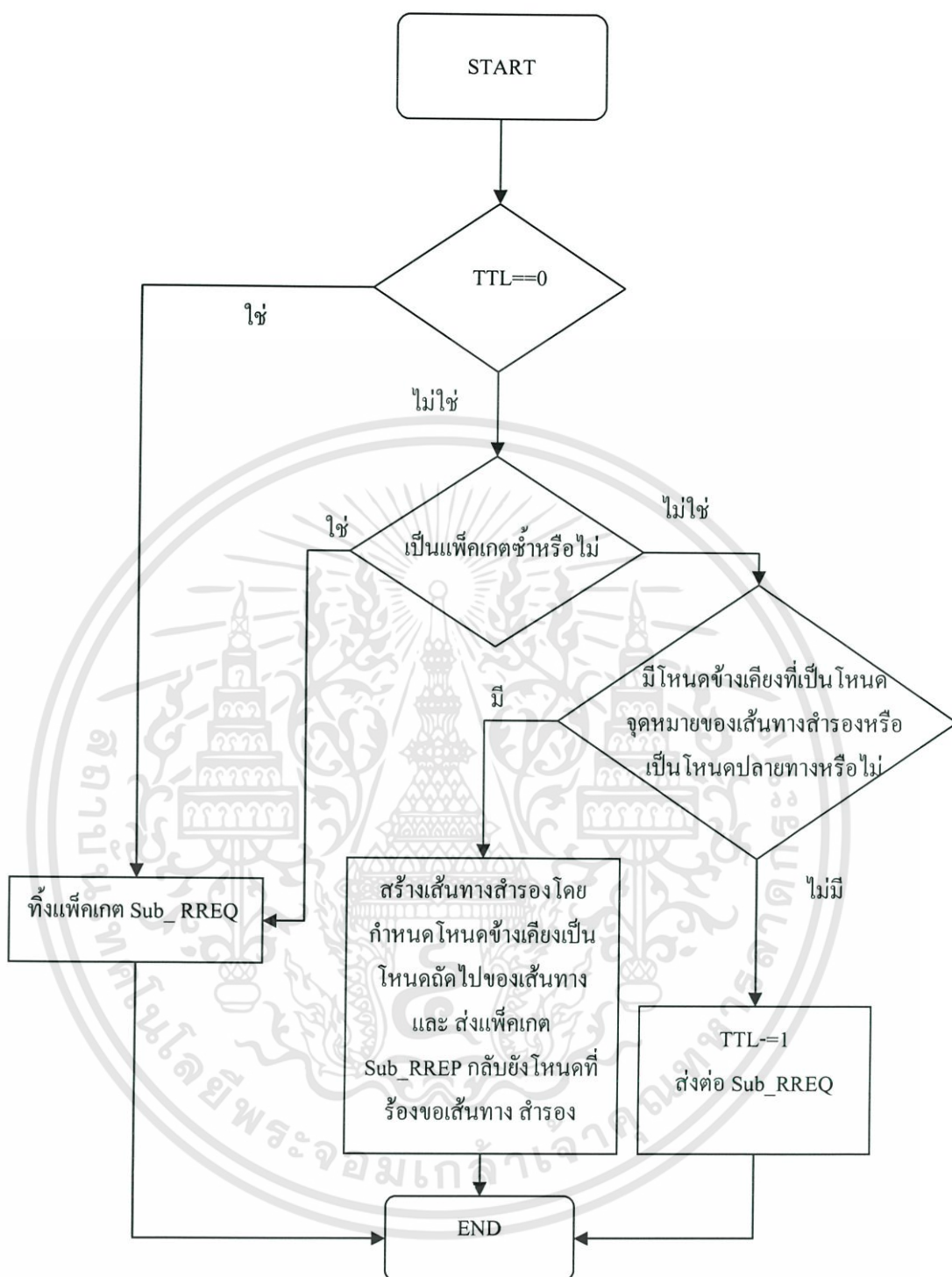
3.2.2 การรับแพ็คเกจร้องขอเส้นทางสำรอง

เมื่อโหนดที่อยู่ในเส้นทางหลักได้รับแพ็คเกจ Sub_RREQ โหนดนั้นก็จะจัดการกับ Sub_RREQ ดังอัลกอริทึมที่แสดงในรูปที่ 3.6 ซึ่งโหนดจะใช้ฟังก์ชันในการหาโหนดเพื่อนบ้านที่ตรงกับที่อยู่ของโหนดจุดหมายของเส้นทางสำรอง (ซึ่งก็คือโหนดที่อยู่ในเส้นทางหลักเดิม) ถ้ามีโหนดดังกล่าวก็จะส่งแพ็คเกจตอบกลับ Sub_RREP ไปยังโหนดที่ร้องขอเส้นทางสำรอง และตัวโหนดเองก็จะบันทึกในตารางเส้นทางสำรองของตัวเองว่าสอดคล้องไปคือโหนดข้างเคียง แต่ถ้าในกรณีที่ไม่มีโหนดเพื่อนบ้านเป็นโหนดในเส้นทางหลัก โหนดจะส่ง Sub_RREQ ต่อไปตามค่าของ TTL



รูปที่ 3.10 แสดงการส่งแพ็คเกจ Sub_RREQ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงการรับแพ็คเก็ต Sub_RREQ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การส่งแพ็คเกจตอบกลับการร้องขอเส้นทางสำรอง

เมื่อโหนดข้างเคียงซึ่งจากรูปที่ 3.5 ก็คือโหนดที่หกได้รับแพ็คเกจร้องขอเส้นทางสำรองแล้ว ถ้าโหนดนั้นสามารถสื่อสารกับโหนดจุดหมายของเส้นทางสำรองซึ่งก็คือโหนดสี่ หรือสามารถสื่อสารกับโหนดปลายทางของเส้นทางหลักได้ แสดงว่าโหนดข้างเคียงนั้นสามารถใช้เป็นเส้นทางสำรองได้ โหนดข้างเคียงก็จะส่งแพ็คเกจตอบกลับการร้องขอเส้นทางสำรองกลับไป ซึ่งจะมีโครงสร้างดังนี้

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Type							Reserve										Hop Count														
Back up Destination Address																															
Back up Destination Sequence Number																															
Originator Address																															

รูปที่ 3.12 แสดงโครงสร้างของแพ็คเกจ Sub_RREP

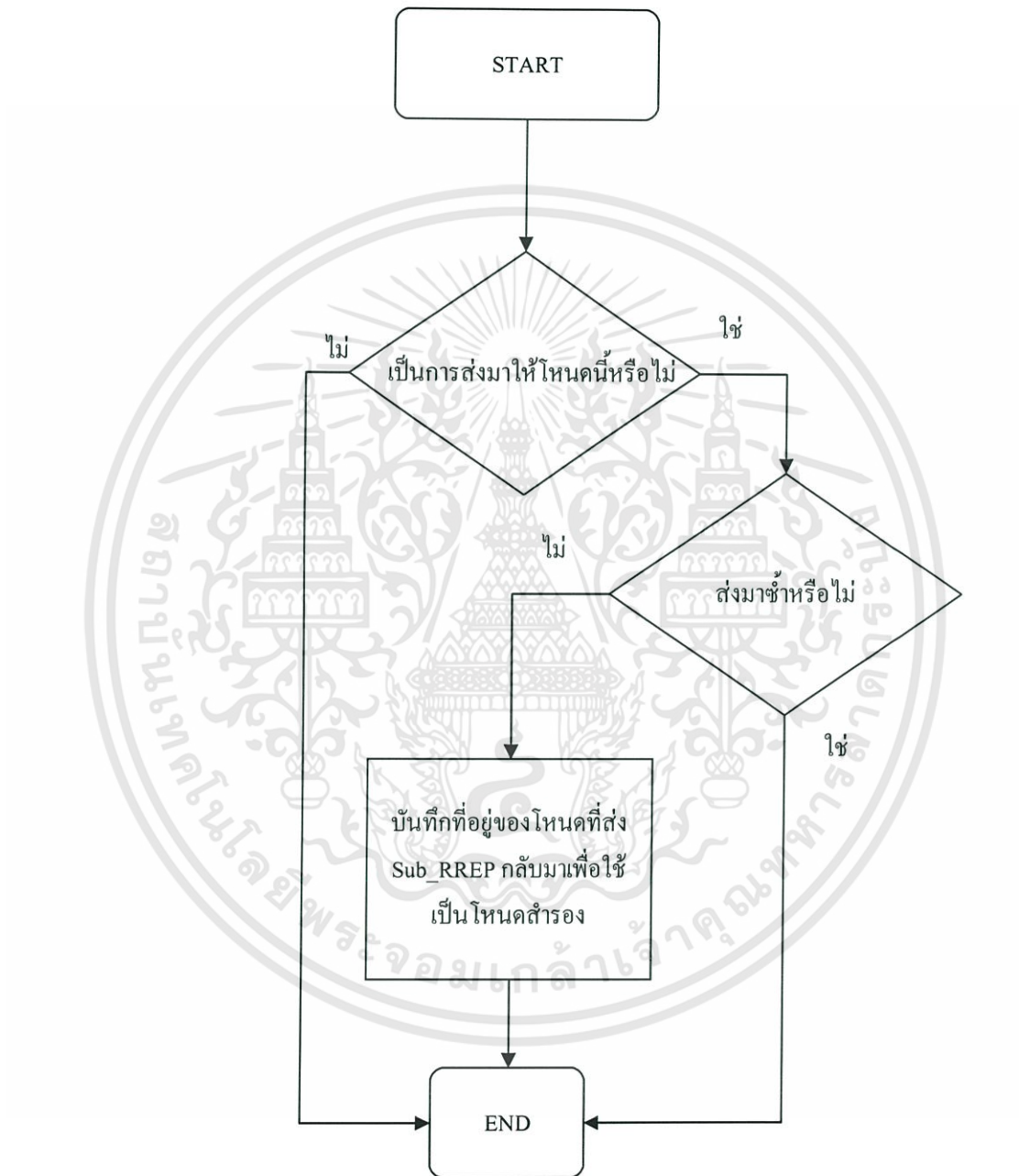
ตารางที่ 3.3 คำอธิบายฟิลด์ต่าง ๆ ในแพ็คเกจตอบกลับการร้องขอเส้นทางสำรอง (Sub_RREP)

Type	7
Reserve	โหนดปลายทางจะไม่สนใจค่านี้เมื่อมีค่าเท่ากับ 0
Hop Count	จำนวนฮอป
Back up Destination Address	หมายเลขไอพีของโหนดที่ใช้เป็นเส้นทางสำรอง
Back up Destination Sequence Number	เลขลำดับเส้นทางสำรอง
Originator Address	หมายเลขไอพีของโหนดที่ส่งแพ็คเกจร้องขอเส้นทางสำรอง

ซึ่ง Originator Address เป็นที่อยู่ของโหนดที่ร้องขอเส้นทางสำรอง ใช้เพื่อให้แพ็คเกจตอบกลับไปถึงโหนดที่ทำการร้องขอเส้นทาง ส่วนข้อมูล Back up Destination Address นั้นจะบอกให้โหนดที่ร้องขอเส้นทางสำรองทราบว่าได้รับการตอบกลับจากการหาเส้นทางสำรองที่ไปยังที่อยู่โหนดดังกล่าวนี้ และ Back up Destination Sequence Number บอกให้โหนดที่ร้องขอเส้นทางสำรอง ทราบถึงความล่าสุดของการร้องขอเส้นทางสำรองเพื่อป้องกันการหาเส้นทางสำรองซ้ำ

3.2.4 การรับแพ็คเกจตอบกลับการร้องขอเส้นทางสำรอง

เมื่อโหนดที่ร้องขอเส้นทางสำรองได้รับแพ็คเกจตอบกลับการร้องขอเส้นทางสำรอง หรือ Sub_RREP กลับมา โหนดที่ร้องขอเส้นทางสำรองจะทำการบันทึกที่อยู่ของโหนดที่ส่งแพ็คเกจ Sub_RREP กลับมาลงไปในตารางเส้นทางไว้ให้เป็นโหนดสำรอง

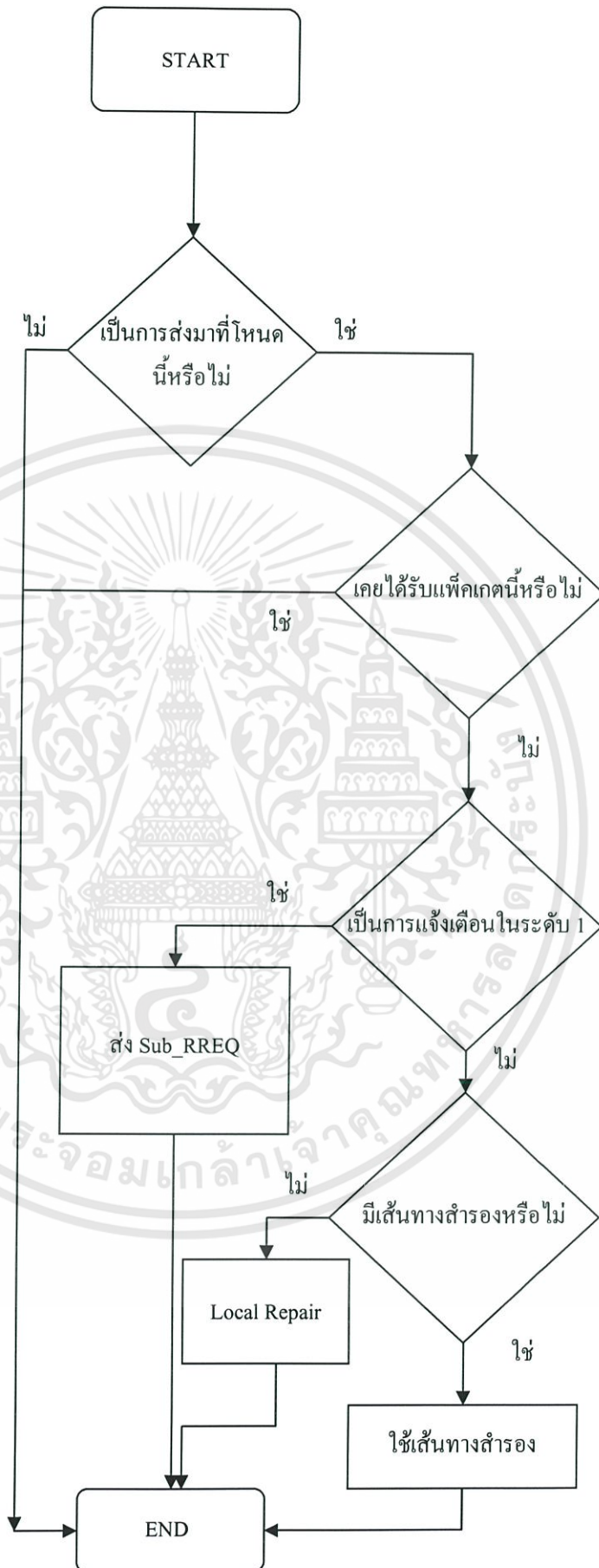


รูปที่ 3.13 แสดงการรับแพ็คเกจ Sub_RREP

3.2.5 การรับแพ็คเกจแจ้งเตือน

เมื่อโหนดที่อยู่ในเส้นทางหลักได้รับแพ็คเกจแจ้งเตือนตอนแรกจะตรวจดูก่อนว่าเคยได้รับแพ็คเกจแจ้งเตือนนี้หรือไม่ ถ้ายังไม่ได้รับก็จะตรวจดูว่าเป็นการแจ้งเตือนในระดับ 1 หรือระดับ 2 ถ้าเป็นการแจ้งเตือนในระดับ 1 โหนดก็จะส่ง Sub_RREQ ออกไปเพื่อหาเส้นทางสำรอง แต่ถ้าเป็นการแจ้งเตือนในระดับ 2 โหนดก็จะตรวจสอบก่อนว่ามีเส้นทางสำรองอยู่หรือไม่ ถ้ามีก็จะเปลี่ยนไปใช้เส้นทางสำรอง แต่ถ้าไม่มีโหนดดังกล่าวจะทำการซ่อมแซมเส้นทางโดยกลับไปทำตามขั้นตอนปกติของการทำโลคอลลีเพอร์





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.14 แสดงการรับแพ็คเกจแจ้งเตือน อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การจำลองการทำงานของระบบ

การจำลองการทำงานของโปรโตคอล Multi-hop AODV-2T (Multi-hop AODV Backup Route Discovery with 2 Levels of Threshold) โปรโตคอล AODV-2T (AODV Backup Route Discovery with 2 Levels of Threshold) และโปรโตคอล AODV แบบมาตรฐาน โดยใช้โปรแกรมจำลองเครือข่าย NS-2 ซึ่งเก็บผลการจำลองในรูปของเทรซไฟล์มาทำการรวบรวมและประมวลผลเพื่อหาผลเปรียบเทียบประสิทธิภาพของการทำงานของทั้งสามโปรโตคอล

4.1 การประมวลผลประสิทธิภาพของเครือข่ายจากการจำลองการทำงาน

ในการวัดประสิทธิภาพของการทำงานของโปรโตคอล Multi-hop AODV-2T (Multi-hop AODV Backup Route Discovery with 2 Levels of Threshold) เปรียบเทียบกับโปรโตคอล AODV-2T (AODV Backup Route Discovery with 2 Levels of Threshold) และโปรโตคอล AODV แบบมาตรฐาน มีพารามิเตอร์ที่ใช้ในการวัดผลดังต่อไปนี้

- **อัตราการรับส่งข้อมูล (Packet Delivery Fraction)** เป็นตัวเลขที่แสดงถึงจำนวนแพ็คเกจที่โหนดต้นทางสามารถส่งไปยังโหนดปลายทางด้วยจำนวนแพ็คเกจที่โหนดต้นทางส่งออกไป ซึ่งไม่นับรวมแพ็คเกจควบคุมการหาเส้นทาง ดังนั้นถ้าหากค่าที่ได้มีค่าใกล้เคียงหนึ่งมากเท่าไร ก็แสดงว่ามีประสิทธิภาพในการรับส่งข้อมูลที่สูงมากเท่านั้น
- **จำนวนแพ็คเกจที่ถูกทิ้ง (Dropped Packets)** เป็นตัวเลขที่แสดงถึงจำนวนแพ็คเกจที่ส่งจากโหนดต้นทางแล้วส่งไปไม่ถึงโหนดปลายทาง
- **จำนวนแพ็คเกจควบคุมที่ใช้ในการค้นหาเส้นทาง (Routing Packets)** เป็นตัวเลขที่แสดงถึงจำนวนแพ็คเกจควบคุมที่ใช้ในการค้นหาเส้นทาง ซึ่งรวมถึงจำนวนแพ็คเกจควบคุมที่ใช้ในการหาเส้นทางสำรองในโปรโตคอล Multi-hop AODV-2T ด้วย โดยจะนับรวมทุก ๆ โหนด แม้ว่าโหนดนั้น ๆ จะไม่เกี่ยวข้องกับการส่งข้อมูลเลยก็ตาม
- **ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูล (Average end-to-end Delay)** เป็นค่าเฉลี่ยของเวลาในการส่งแพ็คเกจข้อมูลจากโหนดต้นทางไปยังโหนดปลายทางโดยคำนวณจากเวลารวมที่แพ็คเกจข้อมูลทั้งหมดถูกส่งจากโหนดต้นทางจนกระทั่งไปถึงโหนดปลายทางด้วย

เอกสารนี้เป็นจำนวนแพ็คเกจข้อมูลที่ส่งไปยังปลายทางได้สำเร็จนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จำนวนครั้งที่เส้นทางการเชื่อมต่อถูกตัดขาด (Number of Route Break) เป็นตัวเลขที่แสดงถึงจำนวนครั้งที่เส้นทางการเชื่อมต่อถูกตัดขาด ซึ่งมีสาเหตุมาจาก 2 กรณี คือกรณีที่โหนดในเส้นทางเคลื่อนห่างออกไปจากเส้นทางการสื่อสาร และกรณีที่พลังงานของโหนดอ่อนลงจนไม่สามารถทำการสื่อสารได้

4.2 การจำลองการทำงานในสภาพแวดล้อมต่าง ๆ

งานวิจัยนี้ได้จำลองการทำงานของเครือข่ายในสภาพแวดล้อมที่แตกต่างกัน ดังในตารางที่ 4.1 ตารางที่ 4.2 และตารางที่ 4.3 เพื่อหาค่าประสิทธิภาพของโปรโตคอล Multi-hop AODV-2T เปรียบเทียบกับโปรโตคอล AODV-2T และโปรโตคอล AODV โดยใช้ค่าพารามิเตอร์ต่าง ๆ ที่กำหนดไว้ในหัวข้อที่ 4.1 โดยแบ่งการจำลองการทำงานของระบบเครือข่ายออกเป็น 3 ลักษณะเพื่อใช้ในการศึกษาประสิทธิภาพของโปรโตคอลดังกล่าว โดยการจำลองจะกำหนดคุณลักษณะต่าง ๆ จากการ์ดเครือข่ายชนิด Orinoco PCMCIA Silver/Gold ภายใต้มาตรฐาน 802.11b โดยจำลองในสภาพแวดล้อมต่าง ๆ ดังนี้คือ

- สภาพแวดล้อมที่ 1 เป็นสภาพแวดล้อมทั่วไปที่ใช้ในการสื่อสาร โดยไม่มีการปรับเปลี่ยนใด ๆ เพื่อเปรียบเทียบประสิทธิภาพการทำงานของทั้งสามโปรโตคอล
- สภาพแวดล้อมที่ 2 เพื่อศึกษาผลของขนาดพื้นที่ของระบบเครือข่ายที่มีผลต่อประสิทธิภาพการทำงานของทั้งสามโปรโตคอล โดยการปรับเปลี่ยนขนาดของพื้นที่ในการจำลองให้เพิ่มขึ้นอีกหนึ่งเท่า
- สภาพแวดล้อมที่ 3 เพื่อศึกษาผลของรูปแบบการเคลื่อนที่ของโหนดสื่อสาร โดยการปรับเวลาหยุดนิ่งอยู่กับที่ที่มีผลต่อประสิทธิภาพการทำงานของทั้งสามโปรโตคอล โดยมีการปรับเปลี่ยนเวลาหยุดนิ่งอยู่กับที่ให้เปลี่ยนแปลงอยู่ในช่วงระหว่าง 0 - 150 วินาทีต่อครั้ง

ตารางที่ 4.1 ข้อกำหนดในการจำลองการทำงานในสภาพแวดล้อมที่ 1

จำนวน โหนด	100
พื้นที่จำลองการทำงาน	500 เมตร x 500 เมตร
ความเร็วในการเคลื่อนที่ของโหนด	10, 20, 30, 40, 50 , 60, 70, 80, 90, 100, 110 เมตรต่อวินาที
จำนวนคู่สนทนา (ไม่ซ้ำกัน)	20 คู่
ชนิดของการรับส่งข้อมูล	ส่งข้อมูลด้วยอัตราคงที่ (CBR)
ขนาดแพ็คเกจ	512 ไบต์
เวลาที่ใช้ในการจำลอง	150 วินาที
ระยะการสื่อสาร	100 เมตร
ค่าความถี่	2.45 GHz
อัตราข้อมูล (Data Rate)	11 Mbps
Pt (Transmit Power)	0.02998531 วัตต์ (14.771dBm)
Gt (Antenna gain ฟังส่ง)	2.4 dBi
Gr (Antenna gain ฟังรับ)	2.4 dBi
ht (Height Antenna ฟังส่ง)	1.0 เมตร
hr (Height Antenna ฟังรับ)	1.0 เมตร
เรขาคณิตของสัญญาณระดับที่ 1	9.07×10^{-10} วัตต์
เรขาคณิตของสัญญาณระดับที่ 2	7.31×10^{-10} วัตต์
พลังงานเริ่มต้นของแบตเตอรี่	150 จูล
เรขาคณิตของพลังงานระดับที่ 1	10 จูล
เรขาคณิตของพลังงานระดับที่ 2	5 จูล

ตารางที่ 4.2 ข้อกำหนดในการจำลองการทำงานในสภาพแวดล้อมที่ 2

จำนวน โหนด	100
พื้นที่จำลองการทำงาน	1000 เมตร x 1000 เมตร
ความเร็วในการเคลื่อนที่ของโหนด	10, 20, 30, 40, 50 , 60, 70, 80, 90, 100, 110 เมตรต่อวินาที
จำนวนคู่สนทนา (ไม่ซ้ำกัน)	20 คู่
ชนิดของการรับส่งข้อมูล	ส่งข้อมูลด้วยอัตราคงที่ (CBR)
ขนาดแพ็คเกจ	512 ไบต์
เวลาที่ใช้ในการจำลอง	150 วินาที
เรขาคณิตของสัญญาณระดับที่ 2	7.31×10^{-10} วัตต์
พลังงานเริ่มต้นของแบตเตอรี่	150 จูล

ตารางที่ 4.2 การจำลองการทำงานในสภาพแวดล้อมที่ 2 (ต่อ)

แรงแม่เหล็กของพลังงานระดับที่ 1	10 จูล
แรงแม่เหล็กของพลังงานระดับที่ 2	5 จูล
ระยะการสื่อสาร	100 เมตร
ค่าความถี่	2.45 GHz
อัตราข้อมูล (Data Rate)	11 Mbps
Pt (Transmit Power)	0.02998531 วัตต์ (14.771dBm)
Gt (Antenna gain ฟังส่ง)	2.4 dBi
Gr (Antenna gain ฟังรับ)	2.4 dBi
ht (Height Antenna ฟังส่ง)	1.0 เมตร
hr (Height Antenna ฟังรับ)	1.0 เมตร
แรงแม่เหล็กของสัญญาณระดับที่ 1	9.07×10^{-10} วัตต์
แรงแม่เหล็กของสัญญาณระดับที่ 2	7.31×10^{-10} วัตต์
พลังงานเริ่มต้นของแบตเตอรี่	150 จูล
แรงแม่เหล็กของพลังงานระดับที่ 1	10 จูล
แรงแม่เหล็กของพลังงานระดับที่ 2	5 จูล

ตารางที่ 4.3 ข้อกำหนดในการจำลองการทำงานในสภาพแวดล้อมที่ 3 (ปรับเปลี่ยนเวลาหยุดนิ่งอยู่กับที่)

จำนวนโหนด	100
พื้นที่จำลองการทำงาน	1000 เมตร x 1000 เมตร
ความเร็วในการเคลื่อนที่ของโหนด	110 เมตรต่อวินาที
จำนวนคู่สนทนา (ไม่ซ้ำกัน)	20 คู่
ชนิดของการรับส่งข้อมูล	ส่งข้อมูลด้วยอัตราคงที่ (CBR)
ขนาดแพ็คเกจ	512 ไบต์
เวลาที่ใช้ในการจำลอง	150 วินาที
เวลาหยุดอยู่กับที่	10 วินาที
ระยะการสื่อสาร	100 เมตร
ค่าความถี่	2.45 GHz
อัตราข้อมูล (Data Rate)	11 Mbps
Pt (Transmit Power)	0.02998531 w (14.771dBm)
Gt (Antenna gain ฟังส่ง)	2.4 dBi

ตารางที่ 4.3 ข้อกำหนดในการจำลองการทำงานในสภาพแวดล้อมที่ 3 (ปรับเปลี่ยนเวลาหยุดนิ่งอยู่กับที่) (ต่อ)

Gr (Antenna gain ฟังรับ)	2.4 dBi
ht (Height Antenna ฟังส่ง)	1.0 เมตร
hr (Height Antenna ฟังรับ)	1.0 เมตร
เชรชโฮลต์ของสัญญาณระดับที่ 1	9.07×10^{-10} วัตต์
เชรชโฮลต์ของสัญญาณระดับที่ 2	7.31×10^{-10} วัตต์
พลังงานเริ่มต้นของแบตเตอรี่	150 จูล
เชรชโฮลต์ของพลังงานระดับที่ 1	10 จูล
เชรชโฮลต์ของพลังงานระดับที่ 2	5 จูล

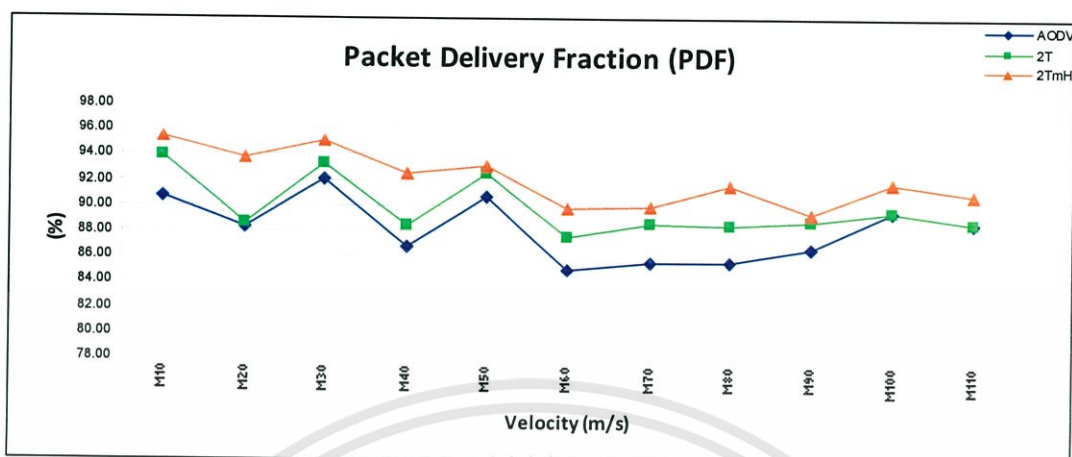
หนึ่งในสภาพแวดล้อมจำลองนี้ เราจะแบ่งช่วงของการอธิบายกราฟที่แสดงผลนี้ออกเป็น 3 ช่วง ช่วงแรกคือการจำลองกำหนดให้ช่วงระยะเวลาที่โหนดหยุดนิ่งอยู่กับที่ค่อนข้างนาน ซึ่งหมายความว่าโหนดมีการเคลื่อนที่น้อย โดยมีเวลาหยุดนิ่งอยู่กับที่ระหว่าง 110 - 150 วินาที ช่วงที่สองคือช่วงที่มีเวลาหยุดนิ่งอยู่กับที่ปานกลาง เวลาหยุดนิ่งอยู่กับที่ระหว่าง 60 - 100 วินาที และช่วงสุดท้ายคือช่วงที่มีเวลาหยุดนิ่งอยู่กับที่น้อย คือมีเวลาหยุดนิ่งอยู่กับที่ระหว่างตั้งแต่ 0 - 50 วินาที ซึ่งหมายความว่าโหนดมีการเคลื่อนที่สูง

4.3 ผลการจำลอง

4.3.1 ผลการจำลองในสภาพแวดล้อมที่ 1

การจำลองในสภาพแวดล้อมที่ 1 เป็นการจำลองเพื่อวัดผลประสิทธิภาพของโปรโตคอล Multi-hop AODV-2T เปรียบเทียบกับโปรโตคอล AODV และโปรโตคอล AODV-2T เมื่อความเร็วในการเคลื่อนที่ของโหนดเปลี่ยนแปลงไป ประสิทธิภาพของการทำงานในโปรโตคอลทั้งสามนั้นจะเป็นอย่างไร ซึ่งค่าความเร็วในการเคลื่อนที่ของโหนดที่ใช้ในการจำลองได้กล่าวไว้แล้วในตารางที่ 4.1 โดยในหัวข้อนี้จะอธิบายกราฟโดยอ้างอิงอัตราของความเร็วในการเคลื่อนที่ของโหนด ซึ่งเมื่อโหนดเคลื่อนที่เร็วขึ้น การสื่อสารจะมีประสิทธิภาพลดลง เพราะการเชื่อมต่อจะถูกตัดขาดได้ง่ายขึ้น โดยการจำลองเราจะกำหนดให้ค่าหยุดพักเป็น 0 วินาที นั่นหมายความว่าโหนดจะเคลื่อนที่ตลอดเวลา

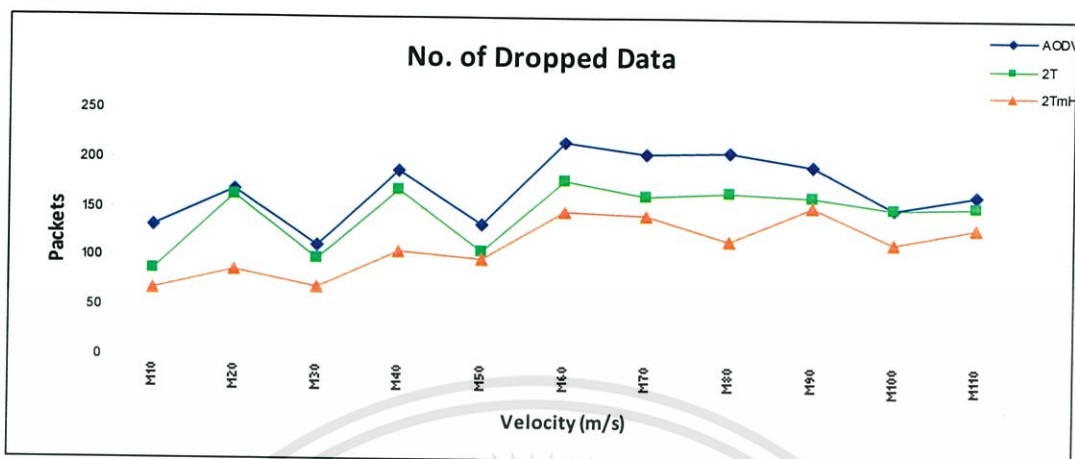
4.3.1.1 อัตราการส่งข้อมูลสำเร็จในสภาพแวดล้อมที่ 1



รูปที่ 4.1 อัตราการส่งข้อมูลสำเร็จในสภาพแวดล้อมที่ 1

จากกราฟในรูปที่ 4.1 เป็นการเปรียบเทียบอัตราการส่งข้อมูลสำเร็จระหว่างโปรโตคอล AODV โปรโตคอล AODV-2T และโปรโตคอล Multi-hop AODV-2T แสดงให้เห็นว่าโปรโตคอล Multi-hop AODV-2T มีอัตราการส่งข้อมูลได้สำเร็จมากกว่าโปรโตคอลอีกสองชนิดถึงแม้ว่าในช่วงแรกที่มีการเคลื่อนที่ของโหนดด้วยความเร็วต่ำการทำงานของโปรโตคอลทั้งสามจะมีผลของการส่งข้อมูลสำเร็จที่ใกล้เคียงกัน ทั้งนี้เพราะว่าในช่วงที่โหนดมีการเคลื่อนที่ด้วยความเร็วต่ำเส้นทางการรับส่งข้อมูลจะไม่เสี่ยงต่อการถูกตัดขาดการเชื่อมต่อ แต่ในช่วงที่มีการเคลื่อนที่ของโหนดด้วยความเร็วที่สูงขึ้นโปรโตคอล Multi-hop AODV-2T กลับแสดงผลการจำลองที่ดีกว่า ทั้งนี้เพราะเมื่อโหนดมีการเคลื่อนที่ด้วยความเร็วที่มากขึ้นย่อมทำให้เส้นทางการเชื่อมต่อจะถูกตัดขาดได้ง่ายขึ้น ดังนั้นเนื่องจากโปรโตคอล Multi-hop AODV-2T มีการค้นหาและสร้างเส้นทางสำรองไว้ก่อนล่วงหน้า ทำให้การสื่อสารยังดำเนินการต่อไปได้อย่างต่อเนื่องและทำให้สามารถส่งข้อมูลไปถึงโหนดปลายทางได้สำเร็จมากกว่าโปรโตคอล AODV และโปรโตคอล AODV-2T

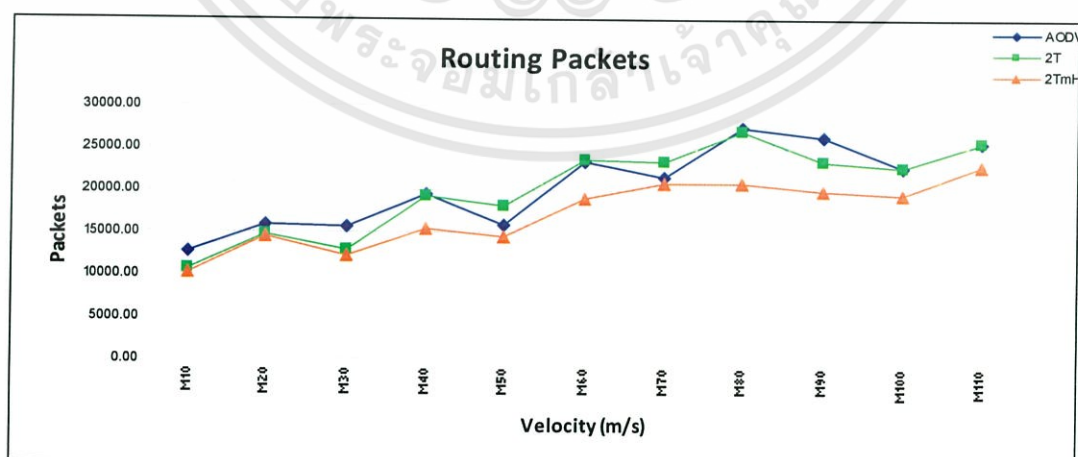
4.3.1.2 จำนวนแพ็คเกจที่ถูกทิ้ง (Dropped Packet) ในสภาพแวดล้อมที่ 1



รูปที่ 4.2 จำนวนแพ็คเกจที่ถูกทิ้งจากผลการจำลองในสภาพแวดล้อมที่ 1

การเปรียบเทียบจำนวนแพ็คเกจที่ถูกทิ้งที่ตรวจวัดได้จากการจำลองสภาพแวดล้อมที่ 1 ระหว่างโปรโตคอล AODV โปรโตคอล AODV-2T และ โปรโตคอล Multi-hop AODV-2T ที่ได้จากกราฟในรูปที่ 4.2 นั้น พบว่าโปรโตคอล Multi-hop AODV-2T สามารถลดจำนวนของแพ็คเกจที่ถูกทิ้งลงได้ถึง 20-30 เปอร์เซ็นต์เมื่อเทียบกับโปรโตคอล AODV-2T และโปรโตคอล AODV

4.3.1.3 จำนวนแพ็คเกจควบคุมที่ใช้ในการค้นหาเส้นทาง (Routing Packets) ในสภาพแวดล้อมที่ 1

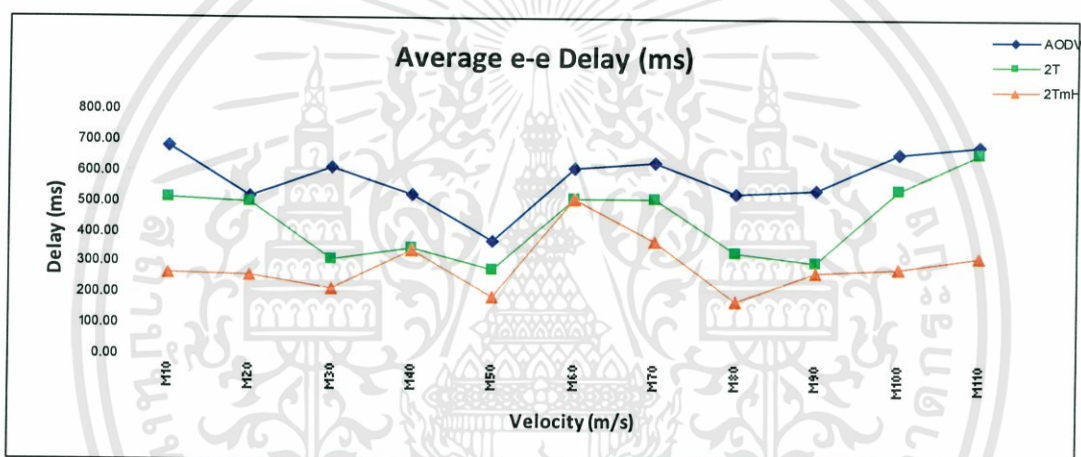


รูปที่ 4.3 จำนวนแพ็คเกจควบคุมในการจำลองสภาพแวดล้อมที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟในรูปที่ 4.3 แสดงการเปรียบเทียบจำนวนแพ็คเก็ตควบคุมทั้งหมดที่ใช้ในการค้นหาเส้นทางของโปรโตคอล AODV โปรโตคอล AODV-2T และโปรโตคอล Multi-hop AODV-2T จะพบว่ามีค่าที่ได้จากโปรโตคอล Multi-hop AODV-2T มีค่าต่ำกว่าค่าที่ได้จากโปรโตคอลอีกสองชนิดที่เหลือ ถึงแม้ว่าในโปรโตคอล Multi-AODV-2T จะมีการส่งแพ็คเก็ตควบคุมการค้นหาเส้นทางสำรองออกมาในระบบก็ตาม ซึ่งแสดงให้เห็นว่าการสร้างเส้นทางสำรองไม่ได้เป็นการเพิ่มจำนวนของแพ็คเก็ตควบคุมมากเกินไปนัก ทั้งนี้เพราะว่าการส่งแพ็คเก็ตควบคุมเพื่อค้นหาเส้นทางสำรองของโปรโตคอล Multi-hop AODV-2T นั้นจะกระทำเฉพาะในตอนทีเส้นทางส่งข้อมูลที่มีโอกาสเสี่ยงต่อการถูกตัดขาดการเชื่อมต่อเท่านั้น

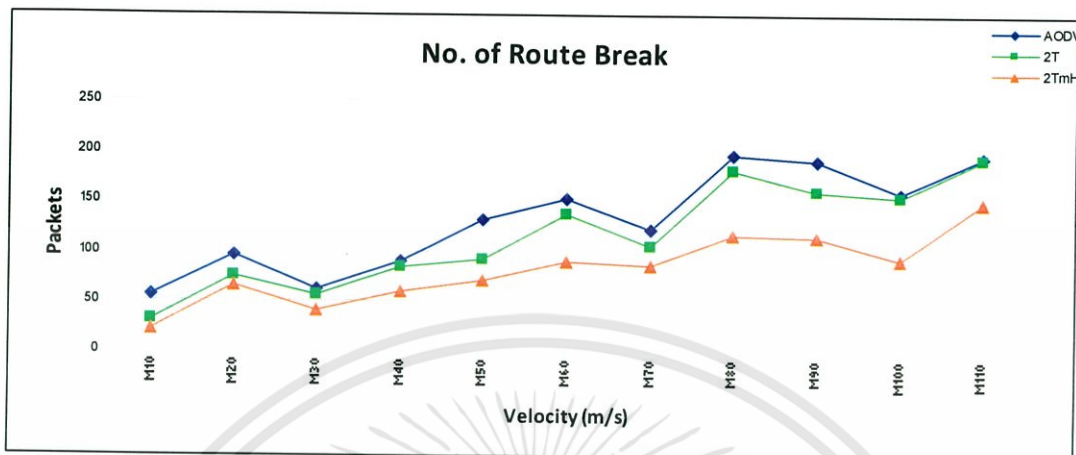
4.3.1.4 ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูล (Average end-to-end Delay) ในสภาพแวดล้อมที่ 1



รูปที่ 4.4 ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลจากการจำลองตามสภาพแวดล้อมที่ 1

จากกราฟเปรียบเทียบค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลระหว่างโปรโตคอล AODV โปรโตคอล AODV-2T และโปรโตคอล Multi-hop AODV 2T ในรูปที่ 4.4 จะพบว่าค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลของโปรโตคอล Multi-hop AODV-2T มีค่าต่ำกว่าค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลของโปรโตคอล AODV และโปรโตคอล AODV-2T ดังนั้นเมื่อโปรโตคอล Multi-hop AODV-2T เริ่มต้นกระบวนการหาเส้นทางสำรองเมื่อเส้นทางเชื่อมต่อมีความเสี่ยงที่จะถูกตัดขาด ซึ่งในขณะที่ระบบกำลังดำเนินการหาเส้นทางสำรองอยู่นั้นการสื่อสารที่กำลังดำเนินการอยู่ในเส้นทางหลักจะไม่ชะงักงัน ทำให้การสื่อสารในระบบเครือข่ายไม่ต้องเสียเวลาในการรอคอยการค้นหาเส้นทางสำรองในสถานการณ์ที่การเชื่อมต่อปัจจุบันถูกตัดขาดออกไป

4.3.1.5 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อ (Number of Route Break) ในสภาพแวดล้อมที่ 1



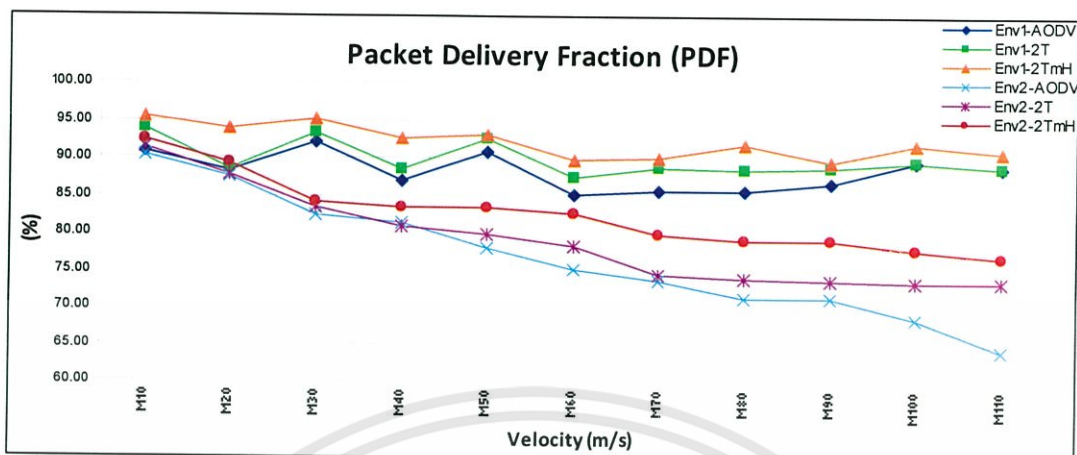
รูปที่ 4.5 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในการจำลองสภาพแวดล้อมที่ 1

จากรูปที่ 4.5 แสดงให้เห็นว่าโปรโตคอล Multi-hop AODV-2T สามารถลดจำนวนเส้นทางที่การเชื่อมต่อที่ถูกตัดขาดลงไปได้ประมาณ 20-30 เปอร์เซ็นต์ เมื่อเทียบกับโปรโตคอล AODV และโปรโตคอล AODV-2T ทั้งนี้เพราะว่าในโปรโตคอล Multi-hop AODV-2T มีการแจ้งเตือนแก่โหนดในเส้นทางเมื่อเส้นทางมีความเสี่ยงต่อการถูกตัดขาดและเปลี่ยนไปใช้เส้นทางสำรองเมื่อเส้นทางที่การเชื่อมต่อกำลังจะถูกตัดขาด ซึ่งทำให้ความเสียหายของการเชื่อมต่อในโปรโตคอล Multi-hop AODV-2T จึงเกิดขึ้นได้น้อยกว่าในโปรโตคอล AODV และโปรโตคอล AODV-2T ซึ่งจะเห็นได้ผลชัดเจนเมื่อโหนดมีการเคลื่อนที่มากขึ้น

4.3.2 ผลการจำลองในสภาพแวดล้อมที่ 2 เมื่อมีการปรับเปลี่ยนพื้นที่ของระบบเครือข่าย

การจำลองในสภาพแวดล้อมที่มีการปรับเปลี่ยนขนาดของพื้นที่การจำลองจากเดิมขนาด 500 x 500 ตารางเมตร เป็น 1000 x 1000 ตารางเมตร ดังในตารางที่ 4.2 เพื่อศึกษาประสิทธิภาพของโปรโตคอล Multi-hop AODV-2T เปรียบเทียบกับโปรโตคอล AODV และโปรโตคอล AODV-2T ในเครือข่ายที่มีลักษณะพื้นที่ที่กว้างขึ้น ซึ่งทำให้โหนดต้นทางกับโหนดปลายทางอยู่ห่างกันมากขึ้น เส้นทางที่ใช้สื่อสารจะต้องมีจำนวนฮอปมากขึ้น และการค้นหาเส้นทางจะต้องใช้แพ็คเกจควบคุมการหาเส้นทางจึงมีมากขึ้นด้วย ซึ่งผลจากการจำลองจะเป็นการเปรียบเทียบกันระหว่างสภาพแวดล้อมที่ 1 และ สภาพแวดล้อมที่ 2 โดยให้ ENV1 แทนกราฟจากสภาพแวดล้อมที่ 1 และ ENV2 แทนกราฟจากสภาพแวดล้อมที่ 2

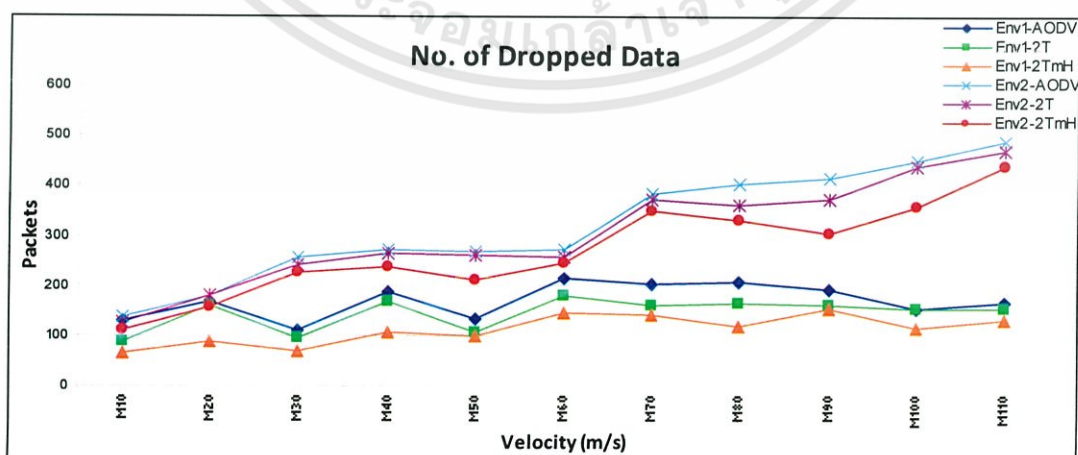
4.3.2.1 อัตราการส่งข้อมูลสำเร็จในสภาพแวดล้อมที่ 2



รูปที่ 4.6 อัตราการส่งข้อมูลสำเร็จในสภาพแวดล้อมที่ 2

จากกราฟในรูปที่ 4.6 แสดงให้เห็นว่าโปรโตคอล Multi-hop AODV-2T มีอัตราการส่งข้อมูลสำเร็จที่สูงกว่าโปรโตคอล AODV และโปรโตคอล AODV-2T ในเครือข่ายที่มีพื้นที่กว้างขึ้น ทั้งนี้เนื่องจากเส้นทางการสื่อสารข้อมูลในเครือข่ายที่มีความกว้างขึ้นนั้น อาจจะมีบางเส้นทางที่โหนดต้นทางอยู่ไกลจากโหนดปลายทางมาก ทำให้เส้นทางการสื่อสารเสี่ยงต่อการถูกตัดขาด การเชื่อมต่อมากขึ้น ยิ่งเมื่อโหนดมีการเคลื่อนที่เร็วขึ้นก็ย่อมมีโอกาสที่เส้นทางการเชื่อมต่อจะถูกตัดขาดมากขึ้น ดังแสดงจากกราฟในรูปที่ 4.6 จะพบว่าอัตราการรับส่งข้อมูลสำเร็จของโปรโตคอล AODV ในช่วงเวลาที่มีความเร็วที่สูงมาก ๆ (ช่วงความเร็วระหว่าง 70-110 m/s) นั้นมีค่าลดลงอย่างมากเมื่อเทียบโปรโตคอล Multi-hop AODV-2T

4.3.2.2 จำนวนแพ็คเกจที่ถูกทิ้งในสภาพแวดล้อมที่ 2

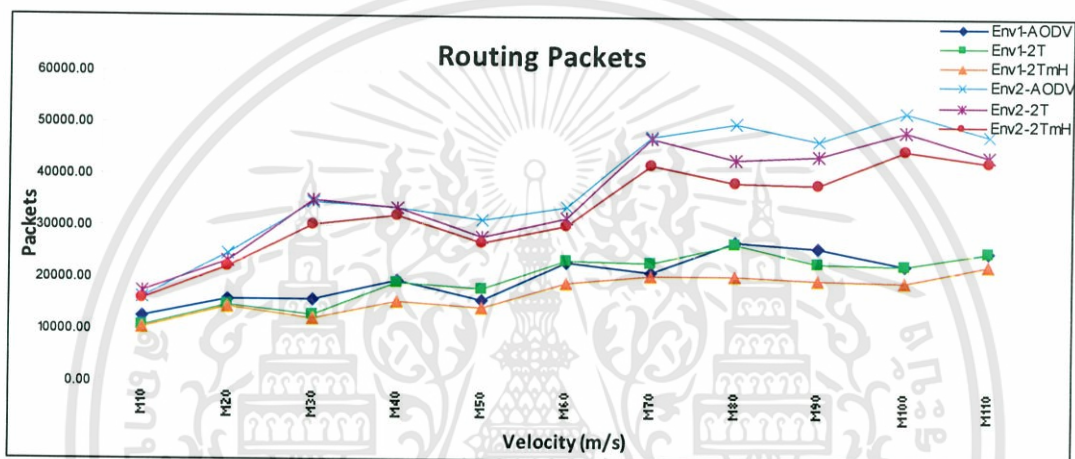


รูปที่ 4.7 จำนวนแพ็คเกจที่ถูกทิ้งในการจำลองสภาพแวดล้อมที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปรูปที่ 4.7 แสดงให้เห็นว่าจำนวนแพ็คเกจที่ถูกทิ้งของทุกโพรโทคอลในสภาพแวดล้อมที่ 2 มีค่ามากกว่าการจำลองในสภาพแวดล้อมที่ 1 เพราะว่าในสถานการณ์ที่พื้นที่เครือข่ายมีขนาดใหญ่ขึ้นจะเป็นการเพิ่มระยะห่างระหว่างโหนดมากขึ้นด้วย ทำให้เส้นทางสื่อสารมีโอกาสเสี่ยงที่จะเสียหายและถูกตัดขาดมากขึ้น แต่การทำงานของโพรโทคอล Multi-hop AODV-2T มีประสิทธิภาพที่มากกว่า ในแง่ที่ทำให้อัตราการทิ้งแพ็คเกจต่ำกว่าเมื่อเทียบกับโพรโทคอล AODV และ AODV-2T ซึ่งก็ตรงตามสมมุติฐาน

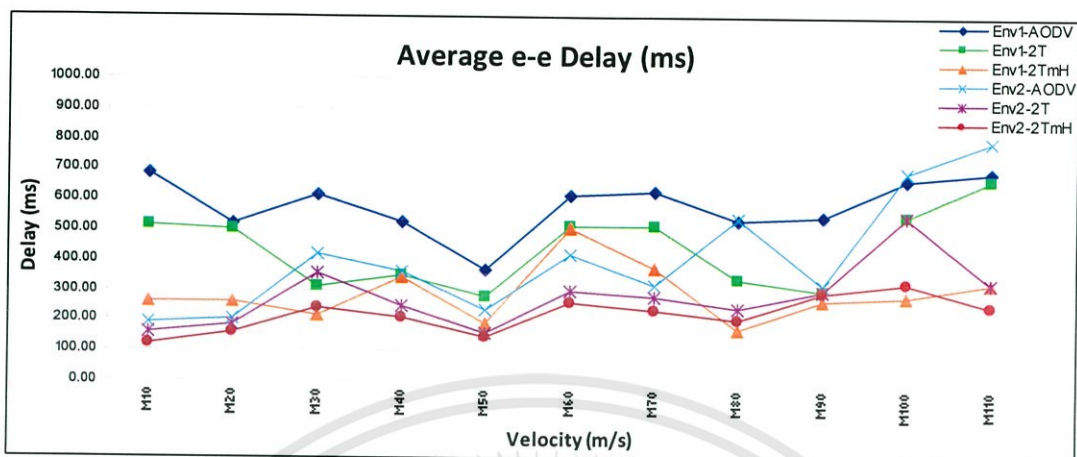
4.3.2.3 จำนวนแพ็คเกจควบคุมที่ใช้ในการค้นหาเส้นทางในสภาพแวดล้อมที่ 2



รูปที่ 4.8 จำนวนแพ็คเกจควบคุมที่ใช้ในการค้นหาเส้นทางในสภาพแวดล้อมที่ 2

ถึงแม้ว่าในโพรโทคอล Multi-hop AODV-2T จะมีการใช้แพ็คเกจควบคุมพิเศษในการค้นหาเส้นทางในระบบสื่อสารเพิ่มขึ้น นอกเหนือจากแพ็คเกจควบคุมการค้นหาเส้นทางของโพรโทคอล AODV เดิม แต่ก็ไม่ได้มีส่วนเพิ่มจำนวนแพ็คเกจควบคุมในระบบเครือข่ายเท่าใดนัก ดังจะเห็นได้จากรูปที่ 4.8 ว่าจำนวนแพ็คเกจควบคุมการค้นหาเส้นทางของโพรโทคอล Multi-hop AODV-2T มีปริมาณน้อยกว่าปริมาณแพ็คเกจควบคุมจากโพรโทคอล AODV และโพรโทคอล AODV-2T เนื่องจากการส่งแพ็คเกจควบคุมเพื่อค้นหาเส้นทางสำรองของโพรโทคอล Multi-hop AODV-2T นั้นจะกระทำเฉพาะในกรณีที่เส้นทางการส่งข้อมูลมีโอกาสเสี่ยงต่อการถูกตัดขาดการเชื่อมต่อเท่านั้น แต่เมื่อสถานการณ์ที่พื้นที่เครือข่ายมีขนาดใหญ่ขึ้นซึ่งจะเป็นการเพิ่มระยะห่างระหว่างโหนดมากขึ้น ทำให้เส้นทางสื่อสารมีโอกาสเสี่ยงที่จะเสียหายและถูกตัดขาดมากขึ้น ดังนั้นจะเห็นได้ว่าจำนวนแพ็คเกจควบคุมการค้นหาเส้นทางของทั้งสามโพรโทคอลก็จะมีจำนวนเพิ่มขึ้นด้วย

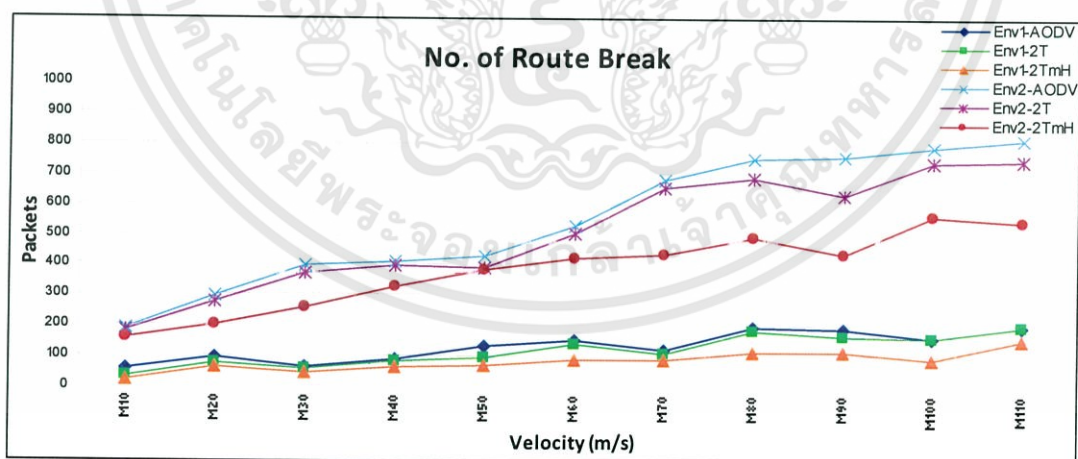
4.3.2.4 ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลในสภาพแวดล้อมที่ 2



รูปที่ 4.9 ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลในสภาพแวดล้อมที่ 2

จากรูปที่ 4.9 ค่าเฉลี่ยเวลาที่ใช้ในการส่งข้อมูลของทั้งสามโปรโตคอลมีค่าที่ใกล้เคียงกัน เพราะเส้นทางสำรองของโปรโตคอล Multi-hop AODV-2T ไม่ได้เพิ่มจำนวนฮอปให้เส้นทางแต่อย่างใด ทำให้เวลาที่แพ็คเก็ตข้อมูลไปถึงโหนดปลายทางมีค่าที่ใกล้เคียงกัน

4.3.2.5 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในสภาพแวดล้อมที่ 2



รูปที่ 4.10 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในสภาพแวดล้อมที่ 2

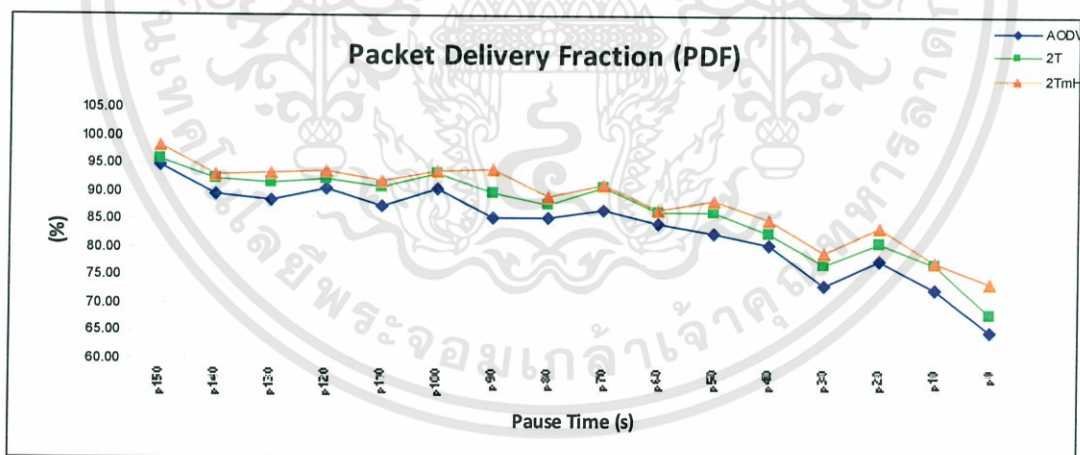
รูปที่ 4.10 แสดงให้เห็นว่าโปรโตคอล Multi-hop AODV-2T สามารถลดจำนวนของเส้นทางที่เชื่อมต่อถูกตัดขาดลงได้เมื่อเทียบกับโปรโตคอล AODV ถึงแม้ว่าขนาดพื้นที่ของการจำลองจะกว้างขึ้น ทั้งนี้เพราะโปรโตคอล Multi-hop AODV-2T มีกระบวนการค้นหาเส้นทาง

สำรองมาเป็นตัวช่วยเมื่อเส้นทางสื่อสารเสี่ยงต่อการถูกตัดขาด ทำให้การส่งข้อมูลไปยังโหนดปลายทางสามารถดำเนินการไปได้อย่างต่อเนื่อง โดยที่เส้นทางที่ใช้ในการส่งข้อมูลไปยังโหนดปลายทางไม่ได้ถูกตัดขาดแต่อย่างใด

4.3.3 ผลการจำลองในสภาพแวดล้อมที่ 3 เมื่อมีการปรับเปลี่ยนเวลาหยุดนิ่งอยู่กับที่ของโหนด

การจำลองในสภาพแวดล้อมที่ 3 เพื่อศึกษาการเปลี่ยนแปลงของเวลาหยุดนิ่งอยู่กับที่ที่มีผลต่อประสิทธิภาพการทำงานของโปรโตคอล Multi-hop AODV-2T เปรียบเทียบกับโปรโตคอล AODV และโปรโตคอล AODV-2T โดยในสภาพแวดล้อมจำลองนี้เราจะแบ่งช่วงของการอธิบายกราฟที่แสดงผลนี้ออกเป็น 3 ช่วง ช่วงแรกคือช่วงระยะเวลาที่โหนดหยุดนิ่งอยู่กับที่ค่อนข้างนาน ซึ่งหมายความว่าโหนดมีการเคลื่อนที่น้อย โดยมีเวลาหยุดนิ่งอยู่กับที่ระหว่าง 110 - 150 วินาที ช่วงที่สองคือช่วงที่มีเวลาหยุดนิ่งอยู่กับที่ปานกลาง เวลาหยุดนิ่งอยู่กับที่ระหว่าง 60 - 100 วินาที และช่วงสุดท้ายคือช่วงที่มีเวลาหยุดนิ่งอยู่กับที่น้อย คือมีเวลาหยุดนิ่งอยู่กับที่ระหว่างตั้งแต่ 0 - 50 วินาที

4.3.3.1 อัตราการส่งข้อมูลสำเร็จในการจำลองในสภาพแวดล้อมที่ 3



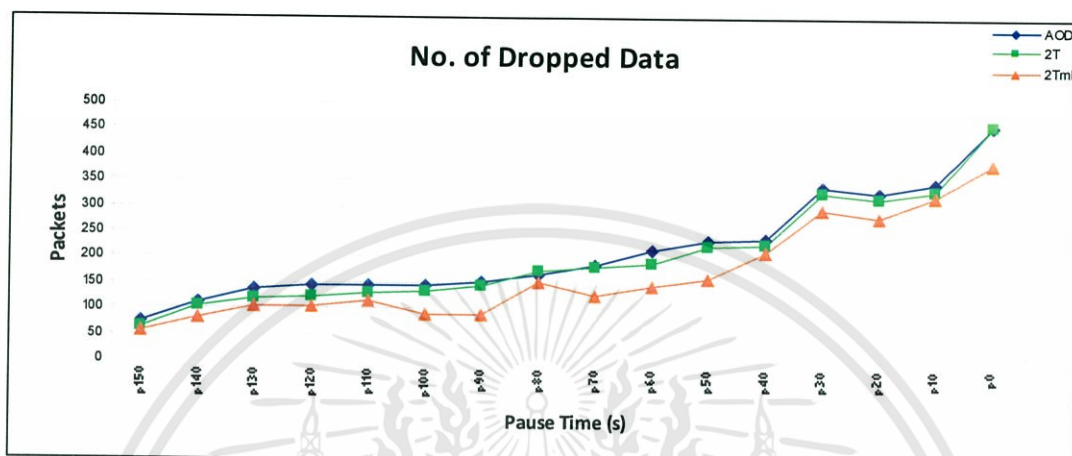
รูปที่ 4.11 อัตราการส่งข้อมูลสำเร็จในการจำลองในสภาพแวดล้อมที่ 3

เมื่อโหนดมีอัตราการเคลื่อนที่ที่สูงขึ้น เส้นทางที่เชื่อมต่อย่อมถูกตัดขาดได้ง่ายขึ้น อัตราการส่งข้อมูลไปถึงปลายทางได้สำเร็จย่อมลดลง ดังที่แสดงในกราฟรูปที่ 4.11 อัตราการส่งข้อมูลสำเร็จซึ่งลดลงมากเมื่อเทียบกับการจำลองในสภาพแวดล้อมที่ 1 และสภาพแวดล้อมที่ 2 เมื่อโหนดเคลื่อนที่ด้วยอัตราที่สูงขึ้นอัตราการส่งข้อมูลสำเร็จยังลดลงมากขึ้น แต่ผลที่ได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอล Multi-hop AODV-2T ก็ยังคงมีอัตราการส่งข้อมูลได้สำเร็จมากกว่าโปรโตคอล AODV และ โปรโตคอล AODV-2T ตามที่ได้ตั้งสมมุติฐานไว้

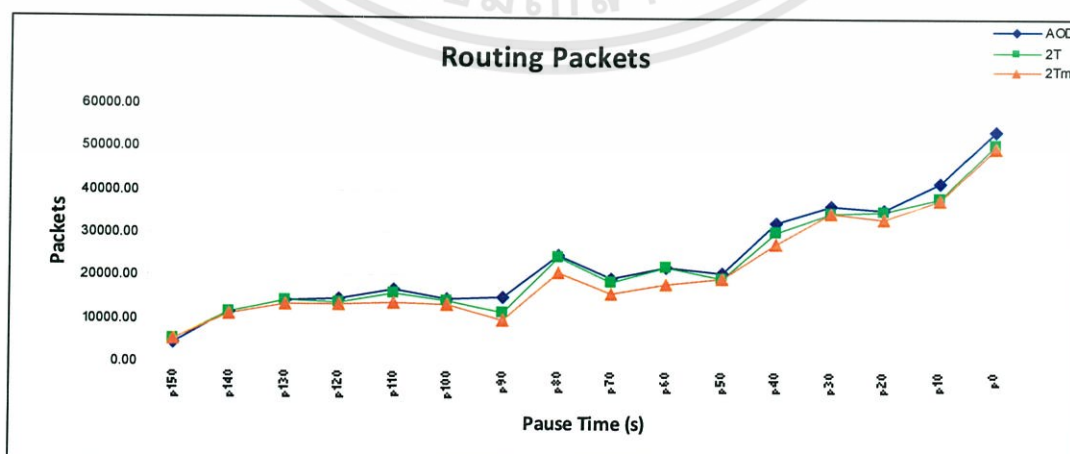
4.3.3.2 จำนวนแพ็คเกจที่ถูกทิ้งในสภาพแวดล้อมที่ 3



รูปที่ 4.12 จำนวนแพ็คเกจที่ถูกทิ้งในสภาพแวดล้อมที่ 3

เมื่อโหนดมีอัตราการเคลื่อนที่สูงมากขึ้น จำนวนแพ็คเกจที่ถูกทิ้งก็จะมากขึ้นตามไปด้วย ดังกราฟรูปที่ 4.12 โดยโปรโตคอล Multi-hop AODV-2T สามารถลดจำนวนแพ็คเกจที่ถูกทิ้งลงได้ประมาณ 15-20 % ยิ่งอัตราการเคลื่อนที่ของโหนดสูงขึ้น ยิ่งทำให้เห็นถึงประสิทธิภาพของโปรโตคอล Multi-hop AODV-2T ได้ชัดเจนกว่าโปรโตคอลอันดิงอื่นเพราะจำนวนแพ็คเกจที่ถูกทิ้งของแต่ละโปรโตคอลจะแตกต่างกันมากขึ้น

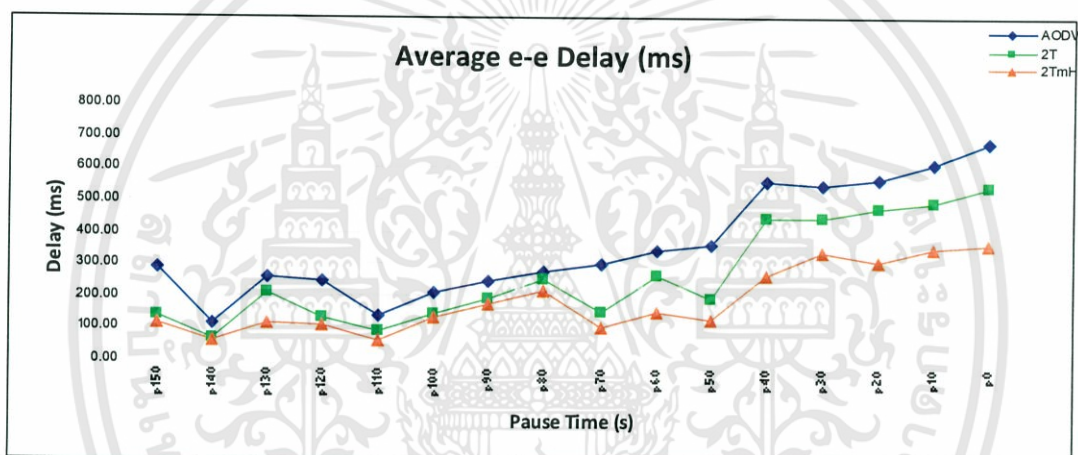
4.3.3.3 จำนวนแพ็คเกจควบคุมที่ใช้ในการค้นหาเส้นทางในสภาพแวดล้อมที่ 3



เอกสารนี้เป็น รูปที่ 4.13 จำนวนแพ็คเกจควบคุมที่ใช้ในการค้นหาเส้นทางในสภาพแวดล้อมที่ 3 ขอบนด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.13 แสดงให้เห็นว่าจำนวนแพ็คเกจควบคุมเส้นทางของทุกโพรโทคอลมีจำนวนมากขึ้นเรื่อย ๆ ตามอัตราการเคลื่อนที่ที่สูงขึ้น ทั้งนี้เพราะเมื่อโหนดมีอัตราการเคลื่อนที่ที่สูงขึ้น การหาเส้นทางไปยังโหนดปลายทางจะยากขึ้น เพราะในขณะที่โหนดระหว่างทางหาเส้นทางไปยังโหนดปลายทางแต่โหนดใดโหนดหนึ่งในเส้นทางสื่อสารอาจจะเคลื่อนที่ออกไปจากระยะสื่อสารก่อนที่จะก่อตั้งการสื่อสารได้สำเร็จ ทำให้ต้องมีการส่งแพ็คเกจร้องขอเส้นทางออกไปใหม่ หรือถึงแม้มีเส้นทางสื่อสารข้อมูลอยู่แล้ว แต่ด้วยการเคลื่อนที่ของโหนดมีอัตราการเคลื่อนที่ที่สูงขึ้นก็อาจจะทำให้การสื่อสารถูกตัดขาดได้โดยง่าย ซึ่งผลการทดลองชี้ให้เห็นว่าปริมาณแพ็คเกจควบคุมของทุกโพรโทคอลก็จะแปรเปลี่ยนไปตามอัตราการเคลื่อนที่ที่สูงขึ้นเรื่อย ๆ

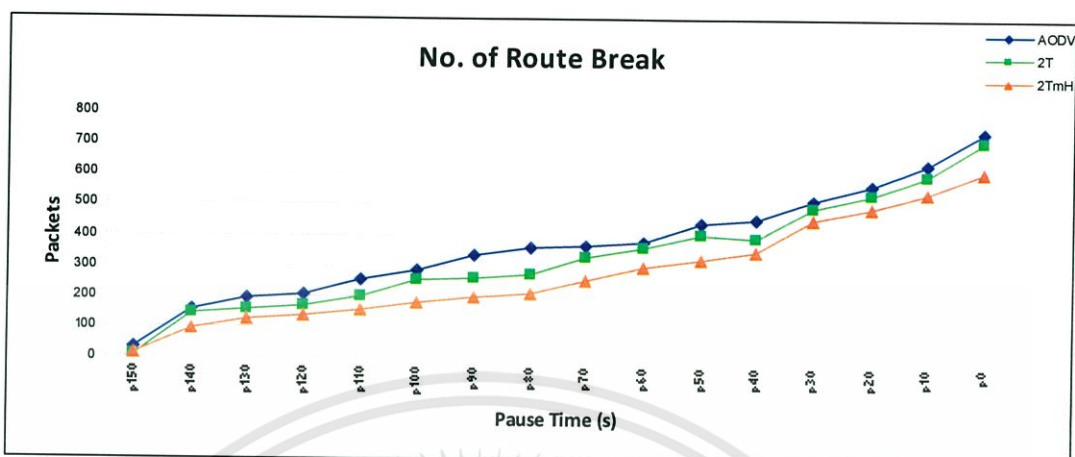
4.3.3.4 ค่าเฉลี่ยของเวลาที่ใช้ในการส่งข้อมูลในสภาพแวดล้อมที่ 3



รูปที่ 4.14 ค่าเฉลี่ยของเวลาที่ใช้ในการส่งข้อมูลในสภาพแวดล้อมที่ 3

จากรูปที่ 4.14 แสดงว่าค่าเฉลี่ยของเวลาที่ใช้ในการส่งข้อมูลของทั้งสามโพรโทคอลในสภาพแวดล้อมที่ 3 นั้นมีค่าที่ใกล้เคียงกัน ซึ่งแสดงให้เห็นว่าการเคลื่อนที่ของโหนดที่มีอัตราการเคลื่อนที่ที่สูงขึ้น มีการเพิ่มกลไกการแจ้งเตือนและกลไกการค้นหาเส้นทางสำรองของโพรโทคอล Multi-hop AODV-2T มากขึ้นนั้น ไม่ได้ทำให้เวลาเฉลี่ยที่ใช้ในการส่งข้อมูลให้กับระบบสื่อสารเพิ่มขึ้นแต่อย่างใด และเส้นทางสำรองที่ใช้ก็ไม่ได้มีส่วนในการเพิ่มจำนวนฮอปของเส้นทางสื่อสารแต่อย่างใด

4.3.3.5 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในสภาพแวดล้อมที่ 3



รูปที่ 4.15 จำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในสภาพแวดล้อมที่ 3

จากรูปที่ 4.15 แสดงให้เห็นว่าจำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อในโปรโตคอล Multi-hop AODV-2T มีจำนวนน้อยกว่าเมื่อเทียบกับโปรโตคอล AODV และโปรโตคอล AODV-2T ซึ่งเมื่อโหนดเคลื่อนที่ด้วยอัตราการเคลื่อนที่ที่สูงขึ้นจำนวนของเส้นทางที่ถูกตัดขาดย่อมเพิ่มขึ้นตามไปด้วย ซึ่งในโปรโตคอล AODV ไม่มีกลไกในการสร้างเส้นทางสำรองล่วงหน้าและไม่มีการแจ้งเตือนถึงอันตรายของเส้นทางที่จะถูกตัดขาดการเชื่อมต่อโดยไม่มีเส้นทางสำรองมารับการสื่อสารข้อมูลที่กำลังดำเนินอยู่ จากกราฟจะเห็นว่าเมื่ออัตราการเคลื่อนที่ของโหนดเพิ่มขึ้น ความแตกต่างของโปรโตคอล Multi-hop AODV-2T AODV และ AODV-2T จะชัดเจนยิ่งขึ้น

4.4 การจำลองโดยการปรับเปลี่ยนค่าเรชโซลต์

ในหัวข้อนี้จะจำลองสภาพแวดล้อมโดยการปรับเปลี่ยนค่าเรชโซลต์ที่ใช้ในการจำลอง เพื่อพิสูจน์ว่าการคำนวณค่าเรชโซลต์ตามสมการที่ (3.3) (3.4) (3.5) (3.6) และ (3.7) เป็นค่าเรชโซลต์ที่เหมาะสมที่สุด

4.4.1 การจำลองโดยการปรับเปลี่ยนเรชโซลต์ของแบตเตอรี่

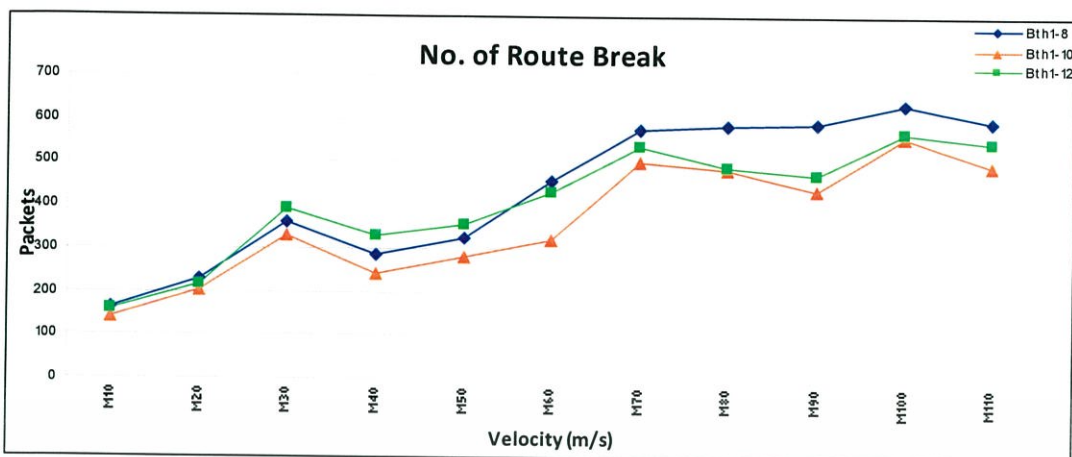
การจำลองนี้ใช้พารามิเตอร์การจำลองในสภาพแวดล้อมที่ 1 แล้วทำการเปลี่ยนค่าของเรชโซลต์ระดับที่หนึ่งของแบตเตอรี่ออกเป็น 3 ระดับคือ 8 จูล 10 จูล และ 12 จูล ตามลำดับ ซึ่งค่าเรชโซลต์ที่ได้จากการคำนวณคือ 10 จูล และกำหนดให้เรชโซลต์ของระดับสัญญาณมีค่าคงที่ไม่มีการปรับเปลี่ยน ซึ่งพารามิเตอร์ที่ใช้ในการจำลองจะเป็นดังตารางที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

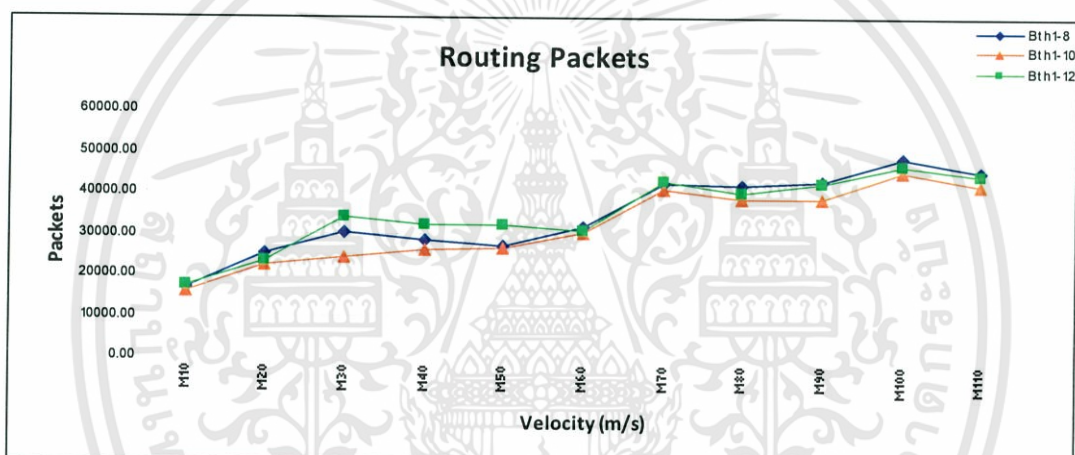
ตารางที่ 4.4 ข้อกำหนดในการจำลองการทำงานในสภาพแวดล้อมที่ 4

จำนวน โหนด	100
พื้นที่จำลองการทำงาน	1000 เมตร x 1000 เมตร
ความเร็วในการเคลื่อนที่ของโหนด	110 เมตรต่อวินาที
จำนวนกลุ่มสนทนา (ไม่ซ้ำกัน)	20 คู่
ชนิดของการรับส่งข้อมูล	ส่งข้อมูลด้วยอัตราคงที่ (CBR)
ขนาดแพ็คเกจ	512 ไบต์
เวลาที่ใช้ในการจำลอง	150 วินาที
เวลาหยุดอยู่กับที่	10 วินาที
ระยะการสื่อสาร	100 เมตร
ค่าความถี่	2.45 GHz
อัตราข้อมูล (Data Rate)	11 Mbps
Pt (Transmit Power)	0.02998531 วัตต์ (14.771dBm)
Gt (Antenna Gain ฟังส่ง)	2.4 dBi
Gr (Antenna Gain ฟังรับ)	2.4 dBi
ht (Height Antenna ฟังส่ง)	1.0 เมตร
hr (Height Antenna ฟังรับ)	1.0 เมตร
เรขาคณิตของสัญญาณระดับที่ 1	9.07×10^{-10} วัตต์
เรขาคณิตของสัญญาณระดับที่ 2	7.31×10^{-10} วัตต์
พลังงานเริ่มต้นของแบตเตอรี่	150 จูล
เรขาคณิตของพลังงานระดับที่ 1	Bth1 = 8 จูล Bth1 = 10 จูล Bth1 = 12 จูล
เรขาคณิตของพลังงานระดับที่ 2	5 จูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 จำนวนเส้นทางที่ถูกตัดขาดในการทดลองปรับเปลี่ยนค่าเรซโซลด์ของแบตเตอรี่



รูปที่ 4.17 จำนวนแพ็คเกจความคลุมในการจำลองโดยการปรับเปลี่ยนค่าเรซโซลด์ของแบตเตอรี่

จากการจำลองจะเห็นได้ว่าหากมีการกำหนดระดับค่าเรซโซลด์ที่ 1 ที่ต่ำหรือสูงเกินไป จะทำให้ไม่สามารถลดจำนวนแพ็คเกจที่ถูกทิ้งลงได้ ดังแสดงในรูปที่ 4.16 ทั้งนี้เพราะโหนดยังไม่สามารถค้นหาเส้นทางสำรองได้ทันเวลาทำให้ไม่สามารถใช้งานเส้นทางสำรองได้ และเมื่อพลังงานแบตเตอรี่ต่ำกว่าค่าเรซโซลด์ระดับที่สองโหนดในเส้นทางสื่อสารจะหันไปใช้การซ่อมแซมเส้นทางแบบโลคอลรีแพร์แทน ซึ่งถ้าทำไม่สำเร็จโหนดในเส้นทางจะต้องทิ้งแพ็คเกจไป

4.4.2 การจำลองโดยการปรับเปลี่ยนเรซโซลด์ของระดับสัญญาณที่ได้รับ

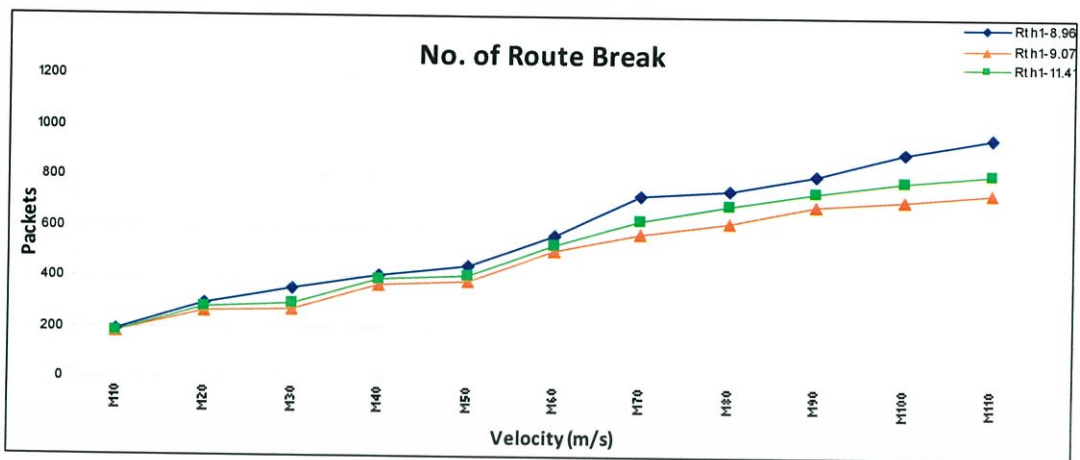
ในการจำลองมีการปรับเปลี่ยนค่าพารามิเตอร์ในการจำลองในสภาพแวดล้อมที่ 1 โดยเปลี่ยนค่าของเรซโซลด์ที่หนึ่งของระดับสัญญาณที่ได้รับ โดยค่าเรซโซลด์ของแบตเตอรี่จะมีค่าคงที่ไม่เปลี่ยนแปลง ซึ่งพารามิเตอร์ที่ใช้ในการจำลองจะเป็นดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

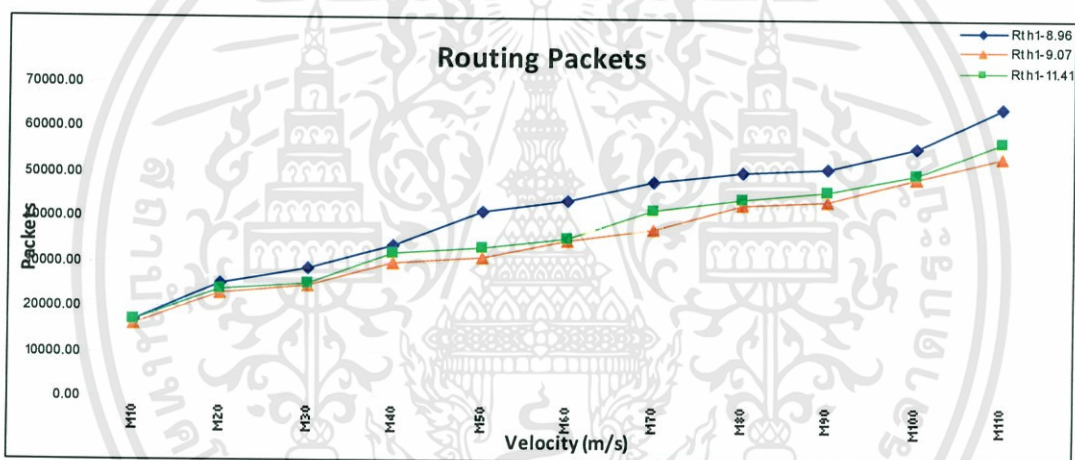
ตารางที่ 4.5 ข้อกำหนดในการจำลองการทำงานในสภาพแวดล้อมที่ 5

จำนวน โหนด	100
พื้นที่จำลองการทำงาน	1000 เมตร x 1000 เมตร
ความเร็วในการเคลื่อนที่ของโหนด	110 เมตรต่อวินาที
จำนวนคู่สนทนา (ไม่ซ้ำกัน)	20 คู่
ชนิดของการรับส่งข้อมูล	ส่งข้อมูลด้วยอัตราคงที่ (CBR)
ขนาดแพ็คเกจ	512 ไบต์
เวลาที่ใช้ในการจำลอง	150 วินาที
เวลาหยุดอยู่กับที่	10 วินาที
ระยะการสื่อสาร	100 เมตร
ค่าความถี่	2.45 GHz
อัตราข้อมูล (Data Rate)	11 Mbps
Pt (Transmit Power)	0.02998531 w (14.771dBm)
Gt (Antenna Gain ฟังส่ง)	2.4 dBi
Gr (Antenna Gain ฟังรับ)	2.4 dBi
ht (Height Antenna ฟังส่ง)	1.0 เมตร
hr (Height Antenna ฟังรับ)	1.0 เมตร
เรชโซลต์ของสัญญาณระดับที่ 1	Rth1=8.96x10 ⁻¹⁰ วัตต์ Rth2=9.07x10 ⁻¹⁰ วัตต์ Rth3=11.41x10 ⁻¹⁰ วัตต์
เรชโซลต์ของสัญญาณระดับที่ 2	7.31x10 ⁻¹⁰ วัตต์
พลังงานเริ่มต้นของแบตเตอรี่	150 จูล
เรชโซลต์ของพลังงานระดับที่ 1	10 จูล
เรชโซลต์ของพลังงานระดับที่ 2	5 จูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 จำนวนเส้นทางที่ถูกตัดขาดในการจำลอง โดยการปรับเปลี่ยนค่าเรซโซลต์ของระดับสัญญาณ



รูปที่ 4.19 จำนวนแพ็คเกจควบคุมในการจำลอง โดยการปรับเปลี่ยนค่าเรซโซลต์ของระดับสัญญาณ

จากการจำลองสภาพแวดล้อมที่ 5 เมื่อทำการปรับเปลี่ยนค่าเรซโซลต์ของระดับสัญญาณที่ 1 พบว่าค่าเรซโซลต์ของระดับสัญญาณที่ 1 ที่เหมาะสมที่สุดคือ 9.07×10^{-10} w ซึ่งเป็นค่าที่ได้จากการคำนวณจากสมการที่ 3.3 ทั้งนี้เพราะสามารถลดจำนวนเส้นทางที่ถูกตัดขาดได้มากกว่าค่าเรซโซลต์ระดับที่ 1 ค่าอื่น ๆ ดังรูปที่ 4.18 โดยสร้างแพ็คเกจควบคุมการสื่อสารในปริมาณที่ใกล้เคียงกับระดับเรซโซลต์ระดับสัญญาณที่ 1 ค่าอื่นที่กำหนดไว้ ดังแสดงในกราฟรูปที่ 4.19

บทที่ 5

สรุปผลการวิจัย

5.1 สรุปผลวิจัย

งานวิจัยนี้ได้นำเสนอถึงแนวทางในการปรับปรุงการทำงานของโปรโตคอล AODV-2T ในการสร้างเส้นทางสำรอง เพื่อแก้ไขปัญหาการซอมแซมเส้นทางเชื่อมต่อของโปรโตคอล AODV ด้วยวิธีการปรับปรุงรองรับการสร้างเส้นทางย่อยสำรองแบบหลาย ฮอป โดยนำเสนอวิธีการดังกล่าวในชื่อโปรโตคอล Multi-hop AODV-2T

การจำลองของโปรโตคอล Multi-hop AODV-2T ได้แสดงให้เห็นแล้วว่าสามารถช่วยเพิ่มประสิทธิภาพในการรับส่งข้อมูล ช่วยลดจำนวนแพ็คเก็ตที่ถูกทิ้งลงโดยเฉลี่ยประมาณ 15 - 20 เปอร์เซ็นต์เมื่อเทียบกับโปรโตคอล AODV และ 10 - 12 เปอร์เซ็นต์เมื่อเทียบกับโปรโตคอล AODV-2T ซึ่งผลที่ได้นำมาจากค่าเฉลี่ยของจำนวนแพ็คเก็ตที่ถูกทิ้งจากการจำลองของทั้งสามโปรโตคอลแล้วนำมาเปรียบเทียบกัน นอกจากนั้นโปรโตคอล Multi-hop AODV-2T ยังช่วยลดจำนวนของเส้นทางที่ถูกตัดขาดการเชื่อมต่อลงได้อีกด้วย โดยที่การทำงานของโปรโตคอลไม่ได้เพิ่มจำนวนแพ็คเก็ตควบคุมการหาเส้นทางเข้าไปในระบบมากนัก ซึ่งก็เป็นไปตามสมมติฐานตามที่คาดการณ์ไว้ตั้งแต่ตอนเริ่มต้นการออกแบบโปรโตคอล

จากผลการเปรียบเทียบอัตราการส่งข้อมูลสำเร็จระหว่างโปรโตคอล AODV โปรโตคอล AODV-2T และโปรโตคอล Multi-hop AODV-2T แสดงให้เห็นว่าโปรโตคอล Multi-hop AODV-2T มีอัตราการส่งข้อมูลได้สำเร็จมากกว่าโปรโตคอลอีกสองชนิด ถึงแม้ว่าในช่วงแรกที่มีการเคลื่อนที่ของโหนดด้วยความเร็วต่ำการทำงานของโปรโตคอลทั้งสามจะมีผลของการส่งข้อมูลสำเร็จที่ใกล้เคียงกัน ทั้งนี้เพราะว่าในช่วงที่โหนดมีการเคลื่อนที่ด้วยความเร็วต่ำเส้นทางการรับส่งข้อมูลจะไม่เสี่ยงต่อการถูกตัดขาดการเชื่อมต่อ แต่ในช่วงที่มีการเคลื่อนที่ของโหนดด้วยความเร็วที่สูงขึ้นโปรโตคอล Multi-hop AODV-2T กลับแสดงผลการจำลองที่ดีกว่า ทั้งนี้เพราะเมื่อโหนดมีการเคลื่อนที่ด้วยความเร็วที่มากขึ้นยอมทำให้เส้นทางเชื่อมต่อจะถูกตัดขาดได้ง่ายขึ้น ดังนั้นเนื่องจากโปรโตคอล Multi-hop AODV-2T มีการค้นหาและสร้างเส้นทางสำรองไว้ก่อนล่วงหน้า ทำให้การสื่อสารยังดำเนินการต่อไปได้อย่างต่อเนื่อง และทำให้สามารถส่งข้อมูลไปถึงโหนดปลายทางได้สำเร็จมากกว่าโปรโตคอล AODV และโปรโตคอล AODV-2T

และจากผลการเปรียบเทียบจำนวนแพ็คเก็ตที่ถูกทิ้งที่ตรวจวัดได้จากการจำลองสภาพแวดล้อมที่ 1 ระหว่างโปรโตคอล AODV โปรโตคอล AODV-2T และ โปรโตคอล Multi-hop AODV-2T ที่ได้ นั้น พบว่าโปรโตคอล Multi-hop AODV-2T สามารถลดจำนวนของแพ็คเก็ตที่ถูกทิ้งลงได้ถึง 20-30 เปอร์เซ็นต์เมื่อเทียบกับโปรโตคอล AODV-2T และโปรโตคอล AODV

บรรณานุกรม

- [1] Akkaphop, O., Mayuree, L., “An Improvement of Ad Hoc Network Route Maintenance”, Proceeding of International Symposium on Communication and Information Technologies, pp. 102-105, October 2008.
- [2] C. E. Perkins and E. M. Royer, “Ad Hoc On-Demand Distance Vector Routing,” IEEE Workshop on Mobile Computing system and Applications, pp. 90-100, February 1999.
- [3] **The Network Simulator ns-2**, available online at <http://www.isi.edu/nsnam/ns>
- [4] C. E. Perkins and P. Bhangwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computer,” ACM SIGCOMM, pp. 234-244, August 1994.
- [5] S. Murthy and J. J. Garcia-Luna-Aceves, “An Efficient Routing Protocol for Wireless Network,” ACG Mobile Network and Application Journal, Vol. 1, pp. 183-197, October 1996.
- [6] C. C. Chiang, H. K. Wu, W. Liu and M. Gerla, “Routing in Clustered Multi-Hop Mobile Wireless Network with Fading Channel,” IEEE SICON, pp. 197-211, April 1997.
- [7] J. J. Garcia-Luna-Aceves and M. Spohn, “Source Tree Routing in Wireless Network,” IEEE ICNP, pp. 273-282, October 1999.
- [8] T. H. Clausen, G. Hansen, L. Christensen, and G. Behrmann, “The Optimized Link State Routing Protocol, Evaluation Through Experiments and Simulation,” IEEE Symposium on Wireless Personal Mobile Communication, September 2001.
- [9] A. Iwata, C. C. Chiang, G. Pei, M. Gerla, and T. W. Chen, “Scalable Routing Strategies for Ad Hoc Wireless Network,” IEEE Journal on Selected Areas in Communication, Vol. 17, pp. 1369-1379, August 1999.
- [10] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Network,” Mobile Computing Kluwer Academic Publisher, Vol. 353, pp. 153-181, November 1996.
- [11] R. Dube, C. D. Rais, K. Y. Wang, and S. K. Tripathi, “Signal Stability-Based Adaptive Routing for Ad Hoc Mobile Network,” IEEE Personal Communication Magazine, pp. 36-45, February 1997.

- [12] W. Su and M. Gerla, “**IPv6 Flow Handoff in Ad Hoc Wireless Networks Using Mobility Prediction,**” IEEE GLOBECOM, pp. 271-275, December 1999.
- [13] R. S. Sisodia, B. S. Manoj, and C. Siva Ram Murthy, “**A Preferred Link-Based Routing Protocol for Ad Hoc Wireless Network,**” Journal of Communications and Networks, Vol. 4 no, 1, pp. 14-21, March 2002.
- [14] P. Sinha, R. Sivakumar, and V. Bharghavan, “**CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm,**” IEEE Journal on selected Areas in Communication, Vol 17, no. 8, pp. 1454-1466, August 1999.
- [15] Z. J. Haas, “**The Routing Algorithm for the Reconfigurable Wireless Networks,**” ICUPC, Vol. 2, pp. 562-566, October 1997.
- [16] M. Joe-Ng and I. T. Lu, “**A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks,**” IEEE Journal on selected Areas in Communication, Vol 17, no. 8, pp. 1415-1425, August 1999.
- [17] B. S. Manoj, R. Ananthapadmanabha, and C. Shiva Ram Murthy, “**Link Life Based Routing Protocol for Ad Hoc Wireless Networks,**” IEEE ICCCN, pp. 573-576, October 2001.
- [18] S. Singh, M. Woo, and C. S. Raghavendra, “**Power-Aware Routing in Mobile Ad Hoc Network,**” ACM MOBICOM, pp. 181-190, October 1998.
- [19] Y. Ko and N. H. Vaidya, “**Location-Aided Routing (LAR) in Mobile Ad Hoc Network,**” ACM MOBICOM, pp 66-75, October 1998.
- [20] M. Tauchi, T. Ideguchi and Y. Muruta, “**Ad Hoc Routing Protocol Avoiding Route Break Based on AODV,**” IEEE Personal Communication, pp.322-329, January 2005.
- [21] MANET, “**Mobile Ad Hoc Network,**” available online at <http://www.ietf.org/html/charters/manet-charter.html>
- [22] Theodore S. Rappaport, “**Wireless Communications Principle and Practice,**” Prentice Hall, December 2001.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

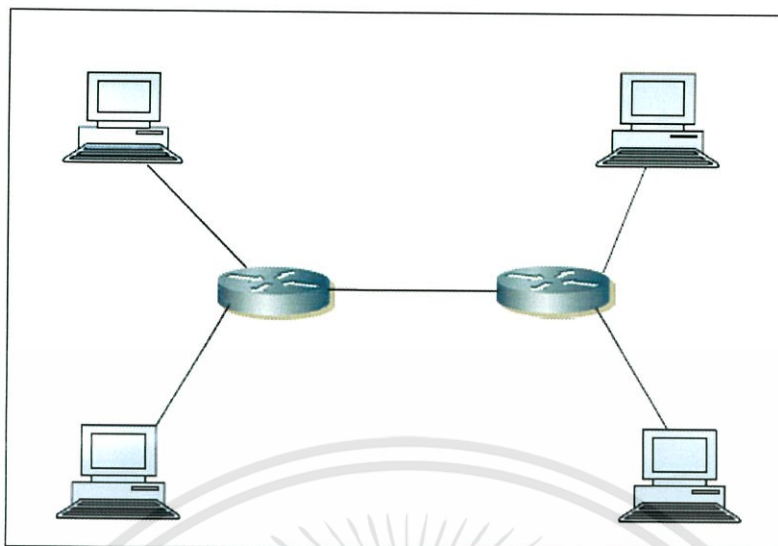
การจำลองการทำงานของเครือข่ายโดยใช้โปรแกรม NS-2

ในส่วนนี้จะกล่าวถึงการจำลองการทำงานของเครือข่ายโดยใช้โปรแกรมจำลองการทำงานชื่อ NS-2 (Network Simulator Version 2) ซึ่งได้ถูกนำมาใช้ในการจำลองระบบเครือข่ายของงานวิจัยนี้ โดยจะแสดงถึงรายละเอียดของทฤษฎีที่เกี่ยวข้อง หลักการทำงานของโปรแกรม และตัวอย่างการใช้งานโปรแกรมจำลองการทำงาน

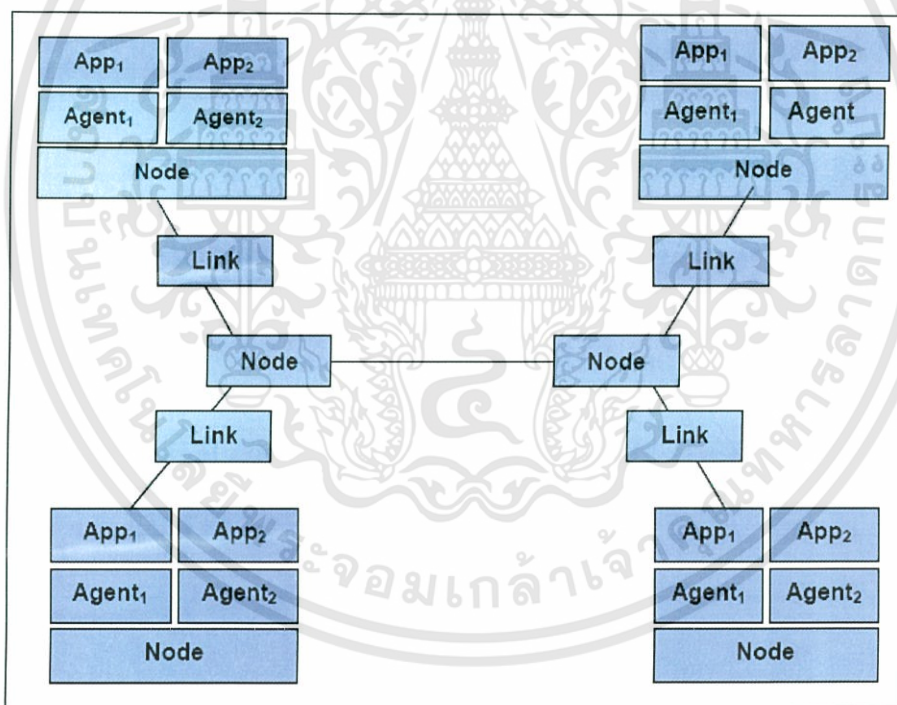
- โปรแกรมจำลองการทำงานของเครือข่าย NS-2

ตั้งแต่อดีตจนถึงปัจจุบันได้มีการพัฒนาโครงสร้างของระบบเครือข่ายอยู่ตลอดเวลา โดยการพัฒนาาระบบเครือข่ายนั้น ๆ จะต้องมีการวางโครงสร้างระบบผ่านอุปกรณ์ต่าง ๆ ที่ใช้ในการเชื่อมต่อเครือข่ายมากมาย โดยก่อนที่จะติดตั้งระบบเครือข่ายสำหรับการใช้งานจริงนั้น จะต้องมีการทดสอบการทำงานของระบบเครือข่ายก่อนจึงเป็นการยากที่จะทดสอบการทำงานของระบบเครือข่ายโดยการทดสอบผ่านอุปกรณ์ที่ใช้งานจริงต่าง ๆ ตามที่ได้ทำการวางโครงสร้างของระบบไว้ เนื่องจากการวางโครงสร้างของระบบเครือข่ายนั้นเป็นสิ่งที่ต้องใช้งบประมาณและเวลามากจึงไม่คุ้มที่จะทดสอบการทำงานของระบบเครือข่ายโดยการทดสอบผ่านอุปกรณ์ที่ใช้งานจริง ดังนั้นจึงมีความจำเป็นที่จะต้องมีการจำลองการทำงานของโครงสร้างระบบเครือข่ายขึ้นก่อน เพื่อที่จะได้ทำการศึกษาการทำงานของระบบเครือข่ายในด้านต่าง ๆ ทั้งพฤติกรรมของการขนส่ง Packet, การศึกษาการทำงานของ Traffic บนระบบเครือข่าย, การศึกษาการทำงานของ Algorithm ที่ใช้ในการส่งผ่าน Packet เป็นต้น ซึ่งในปัจจุบันได้มีโปรแกรมที่ช่วยในการจำลองการทำงานของระบบเครือข่ายมากมาย มีทั้งที่เป็น Freeware และ Commercial เช่น NS-2, OMNet++, OPNet, CSim, Swans และ NCTUns เป็นต้น ดังนั้นเมื่อมีโปรแกรมที่คอยช่วยเหลือในการจำลองการทำงานของระบบเครือข่ายจึงเป็นการดีต่อการศึกษาและงานวิจัยทางด้านระบบเครือข่าย เนื่องจากการโปรแกรมเหล่านี้สามารถทำให้ผู้ใช้งานได้มองเห็นรูปแบบการทำงานจริงของระบบเครือข่ายผ่านการจำลองโดยใช้โปรแกรม โดยไม่ต้องคำนึงถึงค่าใช้จ่ายที่จะต้องสูญเสียไปจากการทดสอบการทำงานของระบบเครือข่าย จึงสามารถที่จะพัฒนาความรู้ในด้านนี้ได้อย่างรวดเร็ว

โปรแกรมการจำลองการทำงานของระบบเครือข่ายที่มีชื่อว่า NS-2 (Network Simulator Version 2) ซึ่งเป็น Discrete Event Simulator โดย NS-2 ได้สนับสนุนการทำแบบจำลองการทำงานของ TCP, routing, multicast protocols over wired และ wireless (local และ satellite) networks จากคุณสมบัติเหล่านี้เราจึงสามารถจำลองการทำงานของ Network ได้เป็นอย่างดี



รูปที่ ก.1.1 แสดงภาพการวางโครงสร้างต้นแบบระบบเน็ตเวิร์ก



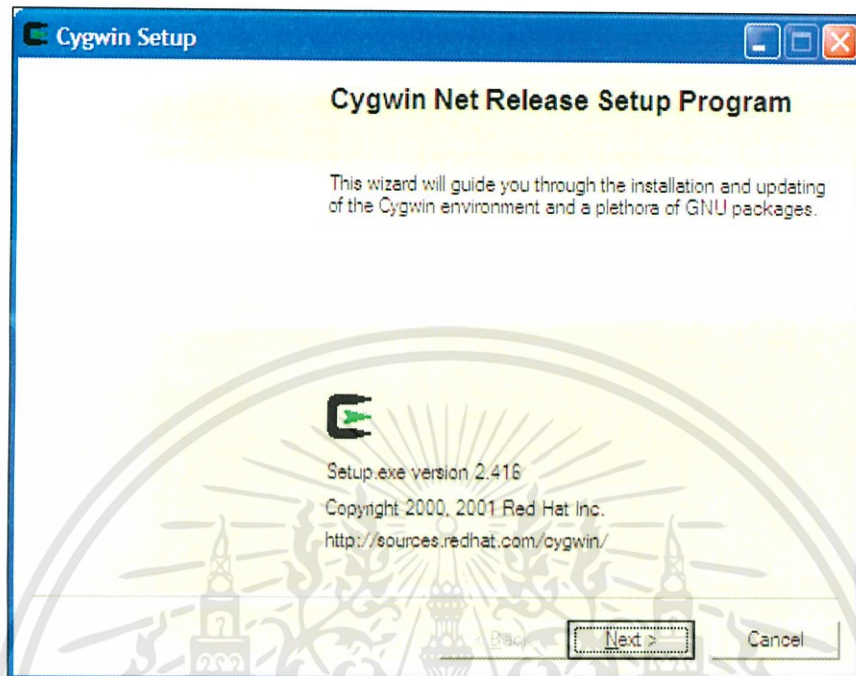
รูปที่ ก.1.2 แสดงภาพโครงสร้างของการจำลองระบบเครือข่าย

จากรูปที่ ก.1.1 นั้นเป็นการวางโครงสร้างต้นแบบของระบบเน็ตเวิร์กในปัจจุบัน ซึ่งเมื่อนำรูปแบบโครงสร้างของระบบเน็ตเวิร์กที่ใช้กันอยู่ในปัจจุบันมาทำการจำลอง เพื่อที่จะใช้ในการทดสอบการทำงานในด้านต่าง ๆ จึงต้องใช้โปรแกรมการจำลองการทำงานของระบบเครือข่าย

NS-2 มาช่วยในการออกแบบการจำลองเน็ตเวิร์ก ดังนั้นรูปที่ ก.1.2 จึงเป็นโครงสร้างของการจำลองระบบเครือข่ายที่เทียบมาจากความเป็นจริง

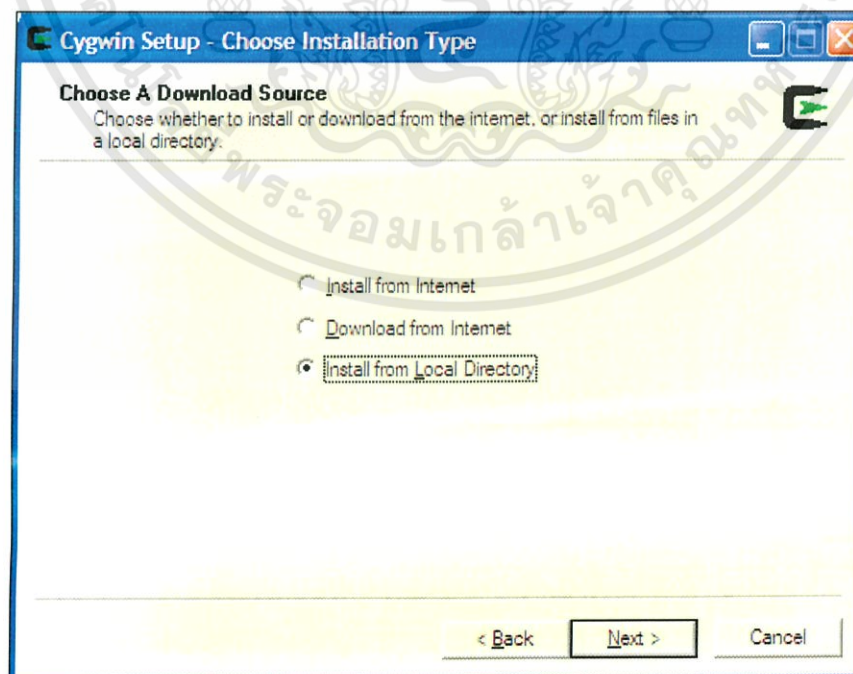
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ให้ดับเบิลคลิกที่ไฟล์ชื่อ `setup.exe` เพื่อเริ่มต้นการติดตั้งโปรแกรม Cygwin จะปรากฏหน้าต่างเริ่มต้นการติดตั้งโปรแกรมขึ้นมา ให้คลิกที่ปุ่ม



รูปที่ ก.1.4 การติดตั้งโปรแกรม Cygwin

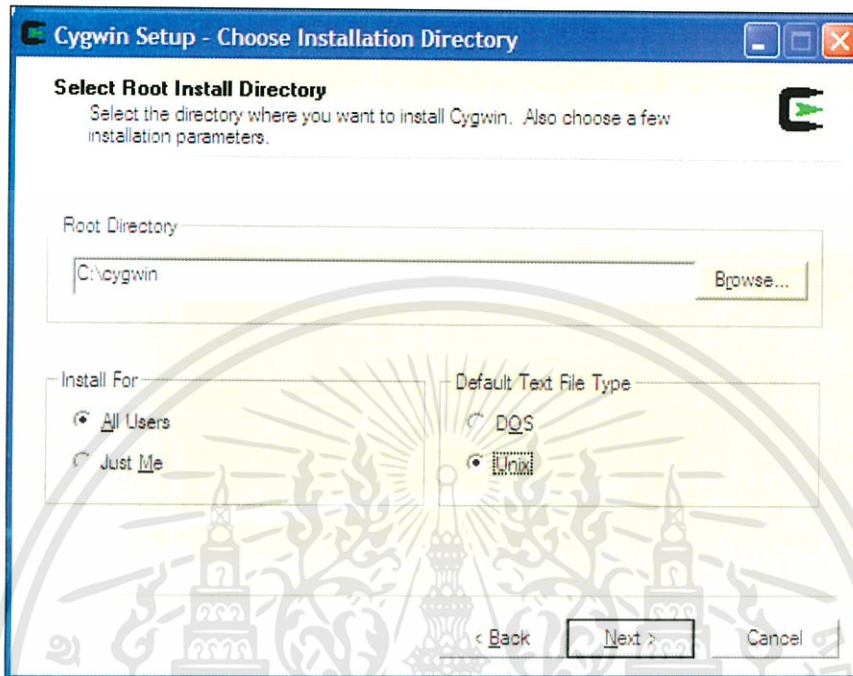
3. เลือกรูปแบบการติดตั้งแบบ **Install From Local Directory** แล้วคลิกที่ปุ่ม



รูปที่ ก.1.5 แสดงรูปแบบต่าง ๆ ให้เลือกติดตั้งโปรแกรม Cygwin

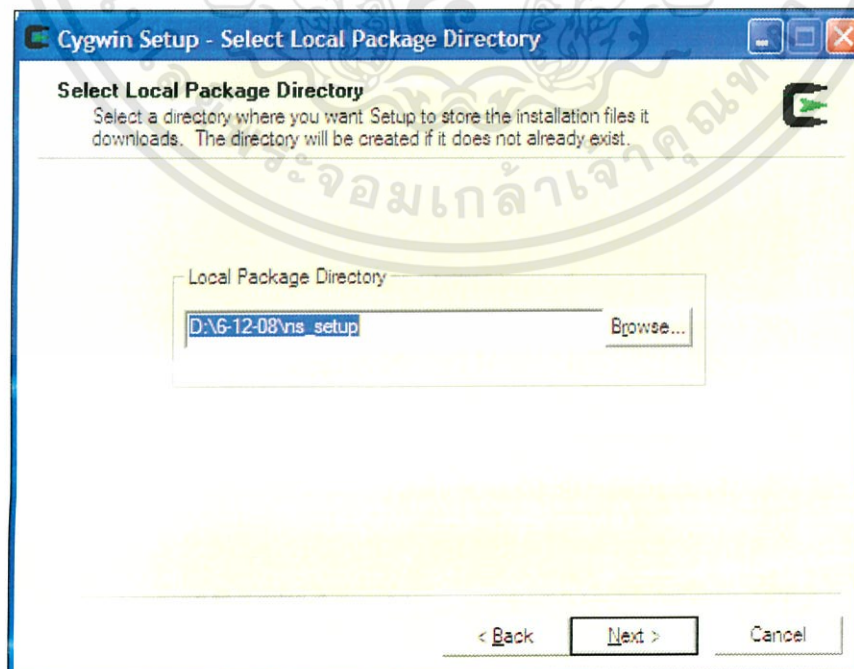
เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สงวนเพื่อการศึกษาเท่านั้น มิใช่สัญญาหรือการรับประกันใดๆ ในประการใดๆ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. จะปรากฏหน้าต่างให้เลือก Root Directory ที่ต้องการติดตั้ง เลือกผู้ที่สามารถใช้งานโปรแกรมได้ และเลือกชนิดของ Text File ที่ต้องการใช้งาน ให้เลือกตามรูปที่ ก.1.6 แล้วคลิกที่ปุ่ม

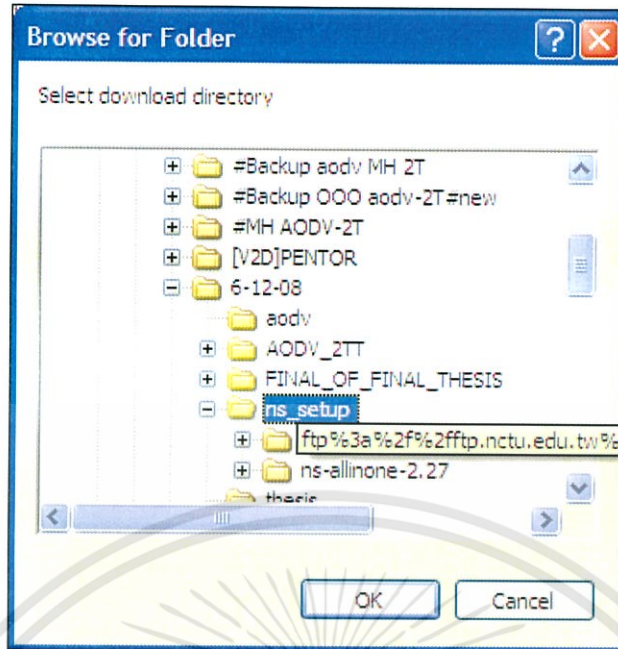


รูปที่ ก.1.6 แสดงการเลือกรายละเอียดติดตั้งโปรแกรม

5. คลิกที่ เพื่อเลือกไดเรกทอรีที่เก็บไฟล์ที่ใช้ในการติดตั้งไว้

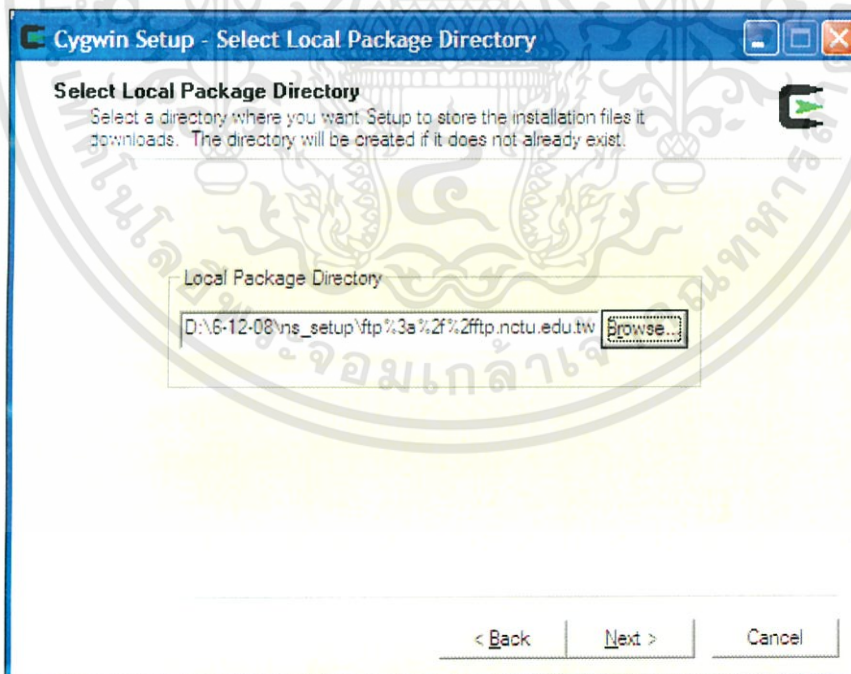


เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การนำข้อมูลไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1.8 แสดงไดเรกทอรีที่ใช้ในการติดตั้ง

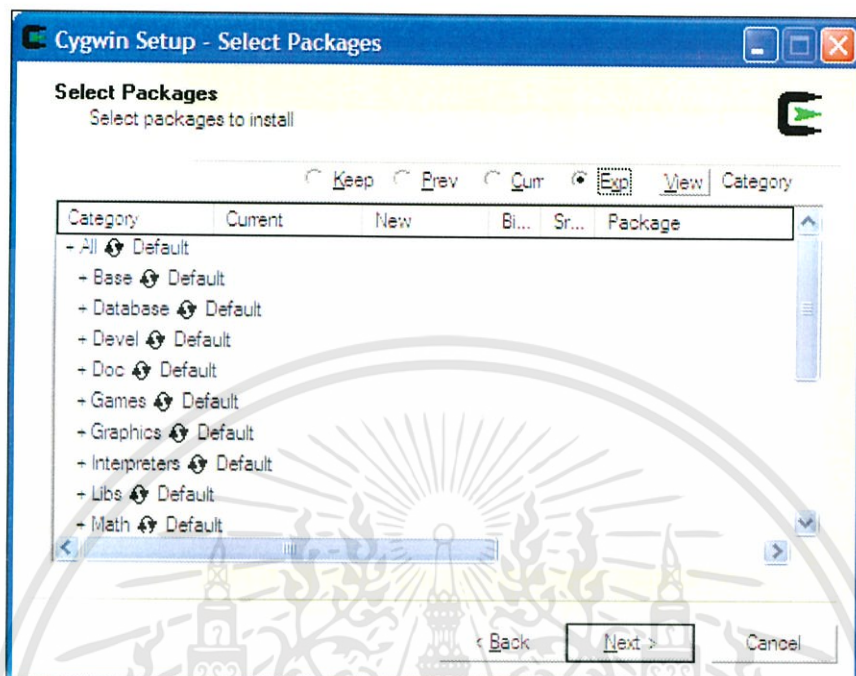
6. จากรูปไฟล์ที่ใช้ในการติดตั้งโปรแกรมจะเก็บไว้ที่ **D:\6-12-08\ns_setup** แล้วคลิกที่ปุ่ม



รูปที่ ก.1.9 แสดงไดเรกทอรีที่ได้เลือกใช้ในการติดตั้ง

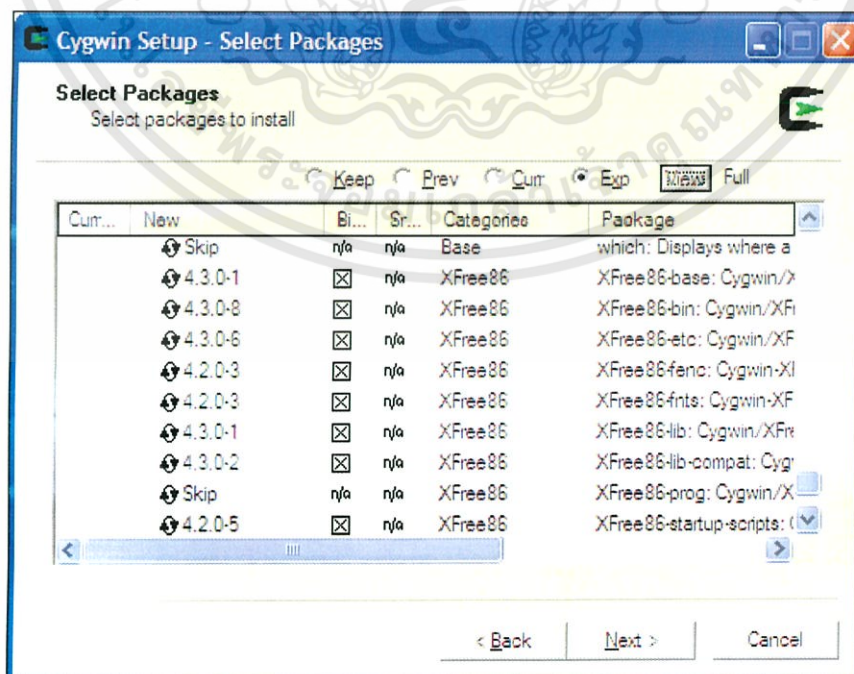
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. จะปรากฏหน้าจอให้เลือกแพ็คเกจที่ต้องการใช้งาน ให้คลิกเลือกที่ View เพื่อให้โปรแกรมแสดงรายละเอียดแพ็คเกจทั้งหมด



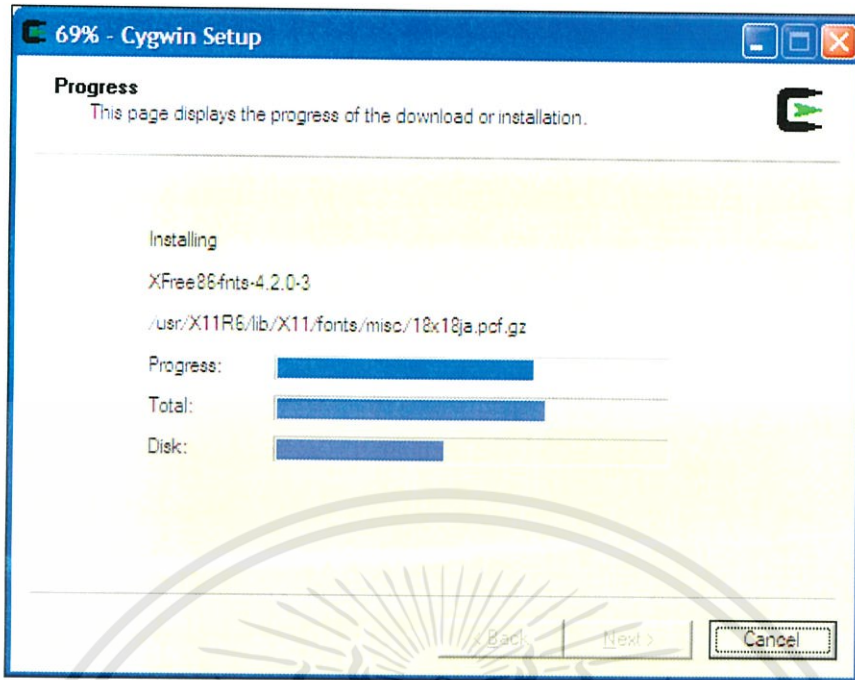
รูปที่ ก.1.10 แสดงรายชื่อของกลุ่มแพ็คเกจ

8. คลิกที่รายชื่อแพ็คเกจในหมวดของ XFree86 ให้เปลี่ยนค่าจาก Skip เป็น 4.3.0-1 ตามรูปแล้วคลิกที่ปุ่ม Next > จะปรากฏหน้าต่างแสดงรายละเอียดการติดตั้งแต่ละแพ็คเกจ



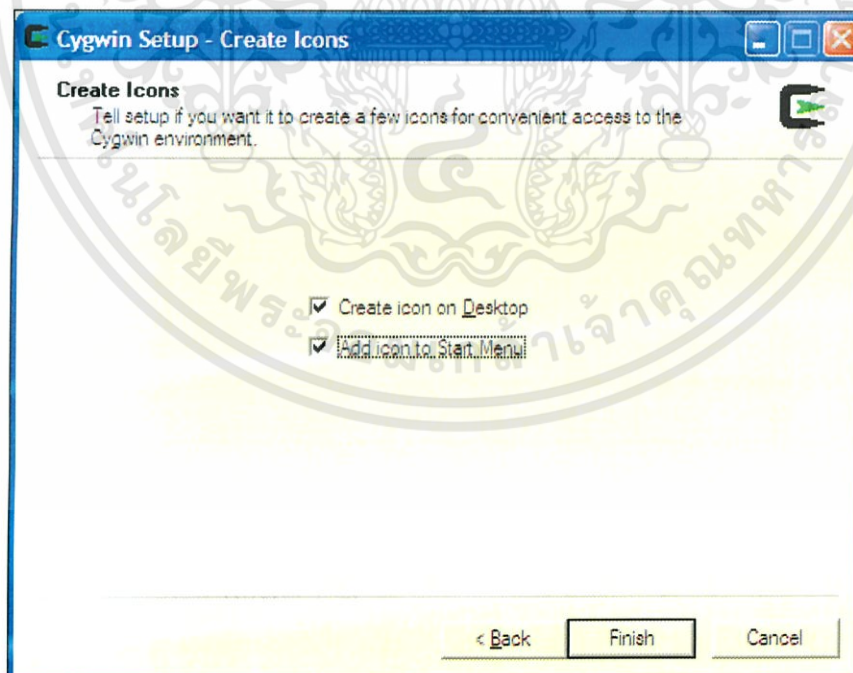
รูปที่ ก.1.11 แสดงรายชื่อของแพ็คเกจต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1.12 แสดงรายละเอียดของแพ็คเกจที่กำลังติดตั้ง

9. ในขั้นตอนสุดท้าย ให้คลิกเลือกเงื่อนไขในการสร้าง icon ตามต้องการ แล้วคลิกที่ปุ่ม 



รูปที่ ก.1.13 แสดงรายละเอียดในการสร้าง icon ที่ต้องการ

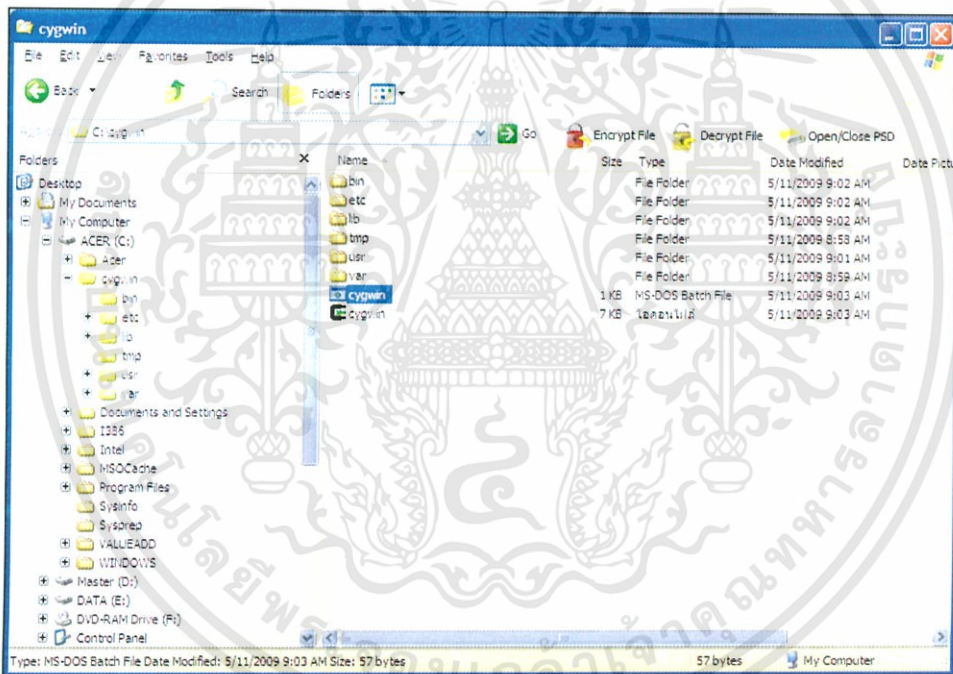
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. เมื่อเสร็จสิ้นการติดตั้งโปรแกรมจะปรากฏหน้าต่างดังรูป ให้คลิกที่ปุ่ม



รูปที่ ก.1.14 แสดงหน้าต่างเสร็จสิ้นการติดตั้งโปรแกรม

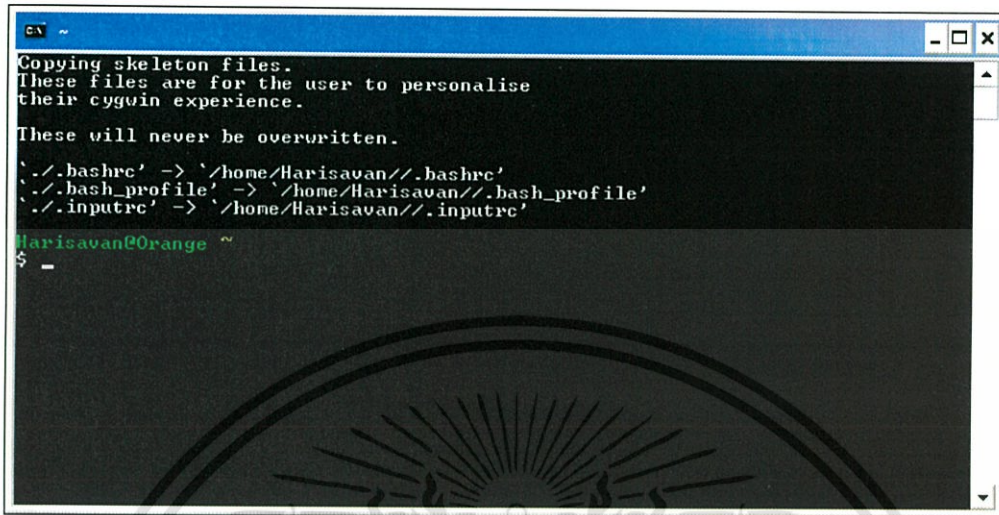
11. เปิดโปรแกรม Windows Explorer แล้วเลือกเข้าไปยังไดเรกทอรีที่ติดตั้งโปรแกรม Cygwin แล้วดับเบิลคลิกที่ไฟล์ชื่อ cygwin.exe เพื่อเรียกใช้งานโปรแกรม



รูปที่ ก.1.15 แสดงไดเรกทอรีที่ติดตั้งโปรแกรม Cygwin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. จะปรากฏหน้าต่างของโปรแกรม Cygwin ขึ้นมา โปรแกรมจะทำการสร้าง profile และ โดเร็กทอรีที่ใช้เก็บข้อมูลให้กับผู้ใช้งาน ดังรูป (สังเกตที่สัญลักษณ์ ~)



```

C:\ ~
Copying skeleton files.
These files are for the user to personalise
their cygwin experience.

These will never be overwritten.

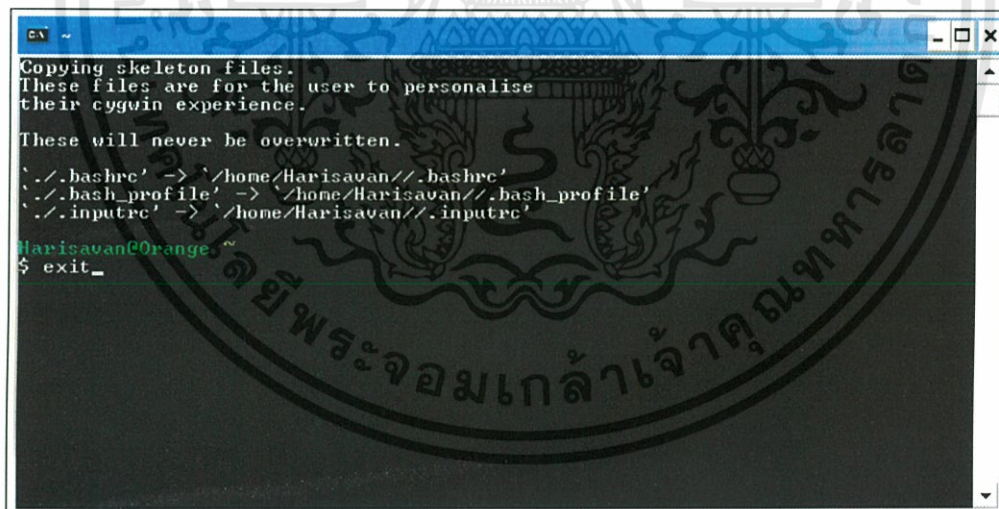
\./bashrc' -> '/home/Harisavan//.bashrc'
\./bash_profile' -> '/home/Harisavan//.bash_profile'
\./inputrc' -> '/home/Harisavan//.inputrc'

Harisavan@Orange ~
$ -

```

รูปที่ ก.1.16 แสดงหน้าต่างของโปรแกรม Cygwin

13. พิมพ์คำสั่ง exit แล้วกดปุ่ม Enter เพื่อออกจากโปรแกรม



```

C:\ ~
Copying skeleton files.
These files are for the user to personalise
their cygwin experience.

These will never be overwritten.

\./bashrc' -> '/home/Harisavan//.bashrc'
\./bash_profile' -> '/home/Harisavan//.bash_profile'
\./inputrc' -> '/home/Harisavan//.inputrc'

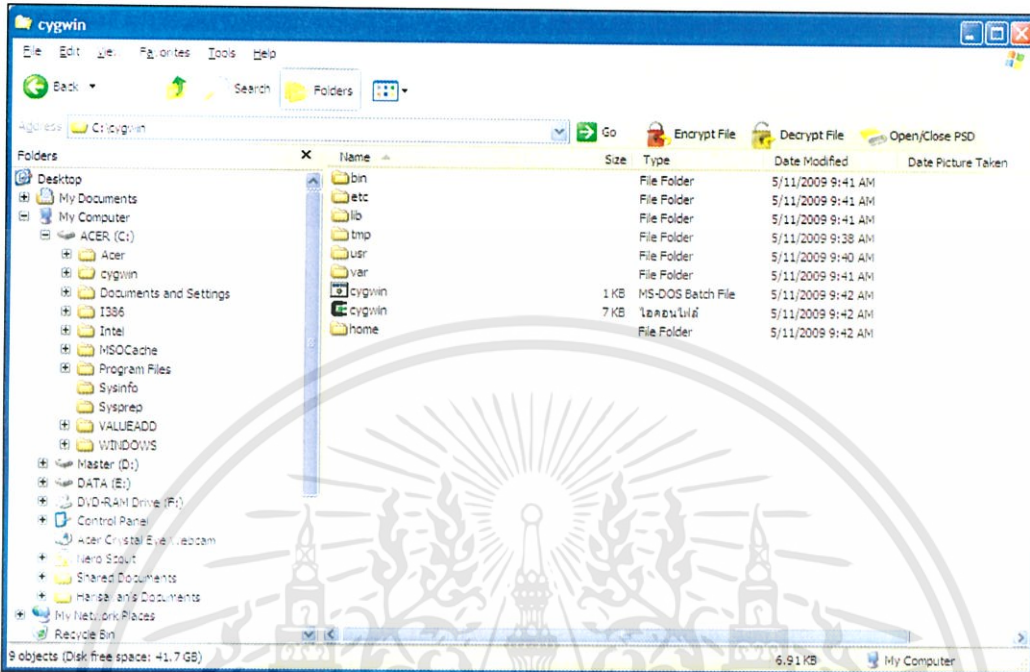
Harisavan@Orange ~
$ exit_

```

รูปที่ ก.1.17 แสดงคำสั่งเพื่อออกจากโปรแกรม

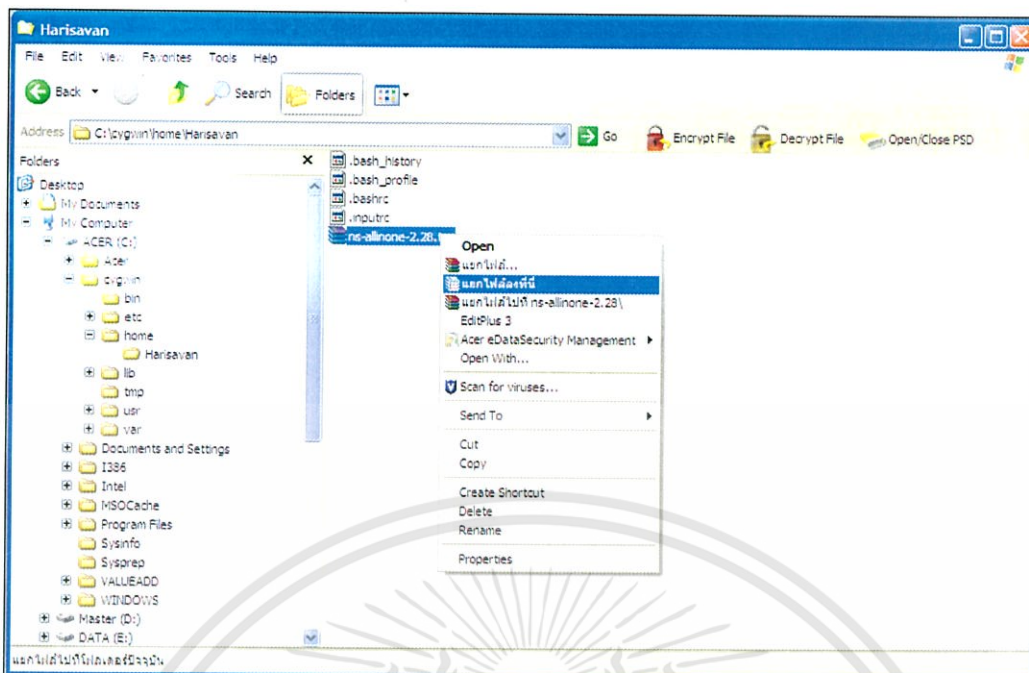
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. สังเกตที่หน้าต่างของโปรแกรม Windows Explorer ที่ไดเรกทอรีที่ติดตั้งโปรแกรม Cygwin อยู่จะปรากฏเพิ่มข้อมูลชื่อ home เพิ่มขึ้นมา

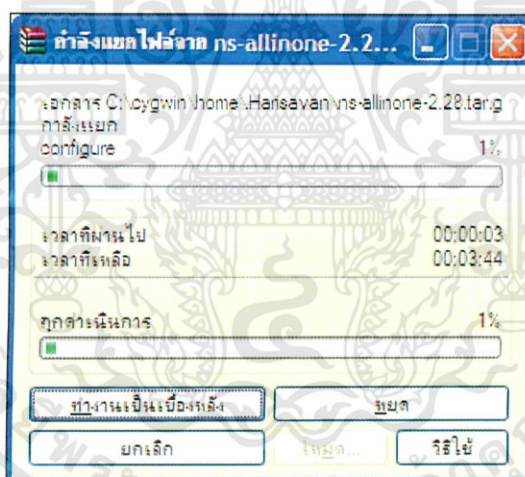


รูปที่ ก.1.18 แสดงไดเรกทอรีชื่อ home ที่เพิ่มขึ้นมา

15. ดาวน์โหลดโปรแกรม NS-2 Version 2.28 จาก <http://140.116.72.80/~smallko/ns2/ns-allinone-2.28.rar> มาจัดเก็บไว้ profile ที่อยู่ในที่ไดเรกทอรีที่ติดตั้งโปรแกรม cygwin ไว้ ซึ่งในที่นี้คือ C:\cygwin\home\Harisavan (Harisavan เป็นชื่อของ User Profile) แล้วทำการแตกไฟล์ออก จะได้เพิ่มข้อมูลชื่อ ns-allinone-2.28 ดังรูป

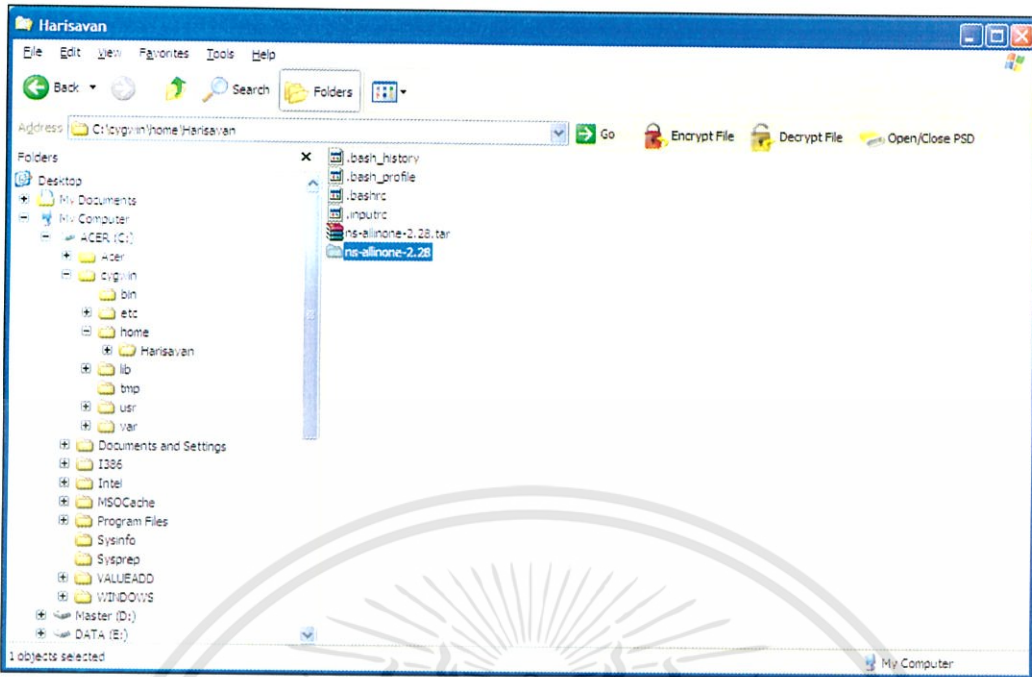


รูปที่ ก.1.19 แสดงการแตกไฟล์ ns-allinone-2.28.rar ออก



รูปที่ ก.1.20 แสดงขั้นตอนระหว่างการแตกไฟล์ ns-allinone-2.28.rar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1.21 แสดงเพิ่มข้อมูลชื่อ ns-allinone-2.28 ที่ได้จากการแตกไฟล์

16. เปิดโปรแกรม Cygwin ขึ้นมาอีกครั้ง แล้วพิมพ์คำสั่ง `cd ns-allinone-2.28/ns-2.28` แล้วกดปุ่ม Enter เพื่อเข้าไปที่ไดเรกทอรีดังกล่าว พิมพ์ `./configure` แล้วกดปุ่ม Enter เพื่อเริ่มต้นการ Config โปรแกรม

```

c:\ ~\ns-allinone-2.28/ns-2.28
Copying skeleton files.
These files are for the user to personalise
their cygwin experience.

These will never be overwritten.
`./bashrc' -> '/home/Harisavan/./bashrc'
`./bash_profile' -> '/home/Harisavan/./bash_profile'
`./inputrc' -> '/home/Harisavan/./inputrc'

Harisavan@Orange ~
$ cd ns-allinone-2.28/ns-2.28
ns-2.28
Harisavan@Orange ~
$ cd ns-allinone-2.28/ns-2.28/
Harisavan@Orange ~\ns-allinone-2.28/ns-2.28
$ ./configure
  
```

รูปที่ ก.1.22 แสดงหน้าต่างของโปรแกรม Cygwin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

c:\ ~/ns-allinone-2.28/ns-2.28
'./inputrc' -> '/home/Harisavan//.inputrc'
Harisavan@Orange ~
$ cd ns-allinone-2.28/ns
ns-2.28
Harisavan@Orange ~
$ cd ns-allinone-2.28/ns-2.28/
Harisavan@Orange ~/ns-allinone-2.28/ns-2.28
$ ./configure
No .configure file found in current directory
Continuing with default options...
checking build system type... i686-pc-cygwin
checking host system type... i686-pc-cygwin
checking target system type... i686-pc-cygwin
checking for gcc... gcc
checking for C compiler default output... a.exe
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables... .exe
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ANSI C...

```

รูปที่ ก.1.23 แสดงขั้นตอนระหว่างการ Confige

17. พิมพ์คำสั่ง make clean แล้วกดปุ่ม Enter

```

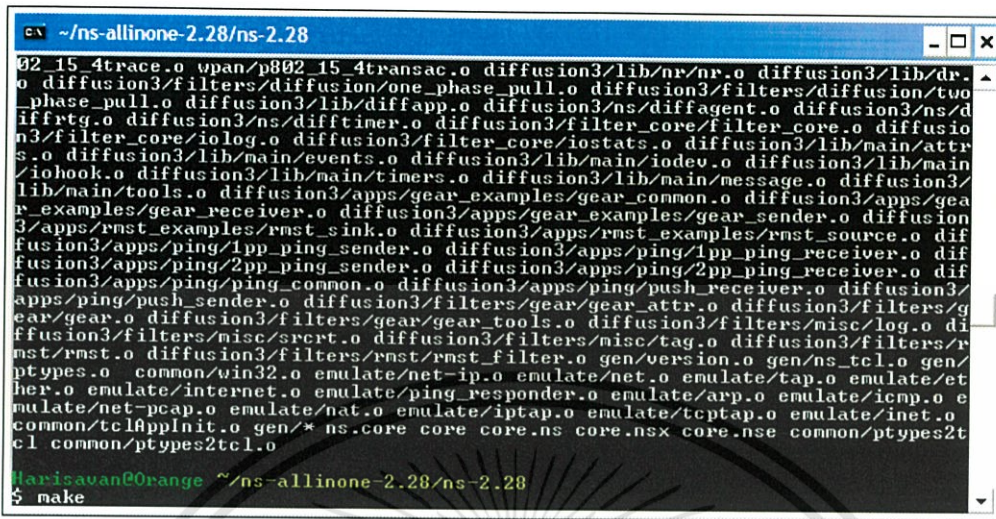
c:\ ~/ns-allinone-2.28/ns-2.28
checking for gethostbyname in -lnsl... no
checking for dcgettext in -lintl... yes
checking for getnodebyname in -ldnet_stub... no
checking that g++ can handle -O2... no
checking if C++ libraries work without any namespace... no
checking if C++ libraries work with namespace std... yes
checking if STL works without any namespace... no
checking if STL works with namespace std... yes
checking should use STL... yes
checking for tcl.h... -I../tcl8.4.5/generic
checking for libtcl8.4... no
checking for init.tcl... ../tcl8.4.5/library
checking for tclsh8.4.5... no
checking for tclsh8.4... no
checking for tclsh... no
checking for tclsh8.3... no
configure: error: Installation of tcl seems incomplete or can't be found automatically.
Please correct the problem by telling configure where tcl is
using the argument --with-tcl=/path/to/package
(perhaps after installing it),
or the package is not required, disable it with --with-tcl=no.
Harisavan@Orange ~/ns-allinone-2.28/ns-2.28
$ make clean

```

รูปที่ ก.1.24 แสดงการใช้คำสั่ง make clean

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18. พิมพ์คำสั่ง make แล้วกดปุ่ม Enter



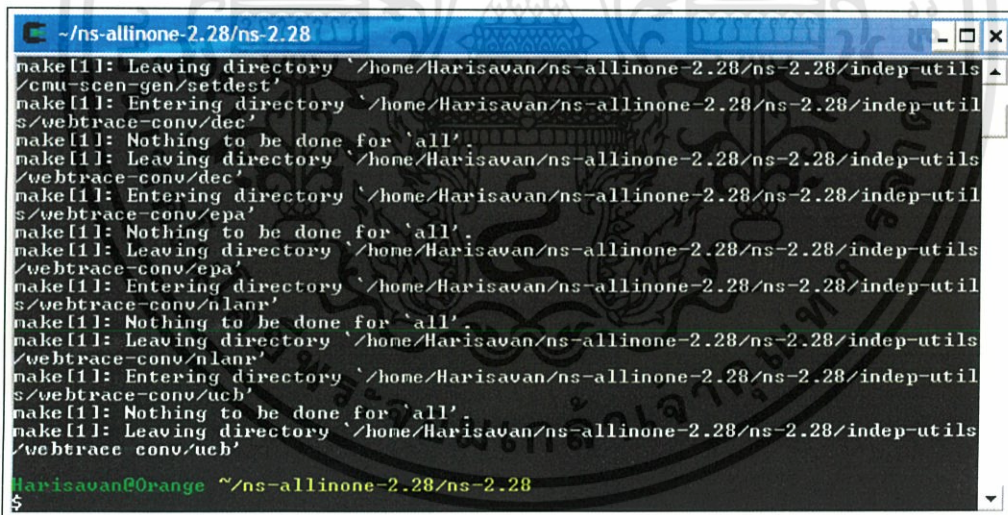
```

~/ns-allinone-2.28/ns-2.28
02_15_4trace.o upan/p802_15_4transac.o diffusion3/lib/nr/nr.o diffusion3/lib/dr...
diffusion3/filters/diffusion/one_phase_pull.o diffusion3/filters/diffusion/two...
phase_pull.o diffusion3/lib/diffapp.o diffusion3/ns/diffagent.o diffusion3/ns/d...
iffrtg.o diffusion3/ns/difftimer.o diffusion3/filter_core/filter_core.o diffusio...
n3/filter_core/iolog.o diffusion3/filter_core/iostats.o diffusion3/lib/main/attr...
s.o diffusion3/lib/main/events.o diffusion3/lib/main/iodev.o diffusion3/lib/main...
/iohook.o diffusion3/lib/main/timers.o diffusion3/lib/main/message.o diffusion3/...
lib/main/tools.o diffusion3/apps/gear_examples/gear_common.o diffusion3/app...
r_examples/gear_receiver.o diffusion3/apps/gear_examples/gear_sender.o diffusi...
on3/apps/rnst_examples/rnst_sink.o diffusion3/apps/rnst_examples/rnst_source.o di...
ffusion3/apps/ping/1pp_ping_sender.o diffusion3/apps/ping/1pp_ping_receiver.o di...
ffusion3/apps/ping/2pp_ping_sender.o diffusion3/apps/ping/2pp_ping_receiver.o di...
ffusion3/apps/ping/ping_common.o diffusion3/apps/ping/push_receiver.o diffusion3...
apps/ping/push_sender.o diffusion3/filters/gear/gear_attr.o diffusion3/filters/g...
ear/gear.o diffusion3/filters/gear/gear_tools.o diffusion3/filters/misc/log.o di...
ffusion3/filters/misc/srpt.o diffusion3/filters/misc/tag.o diffusion3/filters/r...
nst/rnst.o diffusion3/filters/rnst/rnst_filter.o gen/version.o gen/ns_tcl.o gen/...
rtypes.o common/win32.o emulate/net-ip.o emulate/net.o emulate/tap.o emulate/et...
her.o emulate/internet.o emulate/ping_responder.o emulate/arp.o emulate/icmp.o e...
mulate/net-pcap.o emulate/nat.o emulate/iptables.o emulate/tcptap.o emulate/inet.o...
common/tclAppInit.o gen/* ns.core core core.ns core.nsx core.nse common/ptypes2t...
cl common/ptypes2tcl.o
Harisavan@Orange ~/ns-allinone-2.28/ns-2.28
$ make

```

รูปที่ ก.1.25 การใช้คำสั่ง make

19. เมื่อโปรแกรมทำงานตามขั้นตอนดังกล่าวเสร็จเรียบร้อยแล้วจะปรากฏ ดังรูป



```

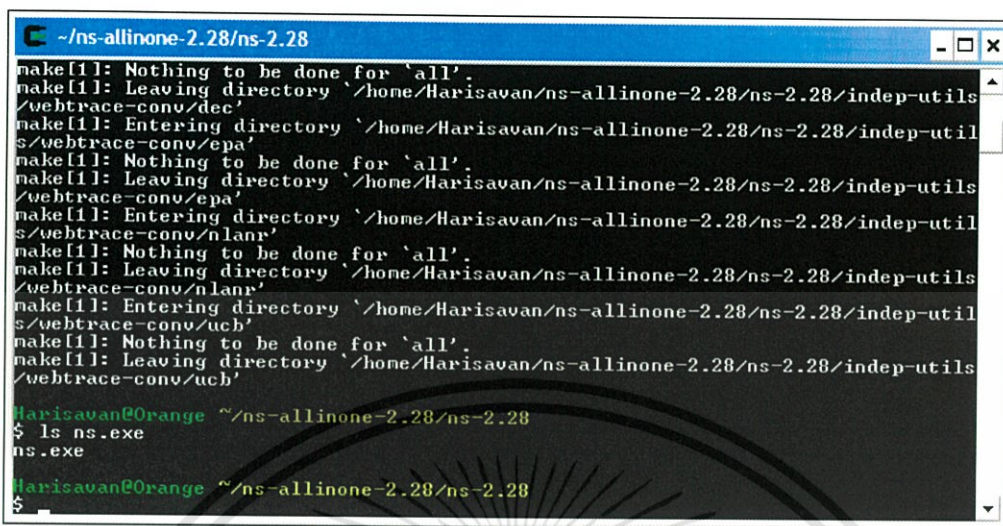
~/ns-allinone-2.28/ns-2.28
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils...
/cmu-scen-gen/setdest'
make[1]: Entering directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-util...
s/webtrace-conv/dec'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-util...
s/webtrace-conv/dec'
make[1]: Entering directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-util...
s/webtrace-conv/epa'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-util...
s/webtrace-conv/epa'
make[1]: Entering directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-util...
s/webtrace-conv/nlanr'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-util...
s/webtrace-conv/nlanr'
make[1]: Entering directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-util...
s/webtrace-conv/uch'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-util...
s/webtrace conv/uch'
Harisavan@Orange ~/ns-allinone-2.28/ns-2.28
$

```

รูปที่ ก.1.26 แสดงหน้าต่างของโปรแกรมเมื่อเสร็จสิ้นขั้นตอนการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20. พิมพ์คำสั่ง `ls ns.exe` เพื่อทดสอบว่าโปรแกรมสามารถทำงานได้



```

~/ns-allinone-2.28/ns-2.28
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/dec'
make[1]: Entering directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/epa'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/epa'
make[1]: Entering directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/nlanr'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/nlanr'
make[1]: Entering directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/uch'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/uch'

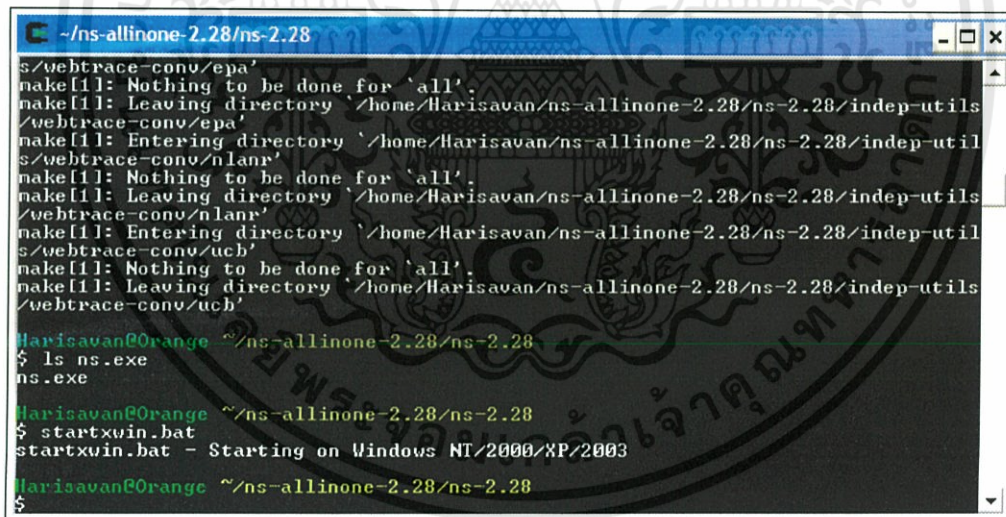
Harisavan@Orange: ~/ns-allinone-2.28/ns-2.28
$ ls ns.exe
ns.exe

Harisavan@Orange: ~/ns-allinone-2.28/ns-2.28
$

```

รูปที่ ก.1.27 แสดงการใช้คำสั่ง `ls ns.exe`

21. เริ่มต้นใช้งาน โปรแกรม NS-2 โดยการพิมพ์คำสั่ง `startxwin.bat` แล้วกดปุ่ม Enter



```

~/ns-allinone-2.28/ns-2.28
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/epa'
make[1]: Entering directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/epa'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/epa'
make[1]: Entering directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/nlanr'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/nlanr'
make[1]: Entering directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/uch'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/Harisavan/ns-allinone-2.28/ns-2.28/indep-utils/vebtrace-conv/uch'

Harisavan@Orange: ~/ns-allinone-2.28/ns-2.28
$ ls ns.exe
ns.exe

Harisavan@Orange: ~/ns-allinone-2.28/ns-2.28
$ startxwin.bat
startxwin.bat - Starting on Windows NT/2000/XP/2003

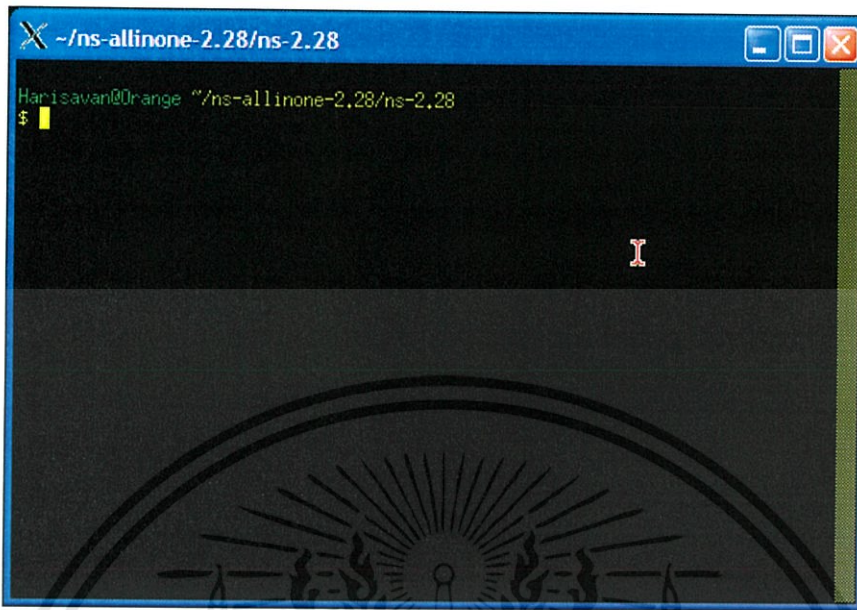
Harisavan@Orange: ~/ns-allinone-2.28/ns-2.28
$

```

รูปที่ ก.1.28 แสดงการใช้คำสั่ง `startxwin.bat`

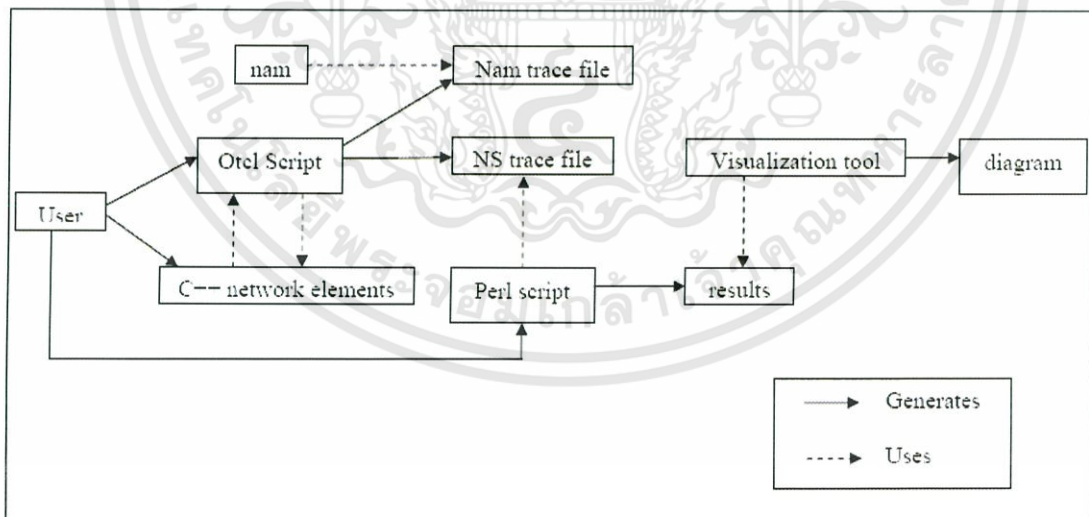
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

22. จะปรากฏหน้าต่างของโปรแกรม ดังรูป



รูปที่ ก.1.29 แสดงหน้าต่างของโปรแกรม

- การทำงานของโปรแกรม NS-2



รูปที่ ก.2 แสดงกระบวนการทำงานของโปรแกรม NS-2

ผู้ใช้งานสามารถสร้าง Otel Script, C++ Network Elements และ Perl Script เพื่อที่จะนำมาใช้ในการออกแบบเพื่อจำลองการทำงานของระบบเครือข่าย ซึ่งผู้ใช้งานต้องสร้างไฟล์ชนิด C++ Network ขึ้นมาเพื่อใช้เป็น Object ที่อยู่ใน Library ที่จะสามารถเรียกใช้ได้โดยเรียกผ่านเอกซามพลีเมนต์ที่สร้างขึ้นใหม่ซึ่งมีทั้งการกำหนดค่าและชื่อของเอกซามพลีเมนต์ที่สร้างขึ้นใหม่ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกซามพลีเมนต์ที่มีการนำไปใช้

Command ของ Otcl Script ซึ่งจะมี Otcl Linkage เป็นตัวเรียก Object ใน Library มาใช้งาน เมื่อ Otcl Script ถูกประมวลผล Otcl Script ก็จะสร้าง Nam Trace file และ NS Trace file ขึ้นมา ซึ่งตัว Nam Trace file ที่ได้นั้นจะถูกเรียกใช้โดย nam เพื่อที่จะนำไปประมวลผลแล้วแสดงผลออกมาในรูปแบบกราฟฟิก ในส่วนของ Perl Script นั้นจะเรียกใช้ NS Trace file เพื่อนำ NS Trace file มาผ่าน Filter เพื่อให้ได้ผลลัพธ์ออกมาเป็น Result ที่จะถูกเรียกใช้โดย Visualization Tool เช่น Xgraph เพื่อที่จะนำมาใช้สร้างกราฟที่แสดงผลการทำงานของการทำงานของระบบเครือข่าย

ดังที่ทราบกันว่าภาษาที่ใช้ในโปรแกรม NS-2 มีอยู่ 2 ภาษา เนื่องจาก Simulator มีลักษณะการทำงาน 2 อย่างที่แตกต่างกัน ซึ่งงานส่วนแรกจะเป็นการจำลองรายละเอียดต่าง ๆ เกี่ยวกับ Protocol จึงจำเป็นต้องใช้ System Programming Language ซึ่งมีประสิทธิภาพในการจัดการข้อมูลที่มีหน่วยของขนาดข้อมูลเป็น Byte , Packet Header และสร้าง Algorithm ที่สามารถประมวลผลข้อมูลที่มีขนาดใหญ่ได้ ซึ่งงานในส่วนนี้จะเน้นและให้ความสำคัญในเรื่องของ Run-time Speed ส่วนในเรื่องของ Turn-around Time (ใช้เพื่อค้นหา Bug , Re-run, Recompile) นั้นจะมีความสำคัญน้อยกว่า ซึ่งส่วนงานในส่วนที่ 2 ส่วนใหญ่จะเป็นเรื่องของการทำวิจัยเกี่ยวกับระบบเครือข่าย ซึ่งจะต้องยุ่งเกี่ยวกับพวกตัวแปร (Parameters) หรือ การทำ Configuration ซึ่ง Iteration time (Change the model and re-run) นั้นเป็นสิ่งสำคัญ เพราะฉะนั้นจึงจำเป็นที่โปรแกรม NS-2 จะต้องใช้ 2 ภาษาในการทำงาน คือ

1. C++

ใช้ในการสร้าง Protocol และ Application Agents ใช้กำหนดคุณสมบัติของกระบวนการทำงานของแต่ละ packet ตามที่ผู้ใช้งานต้องการ และสามารถเปลี่ยนการทำงานของ Object ใน C++ เพื่อเอาไว้ปรับใช้กับ Non-standard Policies ซึ่งมีข้อดีคือสามารถประมวลผลได้เร็ว แต่มีข้อเสียคือจะทำงานช้าเมื่อมีการ run simulation, ค้นหา bug, fix bug, recompile และ re-run

2. Otcl (Object Tool Command Language)

ใช้ในการเขียน Simulation script, Configuration และ setup ใช้เพื่อทดสอบผลของการกำหนด Network Parameter โดยทำการเปลี่ยนแปลงค่าใน object ของ C++ ใช้เพื่อวิเคราะห์ผลของการเปลี่ยนแปลง Parameter โดยใช้การ trace และแสดงผลทางหน้าจอ โดยมีข้อดีคือสามารถทำงานได้อย่างรวดเร็วเมื่อมีการเปลี่ยนแปลง และมีข้อเสียคือประมวลผลช้า

- โปรแกรมที่ใช้ทำการจำลองในงานวิจัย

โปรแกรมที่ใช้เป็นโปรแกรมการจำลองของระบบเครือข่ายไร้สายชนิดแอดฮอค ซึ่งเขียนขึ้นด้วยภาษา Tcl นั้น โดยการทำจำลองจะใช้โปรแกรม Cbrgen.tcl และ setdest เพื่อกำหนดรูปแบบและข้อกำหนด (Scenario) ในการจำลองของเครือข่าย

```
# =====
# Define options
# =====

Antenna/OmniAntenna set Gt_ 2.4
Antenna/OmniAntenna set Gr_ 2.4
#Antenna/OmniAntenna set Z_ 1.5
Phy/WirelessPhy set Pt_ 0.02998531
Phy/WirelessPhy set freq_ 2.45e+9
Phy/WirelessPhy set bandwidth_ 11e+6
Phy/WirelessPhy set L_ 1.0
#Phy/WirelessPhy set CPThresh_ 10.0
#Phy/WirelessPhy set CSThresh_ 1.559e-11
#Phy/WirelessPhy set RXThresh_ 3.652e-10
#Mac/802_11 set basicRate_ 1Mb ;# set this to 0 if want to use bandwidth_ for \n\
#Mac/802_11 set dataRate_ 11Mb ;# both control and data pkts\n

set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
set val(mac) Mac/802_11
set val(ifq) Queue/DropTail/PriQueue
set val(ll) LL
set val(ant) Antenna/OmniAntenna
set val(x) 1000 ;# X dimension of the topography
set val(y) 1000 ;# Y dimension of the topography
set val(ifqlen) 100 ;# max packet in ifq
set val(seed) 0.0
set val(rp) AODV
```

```

set val(nn)      100      ;# how many nodes are simulated
set val(cp)      "cbr100mc20.tcl"
set val(sc)      "sim500x500n100v20p100"
set val(stop)    150.0    ;# simulation time
set opt(energymodel) EnergyModel;
set opt(radiomodel) RadioModel;

set opt(initialenergy) 150;

# =====
# Main Program
# =====

#
# Initialize Global Variables
#
# create simulator instance

set ns_          [new Simulator]

# setup topography object

set topo [new Topography]

# create trace object for ns and nam

set tracefd      [open x_500_n100_c20_v20_p100_ori.tr w]
#set windowVsTime2 [open win1.tr w]
set namtrace     [open nam_500_n50_c20_v15_p120.nam w]

# $ns_ use-newtrace

```

เอกสารนี้จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
ns-random 1.0
# define topology
$topo load_flatgrid $val(x) $val(y)

#
# Create God
#
set god_ [create-god $val(nn)]

#
# define how node should be created
#
#global node setting
$ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
        -energyModel $opt(energymodel)\
        -idlePower 1.0\
    -rxPower 1.0\
    -txPower 1.0\
    -sleepPower 0.001\
    -transitionPower 0.5\
    -transitionTime 0.005\
        -initialEnergy $opt(initialenergy)\
    -phyType $val(netif) \
    -channelType $val(chan) \

```

```

-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF

#
# Create the specified number of nodes [$val(nn)] and "attach" them to the channel.

for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns_node]
    $node_($i) random-motion 0 ;# disable random motion
}

#
# Define node movement model
#
puts "Loading connection pattern..."
source $val(cp)

#
# Define traffic model
#
puts "Loading scenario file..."
source $val(sc)

#=====printring window size=====
#proc plotWindow {tcpSource file} {
#    global ns
#    set time 0.1
#    set now [$ns now]
#    set cwnd [$tcpSource set cwnd_]
#    puts $file "$now $cwnd"
#    $ns at [expr $now+$time] "plotWindow $tcpSource $file" }
#
#นี่เป็น $ns at 160.0 "plotWindow $tcp $windowVsTime2" นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
#ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```

#=====
# Define node initial position in nam

for {set i 0} {$i < $val(nn)} {incr i} {

    # 20 defines the node size in nam, must adjust it according to your scenario
    # The function must be called after mobility model is defined

    $ns_ initial_node_pos $node_($i) 20
}

#
# Tell nodes when the simulation ends
#
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at $val(stop).0 "$node_($i) reset";
}

$ns_ at $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"

puts $tracefd "M 0.0 nn $val(nn) x $val(x) y $val(y) rp $val(rp)"
puts $tracefd "M 0.0 sc $val(sc) cp $val(cp) seed $val(seed)"
puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"

puts "Starting Simulation..."

$ns_ run

```

- การใช้งานโปรแกรม Setdest

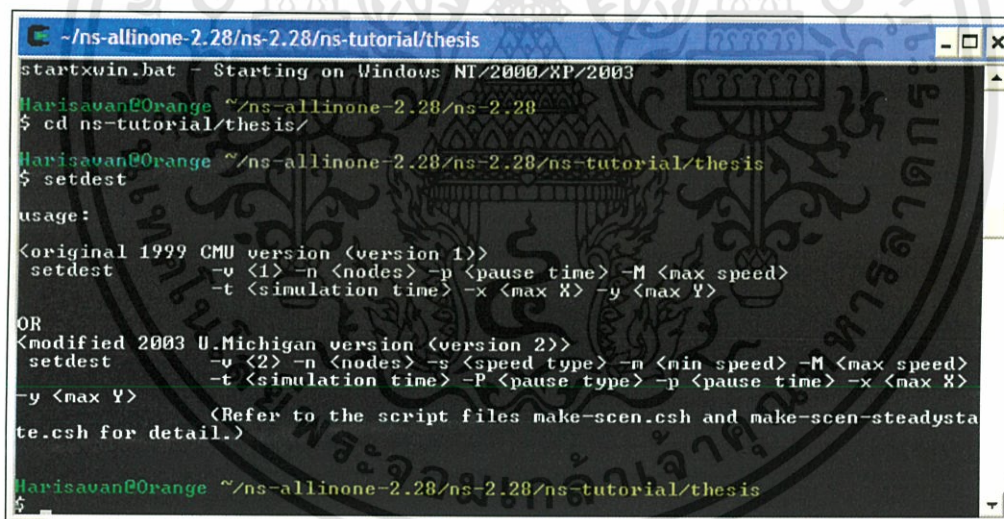
โปรแกรม Setdest เป็นโปรแกรมที่ใช้ทำการกำหนดการเคลื่อนที่ และตำแหน่งทั้งหมดของโหนดในระบบเครือข่ายที่ใช้ในการจำลอง ซึ่งจะมีอยู่ 2 เวอร์ชันสำหรับการใช้งาน ซึ่งรูปแบบคำสั่งแต่ละเวอร์ชันจะเป็นดังนี้

1) คำสั่งของ setdest เวอร์ชัน 1

```
setdest -v1 -n<node> -p<pause time> -M<Maxspeed> -t<simulation time>
```

```
-x<max X> -y<max Y> > output.tcl
```

โดย	n	จำนวนโหนดที่ใช้ในการจำลอง
	p	ค่าหยุดอยู่กับที่
	M	ความเร็วสูงสุด
	t	เวลาที่ใช้ในการจำลอง
	x	ความยาวของพื้นที่ที่ใช้ในการจำลอง
	y	ความกว้างของพื้นที่ที่ใช้ในการจำลอง



```
~/ns-allinone-2.28/ns-2.28/ns-tutorial/thesis
startxwin.bat - Starting on Windows NT/2000/XP/2003
Harisavan@Orange ~/ns-allinone-2.28/ns-2.28
$ cd ns-tutorial/thesis/
Harisavan@Orange ~/ns-allinone-2.28/ns-2.28/ns-tutorial/thesis
$ setdest
usage:
<original 1999 CMU version (version 1)>
setdest
-v <1> -n <nodes> -p <pause time> -M <max speed>
-t <simulation time> -x <max X> -y <max Y>
OR
<modified 2003 U. Michigan version (version 2)>
setdest
-v <2> -n <nodes> -s <speed type> -m <min speed> -M <max speed>
-t <simulation time> -P <pause type> -p <pause time> -x <max X>
-y <max Y>
(Refer to the script files make-scen.csh and make-scen-steadystate.csh for detail.)
Harisavan@Orange ~/ns-allinone-2.28/ns-2.28/ns-tutorial/thesis
$
```

รูปที่ ก.3 แสดงการใช้งานคำสั่ง setdest

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

~/ns-allinone-2.28/ns-2.28/ns-tutorial/thesis
$ cd ns-tutorial/thesis/
Harisavan@Orange ~/ns-allinone-2.28/ns-2.28/ns-tutorial/thesis
$ setdest
usage:
<original 1999 CMU version (version 1)>
setdest      -v <1> -n <nodes> -p <pause time> -M <max speed>
              -t <simulation time> -x <max X> -y <max Y>
OR
<modified 2003 U.Michigan version (version 2)>
setdest      -v <2> -n <nodes> -s <speed type> -m <min speed> -M <max speed>
              -t <simulation time> -P <pause type> -p <pause time> -x <max X>
              -y <max Y>
              (Refer to the script files make-scen.csh and make-scen-steadysta
              te.csh for detail.)
Harisavan@Orange ~/ns-allinone-2.28/ns-2.28/ns-tutorial/thesis
$ setdest -v1 -n50 -p150 -M15 -t150 -x200 -y200 > test
Harisavan@Orange ~/ns-allinone-2.28/ns-2.28/ns-tutorial/thesis
$

```

รูปที่ ก.4 การใช้งานโปรแกรม setdest เวอร์ชัน 1

2) คำสั่งของ setdest เวอร์ชัน 2

```

setdest -v<2> -n<node> -s<speed type> -m<min speed> -M<max speed>
-t<simulation time> -P<pause type> -p<pause time> -x<max X> -y<max Y>
> output.tcl

```

โดย	n	จำนวนโหนดในเครือข่าย
	s	ชนิดของการเคลื่อนที่ แบ่งออกเป็น -1 เป็นการเคลื่อนที่แบบมีรูปแบบ (Uniform) -2 เป็นการเคลื่อนที่แบบธรรมดา (Normal)
	m	ความเร็วในการเคลื่อนที่ต่ำสุด
	M	ความเร็วในการเคลื่อนที่สูงสุด
	t	เวลาที่ใช้ในการจำลอง
	P	ชนิดของการหยุดอยู่กับที่ของโหนด แบ่งออกเป็น -1 เป็นการหยุดแบบคงที่ (Constant) -2 เป็นการหยุดแบบธรรมดา (Normal)
	x	ความยาวของพื้นที่ที่ใช้ในการจำลอง
	y	ความกว้างของพื้นที่ที่ใช้ในการจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

~/ns-allinone-2.28/ns-2.28/ns-tutorial/thesis
$ setdest
usage:
<original 1999 CMU version (version 1)>
setdest      -v <1> -n <nodes> -p <pause time> -M <max speed>
              -t <simulation time> -x <max X> -y <max Y>

OR
<modified 2003 U.Michigan version (version 2)>
setdest      -v <2> -n <nodes> -s <speed type> -m <min speed> -M <max speed>
              -y <max Y>
              -t <simulation time> -P <pause type> -p <pause time> -x <max X>
              (Refer to the script files make-scen.csh and make-scen-steadysta
              te.csh for detail.)

Harisavan@Orange ~/ns-allinone-2.28/ns-2.28/ns-tutorial/thesis
$ setdest -v1 -n50 -p150 -M15 -t150 -x200 -y200 > test

Harisavan@Orange ~/ns-allinone-2.28/ns-2.28/ns-tutorial/thesis
$ setdest -v2 -n50 -s1 -m1 -M10 -t150 -P1 -p2.0 -x500 -y500 > test2

Harisavan@Orange ~/ns-allinone-2.28/ns-2.28/ns-tutorial/thesis
$

```

รูปที่ ก.5 การใช้งาน โปรแกรม setdest เวอร์ชัน 2

ตัวอย่างโค้ด โปรแกรมที่ได้จากการรัน โปรแกรม setdest จะเป็นดังต่อไปนี้

```

test2.tcl

#
# nodes: 50, speed type: 1, min speed: 1.00, max speed: 10.00
# avg speed: 3.79, pause type: 1, pause: 2.00, max x: 500.00, max y: 500.00
#
$node_(0) set X_ 266.515666138208
$node_(0) set Y_ 33.856773060329
$node_(0) set Z_ 0.000000000000
$node_(1) set X_ 35.494420999391
$node_(1) set Y_ 185.805195161952
$node_(1) set Z_ 0.000000000000
$node_(2) set X_ 275.484342663722
$node_(2) set Y_ 97.011232440904
$node_(2) set Z_ 0.000000000000
$node_(3) set X_ 222.893653288921
$node_(3) set Y_ 201.234133182256
$node_(3) set Z_ 0.000000000000
$node_(4) set X_ 344.640658513004
$node_(4) set Y_ 124.999956809914

```

\$node_(4) set Z_ 0.000000000000
 \$node_(5) set X_ 97.903289334822
 \$node_(5) set Y_ 273.111402594292
 \$node_(5) set Z_ 0.000000000000
 \$node_(6) set X_ 48.362949178436
 \$node_(6) set Y_ 99.464166022033
 \$node_(6) set Z_ 0.000000000000
 \$node_(7) set X_ 225.574119114181
 \$node_(7) set Y_ 485.132331123465
 \$node_(7) set Z_ 0.000000000000
 \$node_(8) set X_ 304.144027238236
 \$node_(8) set Y_ 172.605679114810
 \$node_(8) set Z_ 0.000000000000
 \$node_(9) set X_ 456.668940369771
 \$node_(9) set Y_ 236.955938345481
 \$node_(9) set Z_ 0.000000000000
 \$node_(10) set X_ 296.720752310454
 \$node_(10) set Y_ 41.039308075834
 \$node_(10) set Z_ 0.000000000000
 \$node_(11) set X_ 26.001191909483
 \$node_(11) set Y_ 52.621383672871
 \$node_(11) set Z_ 0.000000000000
 \$node_(12) set X_ 394.561114154922
 \$node_(12) set Y_ 34.709990890622
 \$node_(12) set Z_ 0.000000000000
 \$node_(13) set X_ 139.190376259293
 \$node_(13) set Y_ 238.787654244302
 \$node_(13) set Z_ 0.000000000000
 \$node_(14) set X_ 256.853861251288
 \$node_(14) set Y_ 457.497405274646
 \$node_(14) set Z_ 0.000000000000
 \$node_(15) set X_ 220.340775286518
 \$node_(15) set Y_ 397.470725857166

\$node_(15) set Z_ 0.000000000000
 \$node_(16) set X_ 12.404138473808
 \$node_(16) set Y_ 479.863094820530
 \$node_(16) set Z_ 0.000000000000
 \$node_(17) set X_ 387.194569580367
 \$node_(17) set Y_ 421.491671276807
 \$node_(17) set Z_ 0.000000000000
 \$node_(18) set X_ 278.727474383850
 \$node_(18) set Y_ 347.056740356563
 \$node_(18) set Z_ 0.000000000000
 \$node_(19) set X_ 455.497151879456
 \$node_(19) set Y_ 341.069227769598
 \$node_(19) set Z_ 0.000000000000
 \$node_(20) set X_ 182.030474385791
 \$node_(20) set Y_ 112.580760479159
 \$node_(20) set Z_ 0.000000000000
 \$node_(21) set X_ 31.107487033670
 \$node_(21) set Y_ 62.868805852884
 \$node_(21) set Z_ 0.000000000000
 \$node_(22) set X_ 190.095250154802
 \$node_(22) set Y_ 368.825731732825
 \$node_(22) set Z_ 0.000000000000
 \$node_(23) set X_ 202.687421268547
 \$node_(23) set Y_ 479.275296719107
 \$node_(23) set Z_ 0.000000000000
 \$node_(24) set X_ 143.813950804365
 \$node_(24) set Y_ 410.088424640333
 \$node_(24) set Z_ 0.000000000000
 \$node_(25) set X_ 298.366716494848
 \$node_(25) set Y_ 370.395863322155
 \$node_(25) set Z_ 0.000000000000
 \$node_(26) set X_ 8.393935800050

\$node_(26) set Z_ 0.000000000000
 \$node_(27) set X_ 37.790445694796
 \$node_(27) set Y_ 36.493087674156
 \$node_(27) set Z_ 0.000000000000
 \$node_(28) set X_ 25.146640075953
 \$node_(28) set Y_ 98.456500791701
 \$node_(28) set Z_ 0.000000000000
 \$node_(29) set X_ 452.021687994830
 \$node_(29) set Y_ 424.416337925614
 \$node_(29) set Z_ 0.000000000000
 \$node_(30) set X_ 450.955736520419
 \$node_(30) set Y_ 432.078629120336
 \$node_(30) set Z_ 0.000000000000
 \$node_(31) set X_ 64.479521920844
 \$node_(31) set Y_ 167.437812506004
 \$node_(31) set Z_ 0.000000000000
 \$node_(32) set X_ 476.491484816705
 \$node_(32) set Y_ 99.213769574758
 \$node_(32) set Z_ 0.000000000000
 \$node_(33) set X_ 442.773952404266
 \$node_(33) set Y_ 68.708653978864
 \$node_(33) set Z_ 0.000000000000
 \$node_(34) set X_ 119.681244807713
 \$node_(34) set Y_ 10.301545179151
 \$node_(34) set Z_ 0.000000000000
 \$node_(35) set X_ 273.013607898455
 \$node_(35) set Y_ 367.072714108770
 \$node_(35) set Z_ 0.000000000000
 \$node_(36) set X_ 324.829121717358
 \$node_(36) set Y_ 161.490415476730
 \$node_(36) set Z_ 0.000000000000
 \$node_(37) set X_ 452.207831446833
 \$node_(37) set Y_ 299.864396655884

\$node_(37) set Z_ 0.000000000000
 \$node_(38) set X_ 164.950000659935
 \$node_(38) set Y_ 376.242195432628
 \$node_(38) set Z_ 0.000000000000
 \$node_(39) set X_ 20.874611647315
 \$node_(39) set Y_ 136.826656982290
 \$node_(39) set Z_ 0.000000000000
 \$node_(40) set X_ 389.727390502416
 \$node_(40) set Y_ 108.467354453450
 \$node_(40) set Z_ 0.000000000000
 \$node_(41) set X_ 299.372711630893
 \$node_(41) set Y_ 210.381880462969
 \$node_(41) set Z_ 0.000000000000
 \$node_(42) set X_ 34.598425891360
 \$node_(42) set Y_ 330.637823970212
 \$node_(42) set Z_ 0.000000000000
 \$node_(43) set X_ 388.558517633977
 \$node_(43) set Y_ 74.300013355539
 \$node_(43) set Z_ 0.000000000000
 \$node_(44) set X_ 222.842388984565
 \$node_(44) set Y_ 416.578527225259
 \$node_(44) set Z_ 0.000000000000
 \$node_(45) set X_ 200.958900223358
 \$node_(45) set Y_ 228.050454155191
 \$node_(45) set Z_ 0.000000000000
 \$node_(46) set X_ 424.821636607614
 \$node_(46) set Y_ 368.923211990862
 \$node_(46) set Z_ 0.000000000000
 \$node_(47) set X_ 467.010906929679
 \$node_(47) set Y_ 309.614906646288
 \$node_(47) set Z_ 0.000000000000
 \$node_(48) set X_ 20.750308133677
 \$node_(48) set Y_ 22.357243846708

```

$node_(48) set Z_ 0.000000000000
$node_(49) set X_ 108.686014219814
$node_(49) set Y_ 166.057778880936
$node_(49) set Z_ 0.000000000000
$ns_ at 0.000000000000 "$node_(0) setdest 45.615640349699 239.310950454482
1.050714850088"
$ns_ at 0.000000000000 "$node_(1) setdest 35.427249239028 458.865163369094
2.198809584452"
$ns_ at 0.000000000000 "$node_(2) setdest 110.063190430669 291.293219683922
6.925368027174"
$ns_ at 0.000000000000 "$node_(3) setdest 147.683442038986 333.290866535273
5.980585711999"
$ns_ at 0.000000000000 "$node_(4) setdest 45.653096864895 212.523330283550
3.256354710861"
$ns_ at 0.000000000000 "$node_(5) setdest 347.357417282263 143.728819767850
8.199962573962"
$ns_ at 0.000000000000 "$node_(6) setdest 11.616278420246 319.323358223601
9.965918790296"
$ns_ at 0.000000000000 "$node_(7) setdest 126.710796113090 450.059389721684
2.514763742469"
$ns_ at 0.000000000000 "$node_(8) setdest 28.145987506575 161.906016333134
1.121767083725"
$ns_ at 0.000000000000 "$node_(9) setdest 171.856129017210 314.708007839337
1.182113738470"
$ns_ at 0.000000000000 "$node_(10) setdest 206.251836824320 350.822254310136
8.236574770698"
$ns_ at 0.000000000000 "$node_(11) setdest 123.053604177001 299.725505486807
6.010264412764"
$ns_ at 0.000000000000 "$node_(12) setdest 433.268007850215 317.083623365824
4.431713719805"
$ns_ at 0.000000000000 "$node_(13) setdest 467.904990381617 338.895855352827
2.002092732218"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\$ns_ at 0.000000000000 "\$node_(14) setdest 80.172324710489 277.691151890848
 4.595636179413"
 \$ns_ at 0.000000000000 "\$node_(15) setdest 140.984967426600 164.249026021873
 7.345988230539"
 \$ns_ at 0.000000000000 "\$node_(16) setdest 59.406849173043 254.268971478554
 6.576130805266"
 \$ns_ at 0.000000000000 "\$node_(17) setdest 210.570424399024 271.604507578010
 6.541633934868"
 \$ns_ at 0.000000000000 "\$node_(18) setdest 476.511210695452 309.117968728891
 1.007894948507"
 \$ns_ at 0.000000000000 "\$node_(19) setdest 304.090566293066 226.332925510883
 7.910235050677"
 \$ns_ at 0.000000000000 "\$node_(20) setdest 362.246850819174 41.168502430210
 2.895497180350"
 \$ns_ at 0.000000000000 "\$node_(21) setdest 380.505440441224 87.898630249061
 1.508210590740"
 \$ns_ at 0.000000000000 "\$node_(22) setdest 20.521901028360 496.578628148035
 3.062102337349"
 \$ns_ at 0.000000000000 "\$node_(23) setdest 495.339679306060 376.376573645120
 5.261523982370"
 \$ns_ at 0.000000000000 "\$node_(24) setdest 212.088788536952 37.349579103457
 5.957194386073"
 \$ns_ at 0.000000000000 "\$node_(25) setdest 25.423646505950 39.975319945921
 4.295191032439"
 \$ns_ at 0.000000000000 "\$node_(26) setdest 414.341354668839 415.158723563192
 5.017947850920"
 \$ns_ at 0.000000000000 "\$node_(27) setdest 40.053248133295 356.942892597085
 1.465345734915"
 \$ns_ at 0.000000000000 "\$node_(28) setdest 287.381536973491 402.147175429066
 1.275324246676"
 \$ns_ at 0.000000000000 "\$node_(30) setdest 482.323268503705 215.782987392242
 8.465901075142"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\$ns_ at 0.000000000000 "\$node_(31) setdest 103.851760411907 420.526784488366
5.696562931148"
\$ns_ at 0.000000000000 "\$node_(32) setdest 386.868764988310 102.696352140243
6.671166099981"
\$ns_ at 0.000000000000 "\$node_(33) setdest 177.977833435729 316.782469137282
3.172695352445"
\$ns_ at 0.000000000000 "\$node_(35) setdest 361.083510822004 157.935668795979
1.346451921636"
\$ns_ at 0.000000000000 "\$node_(36) setdest 231.910549974394 32.474823238971
4.010742503273"
\$ns_ at 0.000000000000 "\$node_(37) setdest 279.234069767568 196.985982817850
1.364638665888"
\$ns_ at 0.000000000000 "\$node_(38) setdest 444.708497146929 55.720879600836
7.545832234792"
\$ns_ at 0.000000000000 "\$node_(39) setdest 79.856726250192 254.015723321417
1.431005784229"
\$ns_ at 0.000000000000 "\$node_(40) setdest 302.655657393014 66.061131362970
7.002513405989"
\$ns_ at 0.000000000000 "\$node_(41) setdest 337.036250881744 423.181675030335
1.126708010527"
\$ns_ at 0.000000000000 "\$node_(42) setdest 161.901491502186 8.133728916714
2.228126658464"
\$ns_ at 0.000000000000 "\$node_(43) setdest 314.172219100367 403.757036542861
1.248424467554"
\$ns_ at 0.000000000000 "\$node_(44) setdest 92.092827929954 106.464630957843
2.635397823333"
\$ns_ at 0.000000000000 "\$node_(45) setdest 84.077424366983 117.168906580455
8.094846652990"
\$ns_ at 0.000000000000 "\$node_(46) setdest 171.133361313431 410.537478023161
3.737826962133"
\$ns_ at 0.000000000000 "\$node_(47) setdest 120.650489720353 314.015157244902
2.649147883529"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$ns_ at 0.000000000000 "$node_(48) setdest 76.254629497641 278.960830188322
1.806609606085"
$ns_ at 0.000000000000 "$node_(49) setdest 191.048272358729 17.332507903958
2.836474017703"
$god_ set-dist 1 2 2
$god_ set-dist 2 3 1
$god_ set-dist 3 4 1
$god_ set-dist 4 5 2
$god_ set-dist 5 6 1
$god_ set-dist 6 7 2
$god_ set-dist 7 8 2
$god_ set-dist 8 9 1
$god_ set-dist 9 10 2
$god_ set-dist 10 11 2
$god_ set-dist 11 12 2
$god_ set-dist 12 13 2
$god_ set-dist 13 14 1
$god_ set-dist 14 15 1
$god_ set-dist 15 16 1
$god_ set-dist 16 17 2
$god_ set-dist 17 18 1
$god_ set-dist 18 19 1
$god_ set-dist 19 20 2
$god_ set-dist 20 21 1
$god_ set-dist 21 22 2
$god_ set-dist 22 23 1
$god_ set-dist 23 24 1
$god_ set-dist 24 25 1
$god_ set-dist 25 49 2
$god_ set-dist 26 27 1
$god_ set-dist 26 49 1
$god_ set-dist 27 28 1
$god_ set-dist 27 49 1

```

\$god_set-dist 28 29 3

\$god_set-dist 29 30 1

\$god_set-dist 30 31 3

\$god_set-dist 31 32 2

\$god_set-dist 32 33 1

\$god_set-dist 33 34 2

\$god_set-dist 34 35 2

\$god_set-dist 35 36 1

\$god_set-dist 36 37 1

\$god_set-dist 37 38 2

\$god_set-dist 38 39 2

\$god_set-dist 39 40 2

\$god_set-dist 40 41 1

\$god_set-dist 41 42 2

\$god_set-dist 42 43 2

\$god_set-dist 43 44 2

\$god_set-dist 44 45 1

\$god_set-dist 45 48 2

\$god_set-dist 45 49 1

\$god_set-dist 46 47 1

\$god_set-dist 46 48 3

\$god_set-dist 48 49 1

\$ns_at 0.006286322484 "\$god_set-dist 2 5 1"

\$ns_at 0.006286322484 "\$god_set-dist 2 7 2"

\$ns_at 2.000000000000 "\$node_(29) setdest 401.556364266138 342.323757522586
4.183103104840"

\$ns_at 2.000000000000 "\$node_(34) setdest 291.870911770768 173.047219285234
1.241727548716"

\$ns_at 13.444479796041 "\$node_(32) setdest 386.868764988311 102.696352140243
0.000000000000"

\$ns_at 13.462311381379 "\$god_set-dist 5 37 1"

\$ns_at 13.462311381379 "\$god_set-dist 37 48 2"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\$ns_ at 13.830639104160 "\$node_(40) setdest 302.655657393014 66.061131362970
 0.000000000000"
 \$ns_ at 14.053527469152 "\$god_ set-dist 6 25 1"
 \$ns_ at 14.072346208251 "\$god_ set-dist 10 46 1"
 \$ns_ at 15.444479796041 "\$node_(32) setdest 210.864781485841 32.438749505035
 8.756766909130"
 \$ns_ at 15.830639104160 "\$node_(40) setdest 322.989155929942 286.986090404239
 7.476156148370"
 \$ns_ at 19.902619104839 "\$node_(45) setdest 84.077424366983 117.168906580455
 0.000000000000"
 \$ns_ at 21.902619104839 "\$node_(45) setdest 223.331596458557 240.072328232938
 7.269655334784"
 \$ns_ at 22.367120864872 "\$node_(6) setdest 11.616278420246 319.323358223601
 0.000000000000"
 \$ns_ at 23.997558549743 "\$god_ set-dist 17 32 1"
 \$ns_ at 24.015645491717 "\$node_(19) setdest 304.090566293065 226.332925510883
 0.000000000000"
 \$ns_ at 24.367120864872 "\$node_(6) setdest 435.114272638170 433.498730107149
 7.899421462575"
 \$ns_ at 25.036387897400 "\$node_(29) setdest 401.556364266137 342.323757522586
 0.000000000000"
 \$ns_ at 25.410926415519 "\$node_(3) setdest 147.683442038986 333.290866535273
 0.000000000000"
 \$ns_ at 25.430395437292 "\$god_ set-dist 22 41 2"
 \$ns_ at 25.475545683288 "\$god_ set-dist 14 45 1"
 \$ns_ at 25.496106883787 "\$god_ set-dist 31 48 2"
 \$ns_ at 25.816305831353 "\$node_(30) setdest 482.323268503705 215.782987392242
 0.000000000000"
 \$ns_ at 25.873968290653 "\$god_ set-dist 24 33 1"
 \$ns_ at 25.898857044023 "\$god_ set-dist 34 38 1"
 \$ns_ at 26.015645491717 "\$node_(19) setdest 444.051828901931 151.586799190858
 1.997167946866"
 \$ns_ at 26.060126301495 "\$god_ set-dist 5 39 2"

\$ns_ at 101.779957430998 "\$node_(25) setdest 164.822184895867 274.951178159063
4.907389545286"

\$ns_ at 101.782744019607 "\$god_ set-dist 2 26 1"

\$ns_ at 101.960477039696 "\$node_(10) setdest 484.107286481726 413.544290819488
3.409757648881"

\$ns_ at 101.974129098997 "\$node_(7) setdest 13.566137249990 292.869267150947
0.000000000000"

\$ns_ at 102.092592077261 "\$god_ set-dist 10 46 1"

\$ns_ at 102.240854144009 "\$node_(11) setdest 444.634772018538 235.502705207226
0.000000000000"

\$ns_ at 102.278020487812 "\$god_ set-dist 2 25 1"

\$ns_ at 102.328900330633 "\$node_(31) setdest 215.402295534424 80.309995264858
4.723322913389"

\$ns_ at 103.974129098997 "\$node_(7) setdest 338.712160115160 87.277741905714
2.264843434116"

\$ns_ at 104.240854144009 "\$node_(11) setdest 35.430811548049 6.056795306460
4.146930981093"

\$ns_ at 104.359019315429 "\$god_ set-dist 3 6 1"

\$ns_ at 104.599383950255 "\$node_(29) setdest 33.257053447297 418.546825893628
0.000000000000"

\$ns_ at 105.463097186964 "\$node_(19) setdest 444.051828901931 151.586799190858
0.000000000000"

\$ns_ at 106.599383950255 "\$node_(29) setdest 187.472163232558 302.029695204035
2.136992867453"

\$ns_ at 106.674686266213 "\$god_ set-dist 43 44 2"

\$ns_ at 106.708913041940 "\$god_ set-dist 14 47 2"

\$ns_ at 107.463097186964 "\$node_(19) setdest 442.295437794517 288.540671466957
4.179096008011"

\$ns_ at 111.653319973005 "\$node_(20) setdest 298.458389374275 186.813192525214
0.000000000000"

\$ns_ at 113.653319973005 "\$node_(20) setdest 170.123718885177 430.551590061451
3.544784658895"

\$ns_ at 113.675739224317 "\$god_ set-dist 21 31 1" ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

\$ns_ at 122.507758935322 "\$node_(24) setdest 225.551561316178 453.851133561934
9.305090247760"

\$ns_ at 122.587050176306 "\$god_ set-dist 5 38 1"

\$ns_ at 122.650534663468 "\$node_(15) setdest 54.026219699870 453.031162552182
4.070428532001"

\$ns_ at 130.772974920629 "\$god_ set-dist 6 22 1"

\$ns_ at 130.985009936083 "\$god_ set-dist 11 25 1"

\$ns_ at 131.153030533106 "\$node_(4) setdest 257.255924984168 424.809528947897
3.794778487020"

\$ns_ at 132.754635928487 "\$node_(47) setdest 422.931331901280 476.020264325714
9.906747967844"

\$ns_ at 135.045635745269 "\$node_(26) setdest 392.216790020720 165.669785337363
0.000000000000"

\$ns_ at 136.068981284907 "\$node_(30) setdest 171.483285415108 14.871934101303
0.000000000000"

\$ns_ at 137.045635745269 "\$node_(26) setdest 424.341727458658 162.805337543485
4.003589759755"

\$ns_ at 138.068981284907 "\$node_(30) setdest 166.635152921107 406.997291174586
1.383009244144"

\$ns_ at 140.236963625349 "\$node_(19) setdest 442.295437794517 288.540671466957
0.000000000000"

\$ns_ at 141.571009319689 "\$node_(33) setdest 210.487221083916 315.482394099314
0.000000000000"

\$ns_ at 142.236963625349 "\$node_(19) setdest 128.677402335429 132.470521389942
4.540623927361"

\$ns_ at 143.571009319689 "\$node_(33) setdest 88.518459329344 147.807375002637
1.704656370117"

\$ns_ at 144.440855153581 "\$node_(24) setdest 225.551561316178 453.851133561935
0.000000000000"

\$ns_ at 145.101503610352 "\$node_(26) setdest 424.341727458658 162.805337543485
0.000000000000"

\$ns_ at 145.320750790920 "\$node_(48) setdest 76.254629497640 278.960830188322
0.000000000000"

```

$ns_ at 145.943717372580 "$node_(36) setdest 431.848418974427 359.847344189940
0.000000000000"
$ns_ at 146.201371103337 "$god_ set-dist 26 40 2"
$ns_ at 146.440855153581 "$node_(24) setdest 341.802758070401 255.608093381506
8.252109757261"
$ns_ at 146.578279628590 "$god_ set-dist 1 37 1"
$ns_ at 147.101503610352 "$node_(26) setdest 466.161961891150 126.533031887112
1.200817250127"
$ns_ at 147.320750790920 "$node_(48) setdest 472.992364871877 166.118189588325
5.670602916387"
$ns_ at 147.479173945914 "$node_(37) setdest 279.234069767568 196.985982817850
0.000000000000"
$ns_ at 147.943717372580 "$node_(36) setdest 317.215407262744 147.391102662624
5.044733581670"
$ns_ at 149.479173945914 "$node_(37) setdest 180.703284006166 194.109306292314
8.291195422311"
$ns_ at 149.789426371579 "$god_ set-dist 24 40 1"
$ns_ at 149.937191205310 "$god_ set-dist 2 31 2"
#
# Destination Unreachables: 0
#
# Route Changes: 2255
#
# Link Changes: 1894
#
# Node | Route Changes | Link Changes
# 0 | 64 | 58
# 1 | 97 | 59
# 2 | 110 | 104
# 3 | 85 | 83
# 4 | 77 | 76
# 5 | 75 | 71
# 6 | 111 | 93

```

# 7	97	69
# 8	57	57
# 9	64	56
# 10	113	104
# 11	122	100
# 12	95	79
# 13	56	56
# 14	102	92
# 15	96	96
# 16	131	97
# 17	111	95
# 18	54	54
# 19	85	58
# 20	101	89
# 21	80	68
# 22	148	58
# 23	91	73
# 24	104	100
# 25	103	85
# 26	101	91
# 27	80	54
# 28	61	55
# 29	129	95
# 30	114	74
# 31	108	104
# 32	80	64
# 33	77	68
# 34	78	71
# 35	51	51
# 36	81	77
# 37	73	65
# 38	166	150
# 39	70	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# 40	73	70
# 41	45	45
# 42	73	63
# 43	65	50
# 44	77	75
# 45	120	86
# 46	92	77
# 47	99	87
# 48	78	54
# 49	90	72
#		

- **การใช้งาน Cbrgen**

เป็นโปรแกรมที่ช่วยกำหนดการจำลองการสนทนาระหว่างโหนดที่อยู่ในเครือข่าย ซึ่งจะมีคำสั่งดังนี้

```
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [mc macconnection] [-rate rate]
> outputfile.tcl
```

โดย

type	ชนิดของการสื่อสาร สามารถใช้ได้ทั้งแบบ tcp และ cbr
nn	จำนวนของโหนดที่ใช้ในการจำลอง
seed	ใช้คำนวณเวลาระหว่างแต่ละแพ็คเก็ตในการเชื่อมต่อแบบ cbr
mc	จำนวนคู่สนทนาสูงสุด
rate	อัตราการส่งข้อมูล

ตัวอย่างการใช้งานคำสั่ง cbrgen เป็นดังนี้

```
ns cbrgen.tcl -type -nn50 -seed 1 -mc 10 -rate 1 > testcbr.tcl
```

ซึ่งตัวอย่างผลลัพธ์ที่ได้ออกมาเป็นดังนี้

```
testcbr.tcl
```

```
#
```

```
# nodes: 50, max conn: 10, send rate: 1, seed: 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#
# 1 connecting to 2 at time 2.5568388786897245
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 1
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 2.5568388786897245 "$cbr_(0) start"
#
# 4 connecting to 5 at time 56.333118917575632
#
set udp_(1) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(1)
set null_(1) [new Agent/Null]
$ns_ attach-agent $node_(5) $null_(1)
set cbr_(1) [new Application/Traffic/CBR]
$cbr_(1) set packetSize_ 512
$cbr_(1) set interval_ 1
$cbr_(1) set random_ 1
$cbr_(1) set maxpkts_ 10000
$cbr_(1) attach-agent $udp_(1)
$ns_ connect $udp_(1) $null_(1)
$ns_ at 56.333118917575632 "$cbr_(1) start"
#

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การวิเคราะห์ผลที่ได้จากการจำลองการทำงาน

หลังจากการจำลองการทำงานจะได้ผลลัพธ์ออกมาเป็นเทรซไฟล์ (Trace File) ซึ่งจะเก็บข้อมูลระหว่างการจำลองการทำงานลงไป ซึ่งมีข้อมูลของโปรโตคอลหลายเลขอร์แต่ละเลขอร์ถูกแยกออกจากกัน โดยมีรูปแบบของเทรซไฟล์ตามตารางดังต่อไปนี้

ตารางที่ ก.1 รูปแบบของเทรซไฟล์ของแพ็คเก็ตทั่วไปในเครือข่ายไร้สาย

อีเวนต์	ตัวย่อ	ชนิดข้อมูล	ข้อมูลที่เก็บ
Wireless Event	r:Send r:Receive d:Drop f:Forward		%9f%d (%6.2f%6.2f) %3s %4s %d %s %d [%x %x %x %x]
			%.9f_%.d_%.3s%.4s%.d%%.d [%x %x %x %x]
		double	Time
		int	Node ID
		double	X Coordinate
		double	Y Coordinate
		string	Trace Name
		string	Reason
		int	Event Identifier
		string	Packet Type
		int	Packet Size
		hexadecimal	Time to Send Data
		hexadecimal	Destination MAC Address
		hexadecimal	Source Mac Address
hexadecimal	Type (ARP, IP)		

ตารางที่ ก.2 แสดงรูปแบบของเทรซไฟล์ของแพ็คเกจข้อมูลแต่ละเลเยอร์

อีเวนท์	ชนิดข้อมูล	ข้อมูลที่เก็บ
ARP Trace	----[%s %d %d %d %d]	
	string	Request or Reply
	int	Source MAC Address
	int	Source Address
	int	Destination MAC Address
	int	Destination Address
IP Trace	-----[%d:%d %d:%d %d %d]	
	int	Source IP Address
	int	Source Port Number
	int	Destination IP Address
	int	Destination Port Number
	int	TTL Value
	int	Next Hop Address , If Any
TCP Trace	[%d %d]%d%d	
	int	Sequence Number
	int	Acknowledgement Number
	int	Number of Times Packet Was Forwarded
	int	Optimal Number Of Forwards
CBR Trace	[%d]%d%d	
	int	Sequence Number
	int	Number of Times Packet Was Forwarded
	int	Optimal Number Of Forwards
IMEP Trace	[%c %c %c 0x%04x]	
	char	Acknowledgment Flag
	char	Hello Flag
	char	Object Flag
	hexadecimal	Lenght

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.3 แสดงรูปแบบเทรซในการเคลื่อนที่และค่าพลังงานของโหนดไร้สาย

อีเวนต์	ตัวย่อ	ชนิดข้อมูล	ข้อมูลที่เก็บ
Mobile Node Movement	M	%5f%d (%.2f, %.2f, %.2f), (%.2f, %.2f), %.2f	
		double	Time
		int	Address
		double	X Coordinate
		double	Y Coordinate
		double	Z Coordinate
		double	Destination X Coordinate
		double	Destination Y Coordinate
		double	Movement Speed
Mobile Node Energy	N	-t %f -n%d -e %f	
		double	Time
		int	Address
		double	Energy

ตัวอย่างเทรซไฟล์ที่ได้จากการจำลองในงานวิจัยนี้

tracefile
r 8.896141830 _89_ RTR --- 0 AODV 48 [0 ffffffff 17 800] [energy 136.104283] ----- [23:255 -1:255 30 0] [0x2 1 1 [24 0] [23 4]] (REQUEST)
r 8.896141864 _69_ RTR --- 0 AODV 48 [0 ffffffff 17 800] [energy 136.104283] ----- [23:255 -1:255 30 0] [0x2 1 1 [24 0] [23 4]] (REQUEST)
r 8.896141894 _47_ RTR --- 0 AODV 48 [0 ffffffff 17 800] [energy 136.104283] ----- [23:255 -1:255 30 0] [0x2 1 1 [24 0] [23 4]] (REQUEST)
D 40.324386198 _28_ RTR NRTE 40 cbr 532 [0 0 0 0] [energy 104.749189] ----- [28:1 29:1 30 0] [6] 0 2
D 40.324386198 _28_ RTR NRTE 45 cbr 532 [0 0 0 0] [energy 104.749189] ----- [28:1 29:1 30 0] [7] 0 2
D 40.324386198 _28_ RTR NRTE 46 cbr 532 [0 0 0 0] [energy 104.749189] ----- [28:1 29:1 30 0] [8] 0 2
s 124.737067862 _30_ AGT --- 931 cbr 512 [0 0 0 0] [energy 20.347418] ----- [30:1 31:0 32

0] [59] 0 3

s 124.739450131 _28_ AGT --- 932 cbr 512 [0 0 0 0] [energy 20.259803] ----- [28:1 29:1 32

0] [103] 0 1

s 124.806955087 _12_ AGT --- 933 cbr 512 [0 0 0 0] [energy 20.239102] ----- [12:0 13:0 32

0] [76] 0 1

s 124.810252628 _28_ AGT --- 934 cbr 512 [0 0 0 0] [energy 20.189815] ----- [28:2 30:0 32

0] [19] 0 1

r 86.370060773 _7_ AGT --- 658 cbr 532 [13a 7 2 800] [energy 58.629802] ----- [6:1 7:0 27

7] [33] 4 4

r 86.393997734 _13_ AGT --- 659 cbr 532 [13a d 4 800] [energy 58.605865] ----- [11:2 13:0

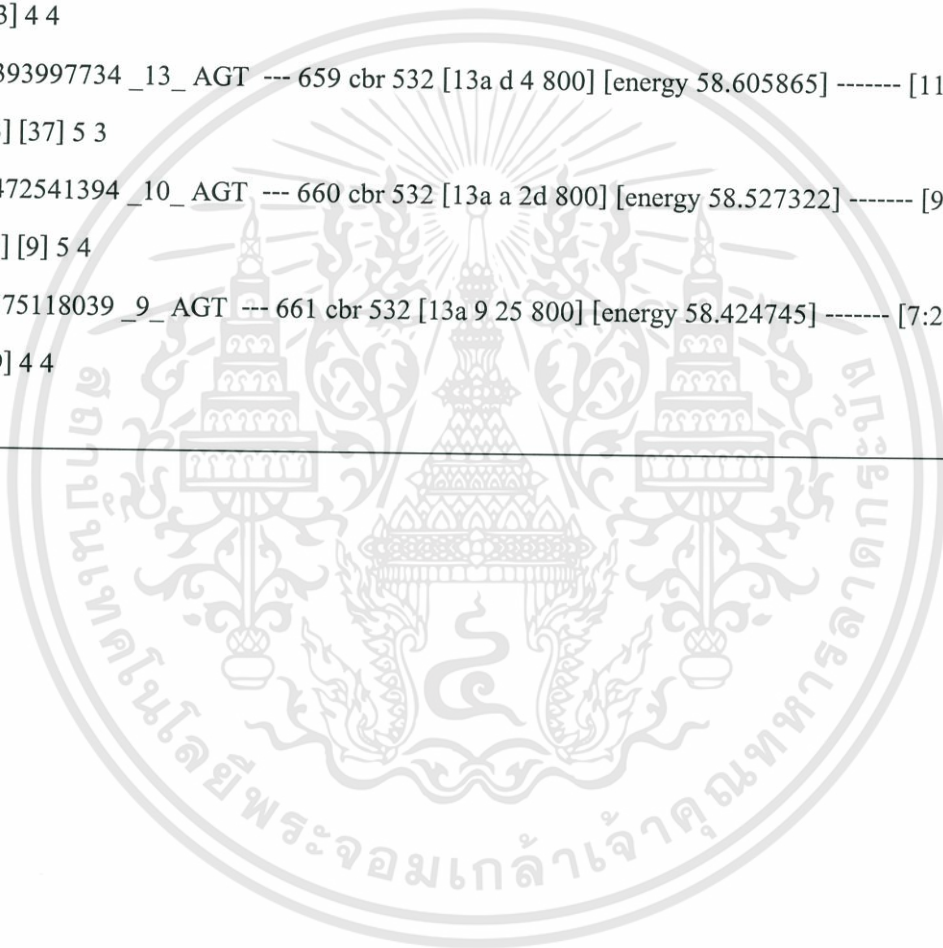
26 13] [37] 5 3

r 86.472541394 _10_ AGT --- 660 cbr 532 [13a a 2d 800] [energy 58.527322] ----- [9:2 10:0

26 10] [9] 5 4

r 86.575118039 _9_ AGT --- 661 cbr 532 [13a 9 25 800] [energy 58.424745] ----- [7:2 9:0 27

9] [79] 4 4



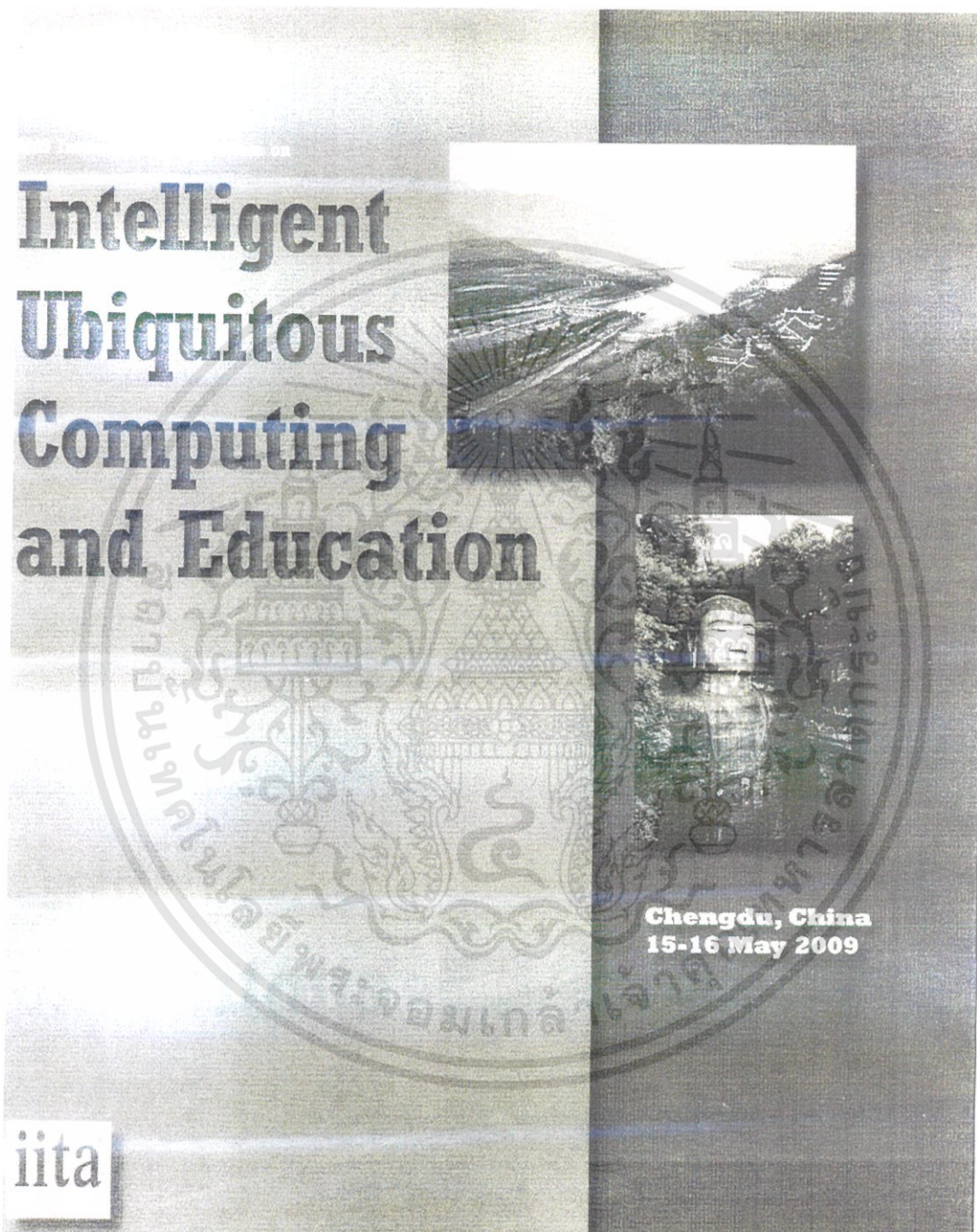
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข.

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. H.Somnuk and M.Lertwatechakul, “Multi-hop AODV-2T”, International Symposium on Intelligent Ubiquitous Computing and Education 2009, May 2009



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**IEEE Computer Society
Conference Publications
Operations Committee**



CPOC Chair
Roy Sterritt
University of Ulster

Board Members

Mike Hinchey, *Co-Director, Lero-the Irish Software Engineering Research Centre*
Larry A. Bergman, *Manager, Mission Computing and Autonomy Systems Research Program Office (982), JPL*
Wenping Wang, *Associate Professor, University of Hong Kong*
Silvia Ceballos, *Supervisor, Conference Publishing Services*
Andrea Thibault-Sanchez, *CPS Quotes and Acquisitions Specialist*

IEEE Computer Society Executive Staff

Evan Butterfield, *Director of Products and Services*
Alicia Stickley, *Senior Manager, Publishing Services*
Thomas Baldwin, *Senior Manager, Meetings & Conferences*

IEEE Computer Society Publications

The world-renowned IEEE Computer Society publishes, promotes, and distributes a wide variety of authoritative computer science and engineering texts. These books are available from most retail outlets. Visit the CS Store at <http://www.computer.org/portal/site/store/index.jsp> for a list of products.

IEEE Computer Society Conference Publishing Services (CPS)

The IEEE Computer Society produces conference publications for more than 250 acclaimed international conferences each year in a variety of formats, including books, CD-ROMs, USB Drives, and on-line publications. For information about the IEEE Computer Society's Conference Publishing Services (CPS), please e-mail: cps@computer.org or telephone +1-714-821-8380, Fax +1-714-761-1784. Additional information about Conference Publishing Services (CPS) can be accessed from our web site at: <http://www.computer.org/cps>

Revised: 1 March 2009



CPS Online is our innovative online collaborative conference publishing system designed to speed the delivery of price quotations and provide conferences with real-time access to all of a project's publication materials during production, including the final papers. The *CPS Online* workspace gives a conference the opportunity to upload files through any Web browser, check status and scheduling on their project, make changes to the Table of Contents and Front Matter, approve editorial changes and proofs, and communicate with their CPS editor through discussion forums, chat tools, commenting tools and e-mail.

The following is the URL link to the *CPS Online* Publishing Inquiry Form:
http://www.ieeeconfpublishing.org/cpir/inquiry/cps_inquiry.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Multi-hop AODV-2T

Harisavan Somnuk
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
e-mail: harisavan@hotmail.com

Mayuree Lertwatechakul
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
e-mail: klmayuree@kmitl.ac.th

Abstract— Communication in Ad Hoc wireless network may generally be uncertain because the established route could be broken anytime. The major cause of link failures in Ad hoc wireless network is the dynamic movement of the mobile nodes themselves and the other might be battery running out. AODV-2T [9] is an improved AODV designed to compromise this problem by applying a conservative and circumspect concept. By investigating received RF signal and the battery power, AODV-2T could overcome the problem by preparing a backup route just-in-time before link breaking. This makes it is able to switch packet transmission to the backup route without any delay with a bit of additive overhead.

Although AODV-2T performs rather well compared to the early proposed protocols [4, 11, 12, 14, 19] in term of overhead and number of route break but AODV-2T is not capable to construct a multi-hop backup route. On the assumption that Multi-hop backup route establishment might be more flexible and gives more chance of success in Ad hoc route maintenance, so we have extended AODV-2T to support multi-hop backup route. The improved protocol called Multi-hop AODV-2T was implemented into a NS-2 model. We simulated Multi-hop AODV-2T function and compared its route maintenance mechanism's performance with AODV and AODV-2T. The results have shown that Multi-hop AODV can reduce number of route break and packet loss obviously.

Keywords- Ad hoc routing, AODV, local repair, Multi-hop, backup route.

I. INTRODUCTION

The main feature of the mobile nodes is its easiness to be used by human or other devices anywhere including on vehicles. That means the mobile nodes could be moved slowly or quickly depended on what and how they are handled. The uncertain location of the mobile node effects to both of the wireless network topology and the routing mechanism design. Due to this problem, many Ad hoc routing protocols [1, 2, 3, 5, 6, 13, 16, 18, 20] was developed to construct an Ad hoc route for a pair of mobile nodes. Some of these protocols prepare Ad hoc routes before hand and called as "proactive" [1, 3, 5] while the others construct routes "on demand" [2, 6, 13, 16, 18]. The proactive type overwhelms mobile node compared to the on demand type.

And this makes on demand type plays important role in Ad hoc networking.

AODV-2T [9] is one of the recently proposed protocols which have improved AODV (Ad hoc On-Demand Distance Vector) which is one of the most popular Ad hoc wireless routing protocol. AODV-2T provides route maintenance strategy on when and how to find a backup route. The concept of AODV-2T is to investigate neighbors' transmission signal power and its own node's battery power regularly. In case that the investigating factor is dropping down to the first level of the specified threshold value, AODV-2T will begin to seek and establish a backup route for the affected communication. After the backup route was constructed successfully AODV-2T will decide to use the backup route at once just before the existing route could be broken. And this route switching is triggered when the second level threshold is met.

Applying 2 level thresholds concept with subroute recovery, AODV-2T showed its performance on reducing the broken route and this also improves the throughput of the Ad hoc network either. Whereas AODV-2T works in circumspect and conservative fashion, the overhead of recovering broken routes is considered under control.

Anyway, AODV-2T is not capable to recover a backup route that may need a Multi-hop subroute to replace of the failed link. Thus, if we improve backup route establishing function of AODV-2T to support Multi-hop backup route, the improved protocol should enhance the Ad hoc networking performance both in term of number of route break decrement which in turn increases the network's throughput.

In this paper, we propose our idea to extend the mechanism of AODV-2T to support Multi-hop backup route. The proposed mechanism performance is shown in the simulation results. And we organize this paper's content by providing the background of Ad hoc wireless network and Ad hoc route maintenance in section 2. We illustrate our proposed idea of Multi-hop AODV-2T in section 3. The simulation scenario and the results are shown in section 4 finally with the conclusion in section 5.

II. AD HOC WIRELESS NETWORK

Routing protocols designed for Ad hoc wireless network could be classified into 2 groups: proactive type and reactive

type (on-demand). Proactive protocol such as DSDV, WRP discovers and prepares ready-for-use route for every current active mobile node. In contrast, reactive protocol such as DSR, AODV seeks a best route and establishes the route just only when it is requested for.

As a result, proactive routing makes an Ad hoc network much fast and much stable compared to reactive protocol. However, the most disadvantage of proactive protocol is to generate a lot of overhead and to burden every mobile node for routing function. The problem consumes much of mobile node's resources and this is suffering and unacceptable for end users. That is the reason why reactive protocol type is more preferable to be used in Ad hoc network.

A. Ad hoc Route Maintenance

It is possible to have a link break during data transmission in Ad hoc wireless network. Thus, route maintenance is considered as an importance function of Ad hoc routing protocol and many research works have proposed for this issue. [2, 4, 9, 11, 12, 14, 15, 16, 17, 18, 19] We consider algorithms [9, 19] those seem appropriated to the AODV network problem. As to give a brief on the concepts, we illustrate the procedure of route maintenance in AODV and AODV-2T in the following subsection.

1) AODV

Local repair is the mechanism of AODV to fix the Ad hoc route break problem. Local repair performs different action depended on the location of link break reference to the communication node. The mechanism could be described as the following.

Near to the destination In this case, triggered node broadcast RREQ downstream to the destination node. If RREQ could reach to the destination, the destination will response to this event by send back a RREP as same as in route setup process. During the local repair process, the triggered node has to buffer all the packets of the affected connections. Immediately after the triggered node receives the corresponding RREP, the damaged route will be reconstructed and the data transmission will be resumed. However, in case that the triggered node does not receive RREP within a timeout period, the local pair process is fail. And the node will response by sending a Route Error Packet (RERR) contained with the unreachable destination address back to notify the source node. As a consequence, the source node broadcasts a RREQ to recover a new route.

Near to the source If the triggered node is close to the source node, the node sends a RERR that contains address of the unreachable destination to the source node. And the source node will do the same process as same as in the case of unsuccessful local repairing.

2) AODV-2T

AODV-2T has improved route maintenance in Ad hoc network with "Just in time backup route" concept. The strategy is to construct a backup route at a proper time. AODV-2T utilizes 2 threshold values of each 2 factors: received signal power and node's battery power. The first level of threshold is set to an appropriate value that could be

used to trigger the corresponding node to recovery a backup route at the right time. The second level of threshold is set to the critical level of the mentioned factors as to decide when AODV-2T should switch all affected communication flows to the backup route. The gap between 2 level thresholds is importance and could effect to the performance of the protocol, thus it must be carefully calculated. In addition, AODV-2T finds a backup route by recovering a partial route between the triggered node and the closest downstream node of the effected route. In contrast to AODV local repair, AODV-2T does not need the source node or the destination node to be involved in every case. The source node or destination node may be involved just only the case they are 1-2 hops far from the triggered node.

By applying the explained idea, AODV-2T constructs a partial route as a backup route at once it senses a risk of link break. The strategy gives a fast response to link break event while reduces unnecessary workload for mobile node as well.

III. MULTI-HOP AODV-2T

The objective of Multi-hop AODV-2T is to enhance AODV-2T performance by extending its ability of creating a backup route to support Multi-hop partial route recovery. The approach is based on local repair mechanism of AODV-2T that has improved route maintenance in Ad hoc network by preparing backup route in a proper time. The mechanism works quite well in decreasing number of route breaks and then gains more network throughput. Thus, we have applied the same local repair mechanism with some modification by adjusting AODV-2T's *recv_sq* (Receiving Subroute Request) process. The improved procedure could be described as the following.

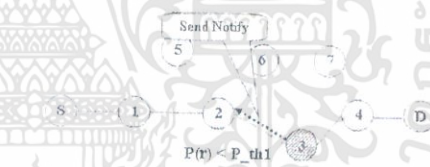


Figure 1 Multi-hop AODV-2T sends a notify in case received signal is weak

Step 1 At the time a mobile node detects an opportunity of a link break where the received signal is dropping to the first level of the defined threshold (or its energy is decreasing to the first level threshold of battery power) The node immediately sends a Notify packet to the upstream neighbor node of the affected route. A Notify packet is sent for every affected existing route. Each Notify packet contains the most essential information to construct a backup route that is addresses of source node and destination node of the affected route.



Figure 2. Triggered node (2) sends SUB_RREQ and each subsequent nodes (6), (7) checks its forwarding table, decreases TTL and forward SUB_RREQ if it does not have an available route to destination

Step 2 After node (2) is triggered by a Notify packet then broadcasts a SUB_RREQ (Subroute Request) packet to all its neighbors as described in Fig. 2. The SUB_RREQ structure is composed of SUB_RREQ sending node address, the source address and the destination address of the damaged route, sequence number of SUB_RREQ and TTL (Time To Live).



Figure 3. RREP is sent back to the triggered node (2) as to confirm the backup route.

Step 3 Any node which receives the SUB_RREQ packet, looks up its own forwarding table whether if the forwarding table contains the routing information for the specified destination address or not. In case that its routing entries has an available route to the destination, it responds back to the previous node with SUB_RREP (Subroute Reply) packet as in Fig. 3. Or else the node decreases TTL and continues to broadcast the received SUB_RREQ packet if the value of decreased TTL field is not equal to zero. (Fig. 2)

Failure case Fig. 4-5 show the failure case of backup route discovering process. After trigger node (2) delivered SUB_RREQ and no RREP is found within the defined timeout, the triggered node (2) sends a RERR to the source node S.



Figure 4. SUB_RREQ can not reach active node (3), (4) or (D)

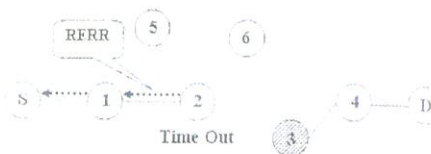


Figure 5. Triggered node tell source node to establish a new route by sending a RERR

IV. SIMULATION RESULTS

In order to investigate the performance of Multi-hop AODV-2T, we simulate the protocol and compare the result with the original AODV and AODV-2T. All simulation scenarios were set as being show in table 1. And the results are measured in term of number of route break, network throughput and number of control packet.

The simulation results could show that Multi-hop AODV-2T gain more performance when compared to AODV and AODV-2T. All figures' Y axis represents speed of the mobile nodes' movement. Fig. 7 shows that, the movement speed effects directly to the frequency of route break. As a consequence, number of route break could increase number of packet lost and then reduce the network throughput.

All the results confirm Multi-hop AODV-2T concept. Multi-hop AODV reduces number of route break and gain more Ad hoc network performance. Especially for high speed movement scenario, the result of Multi-hop AODV-2T is much distinguishable to AODV and AODV-2T.

TABLE 1. SIMULATION PARAMETERS

Number of Node		100
Topology size (m. x m.)		500x500
Velocity (m/s)		20
Number of Connection		20
Traffic Type		CBR
Simulation Time		150 s
1 st Threshold of Radio		8.0c-10W
2 nd Threshold of Radio		4.0c-10W
1 st Threshold of battery		50J
2 nd Threshold of battery		10J

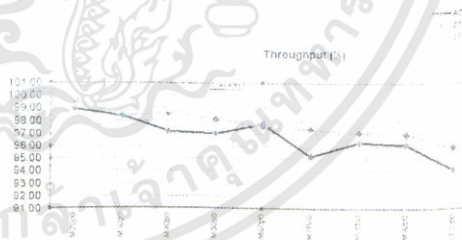


Figure 6. Ac hoc Network Throughput

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

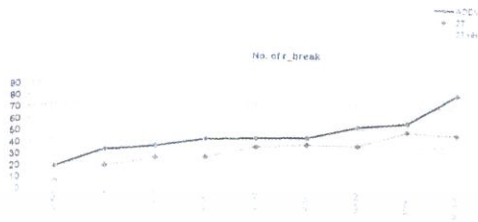


Figure 7 Number of route break

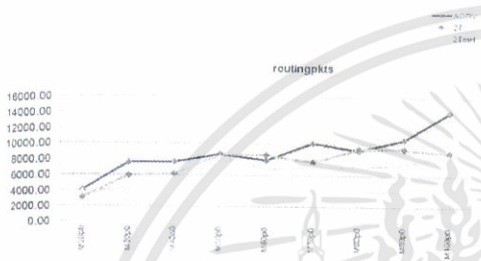


Figure 8 Overhead of routing protocol

V. CONCLUSION

Distinct Ad hoc wireless network nature is its dynamic topology that could be changed frequently. As a consequence Ad hoc network needs a good route maintenance algorithm. We have developed Multi-hop AODV-2T to be capable of discovering a multi-hop partial route. This route maintenance concept inherits most the features of AODV-2T that is proved as a practical idea to be used in reactive Ad hoc network. With its conservative and circumspect principle, Multi-hop AODV-2T gives a better performance compared to either the original AODV or AODV-2T.

REFERENCES

[1] Perkins, C. E. and Bhargava, P. "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers". ACM SIGCOMM 1991, pp. 231-241, August 1991.

[2] Perkins, C. E. and Royer, E. M. "Ad Hoc On-Demand Distance Vector Routing". IETF Workshop on Mobile Computing System and Applications 1999 (February 1999), 90-100.

[3] Murthy, S. and Garcia-Luna-Aceves, J. J. "An Efficient Routing Protocol for Wireless Network". ACG Mobile Network and Application Journal, Vol. 1 (October 1996), 183-197.

[4] Chiang, C. C., Wu, H. K., Liu, W. and Gerla, M. "Routing in Clustered Multi-hop Mobile Wireless Network with Fading Channel". IEEE SICOM 1997 (April 1997), 197-211.

[5] Garcia-Luna-Aceves, J. J. and Spohn, M. "Source Tree Routing in Wireless Network". IEEE ICNP 1999 (October 1999), 273-282.

[6] Clausen, T. H., Hansen, G., Christensen, L., and Behrmann, G. "The Optimized Link State Routing Protocol. Evaluation Through

Experiments and Simulation. IEEE Symposium on Wireless Personal Mobile Communication 2001 (September 2001).

[7] Iwata, A., Chang, C. C., Pei, G., Gerla, M., and Chen, T. W. "Scalable Routing Strategies for Ad Hoc Wireless Network". IEEE Journal on Selected Areas in Communication, Vol. 17 (August 1999), 1369-1379.

[8] Johnson, D. B. and Maltz, D. A. "Dynamic Source Routing in Ad Hoc Wireless Network". Mobile Computing Kluwer Academic Publisher, Vol.353 (November 1996), 153-181.

[9] Akkaphop, O., Mayuree, L. "An Improvement of Ad Hoc Network Route Maintenance". Proceeding of International Symposium on Communication and Information Technologies 2008, (Vientiane, Lao PDR, October 21-23, 2008), 102-105.

[10] Dube, R., Rais, C. D., Wang, K. Y., and Tripathi, S. K. "Signal Stability-Based Adaptive Routing for Ad Hoc Mobile Network". IEEE Personal Communication Magazine, (February 1997), 36-45.

[11] Su, W. and Gerla, M. "IPv6 Flow Handoff in Ad Hoc Wireless Networks Using Mobility Prediction". IEEE GLOBECOM 1999 (December 1999), 271-275.

[12] Sisodia, R. S., Manoj, B. S., and Siva Ram Murthy, C. "A preferred Link-Based Routing Protocol for Ad Hoc Wireless Network". Journal of Communications and Networks, Vol. 4, no. 1 (March 2002), 14-21.

[13] Sinha, P., Sivakumar, R., and Bhargavan, V. "CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm". IEEE Journal on selected Areas in Communication, Vol 17, no. 8 (August 1999), 1454-1466.

[14] Haas, Z. J. "The Routing Algorithm for the Reconfigurable Wireless Networks". ICUPC 1997, Vol. 2 (October 1997), 562-566.

[15] Joe-Ng, M. and Lu, I. T. "A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks". IEEE Journal on selected Areas in Communication, Vol 17, no. 8 (August 1999), 1415-1425.

[16] Manoj, B. S., Ananthapadmanabha, R., and C. Shiva Ram Murthy, "Link Life Based Routing Protocol for Ad Hoc Wireless Networks". IEEE ICCCN 2001 (October 2001), 573-576.

[17] Singh, S., Woo, M., and Raghavendra, C. S. "Power-Aware Routing in Mobile Ad Hoc Network". ACM MOBICOM 1998 (October 1998), 181-190.

[18] Ko, Y., and Vaidya, N. H. "Location-Aided Routing (LAR) in Mobile Ad Hoc Network". ACM MOBICOM 1998 (October 1998), 66-75.

[19] Tsuchi, M., Ideguchi, T. and Murata, Y. "Ad Hoc Routing Protocol Avoiding Route Break Based on AODV". IEEE Personal Communication (January 2005), 322-329.

[20] MANET "Mobile Ad Hoc Network". DOI = <http://www.ietf.org/html/charters/manet-charter.html>

[21] Broch, J. et al. "A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols". Proc. IEEE/ACM MOBICOM '98 (October 1998), 85-97.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้