



การปรับปรุงโพรโทคอลเอโอดีวีสำหรับเครือข่ายเซนเซอร์ไร้สาย  
**Modified AODV Protocol for Wireless Sensor Networks**

ภาคภูมิ มโนยุทธ  
**Parkpoom Manoyut**

วิทยานิพนธ์นี้สำหรับการศึกษิตามหลักสูตรปริญญา  
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์  
มหาวิทยาลัยสงขลานครินทร์

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

ปี พ.ศ. 2555

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์      การปรับปรุงโพรโทคอลเอไอดีวีสำหรับเครือข่ายเซนเซอร์ไร้สาย  
ผู้เขียน              นายภาณุภูมิ มโนยุทธ  
สาขาวิชา            วิศวกรรมคอมพิวเตอร์

---

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

คณะกรรมการสอบ

.....

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต) (ดร.ศกุนา เจริญปัญญาศักดิ์)

.....กรรมการ

(ดร.เดือนเพ็ญ กษกรจารุพงศ์)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้รับวิทยานิพนธ์ฉบับนี้  
สำหรับการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

.....

(ศาสตราจารย์ ดร.อมรรัตน์ พงศ์ดารา)

คณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้เป็นผลมาจากการศึกษาวิจัยของนักศึกษาเอง และ  
ขอขอบคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านไว้ ณ ที่นี้

ลงชื่อ \_\_\_\_\_

( ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต )

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ลงชื่อ \_\_\_\_\_

( นายภาคภูมิ มโนยุทธ )

นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน และไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ \_\_\_\_\_

(นายภาคภูมิ มโนยุทธ)

นักศึกษา

|                 |  |
|-----------------|--|
| ชื่อวิทยานิพนธ์ | การปรับปรุงโพรโทคอลเอโอดีวีสำหรับเครือข่ายเซนเซอร์ไร้สาย |
| ผู้เขียน        | นายภาณุภูมิ มโนยุทธ                                      |
| สาขาวิชา        | วิศวกรรมคอมพิวเตอร์                                      |
| ปีการศึกษา      | 2555   |

### บทคัดย่อ

ในปัจจุบันเครือข่ายเซนเซอร์ไร้สายถูกนำมาประยุกต์ใช้งานกันเป็นจำนวนมาก ได้แก่ การเฝ้าตรวจวัดทางกายภาพในงานด้านการเกษตร และการเฝ้าระวังสุขภาพในงานทางการแพทย์ เป็นต้น แต่เนื่องด้วยข้อจำกัดด้านพลังงานของโหนดซึ่งเป็นอุปกรณ์หลักที่ใช้ในเครือข่าย ทำให้จำเป็นที่จะต้องมีการพัฒนาโพรโทคอลให้ใช้พลังงานอย่างมีประสิทธิภาพ หรือลดจำนวนแพ็กเกจในการสื่อสารลง เพื่อให้ระบบเครือข่ายสามารถทำงานได้นานขึ้น

งานวิจัยนี้จึงได้นำเสนอวิธีการลดจำนวนแพ็กเกจของขั้นตอนการซ่อมแซมเส้นทางในโพรโทคอลค้นหาเส้นทางเอโอดีวี (Ad-hoc On-demand distance vector, AODV) ที่ซึ่งได้รับความนิยมนำมาใช้งานในเครือข่ายเซนเซอร์ไร้สาย โพรโทคอลเอโอดีวีที่ปรับปรุงใหม่นี้ถูกทดสอบการทำงานด้วยโปรแกรมจำลองเครือข่าย Network Simulation 2 (NS-2) และกำหนดให้รูปแบบของเครือข่ายที่ใช้ทดสอบเป็นแบบเคลื่อนที่ได้ จากผลการทดลองพบว่าโพรโทคอลเอโอดีวีใหม่นี้สามารถลดจำนวนแพ็กเกจได้ 17 เปอร์เซ็นต์เมื่อเทียบกับโพรโทคอลเอโอดีวีแบบดั้งเดิม นอกจากนี้โพรโทคอลเอโอดีวีที่ปรับปรุงใหม่นี้ยังช่วยลดค่าหน่วยเวลาของการค้นหาเส้นทางได้สูงถึง 57 เปอร์เซ็นต์

**Thesis Title**            Modified AODV Protocol for Wireless Sensor Networks  
**Author**                    Mr. Parkpoom Manoyut  
**Major Program**        Computer Engineering  
**Academic Year**        2012

### **ABSTRACT**

Wireless sensor networks have been widely deployed in many applications such as physical monitoring in an agricultural system and health care monitoring in a medical system and so on. However, a node which is the major equipment in wireless sensor networks has a power limit. Therefore, most research works in wireless sensor networks are energy efficient network protocol development or package transmission reduction in order to prolong the network life-time.

In this thesis, we propose the method to reduce the number of packet transmission in the local repair of Ad-hoc On-demand distance vector (AODV) protocol which is widely used in wireless sensor networks. Our developed AODV protocol has been simulated using Network Simulation 2 (NS-2). The networks having mobile nodes have been used to experiment our protocol. From the experimental results, we can reduce the number of packet transmission at about 17% compared to the original AODV. In addition, the results also report the routing latency improvement by 57%.

## สารบัญ

หน้า

|   |      |
|---|------|
| บทคัดย่อ.....   | (6)  |
| กิตติกรรมประกาศ.....  | (8)  |
| สารบัญ.....   | (9)  |
| รายการตาราง .....   | (11) |
| รายการภาพประกอบ.....  | (12) |
| บทที่ 1 บทนำ .....  | 1    |
| 1.1 ความสำคัญและที่มาของวิทยานิพนธ์.....                                | 1    |
| 1.2 การตรวจเอกสาร .....   | 2    |
| 1.3 วัตถุประสงค์ของวิทยานิพนธ์.....                                     | 3    |
| 1.4 ขอบเขตการวิจัย.....   | 3    |
| 1.5 ขั้นตอนและวิธีดำเนินงานวิจัย.....                                   | 4    |
| 1.6 ประโยชน์ที่คาดว่าจะได้รับ .....                                     | 4    |
| 1.7 อุปกรณ์และสถานที่วิจัย .....  | 4    |
| บทที่ 2 ทฤษฎีและหลักการ .....   | 5    |
| 2.1 แนะนำเครือข่ายเซนเซอร์ไร้สาย .....                                  | 5    |
| 2.2 โพรโทคอล AODV .....   | 5    |
| 2.3 การตรวจสอบสถานะของการเชื่อมต่อ.....                                 | 23   |
| บทที่ 3 การพัฒนาโพรโทคอล AODV .....                                     | 24   |
| 3.1 การพัฒนากระบวนการซ่อมแซมเส้นทาง Local repair ในโพรโทคอล AODV.....   | 24   |
| 3.2 การปรับปรุงเส้นทางให้มีระยะทางที่สั้นกว่าเส้นทางที่ใช้งานเดิม ..... | 31   |

## สารบัญ (ต่อ)

หน้า

|  |    |
|--|----|
| บทที่ 4 ผลการทดสอบ .....   | 38 |
| 4.1 การหาค่าจำนวน โหนดที่เหมาะสมต่อพื้นที่ที่ทำการทดสอบ .....  | 39 |
| 4.2 การหาค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบ โหนดข้างเคียงที่เหมาะสม .....   | 43 |
| 4.3 การเปรียบเทียบโพรโทคอล AODV แบบดั้งเดิมกับโพรโทคอล AODV ที่พัฒนาการ<br>ซ่อมแซมเส้นทางแบบ Local repair .....                                    | 47 |
| 4.4 การเปรียบเทียบโพรโทคอล AODV แบบดั้งเดิมกับโพรโทคอล AODV ที่พัฒนาการ<br>ซ่อมแซมเส้นทางแบบ Local repair และการเลือกใช้เส้นทางที่สั้นที่สุด ..... | 50 |
| บทที่ 5 บทสรุป ปัญหาและข้อเสนอแนะ .....  | 52 |
| 5.1 บทนำ.....  | 52 |
| 5.2 บทสรุปของการทำวิทยานิพนธ์.....   | 52 |
| 5.3 ปัญหาและอุปสรรคของการทำวิทยานิพนธ์ .....   | 53 |
| 5.4 ข้อเสนอแนะ.....  | 53 |
| บรรณานุกรม.....  | 54 |
| อภิธานศัพท์.....   | 56 |
| ภาคผนวก .....  | 58 |
| ภาคผนวก ก วิธีการติดตั้ง NS-2 .....  | 59 |
| ภาคผนวก ข การใช้งาน NS-2 เพื่อจำลองการทำงานของโพรโทคอล AODV เบื้องต้น .....  | 62 |
| ภาคผนวก ค ผลการทดสอบระบบเพิ่มเติม.....   | 69 |
| ภาคผนวก ง การตีพิมพ์เผยแพร่วิทยานิพนธ์ .....   | 72 |
| ประวัติผู้เขียน .....  | 87 |



## รายการตาราง

|              |   | หน้า |
|--------------|---|------|
| ตารางที่ 2-1 | รายละเอียดการจัดเก็บข้อมูลภายในตารางเส้นทาง .....   | 7    |
| ตารางที่ 2-2 | รายละเอียดแต่ละส่วนภายในข้อความ RREQ .....  | 8    |
| ตารางที่ 2-3 | รายละเอียดแต่ละส่วนภายในข้อความ RREP .....  | 9    |
| ตารางที่ 2-4 | รายละเอียดแต่ละส่วนภายในข้อความ RRER .....  | 11   |
| ตารางที่ 4-1 | พารามิเตอร์พื้นฐานเพื่อใช้ในการหาค่าจำนวนโหนดที่เหมาะสมต่อพื้นที่ทดสอบ 39   |      |
| ตารางที่ 4-2 | พารามิเตอร์พื้นฐานในการหาค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนด<br>ข้างเคียงที่เหมาะสม .....   | 43   |
| ตารางที่ 4-3 | พารามิเตอร์พื้นฐานในการเปรียบเทียบโปรโตคอล AODV แบบดั้งเดิมกับโปรโต<br>คอล AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair .....   | 47   |
| ตารางที่ 4-4 | แสดงการเปรียบเทียบผลที่ได้จากการทดสอบการทำงานของโปรโตคอล AODV<br>แบบดั้งเดิมกับ AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair.....   | 48   |
| ตารางที่ 4-5 | แสดงการเปรียบเทียบผลที่ได้จากการทดสอบการทำงานของโปรโตคอล AODV<br>แบบดั้งเดิมกับ AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair และการ<br>เลือกใช้เส้นทางที่สั้นที่สุด ..... | 50   |
| ตารางที่ ก-1 | อภิธานศัพท์ .....   | 56   |
| ตารางที่ ข-1 | พารามิเตอร์พื้นฐานที่มีการใช้งานในระบบ .....  | 63   |
| ตารางที่ ค-1 | แสดงผลที่ได้จากการทดสอบการทำงานของโปรโตคอล AODV แบบดั้งเดิมกับ<br>AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair และการเลือกใช้เส้นทางที่<br>สั้นที่สุด .....               | 70   |

## รายการภาพประกอบ

หน้า

|                |  |    |
|----------------|--|----|
| ภาพประกอบ 2-1  | ลักษณะของข้อความ RREQ .....  | 8  |
| ภาพประกอบ 2-2  | ลักษณะของข้อความ RREP .....  | 9  |
| ภาพประกอบ 2-3  | ลักษณะของข้อความ RRER.....   | 11 |
| ภาพประกอบ 2-4  | แสดงหลักการทำงานพื้นฐานของโพรโทคอล AODV.....   | 12 |
| ภาพประกอบ 3-1  | ลักษณะการทำงานเมื่อเกิดการเชื่อมต่อเสียหาย .....   | 25 |
| ภาพประกอบ 3-2  | ลักษณะของข้อความตรวจสอบ โหนดข้างเคียง .....  | 26 |
| ภาพประกอบ 3-3  | ตัวอย่างของการทำงานเมื่อโหนดได้รับข้อความตรวจสอบ โหนดข้างเคียงจาก<br>โหนดข้างเคียง.....              | 26 |
| ภาพประกอบ 3-4  | ลักษณะของข้อความ RREP .....  | 27 |
| ภาพประกอบ 3-5  | ลักษณะการทำงานเมื่อมีการส่งข้อความ RREP.....   | 28 |
| ภาพประกอบ 3-6  | ลักษณะของข้อความ RREQ .....  | 29 |
| ภาพประกอบ 3-7  | ลักษณะการทำงานข้อความ RREQ แบบปลายทางเดียว .....   | 30 |
| ภาพประกอบ 3-8  | รูปแบบการเชื่อมต่อตอนเริ่มต้นของระบบ.....  | 31 |
| ภาพประกอบ 3-9  | การเคลื่อนที่ของโหนดภายในเส้นทางที่ใช้งาน ได้มีทิศทางไปยังโหนดต้นทาง                                 | 32 |
| ภาพประกอบ 3-10 | การขอเส้นทางเมื่อโหนดภายในเส้นทางที่ใช้งาน ได้มีทิศทางไปยังโหนดต้น<br>ทาง.....                       | 33 |
| ภาพประกอบ 3-11 | รูปแบบการเชื่อมต่อของเส้นทางที่ใช้งานได้.....  | 33 |
| ภาพประกอบ 3-12 | โหนดใหม่เคลื่อนที่มาอยู่ระหว่างโหนดปลายทางกับเส้นทางที่ใช้งานได้ .....                               | 34 |
| ภาพประกอบ 3-13 | การร้องขอข้อมูลเส้นทางผ่านทางโหนดใหม่ที่เข้ามาอยู่ระหว่างโหนด<br>ปลายทางและเส้นทางที่ใช้งานได้ ..... | 35 |
| ภาพประกอบ 3-14 | รูปแบบการเชื่อมต่อตั้งต้นก่อนที่โหนดต้นทางเคลื่อนที่ .....   | 36 |
| ภาพประกอบ 3-15 | การเคลื่อนที่ของโหนดต้นทางไปในทิศทางเส้นทางที่ใช้งานได้ .....  | 36 |
| ภาพประกอบ 3-16 | การร้องขอข้อมูลเส้นทางใหม่ .....   | 37 |
| ภาพประกอบ 4-1  | กราฟความสัมพันธ์ระหว่างอัตราความสำเร็จในการส่งข้อมูลกับจำนวนโหนด<br>ในเครือข่าย.....                 | 40 |

## รายการภาพประกอบ (ต่อ)

หน้า

|               |  |    |
|---------------|--|----|
| ภาพประกอบ 4-2 | กราฟความสัมพันธ์ระหว่างกำหนดเวลากับจำนวนโหนดในเครือข่าย.....   | 41 |
| ภาพประกอบ 4-3 | กราฟความสัมพันธ์ระหว่างค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลกับจำนวนโหนดในเครือข่าย.....                                 | 42 |
| ภาพประกอบ 4-4 | กราฟความสัมพันธ์ระหว่างอัตราความสำเร็จในการส่งข้อมูลกับค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียง.....                   | 44 |
| ภาพประกอบ 4-5 | กราฟความสัมพันธ์ระหว่างกำหนดเวลากับค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียง.....                                       | 45 |
| ภาพประกอบ 4-6 | กราฟความสัมพันธ์ระหว่างค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลกับค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียง..... | 46 |
| ภาพประกอบ ข-1 | ขั้นตอนการส่งข้อความตรวจสอบโหนดข้างเคียงแบบแพร่กระจาย.....   | 64 |
| ภาพประกอบ ข-2 | การส่งข้อความ RREQ จาก node_(0).....   | 65 |
| ภาพประกอบ ข-3 | การส่งข้อความ RREQ จาก node_(1).....   | 66 |
| ภาพประกอบ ข-4 | การส่งข้อความ RREQ จาก node_(2).....   | 67 |
| ภาพประกอบ ข-5 | การรับ-ส่งข้อความร้องขอ ARP และตอบกลับ ARP.....  | 68 |

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของวิทยานิพนธ์

เครือข่ายเซนเซอร์ไร้สาย [1,2] หมายถึงระบบเครือข่ายที่ประกอบด้วยอุปกรณ์ขนาดเล็กที่เรียกว่า “โหนด” จำนวนมาก โดยที่โหนดมีราคาไม่แพงและมีความสามารถติดต่อสื่อสารระหว่างกันแบบไร้สายในรูปแบบของ Multi-hop<sup>1</sup> เพื่อช่วยเพิ่มระยะทางการส่งข้อมูล ส่งผลให้ในปัจจุบันมีการนำเครือข่ายเซนเซอร์ไร้สายไปประยุกต์ใช้งานในหลายด้าน [3,4] ตัวอย่างเช่น การตรวจสอบสภาพภูมิอากาศและลักษณะทางกายภาพของพืช การตรวจสอบสภาพภูมิประเทศทางการทหาร และการเฝ้าระวังผู้ป่วยในโรงพยาบาล เป็นต้น

โหนดประกอบด้วยการทำงานของส่วนต่างๆ ดังนี้ 1) ส่วนของการรับ-ส่งสัญญาณคลื่นวิทยุบนมาตรฐาน IEEE 802.15.4 ที่กำหนดให้ใช้คลื่นในย่านความถี่ 2.4 GHz 2) ส่วนของตัวตรวจวัดหรือเรียกว่าเซนเซอร์ ทำหน้าที่ในการวัดค่าทางกายภาพต่างๆเช่น ตัวตรวจวัดอุณหภูมิในอากาศ ตัวตรวจวัดความเข้มแสง และตัวตรวจวัดความชื้นในดิน เป็นต้น 3) ส่วนของแหล่งจ่ายไฟ 4) ส่วนของหน่วยความจำ และ 5) ส่วนของหน่วยประมวลผล แต่เนื่องจากการนำไปประยุกต์ใช้งานจริงนั้นมีความต้องการให้โหนดมีขนาดเล็กและราคาไม่แพง จึงทำให้โหนดมีข้อจำกัดในเรื่องของแบตเตอรี่ที่จะต้องมีความเล็กและเบาทำให้จะต้องใช้พลังงานจากแบตเตอรี่ให้มีประสิทธิภาพ นอกจากนี้หน่วยประมวลผลที่ราคาถูกและใช้พลังงานต่ำจะมีความเร็วในการประมวลผลต่ำเช่นกัน ทำให้โหนดไม่เหมาะในการประมวลผลอัลกอริทึมหรือโปรแกรมที่มีความซับซ้อน ดังนั้นงานวิจัยทางด้านเครือข่ายเซนเซอร์ไร้สายจึงเน้นในการใช้พลังงานให้มีประสิทธิภาพ และจากผลการศึกษาในงานวิจัย [5] พบว่าวงจรการรับ-ส่งข้อมูลด้วยคลื่นวิทยุ เป็นส่วนที่มีการใช้พลังงานสูงที่สุด ฉะนั้นวิธีการหนึ่งที่จะช่วยเพิ่มอายุการทำงานของระบบเครือข่ายเซนเซอร์ไร้สาย คือจะต้องลดปริมาณการรับ-ส่งข้อมูลที่ไม่จำเป็นในเครือข่ายหรือจะต้องปรับปรุงการรับ-ส่งข้อมูลให้มีการใช้พลังงานอย่างมีประสิทธิภาพ

ระบบเครือข่ายเซนเซอร์ไร้สายจำเป็นอย่างมากที่จะต้องมีความสามารถในการจัดการเครือข่ายได้ด้วยตนเอง เนื่องจากเมื่อเวลานานโหนดจำนวนมากจะทยอยลงพื้นที่ขนาดใหญ่แล้วระบบจะต้องเริ่มทำงานได้เองอัตโนมัติ และโหนดจะต้องสามารถจัดการเครือข่ายได้ด้วยตนเอง เรียกความสามารถนี้ว่า Self-Organization [7] หากระบบเครือข่ายเซนเซอร์ไร้สายมีความสามารถ

<sup>1</sup> Multi-hop หมายถึงโหนดในเครือข่ายจะมีการทำงานร่วมกับโหนดอื่น ในการส่งต่อข้อมูลระหว่างกันเพื่อให้ข้อมูลถูกส่งไปถึงโหนดปลายทางที่อยู่ไกลออกไปนอกระยะการติดต่อสื่อสารของตนเอง

ดังกล่าว จะทำให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพ เริ่มต้นจัดตั้งเครือข่ายได้ด้วยตนเอง สามารถค้นหาเส้นทางในการส่งข้อมูลและสามารถรักษาการเชื่อมต่อของเครือข่ายได้ด้วยตนเอง

จากการศึกษาการทำงานของการจัดการเครือข่ายด้วยตนเอง [6,7] พบว่าสามารถแบ่งการทำงานออกเป็น 3 ขั้นตอน คือ การตรวจสอบโหนดข้างเคียง (Neighbor Discovery) การค้นหาเส้นทาง (Routing) และการบำรุงรักษาเส้นทาง (Maintenance) ซึ่งในปัจจุบัน โพรโทคอลที่ใช้ในงานในเครือข่ายเซนเซอร์ไร้สายจำนวนมากสามารถรองรับการจัดการเครือข่ายได้ด้วยตนเอง แต่แตกต่างกันที่โพรโทคอลจะถูกพัฒนาเพื่อจุดประสงค์และใช้เทคนิคที่แตกต่างกันออกไปได้แก่ พัฒนาโพรโทคอลเพื่อให้มีการใช้พลังงานอย่างมีประสิทธิภาพด้วยเทคนิคการชนกันของข้อมูลลดจำนวนการรับ-ส่งข้อมูล และการพัฒนาโพรโทคอลเพื่อให้ค้นหาเส้นทางได้รวดเร็วขึ้น เป็นต้น โพรโทคอลที่ถูกพัฒนาขึ้นเหล่านั้นจะมีข้อจำกัดต่างกันไปเช่น สามารถใช้งานได้ดีเฉพาะเครือข่ายที่มีจำนวนโหนดไม่มาก หรือสามารถทำงานได้ดีในเครือข่ายที่ไม่ต้องการการตอบสนองที่รวดเร็ว เป็นต้น

สำหรับงานวิจัยนี้เป็นการศึกษาและปรับปรุงโพรโทคอลค้นหาเส้นทางที่ชื่อว่า Ad-hoc On-Demand Distance Vector (AODV) [11] ซึ่งถูกนำมาใช้งานอย่างแพร่หลายในเครือข่ายเซนเซอร์ไร้สาย การพัฒนาโพรโทคอลในงานวิจัยนี้มีจุดประสงค์เพื่อต้องการลดจำนวนของแพ็กเกจในการสื่อสารช่วงที่เป็นการซ่อมแซมเส้นทางแบบ Local repair ด้วยการใช้ข้อมูลที่ได้จากการตรวจสอบการเชื่อมต่อของโหนดข้างเคียง (Neighbor of neighbor) ทำให้โหนดสามารถซ่อมแซมเส้นทางได้ทันทีโดยไม่ต้องทำการกระจายแพ็กเกจออกไปในเครือข่าย ส่งผลให้ช่วยลดปริมาณของแพ็กเกจในการสื่อสาร ลดการใช้พลังงาน และลดเวลาของการซ่อมแซมเส้นทางในกรณีที่โหนดในเครือข่ายมีการเคลื่อนที่ นอกจากนี้โพรโทคอล AODV ที่ได้รับการปรับปรุงสามารถเลือกเส้นทางที่สั้นกว่าซึ่งพิจารณาจากจำนวนของโหนดในเส้นทาง (Hop count) ในกรณีที่โหนดมีการเคลื่อนที่ ทำให้ใช้เวลาในการส่งข้อมูลเร็วขึ้นด้วย

## 1.2 การตรวจเอกสาร

ในที่นี้สนใจงานวิจัยที่เสนอการลดการใช้พลังงานในระบบเครือข่ายเซนเซอร์ไร้สายในระดับชั้น Network ซึ่งมีดังต่อไปนี้

บทความวิจัยของ S.J. Lee , M. Gerla [8] ได้กล่าวไว้ว่าเส้นทางที่ติดต่อสื่อสารมีโอกาสได้รับความเสียหายเมื่อโหนดในระบบเครือข่ายเซนเซอร์ไร้สายมีการเคลื่อนที่ ซึ่งทำให้เกิดกระบวนการซ่อมแซมเส้นทาง โหนดในระบบจึงต้องทำการส่งแพ็กเกจร้องขอเส้นทางใหม่ ผลจากกระบวนการดังกล่าวก่อให้เกิดจำนวนแพ็กเกจมากขึ้นในระบบ ในงานวิจัย [8] นี้จึงทำการพัฒนา

โพรโทคอล AODV โดยนำเสนอกระบวนการสร้างเส้นทางทางเลือก ดังนั้นเมื่อพบว่าเส้นทางได้รับความเสียหาย โหนดจะใช้เส้นทางสำรองส่งข้อมูลแทน ทำให้ไม่จำเป็นที่จะต้องส่งแพ็กเก็ตกระจายเข้าสู่เครือข่าย โดยขั้นตอนการสร้างเส้นทางเลือกจะเริ่มขึ้นเมื่อโหนดในระบบเริ่มต้นกระบวนการค้นหาเส้นทาง (Route discovery) และเมื่อโหนดข้างเคียงได้รับข้อความ Route reply (RREP) จากโหนดปลายทาง โดยปกติของโพรโทคอล AODV โหนดดังกล่าวจะไม่สนใจข้อความ RREP ที่ไม่ได้ส่งมายังตนเอง แต่ในบทความดังกล่าวเสนอให้โหนดที่ได้รับข้อความ RREP จะทำการบันทึกค่าที่ได้รับลงในตารางเส้นทาง (Routing table) เพื่อใช้เป็นเส้นทางเลือกเมื่อเส้นทางหลักเกิดความเสียหาย และโหนดจะทราบว่าเส้นทางหลักเกิดความเสียหาย เมื่อโหนดได้รับข้อความ Route error (RRER) จากโหนดข้างเคียง โหนดดังกล่าวซึ่งมีค่าข้อมูลเส้นทางเลือกจะทำการส่งข้อมูลเส้นทางที่ตนเองมี เพื่อใช้ในการซ่อมแซมเส้นทางต่อไป กระบวนการดังกล่าวจะช่วยลดปริมาณในการส่งข้อมูลในระบบเมื่อเส้นทางเดิมที่ใช้งานเสียหาย และยังเป็นการช่วยลดระยะเวลาในการส่งข้อมูล

บทความวิจัยของ J. Singh, P. Singh, S. Rani [9] ได้นำเสนอกระบวนการพัฒนาการซ่อมแซมเส้นทางแบบ Local repair ซึ่งเป็นกระบวนการซ่อมแซมเส้นทางประเภทหนึ่งในการทำงานของโพรโทคอล AODV โดยรูปแบบการทำงานจะใช้กระบวนการซ่อมแซมเส้นทางแบบ Local repair ด้วยการส่งข้อความ RREP แบบปลายทางเดียว (Unicast) แทนการส่งแบบแพร่กระจาย (Broadcast) ด้วยข้อมูลที่ได้จากการปรับปรุงกระบวนการตรวจสอบการเชื่อมต่อของโหนดข้างเคียง [10] ซึ่งกระบวนการซ่อมแซมเส้นทางแบบ Local repair ที่ถูกพัฒนาขึ้นจะช่วยลดจำนวนข้อความการจัดการเส้นทาง (Routing overhead) ที่เกิดขึ้นเมื่อมีการซ่อมแซมเส้นทาง และลดค่าหน่วงเวลา (Delay) ให้แก่เครือข่าย

### 1.3 วัตถุประสงค์ของวิทยานิพนธ์

- 1.3.1 เพื่อปรับปรุงอัลกอริทึมในการซ่อมแซมเส้นทางให้มีจำนวนแพ็กเก็ตลดลง
- 1.3.2 เพื่อพัฒนาอัลกอริทึมการบำรุงรักษาเส้นทางให้สามารถทำการเลือกส่งข้อมูลไปยังเส้นทางที่มีจำนวนโหนดที่ใช้ในการส่งข้อมูล (Hop count) ที่น้อยกว่า ด้วยข้อมูลโหนดข้างเคียงที่ตนเองมี

### 1.4 ขอบเขตการวิจัย

- 1.4.1 เป็นวิธีการเลือกการซ่อมแซมเส้นทางของโพรโทคอล AODV ด้วยกระบวนการของ Local repair ซึ่งจะทดสอบผ่าน โปรแกรม Network Simulation 2 (NS-2)
- 1.4.2 ทดสอบโพรโทคอลที่พัฒนาขึ้นในเครือข่ายที่โหนดมีการเคลื่อนที่

## 1.5 ขั้นตอนและวิธีดำเนินงานวิจัย

ขั้นตอนการดำเนินงานวิทยานิพนธ์แบ่งออกเป็น 10 ขั้นตอน เริ่มจากเดือน มิถุนายน 2553 และสิ้นสุดเดือนกุมภาพันธ์ 2555 โดยมีรายละเอียดขั้นตอนการดำเนินงานดังนี้

ขั้นที่ 1: ศึกษาแนวทางและวิธีการดำเนินงานวิจัย

ขั้นที่ 2: ศึกษาการทำงานของระบบเครือข่ายเซเชนเซอร์ไร้สายในระดับชั้น Network และระดับชั้น Media access control (MAC)

ขั้นที่ 3: ศึกษาโพรโทคอลตรวจสอบโหนดข้างเคียง (Neighbor Discovery)

ขั้นที่ 4: ศึกษาการใช้งานโปรแกรมจำลองระบบเครือข่ายเซเชนเซอร์ไร้สาย NS-2

ขั้นที่ 5: พัฒนาและปรับปรุงโพรโทคอลตรวจสอบโหนดข้างเคียง (Neighbor Discovery) และการซ่อมแซมเส้นทางแบบ Local repair ในโพรโทคอล AODV

ขั้นที่ 6: พัฒนาและปรับปรุงโพรโทคอลเพื่อใช้ในการเปลี่ยนแปลงเส้นทางที่ใช้จำนวนโหนดในการส่งข้อมูลที่ลดลง เมื่อระบบมีการเปลี่ยนแปลงลักษณะโครงสร้าง

ขั้นที่ 7: เปรียบเทียบการใช้โพรโทคอลที่ได้พัฒนาขึ้นกับโพรโทคอลที่มีการศึกษาก่อนหน้านี้

ขั้นที่ 8: ปรับปรุงแก้ไขโพรโทคอลที่พัฒนาขึ้นมาใหม่ เพื่อให้มีความเหมาะสมมากยิ่งขึ้น

ขั้นที่ 9: ปรับปรุงและทดสอบระบบทั้งระบบ

ขั้นที่ 10: สรุปผล จัดทำรายงานฉบับสมบูรณ์

## 1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ได้โพรโทคอล AODV เมื่อซ่อมแซมเส้นทางแบบ Local repair จะทำให้จำนวนแพ็กเกจในระบบลดลง

1.6.2 ได้โพรโทคอล AODV ที่มีการเลือกเส้นทางที่มีจำนวนโหนดที่ใช้ในการส่งข้อมูลลดลง เมื่อรูปแบบโครงสร้างของเครือข่ายมีการเปลี่ยนแปลง ซึ่งปกติเส้นทางที่ส่งข้อมูลจะไม่เปลี่ยนแปลงถ้าเส้นทางไม่ได้รับความเสียหาย สำหรับกรณีนี้เส้นทางจะถูกปรับปรุงให้ดีขึ้นตลอดเวลาหากพบว่าจำนวนของโหนดที่ใช้ส่งข้อมูลลดลง

## 1.7 อุปกรณ์และสถานที่วิจัย

1.7.1 อุปกรณ์ คอมพิวเตอร์และโปรแกรมจำลองเครือข่าย NS-2 เวอร์ชัน 2.34 บนระบบปฏิบัติการ Linux Ubuntu เวอร์ชัน 10.10

1.7.2 สถานที่ทำวิจัย ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

## บทที่ 2

### ทฤษฎีและหลักการ

ในบทนี้จะอธิบายทฤษฎีและหลักการที่เกี่ยวข้องกับวิทยานิพนธ์ฉบับนี้ โดยแบ่งออกเป็นสามหัวข้อ คือ แนะนำเครือข่ายเซนเซอร์ไร้สาย การค้นหาเส้นทางด้วยโพรโทคอล Ad-hoc On-Demand Distance Vector Routing (AODV) และการตรวจสอบสถานะของการเชื่อมต่อ

#### 2.1 แนะนำเครือข่ายเซนเซอร์ไร้สาย

ในปัจจุบันระบบเครือข่ายเซนเซอร์ไร้สายถูกนำมาประยุกต์ใช้งานกันอย่างแพร่หลาย ได้แก่ ระบบเครือข่ายเซนเซอร์ไร้สายสำหรับตรวจสอบสภาพอากาศเพื่องานทางด้านเกษตรกรรม ระบบเครือข่ายเซนเซอร์ไร้สายสำหรับตรวจสอบอัตราการเต้นของหัวใจของผู้ป่วย และระบบเครือข่ายเซนเซอร์ไร้สายตรวจสอบการสั่นไหวเพื่อทำการตรวจจับการเกิดแผ่นดินไหว เป็นต้น ระบบเครือข่ายเซนเซอร์ไร้สายประกอบด้วยโหนดขนาดเล็กจำนวนมากกระจายอยู่ในพื้นที่ใช้งาน โดยที่โหนดแต่ละตัวสามารถติดต่อสื่อสารถึงกันในลักษณะรูปแบบ Ad-hoc [11] ซึ่งเป็นการสื่อสารแบบที่ไม่จำเป็นจะต้องมีสถานีฐาน (Base station) ส่งต่อข้อมูลในลักษณะที่เรียกว่า Multi-hop ได้ เครือข่ายสามารถดูแลจัดการตัวเอง (Self-organize) ได้ แม้ว่าจะมีจำนวนโหนดในระบบเพิ่มขึ้นหรือลดลง

โหนดซึ่งเป็นอุปกรณ์พื้นฐานของระบบเซนเซอร์ไร้สายประกอบด้วย แหล่งพลังงาน (Energy storage) หน่วยประมวลผล (CPU) ภาควัดวัดต่างๆ (Sensors) และภาครับ-ส่งข้อมูลแบบไร้สาย (Radio transceiver) โดยที่ภาครับ-ส่งข้อมูลแบบไร้สายจะมีการใช้งานพลังงานมากที่สุดเมื่อเทียบกับการใช้พลังงานของส่วนอื่นๆ [12] เนื่องจากระบบเครือข่ายเซนเซอร์ไร้สายมีข้อจำกัดในเรื่องของพลังงาน ดังนั้นการรับ-ส่งข้อมูลแบบไร้สายจึงใช้โมดูลแบบที่มีกำลังในการรับ-ส่งต่ำ ส่งผลให้สามารถรับ-ส่งข้อมูลได้ในระยะสั้นๆ จึงทำให้เครือข่ายเหมาะสมกับรูปแบบการทำงานแบบ Ad-hoc และ Multi-hop

#### 2.2 โพรโทคอล AODV

Ad-hoc On-Demand Distance Vector (AODV) [13,14] เป็นโพรโทคอลที่ถูกออกแบบเพื่อใช้งานกับระบบเครือข่ายแบบ Ad-hoc ที่โหนดภายในเครือข่ายสามารถเคลื่อนที่ได้ AODV เป็นโพรโทคอลค้นหาเส้นทางแบบ Reactive กล่าวคือเส้นทางจะถูกสร้างขึ้นเฉพาะเมื่อโหนดมีความต้องการในการส่งข้อมูลเท่านั้น (On demand) เส้นทางของการรับ-ส่งข้อมูลจะถูก



บันทึกไว้ในตารางเส้นทาง (Routing table) และมีการใช้งานข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมาย (Route table entry) นอกจากนี้โพรโทคอลดังกล่าวมีการใช้งานหมายเลขลำดับ (Sequence numbers) เพื่อกำหนดว่าข้อมูลของเส้นทางนั้นเป็นข้อมูลเดิมหรือข้อมูลใหม่ที่ได้รับการปรับปรุงแก้ไข เพื่อเป็นการป้องกันการส่งข้อมูลซ้ำซ้อนกัน

โพรโทคอล AODV ได้รับการพัฒนามาจากโพรโทคอล Destination-Sequenced Distance Vector (DSDV) [15] ซึ่งโพรโทคอล DSDV มีข้อเสียคือ จะส่งตารางเส้นทางแบบแพร่กระจาย (Broadcast) ทุกๆ ช่วงเวลา เพื่อเป็นการปรับปรุงข้อมูลภายในตารางเส้นทาง ทำให้เกิดแพ็คเกจของการจัดการเส้นทางในเครือข่ายขึ้นเป็นจำนวนมาก แต่ในการทำงานของโพรโทคอล AODV จะทำการค้นหาและปรับปรุงเส้นทางเฉพาะเมื่อมีความต้องการใช้เส้นทางเท่านั้น นอกจากนี้โพรโทคอล AODV ได้รับการพัฒนาให้เก็บเฉพาะข้อมูลของโหนดถัดไปเพื่อทำการส่งข้อมูลไปยังโหนดปลายทาง (Destination) เท่านั้น ส่งผลให้ประหยัดหน่วยความจำบนโหนดได้ คุณลักษณะสำคัญอีกประการหนึ่งของโพรโทคอล AODV คือมีการจัดการตารางเส้นทางด้วยวิธีการ Time-based state หมายถึงเส้นทางใดที่เคยมีการใช้งาน เมื่อเวลาผ่านไปพบว่าไม่มีมีการใช้งานเส้นทางนั้นภายในระยะเวลาที่กำหนด โหนดจะพิจารณาว่าเส้นทางดังกล่าวหมดอายุ และทำการลบเส้นทางนั้นไปจากตารางเส้นทาง

การทำงานพื้นฐานของโพรโทคอล AODV แบ่งออกเป็นสองส่วนดังต่อไปนี้

- **Route discovery** เริ่มการทำงานเมื่อโหนดมีความต้องการในการส่งข้อมูลไปยังปลายทาง และโหนดต้นทางไม่มีข้อมูลเส้นทางเพื่อใช้ในการส่งข้อมูล ดังนั้นโหนดต้นทางจะทำการส่งข้อความ Route request (RREQ) แบบแพร่กระจาย เพื่อแจ้งให้โหนดข้างเคียงทราบว่าต้องการข้อมูลเส้นทาง และหากโหนดข้างเคียงไม่มีข้อมูลเส้นทาง ก็จะส่งข้อความ RREQ แบบแพร่กระจายต่อไปจนกว่าจะถึงโหนดที่มีข้อมูลเส้นทาง โหนดดังกล่าวจะทำการส่งข้อความ Route reply (RREP) กลับมา ผ่านทางเส้นทางที่ได้ทำการส่งข้อความ RREQ มาก่อนหน้านี้

- **Route maintenance** เป็นกระบวนการบำรุงรักษาให้เส้นทางยังคงสามารถส่งข้อมูลจากโหนดต้นทางไปยังโหนดปลายทางได้ตลอดเวลา แม้ว่าเส้นทางเดิมจะได้รับความเสียหาย

### 2.2.1 ตารางเส้นทาง (Routing table)

ในส่วนของตารางเส้นทางถูกใช้เพื่อจัดเก็บข้อมูลที่ใช้ในการส่งข้อมูลไปยังโหนดปลายทาง โดยมีรายละเอียดการจัดเก็บข้อมูลต่างๆ ดังตาราง ที่ 2-1

ตารางที่ 2-1 รายละเอียดการจัดเก็บข้อมูลภายในตารางเส้นทาง [14]

| ข้อมูลที่จัดเก็บ                       | ความหมาย  |
|--|---|
| Destination IP Address                 | หมายเลขของโหนดปลายทางที่ใช้สำหรับการส่งข้อมูล   |
| Destination Sequence Number            | หมายเลขลำดับของโหนดปลายทาง  |
| Valid Destination Sequence Number flag | ตัวแปรที่ใช้ในการบ่งบอกให้ทราบว่าข้อมูลเส้นทางนั้นๆ มีหมายเลขลำดับของโหนดปลายทางที่ถูกต้องหรือไม่                   |
| Hop Count                              | จำนวนโหนดที่จะต้องทำการส่งต่อข้อมูลไปจนถึงโหนดปลายทาง   |
| Next hop                               | หมายเลขของโหนดตัวถัดไปเพื่อใช้ในการส่งข้อมูลไปยังปลายทาง  |
| List of Precursors                     | รายการของหมายเลขโหนดข้างเคียง (Neighbor nodes) ที่ถูกใช้ในการส่งข้อความ RREP ต่อเพื่อใช้ในการบำรุงรักษาตารางเส้นทาง |
| Life time                              | เวลาที่โหนดใช้ในการพิจารณาว่าข้อมูลเส้นทางดังกล่าวนั้นยังคงสามารถใช้งานได้  |

### 2.2.2 รูปแบบของข้อความ (Message format)

ในโพรโทคอล AODV นั้นจะมีการใช้งานข้อความ (Message) เพื่อใช้ในการจัดการเกี่ยวกับเส้นทางที่โหนดต้องการ แบ่งข้อความได้เป็น 3 ประเภท [14] คือ

#### 2.2.2.1 ข้อความ Route Request (RREQ)

ข้อความ RREQ ถูกนำไปใช้เมื่อโหนดมีการร้องขอเส้นทาง ในกรณีที่โหนดไม่เคยมีเส้นทางที่ต้องการในการส่งข้อมูล โหนดจะเริ่มทำการส่งข้อความ RREQ แบบแพร่กระจายไปยังโหนดข้างเคียง และโหนดข้างเคียงจะทำการส่งต่อข้อมูลไปเรื่อยๆ จนถึงโหนดปลายทางหรือไปจนถึงโหนดที่มีข้อมูลเส้นทางที่ต้องการ โดยข้อความ RREQ มีรูปแบบของข้อมูลเป็นดังภาพประกอบ 2-1 ซึ่งในแต่ละส่วนมีความหมายและการใช้งานดังแสดงไว้ในตารางที่ 2-2

| Type                        | J | R | G | D | U | Reserved | Hop count |
|-----------------------------|---|---|---|---|---|----------|-----------|
| RREQ ID                     |   |   |   |   |   |          |           |
| Destination IP Address      |   |   |   |   |   |          |           |
| Destination Sequence Number |   |   |   |   |   |          |           |
| Originator IP Address       |   |   |   |   |   |          |           |
| Originator Sequence Number  |   |   |   |   |   |          |           |

ภาพประกอบ 2-1 ลักษณะของข้อความ RREQ

ตารางที่ 2-2 รายละเอียดแต่ละส่วนภายในข้อความ RREQ

| ข้อมูลที่จัดเก็บ            | ความหมาย   |
|-----------------------------|--|
| Type                        | ประเภทของแพ็กเกจ ในที่นี้ Type = 1 คือประเภทของแพ็กเกจเป็น RREQ  |
| J                           | Join Flag ถูกสงวนไว้ใช้สำหรับการส่งแบบแพร่กระจาย   |
| R                           | Repair Flag ถูกสงวนไว้ใช้สำหรับการส่งแบบแพร่กระจาย   |
| G                           | Gratuitous RREP Flags หรือเป็นตัวแปรที่บ่งบอกให้ทราบว่าข้อความ RREP ที่ต้องการควรส่งแบบปลายทางเดียวไปยังโหนดปลายทาง    |
| D                           | Destination only flag หรือตัวแปรที่บ่งบอกให้ทราบว่าเฉพาะโหนดปลายทางเท่านั้นที่สามารถทำการตอบกลับข้อความ RREQ ได้       |
| U                           | Unknown Sequence number หรือตัวแปรที่แสดงให้ทราบว่าโหนดที่ต้องการข้อมูลเส้นทางไม่ทราบค่าหมายเลขลำดับของโหนดปลายทาง     |
| Reserved                    | ถูกสงวนไว้โดยการตั้งค่าให้มีค่าเป็น 0  |
| Hop count                   | จำนวนของโหนดที่ใช้ในการส่งต่อจากต้นทางจนถึงโหนดที่ได้รับข้อความ RREQ   |
| RREQ ID                     | หมายเลขลำดับที่ไม่ซ้ำกันเพื่อใช้ในการระบุความใหม่ของข้อความ RREQ เมื่อมีการส่งข้อความ RREQ ในระบบ                      |
| Destination IP Address      | หมายเลขของโหนดปลายทางที่ต้องการค้นหาเส้นทางเพื่อใช้ในการส่งข้อมูล  |
| Destination Sequence Number | หมายเลขลำดับล่าสุดที่ได้รับมาจากอดีต โดยโหนดต้นทางจะทำการพิจารณาสำหรับทุกๆ เส้นทางที่ใช้ในการส่งข้อมูลไปยังโหนดปลายทาง |

ตารางที่ 2-2 รายละเอียดแต่ละส่วนภายในข้อความ RREQ (ต่อ)

| ข้อมูลที่จัดเก็บ           | ความหมาย   |
|----------------------------|--|
| Originator IP Address      | หมายเลขของโหนดต้นทางที่ทำการสร้างข้อความ RREQ เพื่อทำการค้นหาเส้นทางที่ตนมีความต้องการ             |
| Originator Sequence Number | หมายเลขลำดับปัจจุบันที่ถูกใช้ในเส้นทางเพื่อระบุถึงความต้องการของโหนดต้นทางที่ต้องการทำการส่งข้อมูล |

### 2.2.2.2 ข้อความ Route reply (RREP)

ข้อความ RREP จะถูกใช้ในการส่งข้อมูลตอบกลับ ภายหลังจากโหนดที่ได้รับข้อความ RREQ ในการร้องขอข้อมูลเส้นทางก่อนหน้านี้ ซึ่งข้อความ RREP จะถูกส่งกลับแบบปลายทางเดียว โดยที่จะใช้เส้นทางย้อนกลับที่ถูกสร้างขึ้นในระหว่างขั้นตอนของการส่งต่อข้อความ RREQ และเมื่อโหนดตัวถัดไปได้รับข้อความ RREP ก็จะมีกระบวนการสร้างเส้นทางสำหรับการส่งข้อมูลต่อไป ข้อความ RREP มีรูปแบบของข้อมูลดังภาพประกอบ 2-2 และมีรายละเอียดของแต่ละส่วนแสดงดังตารางที่ 2-3

| Type                        | R | A | Reserved | Prefix size | Hop count |
|-----------------------------|---|---|----------|-------------|-----------|
| Destination IP Address      |   |   |          |             |           |
| Destination Sequence Number |   |   |          |             |           |
| Originator IP Address       |   |   |          |             |           |
| Lifetime                    |   |   |          |             |           |

ภาพประกอบ 2-2 ลักษณะของข้อความ RREP

ตารางที่ 2-3 รายละเอียดแต่ละส่วนภายในข้อความ RREP

| ข้อมูลที่จัดเก็บ | ความหมาย  |
|------------------|---|
| Type             | ประเภทของแพ็กเกจ ในที่นี้ Type = 2 นั่นคือแพ็กเกจเป็นข้อความ RREP   |
| R                | Repair Flag ใช้สำหรับการส่งแบบแพร่กระจาย  |
| A                | Acknowledgement required เป็นตัวแปรกำหนดว่าข้อความ RREP ดังกล่าวมีความต้องการในการตอบกลับข้อมูล (Acknowledge) |

ตารางที่ 2-3 รายละเอียดแต่ละส่วนภายในข้อความ RREP (ต่อ)

| ข้อมูลที่จัดเก็บ            | ความหมาย  |
|-----------------------------|---|
| Reserved                    | เมื่อได้รับแพ็กเกจ โหนดจะไม่สนใจแพ็กเกจดังกล่าว   |
| Prefix Size                 | ถ้า Prefix size ไม่เป็น 0 แล้ว 5-bit ในข้อมูลของ Prefix size จะเป็นตัวกำหนดหมายเลขของโหนดถัดไปที่ถูกใช้สำหรับทุกๆ โหนดด้วยเส้นทางในการส่งข้อมูลเส้นทางเดียวกัน  |
| Hop count                   | จำนวนของโหนดที่ใช้ในการส่งต่อข้อมูลจากต้นทางจนกระทั่งถึงปลายทาง สำหรับการส่ง RREQ แบบแพร่กระจาย จำนวนโหนดที่ใช้ในการจัดการเส้นทางจะทำหน้าที่บ่งบอกถึงจำนวนของโหนดที่ใช้ในการส่งข้อมูลไปยัง Multicast tree ที่มีการส่งข้อความ RREP |
| Destination IP Address      | หมายถึงหมายเลขของโหนดปลายทางที่ใช้เส้นทางดังกล่าวในการส่งข้อมูล   |
| Destination Sequence Number | หมายเลขลำดับล่าสุดที่เกี่ยวข้องกับความต้องการเส้นทางนั้นๆ   |
| Originator IP Address       | หมายเลขของโหนดต้นทางที่สร้างข้อความ RREQ สำหรับค้นหาเส้นทางดังกล่าว   |
| Lifetime                    | เวลาในหน่วยมิลลิวินาทีสำหรับโหนดที่ได้รับข้อความ RREP เพื่อทำการพิจารณาว่าเส้นทางดังกล่าวถูกต้องหรือไม่   |

### 2.2.2.3 ข้อความ Route Error (RRER)

การใช้งานข้อความ RRER จะเกิดขึ้นเมื่อโหนดนั้นๆ เคยส่งข้อมูลได้แต่เกิดความเสียหายของเส้นทาง (Link failure) ขึ้นและโหนดไม่สามารถทำการซ่อมแซมเส้นทางด้วยวิธีการ Local repair ได้สำเร็จ ข้อความ RRER จะถูกส่งแบบกระจายเพื่อแจ้งกลับไปยังโหนดต้นทางที่ต้องการส่งข้อมูลให้ทราบว่าเส้นทางที่ต้องการเกิดความเสียหาย และถ้าโหนดต้นทางยังคงมีความต้องการเส้นทางดังกล่าว โหนดต้นทางจะเริ่มกระบวนการค้นหาเส้นทางใหม่ ซึ่งลักษณะของข้อความ RRER มีข้อมูลดังภาพประกอบ 2-3 โดยมีรายละเอียดดังตารางที่ 2-4

| Type   | N | Reserved | DestCount |
|--|---|----------|-----------|
| Unreachable Destination IP Address (1)                         |   |          |           |
| Unreachable Destination Sequence Number (1)                    |   |          |           |
| Additional Unreachable Destination IP Address (if needed)      |   |          |           |
| Additional Unreachable Destination Sequence Number (if needed) |   |          |           |

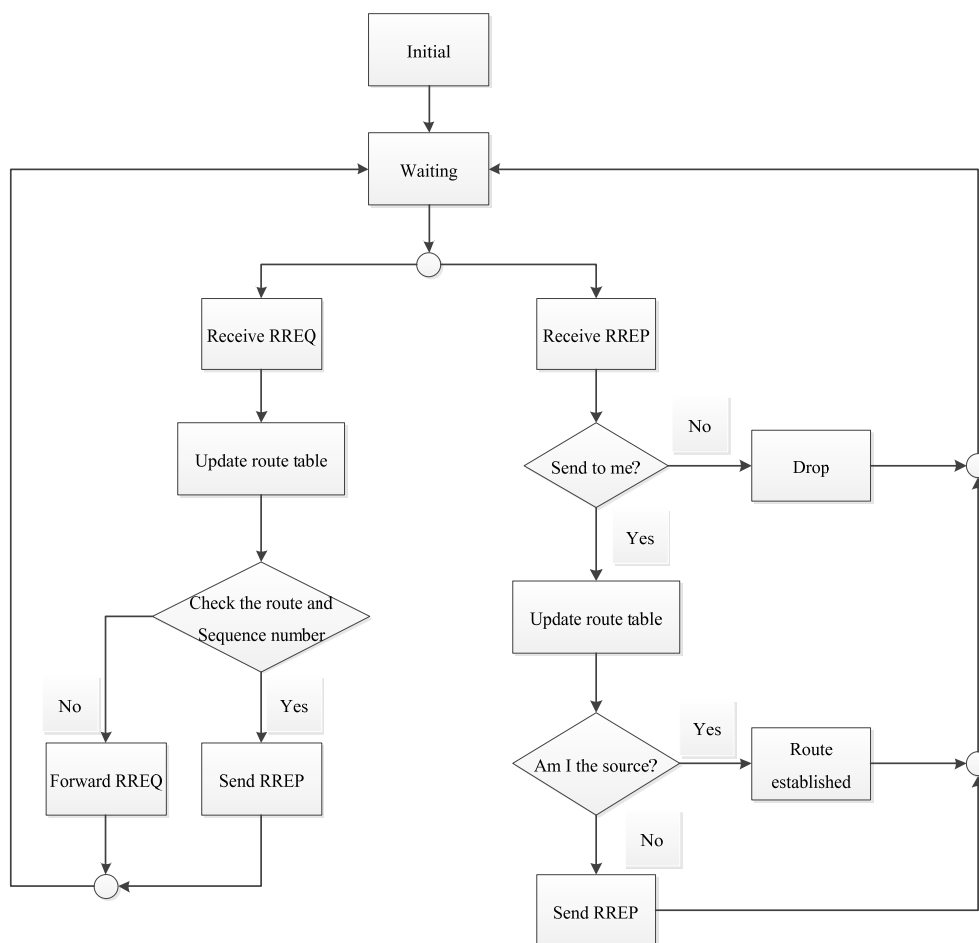
ภาพประกอบ 2-3 ลักษณะของข้อความ RRER

ตารางที่ 2-4 รายละเอียดแต่ละส่วนภายในข้อความ RRER

| ข้อมูลที่จัดเก็บ                        | ความหมาย   |
|---|--|
| Type                                    | ประเภทของแพ็กเกจ ในที่นี้ Type = 3 นั่นคือแพ็กเกจเป็นข้อความ RRER  |
| N                                       | No delete Flag จะถูกตั้งค่าเมื่อ โหนดมีการทำงานในส่วนของ Local repair เกิดขึ้น และ โหนดที่ตรวจพบที่เกิดเส้นทางเสียหายนั้นจะไม่ทำการลบเส้นทางเก่า |
| Reserved                                | เมื่อได้รับแพ็กเกจ โหนดจะไม่สนใจแพ็กเกจดังกล่าว  |
| DestCount                               | จำนวนของ โหนดปลายทางที่ไม่สามารถส่งข้อมูลไปถึงได้ โดยจะต้องมีค่าอย่างน้อยเป็นหนึ่ง   |
| Unreachable Destination IP Address      | หมายเลขของ โหนดปลายทางที่ไม่สามารถส่งข้อมูลไปถึงได้เนื่องจากเกิดเส้นทางเสียหาย   |
| Unreachable Destination Sequence Number | หมายเลขลำดับในข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมาย สำหรับรายชื่อของหมายเลข โหนดปลายทางใน Unreachable Destination IP Address                 |

### 2.2.3 หลักการทำงานของโปรโตคอล AODV

โปรโตคอล AODV เป็นการค้นหาเส้นทางเพื่อใช้ในการส่งข้อมูล โดยใช้แนวคิดของเครือข่ายแบบ Ad-hoc และหลักการของ On demand กล่าวคือ โหนดจะทำการเก็บข้อมูลเส้นทางเฉพาะ โหนดตัวถัดไปสำหรับใช้ในการส่งข้อมูลไปยังปลายทาง และจะทำการค้นหาเส้นทางเฉพาะเมื่อโหนดมีความต้องการเส้นทางเท่านั้น โดยหลักการทำงานพื้นฐานของโปรโตคอล AODV แสดงดังรูป 2-4



ภาพประกอบ 2-4 แสดงหลักการทำงานพื้นฐานของโปรโตคอล AODV

จากภาพประกอบ 2-4 หลักการทำงานพื้นฐานของโปรโตคอล AODV โหนดในระบบจะทำการรอรับข้อความการจัดการเส้นทาง ถ้าหากเป็นข้อความ RREQ โหนดจะเข้าสู่กระบวนการปรับปรุงตารางเส้นทางและตรวจสอบข้อมูลเส้นทางของตนเองว่ามีข้อมูลที่ต้องการหรือไม่ หากไม่มีจะทำการส่งต่อข้อความ RREQ แต่หากมีข้อมูลดังกล่าวจะทำการตอบกลับด้วยข้อความ RREP และในกรณีที่โหนดได้รับข้อความ RREP โหนดจะทำการพิจารณาว่าข้อความ RREP ดังกล่าวมีความต้องการส่งมายังตนเองหรือไม่ หากข้อความดังกล่าวไม่ได้ต้องการส่งมายังตนเอง โหนดจะไม่พิจารณาข้อความนั้น แต่หากข้อความดังกล่าวส่งมายังตนเอง โหนดจะเข้าสู่กระบวนการปรับปรุงตารางเส้นทางและส่งต่อข้อความ RREP กลับไปยังโหนดที่มีความต้องการเส้นทางนั้นๆ ต่อไป โดยหลักการทำงานแบบละเอียดของโปรโตคอล AODV สามารถแบ่งได้เป็น 11 ขั้นตอน ดังต่อไปนี้

### 2.2.3.1 การจัดการเลขลำดับ (Maintaining Sequence Numbers)

ข้อมูลในตารางเส้นทางจะเก็บข้อมูลหมายเลขลำดับสุดท้ายที่ใช้งานได้สำหรับเป็น IP Address ของโหนดปลายทาง โดยที่หมายเลขลำดับ ดังกล่าวจะถูกเรียกว่า หมายเลขลำดับของโหนดปลายทาง (Destination Sequence Number) ซึ่งจะถูกทำการปรับปรุงก็ต่อเมื่อโหนดได้รับหมายเลขลำดับใหม่จากข้อความ RREQ, ข้อความ RREP หรือ ข้อความ RRER ที่มีความเกี่ยวข้องกับโหนดปลายทาง โหนดจะบำรุงรักษาหมายเลขลำดับของโหนดปลายทางของตนเองเพื่อป้องกันการเกิด Loop-freedom และ โหนดปลายทางจะทำการเพิ่มหมายเลขลำดับของตนเองในสองสถานการณ์คือ

- ก่อนที่โหนดต้นทางจะค้นหาเส้นทาง โหนดจะทำการเพิ่มหมายเลขลำดับของตนเองเพื่อหลีกเลี่ยงการซ้ำซ้อนกันของข้อมูลก่อนหน้าที่เคยถูกส่งและใช้งานในการสร้างเส้นทางย้อนกลับ
- ก่อนที่โหนดปลายทางสร้างข้อความ RREP เพื่อตอบกลับข้อความ RREQ ที่ได้รับ โดยโหนดดังกล่าวจะปรับปรุงหมายเลขลำดับของตนเองไปเป็นค่าหมายเลขลำดับปัจจุบันที่ได้รับจากข้อความ RREQ

สำหรับค่าของหมายเลขลำดับในตารางเส้นทางของโหนดจะมีการเปลี่ยนแปลง 3 กรณีดังต่อไปนี้

- เมื่อตนเองเป็นโหนดปลายทาง และมีเส้นทางใหม่ไปยังตนเอง
- โหนดดังกล่าวได้รับข้อความการจัดการเส้นทางซึ่งมีข้อมูลที่ใหม่กว่าสำหรับโหนดปลายทางนั้นๆ
- เส้นทางที่ใช้ในการส่งข้อมูลไปยังโหนดปลายทางหมดอายุหรือเกิดความเสียหาย

### 2.2.3.2 ข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมายและรายการของหมายเลขโหนดที่ใช้ในการส่งข้อความ RREP (Precursor lists)

เมื่อโหนดได้รับข้อความควบคุม (Control packet) จากโหนดข้างเคียง หรือมีการสร้างหรือการปรับปรุงเส้นทางเพื่อใช้ในการส่งข้อมูลไปยังโหนดปลายทาง โหนดดังกล่าวจะทำการตรวจสอบว่าตารางเส้นทางของตนเองมีข้อมูลเส้นทางเพื่อใช้ในการส่งไปยังโหนดปลายทางหรือไม่ ถ้าหากไม่มีข้อมูลเพื่อใช้ในการส่งไปยังโหนดปลายทาง โหนดดังกล่าวจะทำการสร้างข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมายขึ้น โดยที่หมายเลขลำดับจะถูกกำหนดจากข้อมูลที่มีอยู่ในข้อความควบคุมที่ได้รับมา และข้อมูลเส้นทางจะถูกปรับปรุงให้มีหมายเลขลำดับใหม่ด้วยสาเหตุต่อไปนี้



- เมื่อได้รับข้อความควบคุมที่มีค่าของหมายเลขลำดับที่สูงกว่าในหมายเลขลำดับของโหนดปลายทางในตารางเส้นทางของโหนดนั้นๆ
- เมื่อได้รับข้อความควบคุมที่มีค่าของหมายเลขลำดับที่เท่ากับหมายเลขลำดับของโหนดปลายทางในตารางเส้นทางแต่มีค่าของจำนวนโหนดที่ใช้ในการส่งข้อมูลที่น้อยกว่า
- เมื่อโหนดไม่ทราบหมายเลขลำดับที่ใช้ในการส่งข้อมูลไปยังโหนดปลายทาง สำหรับเวลาที่พิจารณาว่าเส้นทางดังกล่าวยังคงใช้งานได้ (Life time) ที่จัดเก็บอยู่ในตารางเส้นทางจะมีการกำหนดค่าจากข้อความควบคุมหรือมีการตั้งค่าเริ่มต้นจากตัวแปร ACTIVE\_ROUTE\_TIMEOUT ซึ่งเส้นทางดังกล่าวสามารถใช้เพื่อทำการส่งข้อมูลได้หากเวลาที่พิจารณาว่าเส้นทางดังกล่าวยังคงใช้งานได้ยังมีค่าที่ถูกต้อง และเมื่อมีการใช้งานเส้นทางในการส่งต่อข้อมูล (Forward) ไปยังโหนดปลายทาง ให้มีค่าไม่น้อยกว่าผลรวมของเวลาปัจจุบัน (current\_time) กับ ACTIVE\_ROUTE\_TIMEOUT

สำหรับทุกเส้นทางที่โหนดพิจารณาให้เป็นเส้นทางที่ยังคงทำงานได้ (Valid route) โหนดดังกล่าวจะต้องทำการเก็บรักษารายการของหมายเลขโหนดที่ถูกใช้ในการส่งข้อความ RREP โดยค่ารายการดังกล่าวจะมีการจัดเก็บโหนดข้างเคียง (Neighboring nodes) ที่ทำการสร้างหรือส่งต่อข้อความ RREP

### 2.2.3.3 การสร้างข้อความ RREQ

เมื่อโหนดต้นทางต้องการส่งข้อมูลไปยังโหนดปลายทาง และโหนดต้นทางดังกล่าวไม่มีข้อมูลเส้นทางเพื่อใช้ในการส่ง โหนดดังกล่าวจะสร้างข้อความ RREQ และส่งต่อไปยังโหนดข้างเคียง ในส่วนของหมายเลขลำดับของโหนดปลายทางในข้อความ RREQ จะเป็นค่าสุดท้ายที่โหนดดังกล่าวเคยมีข้อมูลอยู่ แต่ถ้าไม่มี ค่าตัวแปร Unknown sequence number ในข้อความ RREQ ก็จะถูกใช้งาน ส่วนหมายเลขลำดับของโหนดต้นทาง (Originator sequence number) ในข้อความ RREQ จะเป็นค่าเริ่มต้นหมายเลขลำดับของโหนดนั้นๆ ซึ่งจะถูกเพิ่มครั้งละหนึ่งเหมือนกับค่าหมายเลขลำดับของ RREQ (RREQ ID) ซึ่งทั้งระบบจะมีการเก็บรักษาหมายเลขลำดับ RREQ ร่วมกัน สำหรับค่าจำนวนโหนดที่ใช้ในการส่งข้อมูลจะถูกตั้งค่าเริ่มต้นให้มีค่าเป็นศูนย์ ก่อนที่จะกระจายข้อความ RREQ

โหนดต้นทางจะเก็บค่าชั่วคราวของข้อมูลหมายเลขลำดับ RREQ และหมายเลขของโหนดต้นทางของข้อความ RREQ ในช่วงระยะเวลา PATH\_DISCOVERY\_TIME และเมื่อโหนดดังกล่าวได้รับข้อความ RREQ จากโหนดใกล้เคียงซึ่งมีค่าหมายเลขลำดับ RREQ และหมายเลขของโหนดต้นทางที่ตนเองเป็นผู้ส่งออกไป โหนดจะไม่พิจารณาข้อมูลดังกล่าว นอกจากนี้

โหนดต้นทางจะต้องไม่สร้างข้อความ RREQ มากกว่าค่าที่กำหนดไว้ใน RREQ\_RATELIMIT ต่อวินาที ซึ่งหลังจากส่งข้อความ RREQ แบบแพร่กระจายออกไป โหนดจะรอข้อความ RREP ถ้าไม่ได้รับข้อมูลในช่วงเวลาที่กำหนดไว้ในตัวแปร NET\_TRAVERSEL\_TIME มีหน่วยเป็นมิลลิวินาที โหนดดังกล่าวอาจพยายามค้นหาเส้นทางใหม่อีกครั้งโดยการเพิ่มค่าของหมายเลขลำดับ RREQ และจำนวนครั้งในความพยายามดังกล่าวมีค่ามากที่สุดต้องไม่เกินค่าของ RREQ\_RETRIES และเพื่อเป็นการลดความหนาแน่นในระบบ โหนดต้นทางที่ค้นหาเส้นทางใหม่ใช้หลักการทำงานแบบ Binary exponential back off หมายถึงเมื่อโหนดส่ง RREQ แบบแพร่กระจายและรอข้อความ RREP ในช่วงเวลา NET\_TRAVERSAL\_TIME หากไม่ได้รับข้อความ RREP ที่ต้องการ โหนดต้นทางจะทำการส่งข้อความ RREQ ใหม่โดยการคำนวณเวลาในการรอสำหรับข้อความ RREQ ครั้งต่อไปจะมีค่าเป็น  $2 * \text{NET\_TRAVERSAL\_TIME}$  มิลลิวินาที ถ้าหากยังคงไม่ได้รับข้อความ RREP ในช่วงเวลาดังกล่าว ข้อความ RREQ จะถูกส่งแบบแพร่กระจายใหม่อีกครั้ง และทุกๆ ครั้งที่มีความพยายาม ระยะเวลาในการรอข้อความ RREP จะมีค่าเพิ่มขึ้นเป็นสองเท่าของระยะเวลาในการรอล่าสุด

#### 2.2.3.4 การควบคุมการส่ง Route Request message

เพื่อหลีกเลี่ยงการส่งข้อความ RREQ แบบกระจายไปทั่วทั้งระบบโดยไม่จำเป็น โหนดต้นทางจะใช้เทคนิคการค้นหาแบบ Expanding ring search โดยโหนดต้นทางจะเริ่มทำการกำหนดค่าเริ่มต้นของ Time to life (TTL) ให้มีค่าเท่ากับ TTL\_START ในข้อความ RREQ โดยค่า TTL หมายถึง ค่าจำนวนครั้งของแพ็กเก็ตที่สามารถถูกส่งต่อได้ และตั้งค่าระยะเวลาที่กำหนด (Timeout) สำหรับการรอรับข้อความ RREP ให้มีค่าเป็น RING\_TRAVERSAL\_TIME มิลลิวินาที และถ้าโหนดไม่ได้รับข้อความ RREP ภายในเวลาที่กำหนด โหนดต้นทางจะทำการส่งข้อความ RREQ แบบแพร่กระจายอีกครั้ง และจะเพิ่มค่าของ TTL ภายในข้อความ RREQ โดยค่าที่เพิ่มขึ้นนั้นมีค่าเท่ากับ TTL\_INCREMENT จนกระทั่งค่าของ TTL มีค่าเท่ากับ TTL\_THRESHOLD หลังจากนั้นจะให้ค่า TTL เท่ากับค่า NET\_DIAMETER แทน

ในส่วนของจำนวนโหนดที่ใช้ในการส่งข้อมูลที่ถูกบันทึกอยู่ในตารางเส้นทางว่าเป็นเส้นทางที่ไม่สามารถใช้งานได้ (Invalid Routing table) นั้นหมายถึงค่าจำนวนโหนดที่ใช้ในการส่งข้อมูลล่าสุดที่โหนดดังกล่าวรู้เกี่ยวกับข้อมูลเส้นทางเพื่อใช้ไปยังโหนดปลายทาง และเมื่อมีความต้องการเส้นทางใหม่โดยที่เป็นโหนดปลายทางเดิม ค่าของ TTL ที่ถูกใช้ในการส่งข้อความ RREQ แบบแพร่กระจายจะถูกตั้งค่าเริ่มต้นให้มีค่าเป็นผลรวมของค่าจำนวนโหนดที่ใช้ในการส่งข้อมูลที่มีอยู่ในข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมายกับ TTL\_INCREMENT ในส่วนของตัวแปร Expired routing table entry ไม่ควรถูกลบก่อนผลบออกของเวลาปัจจุบันกับค่าของตัวแปร

DELETE\_PERIOD และข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมาย (Routing table entry) ที่มีการรอข้อความ RREP ไม่ควรถูกลบก่อนเวลาปัจจุบันบวกกับค่า  $2 * NET\_TRAVERSAL\_TIME$

### 2.2.3.5 การพิจารณาข้อมูลที่ได้รับและการส่งต่อข้อความ RREQ

เมื่อโหนดได้รับข้อความ RREQ ขึ้นแรกโหนดดังกล่าวจะสร้างหรือปรับปรุงค่าของโหนดก่อนหน้า โดยที่ไม่สนใจหมายเลขลำดับ หลังจากนั้นโหนดดังกล่าวจะทำการตรวจสอบว่าเคยได้รับข้อความ RREQ ที่มีค่าหมายเลขของโหนดต้นทางและหมายเลขลำดับ RREQ ในช่วงเวลา PAHT\_DISCOVERY\_TIME ก่อนหน้านี้หรือไม่ ถ้าเคยรับข้อความ RREQ โหนดดังกล่าวจะไม่กระทำกระบวนการใดๆ แต่หากยังไม่เคยรับข้อความ RREQ โหนดจะเริ่มกระบวนการดังต่อไปนี้

ขึ้นแรกโหนดจะทำการเพิ่มค่าของจำนวนโหนดที่ใช้ในการส่งข้อมูลที่อยู่ภายในข้อความ RREQ ขึ้นหนึ่งค่าเพื่อใช้ในการส่งต่อไปยังโหนดข้างเคียง หลังจากนั้นโหนดจะทำการค้นหาเส้นทางย้อนกลับ เพื่อส่งค่ากลับไปยังโหนดต้นทางว่ามีหรือไม่ ถ้าหากว่าไม่มีเส้นทางดังกล่าวแล้ว โหนดจะทำการสร้างเส้นทางย้อนกลับ โดยใช้หมายเลขลำดับของโหนดต้นทาง ที่ได้จากข้อความ RREQ ในตารางเส้นทาง เมื่อเส้นทางย้อนกลับถูกสร้างหรือปรับปรุงแล้วจะทำการเปรียบเทียบค่าหมายเลขลำดับของโหนดต้นทางจากข้อความ RREQ กับหมายเลขลำดับของโหนดปลายทางภายในข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมาย ถ้าหากหมายเลขลำดับของโหนดต้นทางมีค่ามากกว่าก็จะทำการคัดลอกค่าเก็บไว้ในข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมาย แทนที่ค่าเดิมและจะทำการตั้งค่าตัวแปรหมายเลขลำดับถูกต้อง (Valid Sequence Number) ในข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมาย ให้มีค่าเป็น TRUE หลังจากนั้น จะทำการตั้งค่าโหนดถัดไปในตารางเส้นทางให้มีค่าเป็นหมายเลขของโหนดที่ทำการส่งข้อความ RREQ ที่ได้รับมา และทำการคัดลอกค่าจำนวนโหนดที่ใช้ในการส่งข้อมูลที่ได้จากข้อความ RREQ

เมื่อใดก็ตามที่โหนดได้รับข้อความ RREQ และทำการปรับปรุงค่าในตารางเส้นทางแล้วพบว่าโหนดดังกล่าวไม่สามารถส่งข้อความ RREP กลับไปได้และถ้าหากค่า TTL ในข้อความ RREQ มีค่ามากกว่า 1 โหนดดังกล่าวจะทำการลดค่า TTL ลงหนึ่งและเพิ่มค่าจำนวนโหนดที่ใช้ในการส่งข้อมูลขึ้นหนึ่ง จากนั้นจะเข้าสู่กระบวนการส่งข้อความ RREQ แบบแพร่กระจายไปยังโหนดข้างเคียง

### 2.2.3.6 การสร้างข้อความ RREP

กรณีที่โหนดจะทำการส่งข้อความ RREP กลับมีขั้นตอนดังต่อไปนี้

- โหนดที่ได้รับข้อความ RREQ ดังกล่าวเป็นโหนดปลายทาง ถ้าหากว่าหมายเลขลำดับของโหนดปลายทางภายในข้อความ RREQ นั้นมีค่ามากกว่าค่าหมายเลขลำดับของตนเอง โหนดจะทำการเพิ่มค่าของหมายเลขลำดับของตนเองขึ้นหนึ่ง แต่ถ้าหากว่าค่าหมายเลขลำดับของโหนดปลายทางภายในข้อความ RREQ ไม่มากกว่าค่าหมายเลขลำดับของตนเอง โหนดดังกล่าวจะไม่กระทำกระบวนการเพิ่มค่าหมายเลขลำดับของตนเอง หลังจากนั้นจะทำการตั้งค่าเริ่มต้นให้ค่าจำนวนโหนดที่ใช้ในการส่งข้อมูลมีค่าเป็นศูนย์ และตั้งค่าให้หมายเลขลำดับของโหนดปลายทางภายในข้อความ RREP มีค่าเท่ากับค่าหมายเลขลำดับของตนเอง และทำการส่งข้อความ RREP กลับผ่านทางเส้นทางย้อนกลับ
- โหนดดังกล่าวมีข้อมูลเส้นทางที่ใช้งานได้ (Active route) เพื่อส่งข้อมูลไปยังโหนดปลายทาง โดยที่หมายเลขลำดับของโหนดปลายทางในตารางเส้นทางนั้นมีค่ามากกว่าหรือเท่ากับหมายเลขลำดับของโหนดปลายทางภายในข้อความ RREQ ที่ได้รับมาและจะทำการคัดลอกค่าหมายเลขลำดับดังกล่าวไปยังหมายเลขลำดับของโหนดปลายทางลงในข้อความ RREP ก่อนทำการส่งกลับผ่านทางเส้นทางย้อนกลับ

ในการสร้างข้อความ RREP โหนดจะทำการคัดลอกค่าของหมายเลขโหนดของโหนดปลายทางและหมายเลขลำดับของโหนดต้นทางที่ได้จากข้อความ RREQ ไปยังข้อความ RREP และการส่งข้อความ RREP จะเป็นการส่งแบบปลายทางเดียวไปยังโหนดตัวถัดไป โดยมีปลายทางเป็นโหนดต้นทางที่ทำการสร้างข้อความ RREQ ขึ้น ส่วนค่าของจำนวนโหนดที่ใช้ในการส่งข้อมูลจะถูกเพิ่มครั้งละหนึ่ง เมื่อมีการส่งข้อมูลผ่านแต่ละโหนดเช่นเดียวกับการส่งข้อความ RREQ ดังนั้นเมื่อข้อความ RREP ถูกส่งไปจนถึงโหนดต้นทางที่มีความต้องการเส้นทาง ค่าของจำนวนโหนดที่ใช้ในการส่งข้อมูลภายในข้อความ จะหมายถึงระยะทางหรือจำนวนของโหนดที่จำเป็นต้องทำการส่งต่อข้อมูลระหว่างโหนดต้นทางและโหนดปลายทาง

### 2.2.3.7 การรับและส่งต่อข้อความ RREP

ขั้นตอนแรกหลังจากเมื่อโหนดได้รับข้อความ RREP จะทำการค้นหาเส้นทางเพื่อทำการส่งข้อมูลไปยังโหนดตัวก่อนหน้า (Previous hop) หลังจากนั้นจะทำการเพิ่มค่าจำนวนโหนดที่ใช้ในการส่งข้อมูลในข้อความ RREP ขึ้นหนึ่ง และจะทำการสร้างเส้นทางไปข้างหน้า (Forward route) ไปยังโหนดปลายทางถ้าโหนดไม่มีการสร้างเส้นทางไปข้างหน้า แต่ถ้าเคยมีเส้นทางไป

ข้างหน้าอยู่ โหนดจะทำการเปรียบเทียบค่าหมายเลขลำดับของ โหนดปลายทางภายในข้อความกับ หมายเลขลำดับของ โหนดปลายทางที่มันเคยเก็บไว้ ซึ่งจะถูกรับปรุงก็ต่อเมื่อมีเหตุการณ์ ดังต่อไปนี้

- หมายเลขลำดับภายในตารางเส้นทางที่ โหนดเก็บค่าอยู่นั้นถูกตั้งค่าให้เป็น เส้นทางที่ไม่สามารถใช้งานได้ (Invalid routing table) ภายในข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมาย
- หากหมายเลขลำดับของ โหนดปลายทางในข้อความ RREP มีค่ามากกว่า หมายเลขลำดับภายในตารางเส้นทาง ค่าของหมายเลขลำดับของ โหนด ปลายทางจะถูกคัดลอกเก็บไว้
- หมายเลขลำดับมีค่าเท่ากัน แต่เส้นทางดังกล่าวถูกตั้งค่าให้เป็นเส้นทางที่ไม่สามารถใช้งานได้
- หมายเลขลำดับมีค่าเท่ากัน แต่ค่าของจำนวน โหนดที่ใช้ในการส่งข้อมูลใหม่ซึ่ง ได้รับมาจากข้อความ RREP มีค่าน้อยกว่าค่าของจำนวน โหนดที่ใช้ในการส่ง ข้อมูลใหม่ที่เคยถูกจัดเก็บไว้ในข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมาย ก่อนหน้า

ถ้าหากข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมายเพื่อใช้ในการส่งข้อมูลไปยัง โหนดปลายทางถูกสร้างหรือถูกปรับปรุงก็จะดำเนินการตามขั้นตอนดังต่อไปนี้

- เส้นทางจะถูกตั้งค่าให้เป็นเส้นทางที่ใช้งานได้
- หมายเลขลำดับของ โหนดปลายทางจะถูกตั้งค่าให้เป็นค่าที่ใช้งานได้
- โหนดตัวถัดไปในตารางเส้นทางจะถูกตั้งค่าให้เป็นค่าของหมายเลข โหนดที่ได้ ทำการส่งข้อความ RREP มาสู่โหนดที่ได้รับข้อมูลดังกล่าว
- จำนวนโหนดที่ใช้ในการส่งข้อมูลจะได้จากการเพิ่มค่าขึ้นหนึ่งค่าจากค่าที่ได้ จากข้อความ RREP
- ส่วนของค่าเวลาหมดอายุ (Expiry time) จะถูกตั้งค่าให้มีค่าเท่ากับค่าของเวลา ปัจจุบันบวกกับค่าของเวลาที่ใช้พิจารณาว่าเส้นทางดังกล่าวยังคงใช้งานได้ (Lifetime) ที่ได้จากข้อความ RREP
- หมายเลขลำดับของ โหนดปลายทางจะเป็นค่าเดียวกับหมายเลขลำดับของ โหนดปลายทางที่อยู่ภายในข้อความ RREP

เมื่อใดก็ตามที่โหนดมีการส่งข้อความ RREP ส่วนของรายการของหมายเลข โหนด ที่ถูกใช้ในการส่งข้อความ RREP สำหรับโหนดปลายทางจะถูกปรับปรุงโดยการเพิ่มหมายเลข

โหนดถัดไปที่จะทำการส่งข้อความ RREP ต่อ และเส้นทางที่ใช้ในการส่งหมายเลข RREP กลับคือเส้นทางย้อนกลับ

### 2.2.3.8 ข้อความตรวจสอบโหนดข้างเคียง (Hello messages)

โหนดอาจใช้วิธีการในการตรวจสอบการเชื่อมต่อกันระหว่างสองโหนดโดยวิธีการส่งข้อความตรวจสอบโหนดข้างเคียงแบบแพร่กระจาย ซึ่งโหนดควรจะมีการใช้งานข้อความตรวจสอบโหนดข้างเคียงเฉพาะในเส้นทางที่ใช้งานเท่านั้น โดยจะทำการส่งแบบแพร่กระจายในทุกๆช่วงเวลา HELLO\_INTERVAL มิลลิวินาที ซึ่งข้อความตรวจสอบโหนดข้างเคียงที่ทำการส่งแบบแพร่กระจายนั้นจัดอยู่ในประเภทของข้อความ RREP ที่มีค่า TTL = 1 โดยที่มีการแก้ไขข้อมูลบางส่วนดังนี้

- หมายเลขของโหนดปลายทางคือหมายเลขของโหนดที่ทำการส่งข้อความตรวจสอบโหนดข้างเคียงแบบแพร่กระจาย
- หมายเลขลำดับของโหนดปลายทางคือหมายเลขลำดับล่าสุดของโหนดที่ส่งแบบแพร่กระจาย
- เวลาที่พิจารณาว่าเส้นทางดังกล่าวยังคงใช้งานได้มีค่าจากการคำนวณของ  $ALLOWED\_HELLO\_LOSS * HELLO\_INTERVAL$

โหนดอาจจะตรวจสอบการเชื่อมต่อโดยคอยรับฟังข้อความที่ได้จากโหนดข้างเคียง ถ้าโหนดดังกล่าวได้รับข้อความตรวจสอบโหนดข้างเคียงจากโหนดข้างเคียง และภายในช่วงเวลา  $ALLOWED\_HELLO\_LOSS * HELLO\_INTERVAL$  มิลลิวินาที โหนดที่ไม่ได้รับข้อความตรวจสอบโหนดข้างเคียงจากโหนดข้างเคียงจากโหนดเดิม โหนดดังกล่าวจะสมมติว่าเกิดการเชื่อมต่อเสียหาย (Link failure) ระหว่างตนเองกับโหนดข้างเคียงขึ้น

เมื่อโหนดได้รับข้อความตรวจสอบโหนดข้างเคียงจากโหนดข้างเคียง และโหนดดังกล่าวมีเส้นทางที่ใช้งานได้และต้องใช้งานโหนดข้างเคียงเป็นโหนดถัดไป โหนดจะเพิ่มค่าเวลาที่พิจารณาว่าเส้นทางดังกล่าวยังคงใช้งานได้ให้แก่เส้นทางนั้นๆ โดยมีค่าในการเพิ่มอย่างน้อยที่สุดคือผลบวกระหว่างค่าของ  $ALLOWED\_HELLO\_LOSS * HELLO\_INTERVAL$

### 2.2.3.9 การบำรุงรักษาการเชื่อมต่อระหว่างโหนด (Maintaining Local Connectivity)

โหนดแต่ละตัวที่ทำหน้าที่ในการส่งข้อมูลต่อในระบบ (โหนดที่อยู่ระหว่างเส้นทางที่ใช้งานได้) ควรจะทำการตรวจสอบการเชื่อมต่อสื่อสารไปยังโหนดถัดไปตลอดจนโหนดข้างเคียงที่เคยมีการส่งข้อความตรวจสอบโหนดข้างเคียง ภายในช่วงระยะเวลาของ

ALLOWED\_HELLO\_LOSS \*HELLO\_INTERVAL ซึ่งโหนดอาจทำการตรวจสอบการเชื่อมต่อ กับโหนดถัดไปได้สองวิธีคือ

- Link layer notification เช่นใน IEEE 802.11 มีการใช้งานโดยในแต่ละเวลาที่ แพ็กเกจถูกส่งไปยังโหนดถัดไป จะมีการส่งข้อความ RTS (Request to send message) เพื่อทำการแจ้งให้โหนดถัดไปทราบว่ามีความต้องการที่จะทำการส่ง ข้อมูล และโหนดตัวถัดไป หากพร้อมรับข้อมูลจะทำการส่งข้อความ CTS (Clear to send message) ตอบกลับมา ซึ่งหากโหนดที่ต้องการส่งข้อมูล ไม่ได้ รับข้อความ CTS ตอบกลับมาภายหลังจากจำนวนครั้งที่มากที่สุดใน การพยายามส่งข้อมูล โหนดดังกล่าวจะสมมติว่าการเชื่อมต่อระหว่างสองโหนด ข้างต้นเกิดการเชื่อมต่อเสียหาย
- กรณีที่ไม่มี Link layer notification โหนดอาจใช้วิธีการคือการรอรับฟังข้อมูล จากช่องสัญญาณ ซึ่งอาจจะอยู่ในรูปแบบของแพ็กเกจที่ได้รับจากโหนดถัดไป (รวมไปถึงข้อความตรวจสอบโหนดข้างเคียง) ถ้าหากว่าโหนดไม่ได้รับข้อมูล ใดๆ จากโหนดถัดไปภายในช่วงเวลา ALLOWED\_HELLO\_LOSS \* HELLO\_INTERVAL โหนดจะสมมติว่าเกิดการเชื่อมต่อเสียหาย และจะเริ่มทำ การส่งข้อความ RRER ในขั้นตอนถัดไป

#### 2.2.3.10 การจัดการข้อความ RERR และการลบข้อมูลเส้นทาง

โดยทั่วไปข้อความ RERR และการจัดการเมื่อเกิดการเชื่อมต่อเสียหายมี กระบวนการดังต่อไปนี้

- เส้นทางที่มีอยู่เดิมเกิดความไม่ถูกต้อง
- ตรวจสอบโหนดปลายทางที่ได้รับผลกระทบจากเส้นทางที่เสียหาย
- ทำการกำหนดว่าโหนดข้างเคียงใดบ้างที่ได้รับผลกระทบ
- ทำการส่งข้อความ RERR ที่เหมาะสมไปยังโหนดข้างเคียง

ข้อความ RERR อาจจะถูกส่งได้ทั้งในรูปแบบแพร่กระจายหรือแบบปลายทาง เดียว และโหนดจะไม่ทำการสร้างข้อความ RERR เกินค่าของ RERR\_RATELIMIT ข้อความต่อ หนึ่งวินาที

โหนดจะเริ่มทำการสร้างข้อความ RERR ใน 3 กรณีต่อไปนี้

- ถ้าโหนดตรวจพบได้ว่าการเชื่อมต่อเสียหายระหว่างตนเองกับโหนดถัดไปที่ต้องใช้งานในเส้นทางที่ใช้งานได้ โหนดจะทำการสร้างรายชื่อของโหนดปลายทางที่ตนเองไม่สามารถส่งข้อมูลไปถึงได้ลงในข้อความ RERR
- ถ้าโหนดได้รับข้อมูลและตนเองไม่มีเส้นทางที่ใช้งานได้ (ในกรณีที่ไม่มีการทำงาน Local repair) ซึ่งหมายถึงโหนดดังกล่าวจะมีรายชื่อของโหนดปลายทางที่ไม่สามารถส่งข้อมูลได้เพียงแค่โหนดเดียว
- ถ้าโหนดได้รับข้อความ RERR จากโหนดข้างเคียง ในกรณีดังกล่าวรายชื่อของโหนดปลายทางที่ไม่สามารถส่งข้อมูลได้จะทำการคัดลอกค่าจากข้อความ RERR ที่ได้รับมา

และก่อนจะมีการส่งข้อความ RERR จะต้องทำการปรับปรุงค่าหมายเลขลำดับของโหนดปลายทางภายในตารางเส้นทางสำหรับโหนดปลายทางที่ไม่สามารถส่งข้อมูลถึงได้ โดยกระบวนการในการปรับปรุงค่าในข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมายมีดังต่อไปนี้

- หมายเลขลำดับของโหนดปลายทางในข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมาย มีค่าถูกต้องจะทำการเพิ่มค่าหมายเลขลำดับขึ้นหนึ่ง ในกรณีที่โหนดต้องการเส้นทางเพื่อส่งข้อมูลไปยังปลายทางที่ไม่สามารถส่งข้อมูลถึงได้ ส่วนในกรณีที่ได้รับข้อความ RERR จะทำการคัดลอกค่าหมายเลขลำดับของโหนดปลายทางมาจากข้อความ RERR แทน
- ข้อมูลเส้นทางภายในตารางเส้นทางจะถูกตั้งค่าให้เป็นเส้นทางที่ไม่สามารถใช้งานได้ นั่นคือจะตั้งค่าให้ข้อมูลเส้นทางแต่ละเส้นทางต่อหนึ่งเป้าหมายดังกล่าวเป็นเส้นทางที่ไม่สามารถใช้งานได้
- ส่วนค่าเวลาที่พิจารณาว่าเส้นทางดังกล่าวยังคงใช้งานได้จะมีค่าเท่ากับผลรวมระหว่างเวลาปัจจุบันกับ DELETE\_PERIOD เพื่อกำหนดให้ทราบว่าเมื่อเกิดเส้นทางเสียหายดังกล่าวขึ้น โหนดจะเก็บเส้นทางดังกล่าวไว้ระยะเวลาหนึ่งก่อนที่จะทำการลบข้อมูลดังกล่าวทิ้ง

#### 2.2.3.11 Local Repair

เมื่อการเชื่อมต่อในเส้นทางเกิดความเสียหายขึ้น โหนดที่ตรวจพบว่าเส้นทางเกิดความเสียหายอาจเลือกใช้วิธีการซ่อมแซมเส้นทางด้วยตนเองหรือ Local repair โดยจะทำการพิจารณาว่า โหนดปลายทางอยู่ใกล้กับตนเองมากกว่าโหนดต้นทางหรือไม่ หากว่าโหนดปลายทางอยู่ใกล้กว่า โหนดที่ตรวจพบว่าเกิดการเชื่อมต่อเสียหายจะทำการเพิ่มหมายเลขลำดับของโหนด



ปลายทางและทำการส่งข้อความ RREQ แบบแพร่กระจายไปยัง โหนดปลายทางด้วยตนเอง ซึ่งเมื่อมีการทำงานในการซ่อมแซมเส้นทางดังกล่าว โหนดต้นทางจะไม่ทราบหรือไม่ได้รับข้อความ RREQ ที่โหนดดังกล่าวส่งออกมา เนื่องจากโหนดจะทำการกำหนดค่าข้อความ RREQ ที่ตนเองทำการส่งแบบแพร่กระจายให้มีค่าเป็นค่าระหว่าง MIN\_REPAIR\_TTL ถึง ครึ่งหนึ่งของจำนวน โหนดที่ใช้ในการส่งข้อมูลจากตนไปยังโหนดต้นทาง และจะทำการเพิ่มค่าขึ้นครึ่งละ LOCAL\_ADD\_TTL และภายหลังจากโหนดที่ทำการตรวจพบว่าการเชื่อมต่อเสียหายทำการส่งข้อความ RREQ แบบแพร่กระจายออกมา โหนดจะทำการรอข้อความ RREP ซึ่งในระหว่างเวลาดังกล่าวข้อมูลต่างๆ ที่โหนดได้รับจะถูกเก็บไว้ชั่วคราว ถ้าภายหลังจากช่วงเวลาการรอ และโหนดไม่ได้รับข้อความ RREP สำหรับใช้ในการส่งข้อมูลไปยังปลายทาง โหนดจะเริ่มกระบวนการในการส่งข้อความ RERR ทันที แต่ถ้าโหนดได้รับข้อความ RREP ในระหว่างช่วงระยะเวลาในการรอที่กำหนดไว้ โหนดจะทำการตรวจสอบว่าข้อความ RREP แต่ละข้อความที่ได้รับมามีค่าจำนวน โหนดที่ใช้ในการส่งข้อมูลดีกว่าค่าที่ตนเองเคยบันทึกไว้ในตารางเส้นทางหรือไม่หากค่าภายในข้อความ RREP ที่ได้รับมามีค่ามากกว่าที่ตนเองบันทึกไว้ กระบวนการส่งข้อความ RERR ก็จะเกิดขึ้น ซึ่งกระบวนการทำ Local repair ดังกล่าวจะเป็นการเพิ่มความสำเร็จในการส่งข้อมูล เนื่องจาก ข้อมูลจะไม่ถูกละทิ้งไปในระหว่างการส่งข้อความ RERR เมื่อเกิดการเชื่อมต่อเสียหายขึ้น

เมื่อการเชื่อมต่อเสียหายหรือเกิดการเชื่อมต่อเสียหายขึ้นภายในเส้นทางที่ใช้งานได้ ซึ่งในบางกรณีอาจทำให้ไม่สามารถส่งข้อมูลไปยังโหนดปลายทางหลายๆ ตัวได้ เนื่องจากอาจมีการใช้งาน โหนดตัวเดียวกันในการส่งข้อมูลไปยังโหนดปลายทางหลายๆ ตัว โหนดที่ทำการตรวจพบว่าการเชื่อมต่อเสียหายในการเชื่อมต่อดังกล่าว เมื่อเริ่มกระบวนการทำ Local repair ขึ้น จะดำเนินการค้นหาเส้นทางใหม่ครึ่งละหนึ่ง โหนดปลายทางเท่านั้น ซึ่งแสดงว่าเส้นทางที่ถูกใช้ในการส่งข้อมูลไปยังโหนดปลายทางอื่นๆ จะต้องถูกตั้งค่าให้เป็นเส้นทางที่ไม่สามารถใช้งานได้ แต่โหนดที่อยู่ในกระบวนการทำ Local repair อาจจะถูกตั้งค่าสถานะว่ากำลังทำ Local repair อยู่ โดยการตั้งค่าสถานะดังกล่าวจะถูกรีเซ็ตภายในเวลาที่กำหนดไว้ คือหลังจากไม่ได้รับการปรับปรุงภายในเวลา ACTIVE\_ROUTE\_TIMEOUT ซึ่งก่อนจะหมดเวลาดังกล่าวอาจมีข้อมูลที่ต้องการส่งไปยังโหนดปลายทางอื่นๆ ดังนั้นเส้นทางที่ทำ Local repair นั้นจะต้องเกิดจากการที่มีข้อมูลที่ต้องการส่งไปยังโหนดปลายทางนั้นๆ เท่านั้น หากไม่มีความต้องการในการส่งข้อมูลไปยังโหนดปลายทาง กระบวนการทำ Local repair เพื่อค้นหาเส้นทางในการส่งข้อมูลไปยังโหนดปลายทางดังกล่าวก็จะไม่เกิดขึ้น

## 2.3 การตรวจสอบสถานะของการเชื่อมต่อ

วิธีการตรวจสอบสถานะของการเชื่อมต่อมี 2 ประเภทคือการใช้งานข้อความตรวจสอบโหนดข้างเคียง (Hello messages) และการใช้งาน Link layer feedback

### 2.3.1 ข้อความตรวจสอบโหนดข้างเคียง

การใช้งานข้อความตรวจสอบโหนดข้างเคียงจะใช้งานโดยให้โหนดในระบบที่อยู่ในเส้นทางที่ใช้งานได้มีการส่งข้อความตรวจสอบโหนดข้างเคียงแบบแพร่กระจายออกมาเป็นระยะๆ เพื่อให้โหนดที่เคยได้รับข้อความดังกล่าวทราบว่าโหนดที่ทำการส่งข้อความนั้นยังคงอยู่ในระยะที่สามารถทำการติดต่อสื่อสารได้ โดยการส่งข้อความตรวจสอบโหนดข้างเคียงจะทำการส่งในช่วงเวลา HELLO\_INTERVAL วินาที และโหนดจะทราบว่าเกิดการเชื่อมต่อเสียหายเมื่อโหนดซึ่งเคยได้รับข้อความจากโหนดข้างเคียงมาก่อน และไม่ได้รับข้อความจากโหนดข้างเคียงนั้นๆ ในช่วงเวลาที่ไต่จากการคำนวณค่าของ  $HELLO\_INTERVAL * ALLOWED\_HELLO\_LOSS$  วินาที

### 2.3.2 Link layer feedback

กระบวนการตรวจสอบสถานะของการเชื่อมต่อแบบ Link layer feedback จะได้รับการสนับสนุนจากชั้น Media access control (MAC) ซึ่งการทำงานดังกล่าวจะสามารถแจ้งเตือนได้เร็วกว่าการตรวจสอบแบบใช้งานข้อความตรวจสอบโหนดข้างเคียง โดยการทำงานคือเมื่อโหนดมีความต้องการส่งข้อมูล โหนดจะทำการส่งความต้องการในการส่งข้อมูลไปยังชั้น MAC ซึ่งเก็บค่าไว้เพื่อส่งข้อมูลไปยังโหนดที่ต้องการ และเมื่อทำการส่งข้อมูลไปยังโหนดที่ต้องการไม่สำเร็จ เช่น ไม่ได้รับข้อความตอบกลับ (Acknowledge messages) จากโหนดที่ตนเองต้องการส่งข้อมูลไป โหนดจะพยายามทำการส่งจนครบจำนวนครั้งในความพยายาม โหนดก็จะทราบได้ทันทีว่าเกิดการเชื่อมต่อเสียหายขึ้น และจะทำการแจ้งเตือนว่าเกิดการเชื่อมต่อเสียหายต่อไป

## บทที่ 3

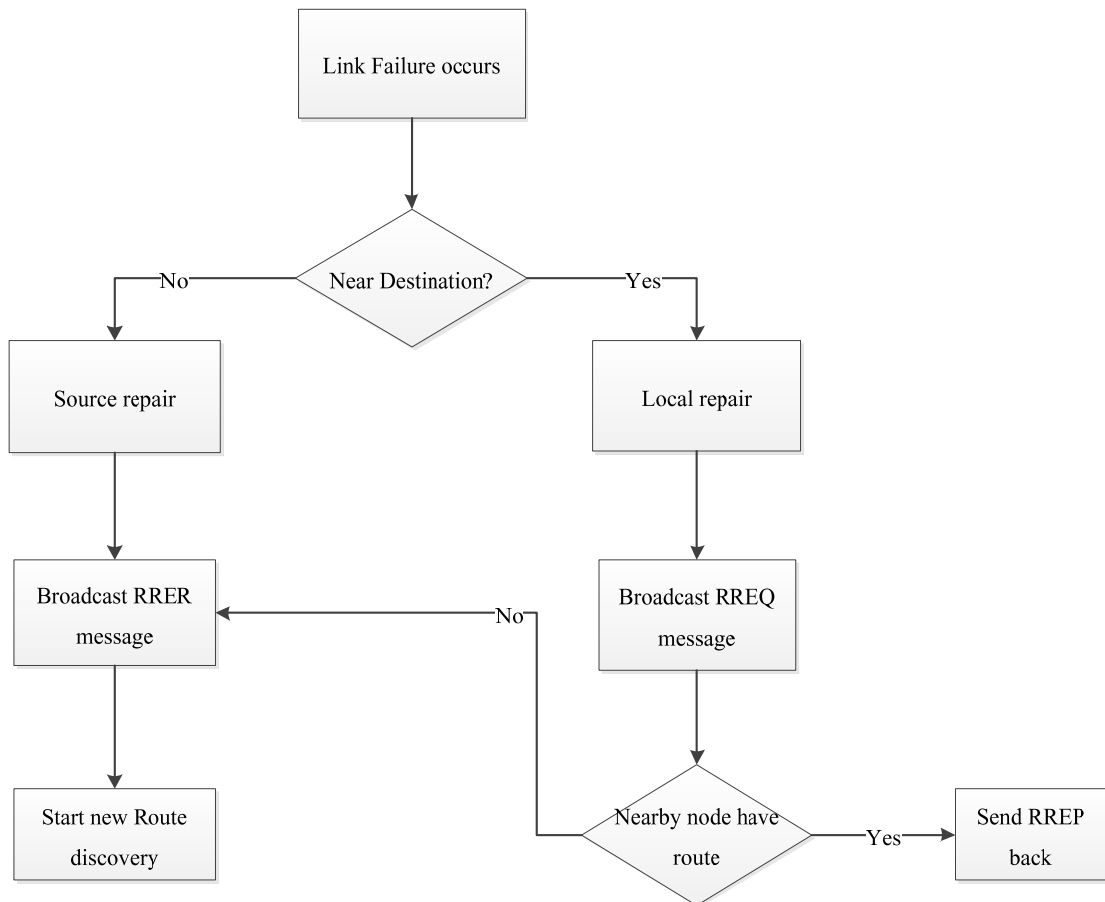
### การพัฒนาโพรโทคอล AODV

บทนี้เป็นการนำเสนอรายละเอียดของการพัฒนาการทำงานโพรโทคอล AODV ซึ่งวิทยานิพนธ์นี้แบ่งการพัฒนาออกเป็นสองส่วนคือ การพัฒนากระบวนการซ่อมแซมเส้นทาง Local repair ในโพรโทคอล AODV และการปรับปรุงเส้นทางให้มีระยะทางที่สั้นกว่าเส้นทางที่ใช้งานเดิม (Active route)

#### 3.1 การพัฒนากระบวนการซ่อมแซมเส้นทาง Local repair ในโพรโทคอล AODV

ในส่วนของพัฒนากระบวนการซ่อมแซมเส้นทาง Local repair เป็นการต่อขยายงานวิจัย [9] แต่ในงานที่ทำการพัฒนาขึ้นนี้มีการทำงานที่แตกต่างออกไป คืองานในวิทยานิพนธ์นี้จะใช้ข้อความตรวจสอบโหนดข้างเคียงที่เรียกว่า Hello messages เพื่อทำหน้าที่เป็นตัวตรวจสอบว่าการเชื่อมต่อได้รับความเสียหาย (Link failure) หรือไม่ โดยเฉพาะโหนดที่อยู่ในเส้นทางที่ใช้งานได้ในปัจจุบัน (Active route) และใช้วิธีการของ Link layer feedback ตรวจสอบการเชื่อมต่อเมื่อมีการรับ-ส่งข้อมูล นอกจากนี้ได้มีการใช้ Hello message เพื่อตรวจสอบข้อมูลโหนดเพื่อนบ้านของโหนดข้างเคียงหรือเรียกว่า Neighbor of neighbor node

การทำงานของโพรโทคอล AODV ดั้งเดิมเมื่อมีเหตุการณ์ที่ทำให้เกิดการเชื่อมต่อเสียหาย จะมีการทำงานดังภาพประกอบ 3-1 โดยขั้นตอนแรกโหนดที่ตรวจพบว่าการเชื่อมต่อได้รับความเสียหาย จะทำการตรวจสอบว่าตนเองอยู่ใกล้กับโหนดต้นทางหรือโหนดปลายทาง ถ้าพบที่อยู่ใกล้กับโหนดต้นทาง จะเลือกทำการซ่อมแซมเส้นทางด้วยวิธีการ Source repair คือ จะเริ่มทำการส่งข้อความ RERR เพื่อแจ้งให้แก่โหนดต้นทางทราบว่าเส้นทางที่โหนดต้นทางมีความต้องการเพื่อทำการส่งข้อมูลไปยังโหนดปลายทางเกิดความเสียหาย มีความจำเป็นต้องทำการหาเส้นทางใหม่ แต่ในกรณีตรงกันข้าม หากโหนดซึ่งตรวจพบว่าการเชื่อมต่อเสียหายอยู่ใกล้กับโหนดปลายทาง โหนดดังกล่าวจะซ่อมแซมเส้นทางด้วยตนเองหรือเรียกว่า Local repair โดยโหนดที่ทำการตรวจพบว่าการเชื่อมต่อเสียหายจะเริ่มกระบวนการส่งข้อความ RREQ แบบแพร่กระจายไปยังโหนดใกล้เคียง เนื่องจากเชื่อว่าโหนดใกล้เคียงน่าจะมีข้อมูลเส้นทางที่ตนเองมีความต้องการ และโหนดซึ่งตรวจพบว่าการเชื่อมต่อเสียหายดังกล่าวจะทำการรอข้อความ RREP หากไม่ได้รับข้อความ RREP โหนดจะซ่อมแซมเส้นทางโดยการใช้ Source repair แทน



ภาพประกอบ 3-1 ลักษณะการทำงานเมื่อเกิดการเชื่อมต่อเสียหาย

ในกระบวนการทำ Local repair ของโพรโทคอล AODV ดั้งเดิม โหนดจะส่งข้อความแบบแพร่กระจายเพิ่มขึ้น ซึ่งแน่นอนว่าการส่งข้อมูลในลักษณะนี้ โหนดรอบข้างจะทำการส่งข้อความต่อออกไปเป็นวงกว้าง ซึ่งข้อมูลดังกล่าวอาจไม่มีความจำเป็น และกรณีที่เครือข่ายมีเหตุการณ์ที่ทำให้เกิดการเชื่อมต่อเสียหายเกิดขึ้นบ่อยครั้งและลักษณะเช่นนี้เกิดขึ้นเป็นธรรมชาติของเครือข่ายเช่นเซอร์ไวส์สาย เมื่อโหนดเลือกใช้งาน Local repair จะทำให้ในเครือข่ายจะมีข้อความจัดการเส้นทาง (Routing overhead) เกิดขึ้นเป็นจำนวนมาก รายละเอียดของกระบวนการพัฒนา Local repair มีสามขั้นตอนหลักดังต่อไปนี้

### 3.1.1 การส่งข้อความตรวจสอบโหนดข้างเคียงแบบแพร่กระจาย

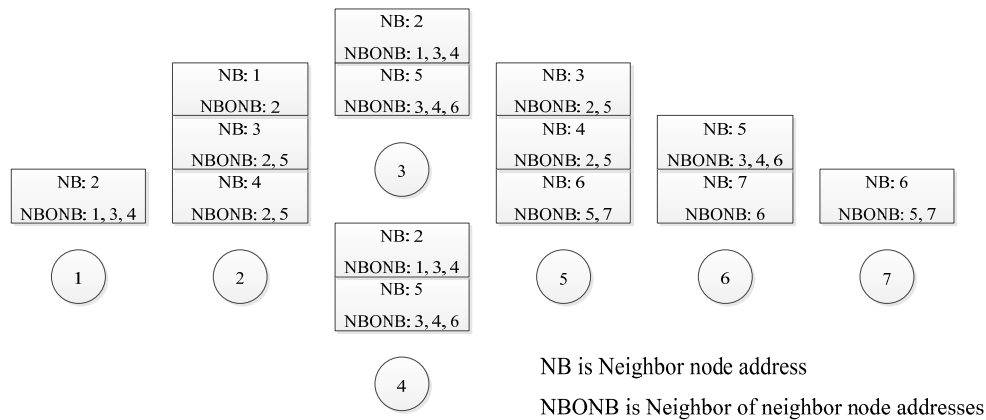
โหนดทุกตัวภายในเครือข่ายมีการใช้งาน Link layer feedback เพื่อทำการตรวจสอบการเชื่อมต่อเสียหาย นอกจากนั้น โหนดทุกตัวภายในเครือข่ายจะมีการใช้งานข้อความตรวจสอบโหนดข้างเคียงเพื่อใช้ในการกระจายข้อมูลเกี่ยวกับตนเองและโหนดข้างเคียง ซึ่งข้อมูลที่ได้ดังกล่าวจะนำไปใช้ในกระบวนการพัฒนาโพรโทคอล AODV ต่อไป ซึ่งจะอธิบายในหัวข้อ 3.1.3

รูปแบบของข้อความตรวจสอบโหนดข้างเคียง ที่มีการใช้งานได้ถูกพัฒนามาจากข้อความตรวจสอบโหนดข้างเคียงเดิม ซึ่งรูปแบบของข้อความตรวจสอบโหนดข้างเคียงแสดงดังภาพประกอบ 3-2

| Type                            | Reserved | Number of Neighbors |
|---------------------------------|----------|---------------------|
| Sequence Number                 |          |                     |
| Neighbor IP Addr (1)            |          |                     |
| Neighbor IP Address (if needed) |          |                     |

ภาพประกอบ 3-2 ลักษณะของข้อความตรวจสอบโหนดข้างเคียง [9]

จากภาพประกอบ 3-2 ลักษณะของข้อความตรวจสอบโหนดข้างเคียงที่มีการใช้งาน จะใช้รูปแบบของข้อความตรวจสอบโหนดข้างเคียงรูปแบบเดิม ซึ่งเป็นส่วนที่อยู่ในกรอบสีขาว และส่วนที่เป็นการเพิ่มข้อมูลเข้าไปใหม่จากรูปแบบของข้อความเดิมจะอยู่ในกรอบเป็นสีเข้ม ซึ่งข้อมูลที่เพิ่มมีสองส่วนคือส่วนของจำนวนโหนดข้างเคียง (Number of Neighbors) และส่วนของหมายเลขของโหนดข้างเคียง (Neighbor IP Address) โดยที่ส่วนของจำนวนโหนดข้างเคียงจะเป็นค่าของจำนวนโหนดข้างเคียงของตนเอง ส่วนหมายเลขของโหนดข้างเคียงจะเป็นหมายเลขของโหนดข้างเคียงแต่ละตัว



ภาพประกอบ 3-3 ตัวอย่างของการทำงานเมื่อโหนดได้รับข้อความตรวจสอบโหนดข้างเคียงจากโหนดข้างเคียง

ภาพประกอบ 3-3 แสดงตัวอย่างของการทำงานเมื่อโหนดได้รับข้อความตรวจสอบโหนดข้างเคียงจากโหนดข้างเคียง มีรายละเอียดการทำงานดังนี้ โหนด 3 เมื่อได้รับข้อความตรวจสอบโหนดข้างเคียงจากโหนดข้างเคียงในที่นี้คือโหนด 2 และ 5 เมื่อโหนด 3 จะส่งข้อความตรวจสอบโหนดข้างเคียงครั้งถัดไป โหนด 3 จะนำข้อมูลที่เกี่ยวข้องกับโหนดรอบข้างของตนเอง มาเพิ่มข้อมูลภายในข้อความตรวจสอบโหนดข้างเคียงในส่วนของจำนวนโหนดข้างเคียงเท่ากับ 2 และเพิ่มหมายเลขของโหนดข้างเคียงให้มีค่าเป็นหมายเลขโหนด 2 และ 5 เมื่อโหนด 2 ได้รับข้อความตรวจสอบโหนดข้างเคียงที่ถูกส่งมาจากโหนด 3 โหนด 2 จะสามารถทราบได้ทันทีว่าโหนดที่ใกล้เคียงโหนด 3 มีจำนวนเท่าใด และมีหมายเลขใดบ้าง ในที่นี้คือโหนด 2 และ 5

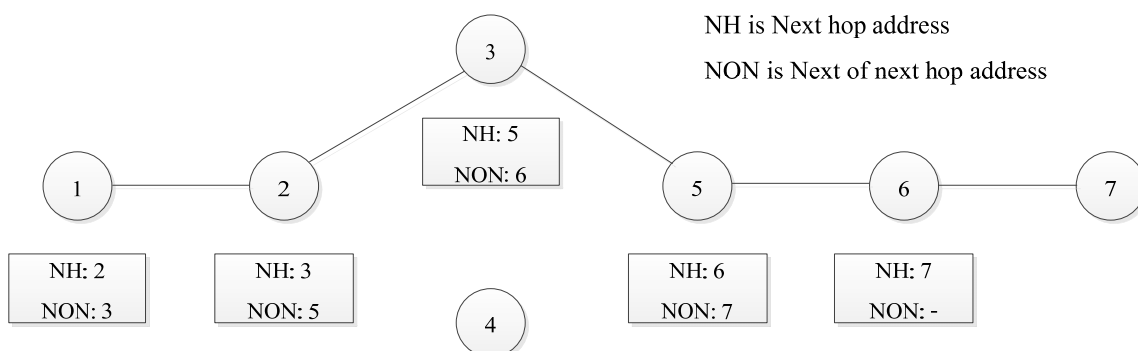
### 3.1.2 หมายเลขของโหนดถัดจากโหนดถัดไป (Next of next hop address)

ข้อความ RREP ที่ได้ปรับปรุงขึ้นมีการเพิ่มข้อมูลต่อท้ายจากรูปแบบของข้อมูลเดิมคือ มีการเพิ่มหมายเลขของโหนดถัดจากโหนดถัดไป และข้อมูลดังกล่าวจะถูกคัดลอกลงในตารางเส้นทางดังภาพประกอบ 3-4

| Type                        | R | A | Reserved | Prefix size | Hop count |
|-----------------------------|---|---|----------|-------------|-----------|
| Destination IP Address      |   |   |          |             |           |
| Destination Sequence Number |   |   |          |             |           |
| Originator IP Address       |   |   |          |             |           |
| Lifetime                    |   |   |          |             |           |
| Next of next hop address    |   |   |          |             |           |

ภาพประกอบ 3-4 ลักษณะของข้อความ RREP [9]

ลักษณะของข้อความ RREP จะมีการเก็บค่าหมายเลขโหนดถัดจากโหนดถัดไปเพิ่มอีกหนึ่งค่า โดยปกติค่าหมายเลขของโหนดถัดไป ที่โหนดบันทึกไว้เพื่อใช้ในการส่งข้อมูลจะเป็นค่าหมายเลขของโหนดตัวที่ส่งข้อความ RREP มาให้ตนเอง ส่วนค่าของหมายเลขโหนดถัดจากโหนดถัดไปจะได้จากการนำค่าหมายเลขโหนดถัดไปภายในตารางเส้นทางของโหนดที่ส่งข้อความ RREP คัดลอกลงในหมายเลขของโหนดถัดจากโหนดถัดไปลงในข้อความ RREP และหลังจากนั้นจึงส่งข้อความ RREP กลับมาผ่านทางเส้นทางย้อนกลับ



ภาพประกอบ 3-5 ลักษณะการทำงานเมื่อมีการส่งข้อความ RREP

ภาพประกอบ 3-5 แสดงลักษณะการทำงานเมื่อมีการส่งข้อความ RREP เมื่อเครือข่ายมีการทำการค้นหาเส้นทาง โดยมีโหนดปลายทางคือโหนด 7 เมื่อโหนด 3 ทำการส่งข้อความ RREP กลับไปยังโหนดต้นทางตามเส้นทางย้อนกลับ โหนดตัวที่จะต้องทำการรับข้อความ RREP แบบปลายทางเดียวจากโหนด 3 คือโหนด 2 สำหรับขั้นตอนของการสร้างข้อความ RREP ของโหนด 3 ได้มีการเพิ่มค่าข้อมูลหมายเลขของโหนดถัดไป (ในที่นี้คือโหนด 5) ลงในหมายเลขของโหนดจากโหนดถัดไปภายในข้อความ RREP และทำการส่งไปยังโหนด 2 เมื่อโหนด 2 ได้รับ

ข้อความ RREP ก็จะทราบทันทีว่าเส้นทางที่ใช้ในการส่งข้อมูลไปยังโหนด 7 มีหมายเลขของโหนดถัดไปคือ โหนด 3 และมีหมายเลขของโหนดถัดจากโหนดถัดไปคือ โหนด 5 เป็นต้น

### 3.1.3 การส่งข้อความ RREQ แบบปลายทางเดียว

เมื่อการเชื่อมต่อเกิดเสียหายขึ้น โหนดที่ตรวจสอบได้ว่าการเชื่อมต่อเสียหาย จะส่งข้อความ RREQ แบบปลายทางเดียว (Unicast) โดยมีเป้าหมายไปยังโหนดที่คาดว่าจะมีข้อมูลเส้นทางที่ตนเองต้องการ โดยข้อมูลที่ใช้จะได้มาจากการรับข้อความตรวจสอบโหนดข้างเคียงและหมายเลขของโหนดถัดจากโหนดถัดไปที่ได้จัดเก็บไว้ในตารางเส้นทางก่อนหน้านี้ ซึ่งรูปแบบของข้อความ RREQ ที่ใช้เป็นดังภาพประกอบ 3-6

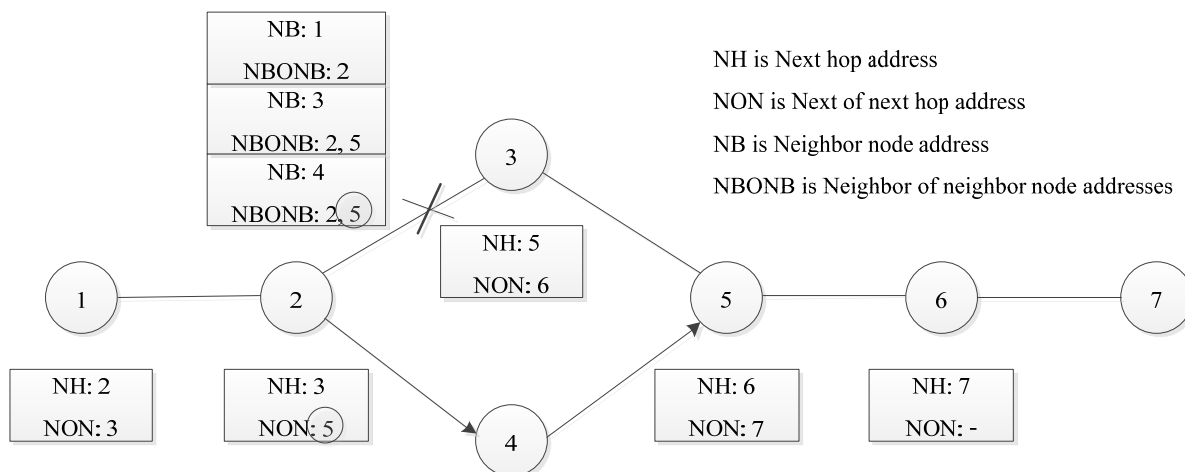
| Type                        | J | R | G | D | U | Reserved | Hop count |
|-----------------------------|---|---|---|---|---|----------|-----------|
| RREQ ID                     |   |   |   |   |   |          |           |
| Destination IP Address      |   |   |   |   |   |          |           |
| Destination Sequence Number |   |   |   |   |   |          |           |
| Originator IP Address       |   |   |   |   |   |          |           |
| Originator Sequence Number  |   |   |   |   |   |          |           |
| Broken next hop             |   |   |   |   |   |          |           |

ภาพประกอบ 3-6 ลักษณะของข้อความ RREQ [9]

ขั้นตอนของการทำ Local repair เริ่มจากการนำข้อมูลที่ได้จากระบวนการส่งข้อความตรวจสอบโหนดข้างเคียงและกระบวนการส่งข้อความ RREP จากเส้นทางที่ใช้งานได้มาใช้งาน โดยการเพิ่มโหนดถัดไปที่ใช้ในการซ่อมแซม (Broken next hop) ไว้ในข้อความ RREQ ซึ่งค่าของโหนดถัดไปที่ใช้ในการซ่อมแซมจะเป็นค่าเดียวกันกับค่าที่ได้ทำการเปรียบเทียบระหว่างค่าของโหนดถัดจากโหนดถัดไปและค่าหมายเลขของโหนดข้างเคียงของโหนดข้างเคียง (Neighbor of neighbor node address) ว่ามีค่าเท่ากันหรือไม่ หากมีค่าเท่ากันหมายความว่าโหนดถัดจากโหนดถัดไปนั้นมีข้อมูลเส้นทางที่ต้องการ ซึ่งเมื่อทำการเปรียบเทียบ โหนดที่พบการเชื่อมต่อเสียหายจะทำการส่งหมายเลข RREQ แบบปลายทางเดียว โดยมีเป้าหมายในการส่งไปยังโหนดที่มีค่าเท่ากับค่าของหมายเลขโหนดข้างเคียงที่ถูกพบว่ามีค่าหมายเลขของโหนดข้างเคียงของโหนดข้างเคียงเท่ากับค่าหมายเลขโหนดถัดจากโหนดถัดไป โดยค่าของหมายเลขโหนดถัดไปที่ใช้ในการซ่อมแซมจะมีค่า



เป็นหมายเลขของโหนดถัดจากโหนดถัดไป จากนั้นจึงทำการส่งข้อความ RREQ แบบปลายทางเดียวดังแสดงในภาพประกอบ 3-7



ภาพประกอบ 3-7 ลักษณะการทำงานข้อความ RREQ แบบปลายทางเดียว

ลักษณะการส่งข้อความ RREQ แบบปลายทางเดียว เริ่มจากเมื่อโหนด 1 มีความต้องการส่งข้อมูลไปยังโหนด 7 และกระบวนการค้นหาเส้นทางเสร็จสิ้น จะได้เส้นทางที่ใช้งานได้คือ 1-2-3-5-6-7 แต่เมื่อโหนด 2 ตรวจพบว่าเกิดการเชื่อมต่อเสียหายขึ้นระหว่างตนเองกับโหนด 3 โหนด 2 จะเข้าสู่กระบวนการซ่อมแซมเส้นทางแบบ Local repair โดยข้อมูลที่โหนด 2 มีอยู่คือข้อมูลหมายเลขโหนดข้างเคียง, หมายเลขของโหนดข้างเคียงของโหนดข้างเคียง, หมายเลขของโหนดถัดไปและหมายเลขของโหนดถัดจากโหนดถัดไป โหนด 2 จะทำการเปรียบเทียบค่าของหมายเลขของโหนดข้างเคียงของโหนดข้างเคียงในแต่ละโหนดข้างเคียงกับหมายเลขของโหนดถัดจากโหนดถัดไปว่ามีค่าเท่ากันหรือไม่ ในกรณีนี้ค่าของหมายเลขโหนดข้างเคียงของโหนดข้างเคียงของโหนดข้างเคียง 4 มีค่าเท่ากับหมายเลขของโหนดถัดจากโหนดถัดไป ในที่นี้คือ 5 ซึ่งแสดงว่าโหนดตัวที่อยู่ใกล้เคียงกับตนเอง ในที่นี้คือโหนด 4 ซึ่งมีโหนดข้างเคียงที่มีข้อมูลเส้นทางที่ตนเองต้องการ ดังนั้นโหนด 2 จะทำการส่งข้อความ RREQ แบบปลายทางเดียว โดยมีหมายเลขโหนดในการส่งเป็นโหนด 4 และมีค่าหมายเลขโหนดถัดไปที่ใช้ในการซ่อมแซมภายในข้อความ RREQ มีค่าเท่ากับโหนด 5 และเมื่อโหนด 2 ส่งข้อความ RREQ แบบปลายทางเดียวไปยังโหนด 4 จะเริ่มทำการตรวจสอบข้อมูลภายในข้อความ RREQ พบว่าค่าของหมายเลขโหนดถัดไปที่ใช้ในการซ่อมแซมมีค่าเท่ากับหมายเลขโหนดข้างเคียงของตนเอง โหนด 4 จะทำการส่งข้อความ RREQ แบบปลายทางเดียวต่อไปยังโหนด 5 และเมื่อโหนด 5 ได้รับข้อความ RREQ ดังกล่าวโหนด 5 ซึ่งมีข้อมูลเส้นทางในการส่งข้อมูลไปยังโหนดปลายทางหรือโหนด 7 จะทำการส่งข้อความ RREP กลับไปยัง

โหนด 4 และโหนด 4 จะทำการส่งข้อความ RREP กลับไปยังโหนด 2 ข้อมูลเส้นทางที่โหนด 2 ต้องการก็จะได้รับ และโหนด 2 จะทำการส่งข้อมูลด้วยเส้นทางใหม่ที่ได้จากกระบวนการดังกล่าว คือ 1-2-4-5-6-7

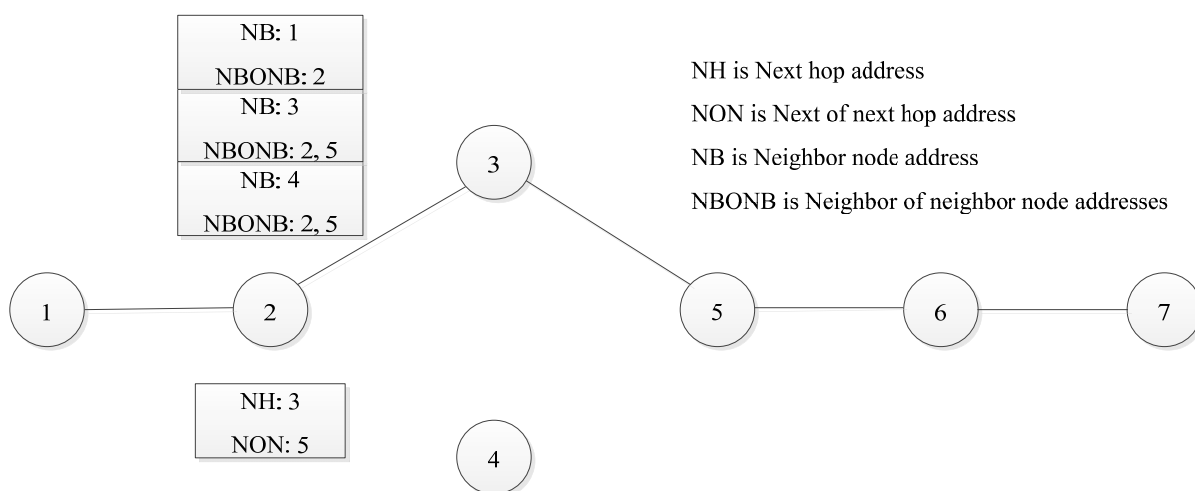
### 3.2 การปรับปรุงเส้นทางให้มีระยะทางที่สั้นกว่าเส้นทางที่ใช้งานเดิม

ในโพรโทคอล AODV ดังเดิมหากเส้นทางไม่มีความเสียหายหรือยังคงสามารถส่งข้อมูลด้วยเส้นทางเดิม โหนดจะไม่ส่งหรือค้นหาเส้นทางใหม่ แม้ว่าในบางกรณีอาจจะมีโหนดบางโหนดเคลื่อนที่แล้วทำให้มีโอกาสที่จะส่งข้อมูลด้วยจำนวนโหนดที่ลดลง ซึ่งในวิทยานิพนธ์นี้ได้ทำการพัฒนาการค้นหาเส้นทางที่สั้นลงและคาดว่าจะสามารถลดค่าหน่วยเวลาได้ใน 3 กรณี ซึ่งรูปแบบเครือข่ายทั้ง 3 กรณีนี้จะถูกนำไปใช้ในการทดสอบด้วยวิธีจำลองการทำงานเครือข่ายในบทที่ 4 ด้วย โดยรูปแบบทั้ง 3 มีรายละเอียดดังต่อไปนี้

- เมื่อโหนดภายในเส้นทางที่ใช้งานได้เคลื่อนที่มีทิศทางไปยังโหนดต้นทาง
- เมื่อโหนดใหม่เคลื่อนที่มาอยู่ระหว่างโหนดปลายทางกับเส้นทางที่ใช้งานได้
- เมื่อโหนดต้นทางเคลื่อนที่ไปในทิศทางของโหนดปลายทางผ่านทางเส้นทางที่ใช้งานได้

#### 3.2.1 เมื่อโหนดภายในเส้นทางที่ใช้งานได้เคลื่อนที่โดยมีทิศทางไปยังโหนด

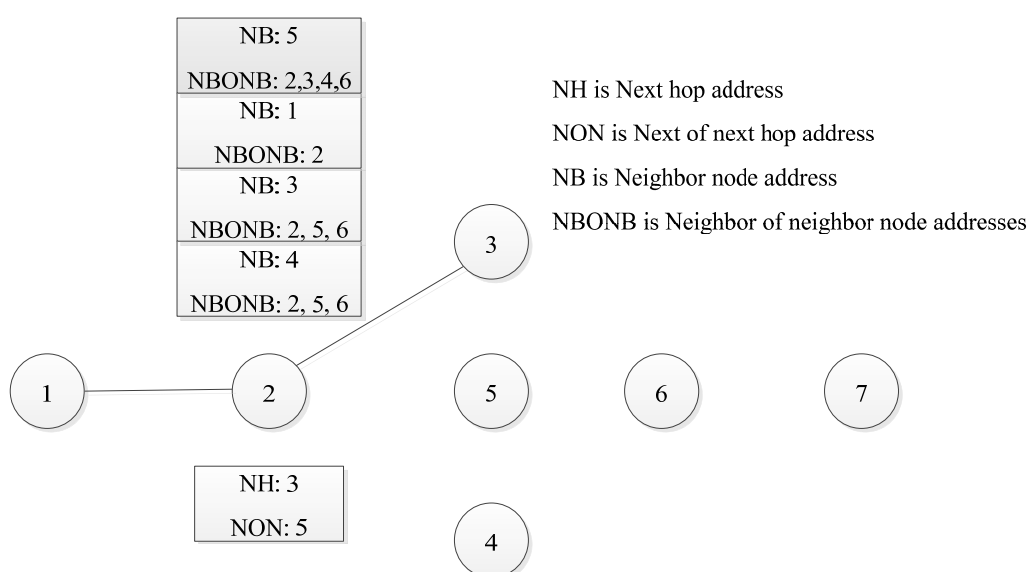
##### ต้นทาง



ภาพประกอบ 3-8 รูปแบบการเชื่อมต่อตอนเริ่มต้นของระบบ

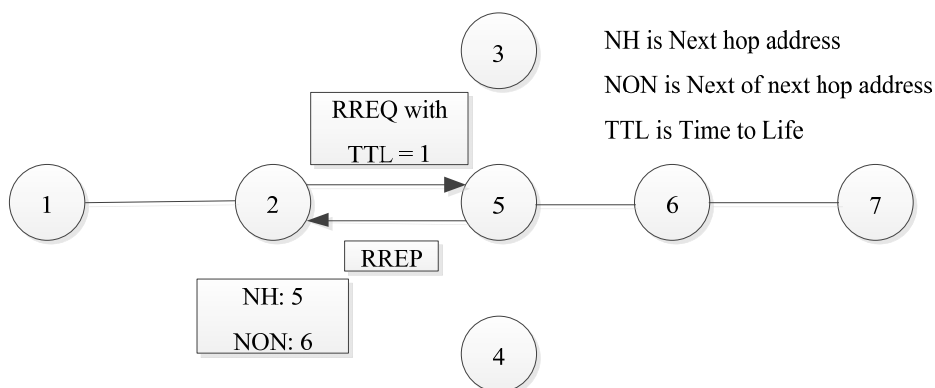
ภาพประกอบ 3-8 เป็นรูปแบบการเชื่อมต่อในเวลาเริ่มต้นของเครือข่ายโดยโหนด 1 จะส่งข้อมูลไปยังโหนด 7 ผ่านทางเส้นทาง 1-2-3-5-6-7 และในกรณีนี้ให้ความสนใจเมื่อโหนด 2

ซึ่งมีการจัดเก็บข้อมูลเกี่ยวกับโหนดปลายทาง 7 คือมีหมายเลขของโหนดถัดไปคือ 3 และมีหมายเลขของโหนดถัดจากโหนดถัดไป 5 นอกเหนือจากนี้โหนด 2 มีการจัดเก็บค่าหมายเลขโหนดข้างเคียงและหมายเลขโหนดข้างเคียงของโหนดข้างเคียงดังนี้มีหมายเลขโหนดข้างเคียงเป็น 1, 3, 4 และมีหมายเลขโหนดข้างเคียงของโหนดข้างเคียงของ 1 เป็น 2 มีหมายเลขโหนดข้างเคียงของโหนดข้างเคียงของ 3 เป็น 2, 5 และมีหมายเลขโหนดข้างเคียงของโหนดข้างเคียงของ 4 เป็น 2, 5 และเมื่อเวลาผ่านไปโหนด 5, 6 และ 7 เคลื่อนที่เข้ามาโหนดต้นทาง ทำให้โครงสร้างการเชื่อมต่อเปลี่ยนไปเป็นดังแสดงในภาพประกอบ 3-9



ภาพประกอบ 3-9 การเคลื่อนที่ของโหนดภายในเส้นทางที่ใช้งานได้มีทิศทางไปยังโหนดต้นทาง

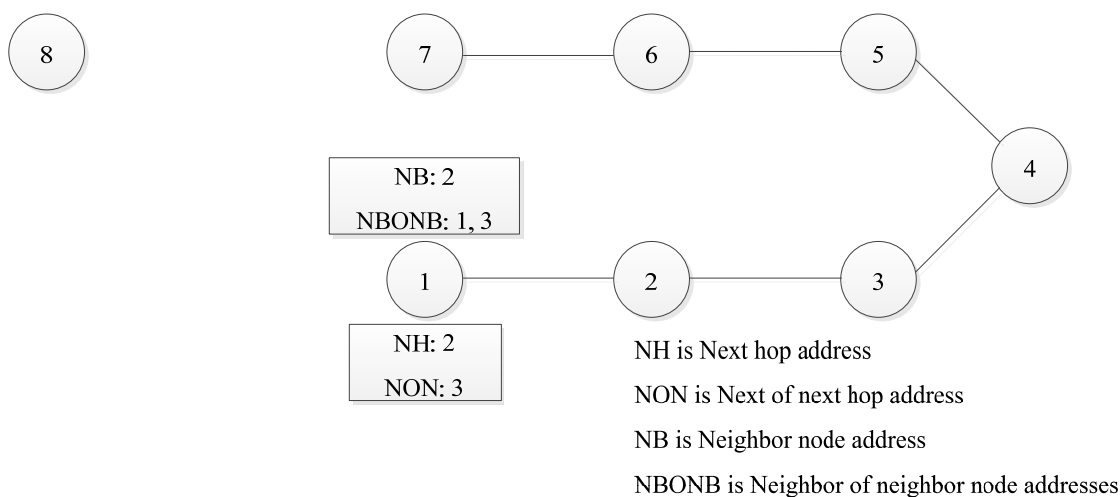
การเคลื่อนที่ของโหนดภายในเส้นทางที่ใช้งานได้ในที่นี้ได้แก่โหนด 5, 6, 7 มีทิศทางไปยังโหนดต้นทาง เมื่อโหนด 2 ตรวจสอบได้ว่าโหนดในเส้นทางที่ใช้งานได้ดังกล่าวเป็นโหนดถัดจากโหนดถัดไปเคลื่อนที่เข้ามาอยู่ในระยะที่สามารถส่งข้อมูลได้ โดยใช้การตรวจสอบจากกระบวนการของการรับข้อความตรวจสอบโหนดข้างเคียง ในที่นี้หมายถึงโหนด 2 สามารถรับข้อความตรวจสอบโหนดข้างเคียงจากโหนด 5 ได้และเมื่อโหนด 2 ทำการพิจารณาข้อมูลในตารางเส้นทางพบว่าโหนดถัดจากโหนดถัดไปมีค่าเป็นโหนด 5 ซึ่งเคลื่อนที่เข้ามาใกล้กับตนเอง กระบวนการในการปรับปรุงข้อมูลเส้นทางที่สั้นกว่าเดิมจึงเริ่มขึ้นดังแสดงในภาพประกอบ 3-10



ภาพประกอบ 3-10 การขอเส้นทางเมื่อโหนดภายในเส้นทางที่ใช้งานได้มีทิศทางไปยังโหนดต้นทาง

ขั้นตอนการปรับปรุงข้อมูลเส้นทางดังกล่าวกระทำโดยโหนด 2 เมื่อได้รับข้อความตรวจสอบโหนดข้างเคียงจากโหนดที่ตนเองบันทึกไว้ว่าเป็นโหนดถัดจากโหนดถัดไป (ในที่นี้คือโหนด 5) โหนด 2 จึงทำการส่งข้อความ RREQ ด้วยค่า TTL = 1 และใช้วิธีการส่งเป็นแบบปลายทางเดียวไปยังโหนด 5 เมื่อโหนดดังกล่าวได้รับข้อความ RREQ จะทำการตอบกลับด้วยข้อความ RREP ไปยังโหนด 2 จากนั้นโหนด 2 จะทำการปรับปรุงข้อมูลเส้นทางที่ใช้ในการส่งข้อมูลไปยังโหนด 7 คือมีค่าหมายเลขโหนดถัดไปคือ 5 และมีค่าหมายเลขโหนดถัดจากโหนดถัดไปคือ 6 กระบวนการดังกล่าวจะช่วยลดจำนวนโหนดที่ใช้ในการส่งข้อมูล

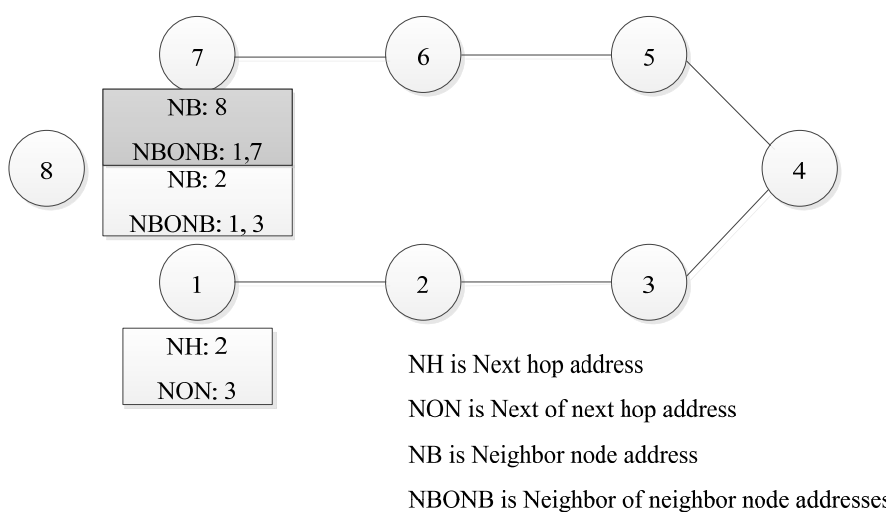
### 3.2.2 เมื่อโหนดใหม่เคลื่อนที่มาอยู่ระหว่างโหนดปลายทางกับเส้นทางที่ใช้งานได้



ภาพประกอบ 3-11 รูปแบบการเชื่อมต่อของเส้นทางที่ใช้งานได้

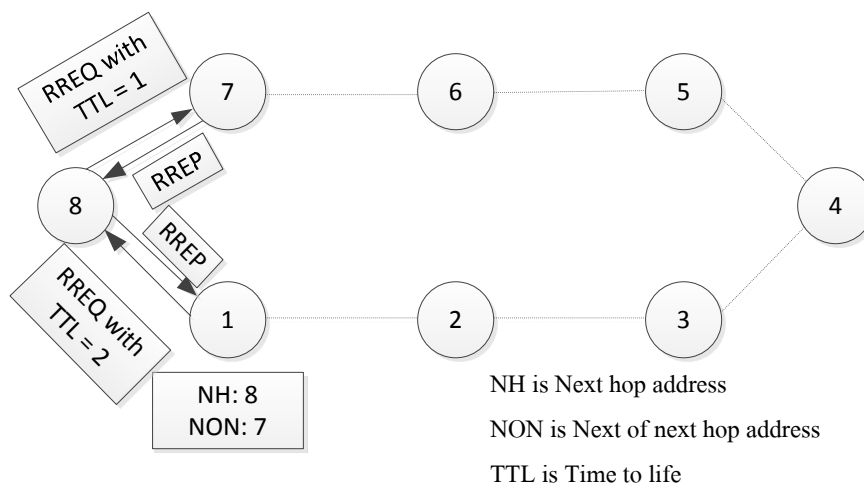
จากภาพประกอบ 3-11 รูปแบบการเชื่อมต่อของเส้นทางที่ใช้งานได้ที่โหนด 1 มีการส่งข้อมูลไปยังโหนดปลายทาง 7 ในที่นี้ทำการพิจารณาข้อมูลตารางเส้นทางภายในโหนด 1

พบว่าค่าหมายเลขโหนดถัดไปคือ 2 และหมายเลขโหนดถัดจากโหนดถัดไปคือ 3 นอกจากนี้ยังมีการเก็บค่าหมายเลขโหนดข้างเคียงคือ 2 และหมายเลขโหนดข้างเคียงของโหนดข้างเคียงคือ 1 และ 3 ซึ่งเส้นทางการทำงานในการส่งข้อมูลจากโหนด 1 ไปยังโหนด 7 เป็นดังนี้ 1-2-3-4-5-6-7 เมื่อเวลาผ่านไปมีโหนดเคลื่อนที่เข้ามาอยู่ระหว่างโหนดต้นทางและโหนดปลายทางดังแสดงในภาพประกอบ 3-12



ภาพประกอบ 3-12 โหนดใหม่เคลื่อนที่มาอยู่ระหว่างโหนดปลายทางกับเส้นทางที่ใช้งานได้

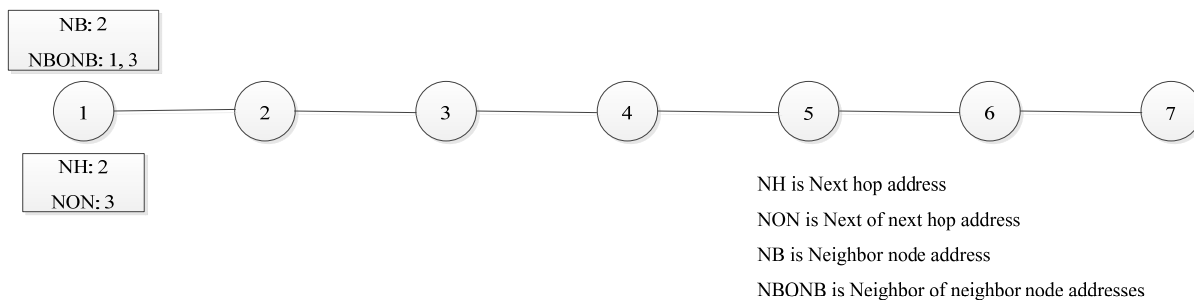
ในกรณีการเชื่อมต่อดังภาพประกอบ 3-12 พบว่ามีโหนด 8 เคลื่อนที่เข้ามาอยู่ระหว่างโหนด 1 ซึ่งเป็นโหนดต้นทางกับโหนด 7 ซึ่งเป็นโหนดปลายทางเมื่อโหนด 1 ได้รับข้อความตรวจสอบโหนดข้างเคียงจากโหนด 8 ที่มีข้อมูลค่าโหนดข้างเคียงของโหนดข้างเคียงคือ 1 และ 7 โหนด 1 จะทำการปรับปรุงข้อมูลเส้นทางใหม่ทันที เพื่อลดจำนวนโหนดที่ใช้ในการส่งข้อมูลดังแสดงในภาพประกอบ 3-13



ภาพประกอบ 3-13 การร้องขอข้อมูลเส้นทางผ่านทางโหนดใหม่ที่เข้ามาอยู่ระหว่างโหนดปลายทาง และเส้นทางที่ใช้งานได้

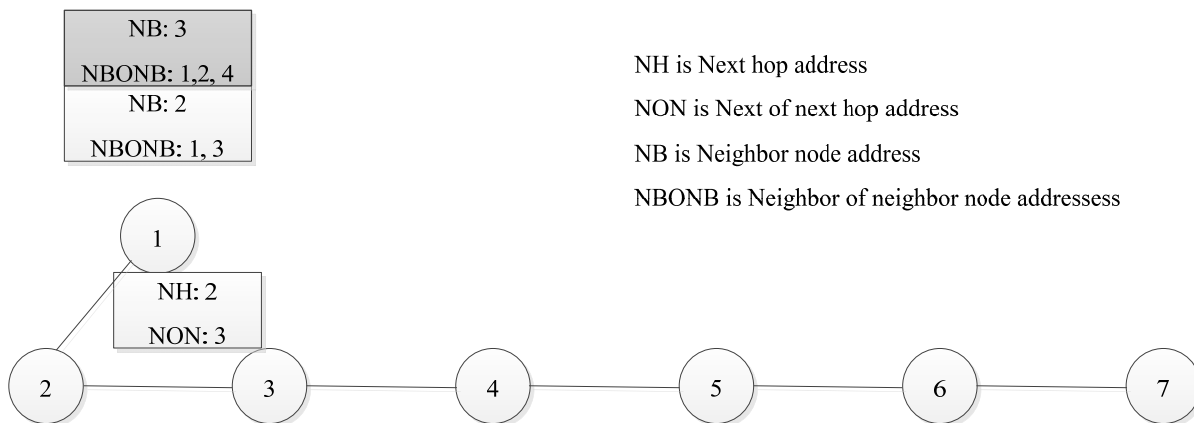
จากภาพประกอบ 3-13 แสดงการร้องขอข้อมูลเส้นทางเมื่อเกิดกรณีที่โหนดใหม่เข้ามาอยู่ระหว่างโหนดปลายทางกับโหนดที่อยู่ในเส้นทางที่ใช้งานได้ เมื่อโหนดใหม่มีค่าหมายเลขของโหนดข้างเคียงเป็นโหนดปลายทาง โหนดตัวที่ตรวจสอบได้จะส่งข้อความ RREQ ด้วยค่า TTL = 2 และใช้ลักษณะการส่งเป็นแบบปลายทางเดียวไปยังโหนดใหม่ที่เข้ามาในเครือข่าย หลังจากที่โหนดใหม่ได้รับข้อความ RREQ โหนดดังกล่าวจะตรวจสอบความต้องการของข้อความ RREQ ว่าความต้องการส่งข้อมูลไปยังโหนดปลายทางดังกล่าวเป็นโหนดข้างเคียงของตนเองหรือไม่ ถ้าใช่จะทำการส่งข้อความ RREQ แบบปลายทางเดียวไปยังโหนดปลายทาง และเมื่อโหนดปลายทางได้รับข้อความ RREQ โหนดปลายทางจะทำการส่งข้อความ RREP ตอบกลับไปยังโหนดใหม่ดังกล่าว โหนดในเส้นทางที่ใช้งานได้ที่พบเหตุการณ์ดังกล่าวก็จะสามารถทำการส่งข้อมูลไปยังปลายทางด้วยเส้นทางที่สั้นลงได้ ดังในภาพประกอบ 3-13 พบว่าโหนด 1 สามารถส่งข้อมูลไปยังโหนด 7 ผ่านทางโหนด 8 ซึ่งใช้จำนวนโหนดที่ใช้ในการส่งข้อมูลเท่ากับ 2 ซึ่งน้อยกว่าเส้นทางเดิมที่มีความต้องการใช้จำนวนโหนดที่ใช้ในการส่งข้อมูลถึง 6

### 3.2.3 เมื่อโหนดต้นทางเคลื่อนที่ไปในทิศทางของโหนดปลายทางผ่านทางเส้นทางที่ใช้งานได้



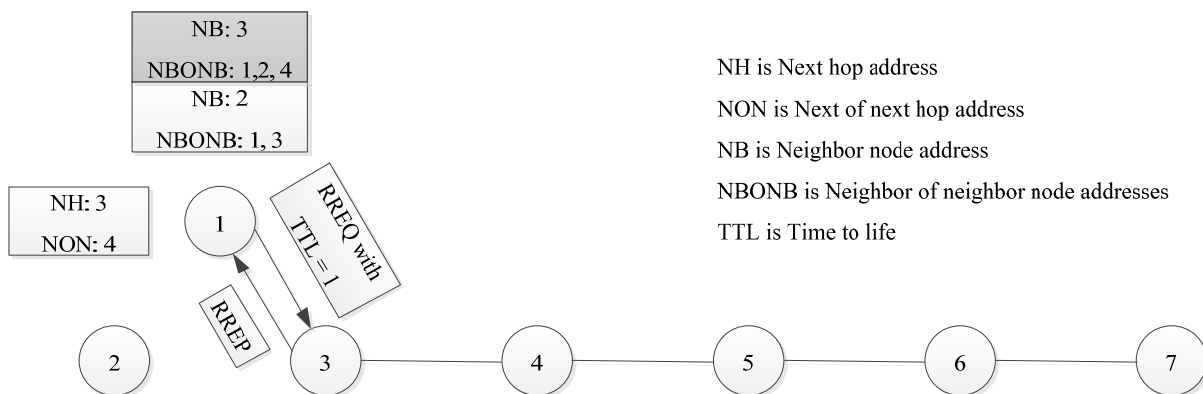
ภาพประกอบ 3-14 รูปแบบการเชื่อมต่อตั้งต้นก่อนที่โหนดต้นทางเคลื่อนที่

ภาพประกอบ 3-14 โหนด 1 เป็นโหนดต้นทาง ต้องการส่งข้อมูลไปยังโหนดปลายทางหมายเลข 7 ผ่านทางเส้นทาง 1-2-3-4-5-6-7 โดยในที่นี้แสดงให้เห็นว่าโหนดถัดไปเพื่อใช้ในการส่งข้อมูลไปยังโหนดปลายทาง 7 ซึ่งโหนด 1 เก็บค่ามีค่าเป็น 2 และโหนดถัดจากโหนดถัดไปมีค่าเป็น 3 และมีค่าโหนดข้างเคียงของโหนดข้างเคียงคือ 1 และ 3



ภาพประกอบ 3-15 การเคลื่อนที่ของโหนดต้นทางไปในทิศทางเส้นทางที่ใช้งานได้

เมื่อมีการเคลื่อนที่ของโหนดต้นทางไปในทิศทางของเส้นทางที่ใช้งานได้ ในที่นี้คือโหนดต้นทางหมายเลข 1 เคลื่อนที่เข้าไปในทิศทางของโหนดในเส้นทางที่ใช้งานได้คือโหนด 3 ซึ่งค่าหมายเลขโหนดถัดจากโหนดถัดไปที่โหนด 1 เก็บค่าไว้มีค่าเป็น 3 ดังนั้นโหนด 1 จึงเริ่มกระบวนการในการร้องขอเส้นทางที่สั้นลงดังแสดงในภาพประกอบ 3-16



ภาพประกอบ 3-16 การร้องขอข้อมูลเส้นทางใหม่

เมื่อมีการร้องขอเส้นทางที่สั้นกว่าโหนด 1 จะทำการส่งข้อความ RREQ ด้วยค่า TTL = 1 ไปยังโหนด 3 ถ้าโหนด 3 มีข้อมูลเส้นทางเพื่อใช้ในการส่งข้อมูลไปยังโหนดปลายทางหมายเลข 7 จะทำการตอบกลับด้วยข้อความ RREP ทำให้เส้นทางใหม่ที่เกิดขึ้นมีค่าจำนวนโหนดที่ใช้ในการส่งข้อมูลที่น้อยลงกว่าเดิม



## บทที่ 4

### ผลการทดสอบ

บทนี้เป็นการแสดงและวิเคราะห์ผลการทดสอบโพรโทคอล AODV ที่ได้รับการพัฒนาในงานวิจัยนี้ โดยแบ่งเนื้อหาออกเป็น 4 ส่วนดังนี้ 1) การหาค่าจำนวนโหนดที่เหมาะสมต่อพื้นที่ที่ทำการทดสอบ 2) การหาค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงที่เหมาะสม 3) การเปรียบเทียบโพรโทคอล AODV แบบดั้งเดิมกับโพรโทคอล AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair และ 4) การเปรียบเทียบโพรโทคอล AODV แบบดั้งเดิมกับโพรโทคอล AODV ที่ทำการพัฒนาการซ่อมแซมเส้นทางแบบ Local repair และการเลือกใช้เส้นทางที่ดีที่สุด

พารามิเตอร์ต่างๆ ที่ใช้ในการทดสอบจะถูกพิจารณาบนสมมติฐานที่ต้องการนำเครือข่ายเซนเซอร์ไร้สายไปใช้ในการเฝ้าระวังสุขภาพของสถานพยาบาล เพื่อช่วยในการดูแลผู้ป่วยที่เข้ารับการรักษาภายในสถานพยาบาล ขนาดกลางที่มีพื้นที่กว้าง 200 เมตร ขนาดความยาว 200 เมตร และให้โหนดในระบบเคลื่อนที่ได้ (ติดกับตัวผู้ป่วย) ด้วยความเร็ว 1 เมตรต่อวินาที โดยในระบบจะมีจำนวนการติดต่อสื่อสารคิดเป็นร้อยละ 10 ของจำนวนโหนดทั้งหมดในสถานพยาบาล

ในการทดสอบประสิทธิภาพของโพรโทคอล AODV ที่พัฒนาขึ้นใช้งานในเครือข่ายเซนเซอร์ไร้สาย จะพิจารณาจาก 3 ค่าดังต่อไปนี้

- อัตราความสำเร็จในการส่งข้อมูล (Packet Delivery Ratio, PDR) คือค่าความสำเร็จในการส่งข้อมูลของโหนดในเครือข่าย โดยคำนวณได้จากจำนวนแพ็กเก็ตที่โหนดปลายทางได้รับต่อจำนวนแพ็กเก็ตที่โหนดต้นทางทำการส่ง
- ค่าหน่วงเวลา (Delay) คือค่าความล่าช้าในการส่งแพ็กเก็ตในเครือข่าย ในที่นี้จะ เป็นค่าเฉลี่ยของค่าหน่วงเวลาที่ได้จากการการทำงานของทุกๆ โหนดในเครือข่าย โดยคำนวณได้จากเวลาที่แพ็กเก็ตถึงปลายทางลบด้วยเวลาที่แพ็กเก็ตเริ่มทำการส่งออกจากโหนดต้นทาง
- ค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูล (Normalized Routing Load, NRL) คือค่าที่ใช้ในการเปรียบเทียบปริมาณแพ็กเก็ตที่ใช้ในการค้นหาเส้นทางเพื่อใช้ในการส่งข้อมูล สามารถคำนวณได้จากอัตราส่วนระหว่างข้อความการจัดการเส้นทางต่อแพ็กเก็ตที่โหนดปลายทางได้รับสามารถคำนวณได้จากสมการ 
$$NRL = (\text{จำนวนRREQ} + \text{จำนวนRREP} + \text{จำนวนRRER}) / \text{จำนวนแพ็กเก็ตที่โหนดปลายทางได้รับ}$$

#### 4.1 การหาค่าจำนวนโหนดที่เหมาะสมต่อพื้นที่ที่ทำการทดสอบ

ในส่วนของการหาค่าจำนวนโหนดที่เหมาะสมต่อพื้นที่ที่ทำการทดสอบเพื่อให้ทราบถึงจำนวนโหนดที่เหมาะสมที่สุดในการทำงานของโพรโทคอล AODV แบบดั้งเดิม ซึ่งมีการกำหนดพารามิเตอร์พื้นฐานสำหรับจำลองการทำงานของเครือข่ายดังแสดงในตารางที่ 4-1

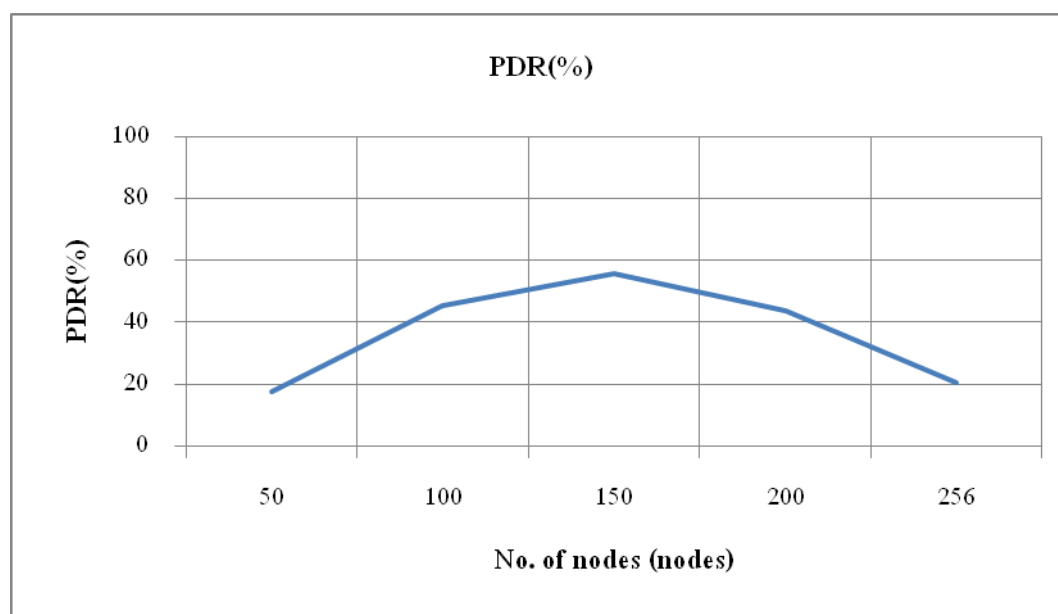
ตารางที่ 4-1 พารามิเตอร์พื้นฐานเพื่อใช้ในการหาจำนวนโหนดที่เหมาะสมต่อพื้นที่ที่ทดสอบ

| พารามิเตอร์               | ค่าที่ใช้งาน                    |
|---------------------------|---------------------------------|
| Mac Layer                 | Mac/802_15_4                    |
| Network Interface         | Phy/WirlessPhy/802_15_4         |
| Routing protocol          | Defaulted AODV routing protocol |
| No. of nodes              | 50, 100, 150, 200, 256          |
| HELLO_INTERVAL            | 1 seconds(Defaulted AODV)       |
| Number of simulation runs | 10 times                        |
| Simulation time           | 1000 seconds                    |
| X dimension               | 200 meters                      |
| Y dimension               | 200 meters                      |
| Packet type               | CBR packet                      |
| Connection                | 10% of No. of nodes             |
| Rx range                  | 20 meters                       |
| Tx range                  | 20 meters                       |
| Rx power                  | 47 mW                           |
| Tx power                  | 47 mW                           |
| CBR interval              | 1 second                        |

จากตารางที่ 4-1 เป็นค่าพารามิเตอร์พื้นฐานเพื่อใช้ในการจำลองการทำงานเพื่อหาจำนวนโหนดที่เหมาะสมต่อพื้นที่ที่ทดสอบด้วยการทดสอบในพื้นที่ขนาด 40,000 ตารางเมตร (200เมตร\*200เมตร) โดยการกำหนดให้มีการส่งข้อความตรวจสอบโหนดข้างเคียงในทุกๆ 1 วินาที ซึ่งเป็นค่าดั้งเดิมของโพรโทคอล AODV และมีจำนวนโหนดสูงสุดที่ใช้ในการทดสอบมีค่า 256 โหนด เนื่องจากในที่นี้ทำการพิจารณาเฉพาะเครือข่ายที่มีการใช้งานหมายเลขโหนดแบบ 8 บิต (หมายเลข ID ที่สามารถทำได้บนโหนดจริง) ซึ่งจะสามารถทำการจัดเก็บหมายเลขที่แตกต่างกัน

กันได้สูงสุด 256 หมายเลข และการทดสอบจะใช้โปรโตคอล AODV แบบดั้งเดิม ซึ่งผลลัพธ์ที่ได้จากการจำลองเครือข่ายเป็นดังต่อไปนี้

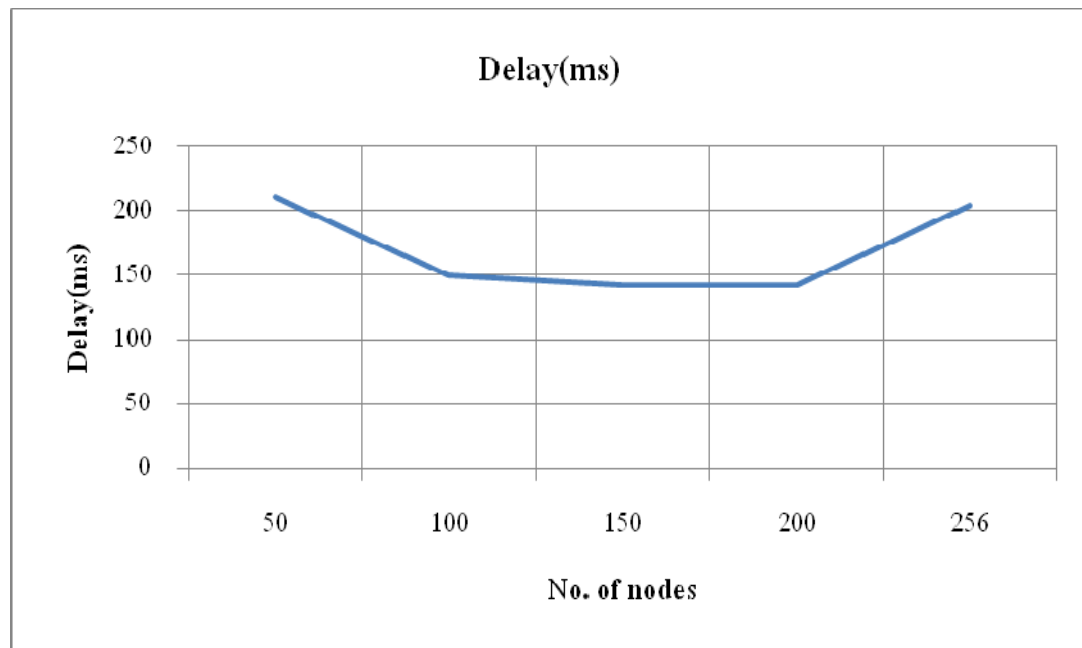
#### 4.1.1 อัตราความสำเร็จในการส่งข้อมูลเมื่อมีการเปลี่ยนแปลงจำนวนโหนดในเครือข่าย



ภาพประกอบ 4-1 กราฟความสัมพันธ์ระหว่างอัตราความสำเร็จในการส่งข้อมูลกับจำนวนโหนดในเครือข่าย

ภาพประกอบ 4-1 แสดงกราฟความสัมพันธ์ระหว่างอัตราความสำเร็จในการส่งข้อมูลที่เปลี่ยนแปลงเมื่อมีการเปลี่ยนแปลงจำนวนโหนดในเครือข่ายเริ่มจาก 50, 100, 150, 200 และ 256 โหนดตามลำดับ อัตราความสำเร็จในการส่งข้อมูลเปลี่ยนแปลงในลักษณะของกราฟพาราโบลาคว่ำคือสามารถให้ค่าอัตราความสำเร็จในการส่งข้อมูลสูงที่สุด 55 เปอร์เซ็นต์ เมื่อมีจำนวนของโหนดในเครือข่าย 150 ตัว โดยที่เมื่อจำนวนโหนดมีค่าน้อยในที่นี้คือ 50 และ 100 อัตราความสำเร็จในการส่งข้อมูลจะมีค่าน้อยด้วยเนื่องจากปริมาณโหนดที่มีอยู่ในเครือข่ายไม่สามารถทำงานครอบคลุมพื้นที่ได้ทั้งหมด ทำให้ในบางกรณีไม่สามารถติดต่อสื่อสารไปยังโหนดที่ต้องการได้ และหากมีจำนวนโหนดในเครือข่ายมากเกินไปในที่นี้คือ 200 และ 256 จะทำให้อัตราความสำเร็จในการส่งข้อมูลลดต่ำลง เนื่องจากในช่วงการติดต่อสื่อสารมีปริมาณโหนดหนาแน่นเกินไป จึงมีโอกาสเกิดการชนกันของข้อมูลได้สูง ทำให้อัตราความสำเร็จในการส่งข้อมูลลดลง ดังนั้นจำนวนโหนดที่กระจายตัวแบบสุ่มบนพื้นที่ 200x200 ตารางเมตรจะให้ค่าอัตราความสำเร็จในการส่งข้อมูลที่ดีที่สุด

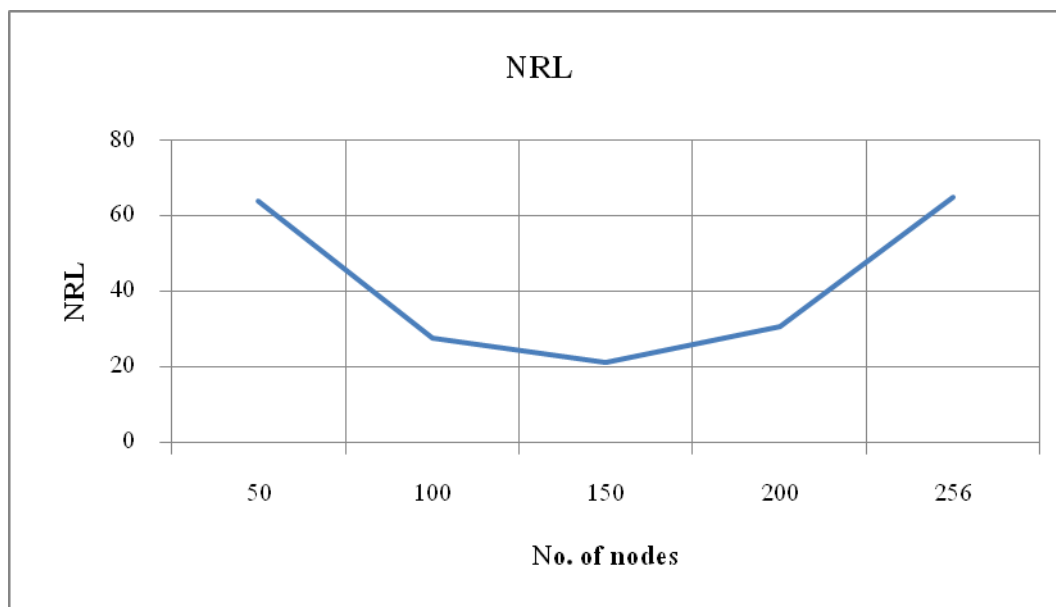
#### 4.1.2 ค่าหน่วยเวลาเมื่อมีการเปลี่ยนแปลงจำนวนโหนดในเครือข่าย



ภาพประกอบ 4-2 กราฟความสัมพันธ์ระหว่างค่าหน่วยเวลากับจำนวนโหนดในเครือข่าย

จากภาพประกอบ 4-2 เป็นกราฟแสดงความสัมพันธ์ระหว่างค่าหน่วยเวลากับจำนวนโหนดในเครือข่ายอยู่ในรูปพาราโบลาแบบหงาย มีค่าหน่วยเวลาที่น้อยที่สุดเมื่อมีจำนวนโหนดในช่วงตั้งแต่ 150 -200 โหนด ซึ่งเมื่อค่าจำนวนโหนดมีค่าน้อยนั่นคือที่จำนวนโหนดมีค่า 50 โหนด ค่าหน่วยเวลาที่เกิดขึ้นจะมีค่าสูง สาเหตุเนื่องจากโหนดไม่สามารถทำการติดต่อสื่อสารได้ทันทีเมื่อมีความต้องการ ซึ่งต้องใช้เวลาเพื่อทำการค้นหาเส้นทางหรือต้องใช้เวลาให้โหนดเคลื่อนที่เข้ามาอยู่ในช่วงที่สามารถติดต่อสื่อสารได้ และเมื่อมีจำนวนโหนดมากเกินไปในที่นี้คือเมื่อจำนวนโหนดมีค่าเท่ากับ 256 โหนด ค่าหน่วยเวลาจะมีค่ามากขึ้นเนื่องจากเมื่อมีความหนาแน่นมากขึ้นโอกาสที่โหนดส่งข้อมูลและเกิดการชนกันจึงมีสูง จึงทำให้โหนดต้องพยายามทำการส่งข้อมูลใหม่ ซึ่งทำให้การส่งข้อมูลดังกล่าวล่าช้ายิ่งขึ้น ดังนั้นจำนวนโหนดที่ 150 ในพื้นที่ 200x200 ตารางเมตร นอกจากจะให้ค่าอัตราความสำเร็จในการส่งข้อมูลที่ดีที่สุดแล้วยังจะให้ค่าหน่วยเวลาที่น้อยที่สุดอีกด้วย

#### 4.1.3 ปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลเมื่อมีการเปลี่ยนแปลงจำนวนโหนดในเครือข่าย



ภาพประกอบ 4-3 กราฟความสัมพันธ์ระหว่างค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลกับจำนวนโหนดในเครือข่าย

ในส่วนนี้เป็นการหาความสัมพันธ์ระหว่างปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลกับจำนวนโหนดในเครือข่าย พบว่าเป็นความสัมพันธ์ในรูปแบบของกราฟพาราโบลาหงาย ในกรณีที่จำนวนโหนดมีค่าน้อยค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลจะมีค่าสูง เนื่องจากโหนดไม่สามารถติดต่อสื่อสารถึงโหนดที่ต้องการ โหนดจึงพยายามทำการค้นหาเส้นทางอยู่เป็นระยะๆ ประกอบกับการคำนวณค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลจะนำค่าจำนวนในการรับข้อมูลของโหนดปลายทางมาใช้ในการพิจารณา แต่เมื่อจำนวนโหนดในเครือข่ายสูง ค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลจะมีค่าสูงขึ้น เนื่องจากการเกิดการชนกันของข้อมูล จึงจำเป็นต้องเกิดกระบวนการในความพยายามส่งข้อมูลใหม่และเมื่อความพยายามไม่สำเร็จ โหนดจำเป็นต้องซ่อมแซมเส้นทางเพื่อใช้ในการเชื่อมต่อ ซึ่งกระบวนการดังกล่าว โหนดในเครือข่ายจำเป็นต้องมีการส่งข้อความการจัดการเส้นทาง จึงทำให้ปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลมีค่าสูงขึ้น

เมื่อทำการพิจารณาจากค่าอัตราความสำเร็จในการส่งข้อมูลพบว่าจำนวนโหนดที่ให้ค่าที่ดีที่สุดคือ 150 โหนด เมื่อพิจารณาค่าหน่วยเวลาพบว่าจำนวนโหนดที่ทำให้ค่าหน่วยเวลามีค่าน้อยคือ 100, 150 และ 200 โหนด และเมื่อพิจารณาค่าปริมาณข้อความการจัดการเส้นทางต่อการ

รับพบว่ามีค่าที่น้อยที่สุดคือ 150 โหนด ดังนั้นจึงนำจำนวนโหนดเท่ากับ 150 โหนดมาใช้ในการพิจารณาในการจำลองเครือข่ายต่อไป

#### 4.2 การหาค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงที่เหมาะสม

ในส่วนของ การหาค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงที่เหมาะสม โดยจะใช้พารามิเตอร์จำนวนโหนดเท่ากับ 150 โหนด ซึ่งได้มาจากการทดสอบก่อนหน้า ซึ่งการทดสอบเพื่อหาค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงดังกล่าวจะใช้งานโปรโตคอล AODV แบบดั้งเดิม และมีการเปลี่ยนแปลงค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงตั้งแต่ 1 ไปจนถึง 12 วินาที

ตารางที่ 4-2 พารามิเตอร์พื้นฐานในการหาค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงที่เหมาะสม

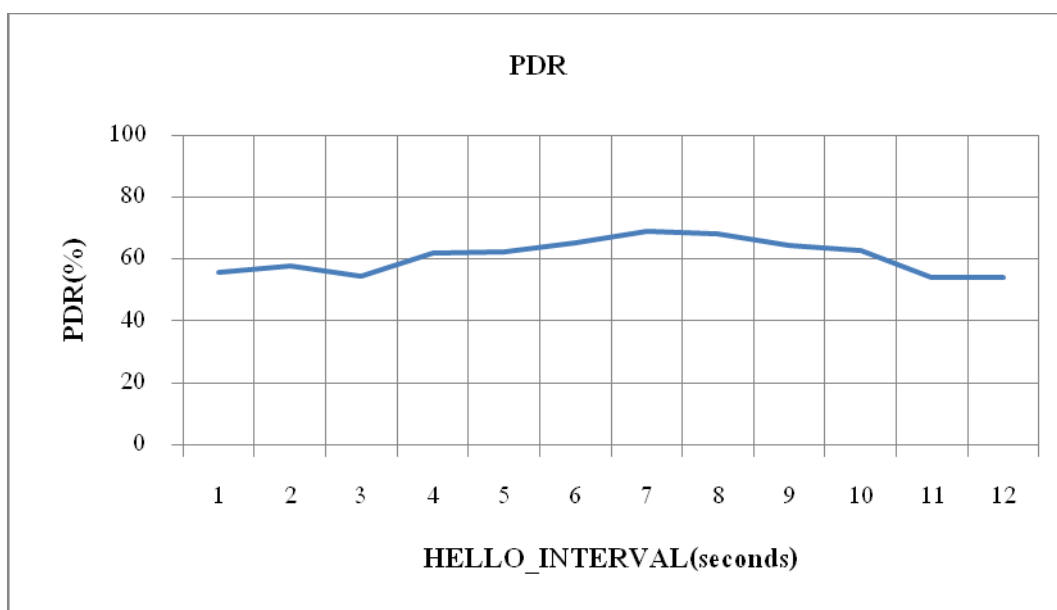
| พารามิเตอร์               | ค่าที่ใช้งาน                    |
|---------------------------|---------------------------------|
| Mac Layer                 | Mac/802_15_4                    |
| Network Interface         | Phy/WirlessPhy/802_15_4         |
| Routing protocol          | Defaulted AODV routing protocol |
| No. of nodes              | 150 nodes                       |
| HELLO_INTERVAL            | 1 – 12 seconds                  |
| Number of simulation runs | 10 times                        |
| Simulation time           | 1000 seconds                    |
| X dimension               | 200 meters                      |
| Y dimension               | 200 meters                      |
| Packet type               | CBR packet                      |
| Connection                | 10% of No. of nodes             |
| Rx range                  | 20 meters                       |
| Tx range                  | 20 meters                       |

ตารางที่ 4-2 พารามิเตอร์พื้นฐานในการหาค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงที่เหมาะสม (ต่อ)

| พารามิเตอร์  | ค่าที่ใช้งาน |
|--------------|--------------|
| Rx power     | 47 mW        |
| Tx power     | 47 mW        |
| CBR interval | 1 second     |

จากตารางที่ 4-2 พารามิเตอร์พื้นฐานเพื่อใช้ในการหาค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงที่เหมาะสม โดยการเปลี่ยนแปลงค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงตั้งแต่ 1 วินาที ไปจนถึง 12 วินาที และกำหนดจำนวนโหนดที่ 150 โหนด ซึ่งเป็นค่าที่ได้จากการทดสอบก่อนหน้า และการทดสอบดังกล่าวจะใช้โปรโตคอล AODV แบบดั้งเดิม ซึ่งผลลัพธ์ที่ได้จากการจำลองเครือข่ายเป็นดังต่อไปนี้

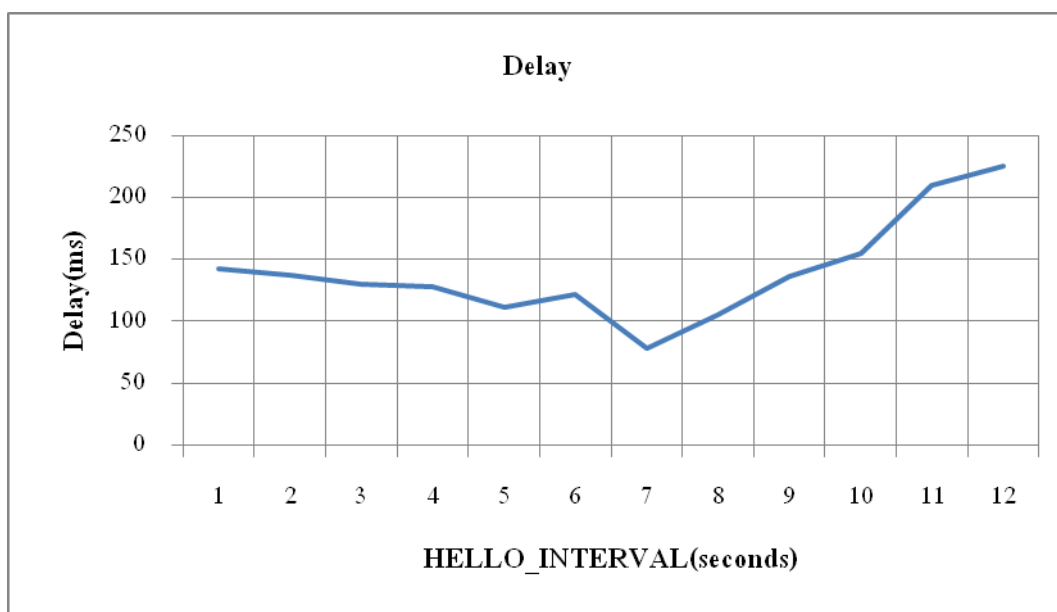
#### 4.2.1 อัตราความสำเร็จในการส่งข้อมูลเมื่อมีการเปลี่ยนแปลงค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียง



ภาพประกอบ 4-4 กราฟความสัมพันธ์ระหว่างอัตราความสำเร็จในการส่งข้อมูลกับค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียง

จากภาพประกอบ 4-4 กราฟความสัมพันธ์ระหว่างค่าอัตราความสำเร็จในการส่งข้อมูลกับค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงพบว่า เมื่อค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงมีค่าน้อยนั่นคือตั้งแต่ 1 วินาทีถึง 5 วินาที ค่าอัตราความสำเร็จในการส่งข้อมูลจะมีค่าน้อยกว่าเล็กน้อยเนื่องจากในกรณีที่ค่าช่วงเวลามีค่าน้อยจะส่งผลให้โหนดมีความถี่ในการส่งข้อความตรวจสอบโหนดข้างเคียงสูง ทำให้ปริมาณข้อความดังกล่าวในเครือข่ายมีมาก ซึ่งอาจส่งผลให้เกิดการชนกันของการส่งข้อมูลขึ้นทำให้ความสำเร็จในการส่งข้อมูลลดต่ำลง และเมื่อค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงมีค่าสูงขึ้นในที่นี้คือที่ 10 วินาทีเป็นต้นไป ค่าอัตราความสำเร็จในการส่งข้อมูลจะมีค่าลดลง เนื่องจากในบางกรณีโหนดข้างเคียงเคลื่อนที่ออกจากกระบวนการติดต่อสื่อสาร แต่โหนดซึ่งเคยจัดเก็บเส้นทางข้อมูลซึ่งใช้โหนดข้างเคียงเป็นโหนดถัดไปไม่ทราบว่าโหนดข้างเคียงดังกล่าวไม่อยู่ในกระบวนการติดต่อสื่อสาร จึงยังคงพยายามติดต่อสื่อสารอยู่ด้วยข้อมูลโหนดข้างเคียงเดิม ซึ่งส่งผลให้อัตราความสำเร็จในการส่งข้อมูลลดต่ำลง

#### 4.2.2 ค่าหน่วงเวลาเมื่อมีการเปลี่ยนแปลงค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียง



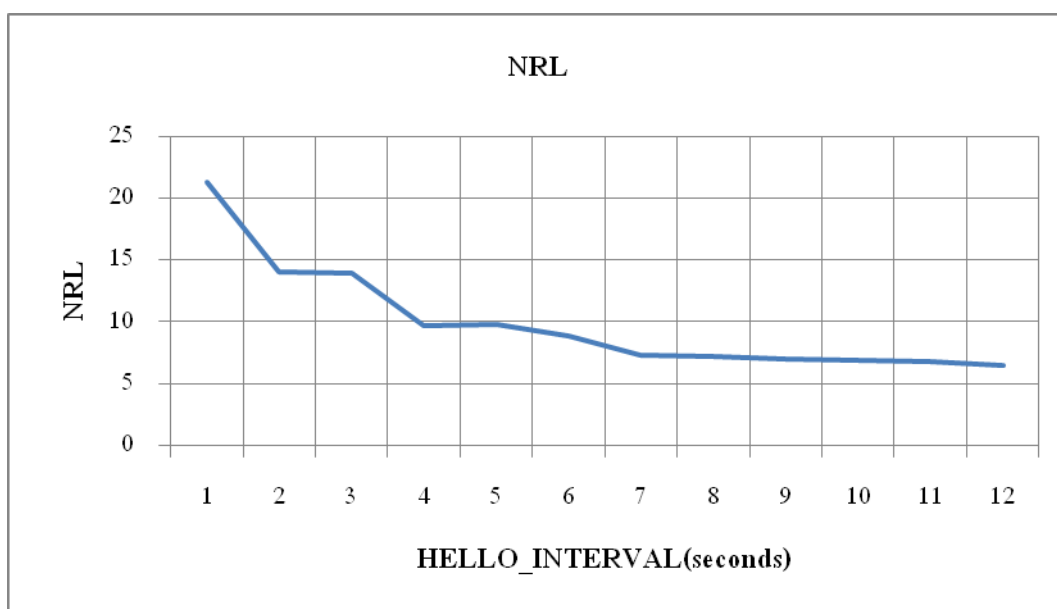
ภาพประกอบ 4-5 กราฟความสัมพันธ์ระหว่างค่าหน่วงเวลากับค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียง

จากกราฟความสัมพันธ์ระหว่างค่าหน่วงเวลากับค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงดังภาพประกอบ 4-5 พบว่าเมื่อค่าช่วงเวลามีค่าน้อยนั่นคือที่ 1



วินาทีถึง 4 วินาที ส่งผลให้ค่าหน่วยเวลาในการส่งข้อมูลสูงเนื่องจากการชนกันของการส่งข้อมูล จึงจำเป็นต้องมีการซ่อมแซมเส้นทางซึ่งทำให้ค่าหน่วยเวลามีค่าเพิ่มขึ้น และเมื่อค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงมีมากขึ้นคือที่ 9 วินาทีเป็นต้นไป ค่าหน่วยเวลามีค่าเพิ่มขึ้นอย่างชัดเจน เนื่องจากโหนดซึ่งได้ทำการจัดเก็บข้อมูลโหนดข้างเคียงกว่าจะทราบว่าโหนดข้างเคียงเคลื่อนที่ออกนอกระยะการติดต่อสื่อสารของตนเองจะต้องใช้เวลาค่อนข้างนาน ซึ่งในระยยะเวลาดังกล่าวโหนดอาจทำการส่งข้อมูลออกไปและโหนดข้างเคียงไม่ได้รับข้อมูลนั้น ทำให้การติดต่อสื่อสารไม่ถูกต้อง ส่งผลให้ต้องพยายามค้นหาเส้นทางใหม่ ซึ่งต้องใช้ระยะเวลาที่ยาวนานยิ่งขึ้น

#### 4.2.3 ค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลเมื่อมีการเปลี่ยนแปลงค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียง



ภาพประกอบ 4-6 กราฟความสัมพันธ์ระหว่างค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลกับค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียง

จากภาพประกอบ 4-6 เป็นกราฟความสัมพันธ์ระหว่างค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลกับค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียง แสดงให้เห็นว่าเมื่อค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงมีค่าน้อย นั่นคือ โหนดจะทำการส่งข้อความตรวจสอบโหนดข้างเคียงถี่มากขึ้น ทำให้เกิดข้อความการจัดการเส้นทางในเครือข่ายมาก ส่งผลให้ค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลมีค่าสูงอย่างเห็นได้ชัด และเมื่อค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงมีค่ามากขึ้น จะทำให้ค่า

ปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลมีค่าลดลง เนื่องจากความถี่ที่โหนดต้องใช้ในการส่งข้อความตรวจสอบ โหนดข้างเคียงมีค่าลดลง

เมื่อทำการพิจารณาจากค่าอัตราความสำเร็จในการส่งข้อมูลพบว่าค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบ โหนดข้างเคียงที่ให้ผลดีที่สุดอยู่ที่ช่วง 7 ถึง 9 วินาที และเมื่อพิจารณาค่าช่วงเวลาพบว่าค่าเวลาน้อยอยู่ที่ 7 วินาที และเมื่อพิจารณาค่าปริมาณข้อความการจัดการเส้นทางต่อการรับพบว่าค่าที่น้อยจะเริ่มจากค่าตั้งแต่ 7 วินาทีเป็นต้นไป ดังนั้นจึงทำการเลือกค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบ โหนดข้างเคียงที่มีค่าเท่ากับ 7 วินาทีมาใช้ในการทดสอบต่อไป

#### 4.3 การเปรียบเทียบโปรโตคอล AODV แบบดั้งเดิมกับโปรโตคอล AODV ที่พัฒนาการซ่อมแซม

##### เส้นทางแบบ Local repair

ในที่นี้ทำการทดสอบระหว่าง AODV แบบดั้งเดิมกับโปรโตคอล AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair โดยการนำพารามิเตอร์ที่ได้ทำการทดสอบก่อนหน้านี้มาใช้ในการทดสอบ ซึ่งค่าพารามิเตอร์แสดงดังตาราง 4-3

ตารางที่ 4-3 พารามิเตอร์พื้นฐานในการเปรียบเทียบโปรโตคอล AODV แบบดั้งเดิมกับโปรโตคอล AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair

| พารามิเตอร์               | ค่าที่ใช้งาน            |
|---------------------------|-------------------------|
| Mac Layer                 | Mac/802_15_4            |
| Network Interface         | Phy/WirlessPhy/802_15_4 |
| No. of nodes              | 150 nodes               |
| HELLO_INTERVAL            | 7 seconds               |
| Number of simulation runs | 10 times                |
| Simulation time           | 1000 seconds            |
| X dimension               | 200 meters              |
| Y dimension               | 200 meters              |
| Packet type               | CBR packet              |
| Connection                | 10% of No. of nodes     |
| Rx range                  | 20 meters               |
| Tx range                  | 20 meters               |

ตารางที่ 4-3 พารามิเตอร์พื้นฐานในการเปรียบเทียบโพรโทคอล AODV แบบดั้งเดิมกับโพรโทคอล AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair (ต่อ)

| พารามิเตอร์         | ค่าที่ใช้งาน |
|---------------------|--------------|
| <b>Rx power</b>     | 47 mW        |
| <b>Tx power</b>     | 47 mW        |
| <b>CBR interval</b> | 1 second     |

จากตารางที่ 4-3 พารามิเตอร์พื้นฐานในการเปรียบเทียบโพรโทคอล AODV แบบดั้งเดิมกับโพรโทคอล AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair ซึ่งค่าจำนวนโหนดและค่าช่วงเวลาที่ใช้ในการส่งข้อความตรวจสอบโหนดข้างเคียงได้มาจากการทดสอบก่อนหน้าซึ่งการทดสอบจะเป็นการสุ่มตำแหน่งของโหนดในพื้นที่และให้มีโหนด 15 โหนดมีการเคลื่อนที่ผลลัพธ์ที่ได้จากการเปรียบเทียบแสดงได้ดังตารางที่ 4-4

ตารางที่ 4-4 แสดงการเปรียบเทียบผลที่ได้จากการทดสอบการทำงานของโพรโทคอล AODV แบบดั้งเดิมกับ AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair

| Value   | AODV แบบดั้งเดิม | AODV ที่พัฒนา Local repair | ผลการเปรียบเทียบ   |
|---|------------------|----------------------------|--------------------|
| <b>PDR(%)</b>                                 | 68.98            | 76.52                      | (+) Better 9.85    |
| <b>Delay(ms)</b>                              | 77.86            | 55.73                      | (+) Faster 22.13   |
| <b>NRL</b>                                    | 7.29             | 5.36                       | (+) Better 1.93    |
| <b>All Energy usage for Sending (mW)</b>      | 11059.66         | 10739.69                   | (+) Better 319.97  |
| <b>All Energy usage for Receiving (mW)</b>    | 16498.46         | 15057.95                   | (+) Better 1440.51 |
| <b>Energy usage for header sending (mW)</b>   | 1955.39          | 1734.16                    | (+) Better 221.23  |
| <b>Energy usage for header receiving (mW)</b> | 7401.88          | 6217.63                    | (+) Better 1184.25 |
| <b>Number of Local Repair</b>                 | 33               | 32.2                       | (+) Better 0.8     |
| <b>Number of Source Repair</b>                | 36.88            | 26.85                      | (+) Better 10.03   |

จากตารางที่ 4-4 แสดงการเปรียบเทียบผลที่ได้จากการทดสอบการทำงานของโพรโทคอล AODV แบบดั้งเดิมกับ AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair เมื่อพิจารณาค่าอัตราความสำเร็จในการส่งข้อมูล (PDR) ที่ได้จาก AODV ที่พัฒนา Local repair มีค่าสูง

กว่า 9.85 เปอร์เซ็นต์ และค่าหน่วงเวลา (Delay) ลดลง 28.24 เปอร์เซ็นต์ ทั้งนี้เนื่องมาจากการทำงานของ AODV ที่มีการพัฒนา Local repair จะใช้ข้อมูลที่ได้จากข้อความตรวจสอบโหนดข้างเคียงเข้ามาช่วยในการทำงาน และเมื่อเกิดกรณีที่การเชื่อมต่อเสียหาย โหนดจะมีข้อมูลของโหนดข้างเคียงทำให้สามารถทำการซ่อมแซมเส้นทางแบบ Local repair ด้วยการส่งแบบปลายทางเดียวแทนการส่งแบบแพร่กระจาย และทำการส่งข้อความร้องขอดังกล่าวไปยังโหนดที่คาดว่าจะมีข้อมูลเส้นทางที่ต้องการ จึงส่งผลให้อัตราความสำเร็จในการส่งข้อมูลมีค่าที่ดีขึ้นและลดระยะเวลาในการซ่อมแซมให้รวดเร็วยิ่งขึ้น ส่วนค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูล (NRL) การทดสอบโพรโทคอล AODV ที่พัฒนา Local repair จะให้ค่าที่ลดลง 26.47 เปอร์เซ็นต์ เนื่องจากเมื่อเครือข่ายมีการซ่อมแซมเส้นทางด้วย Local repair ที่ได้พัฒนาขึ้น โหนดจะใช้วิธีการส่งแบบปลายทางเดียวแทนการส่งแบบแพร่กระจาย ส่งผลให้ปริมาณข้อความการจัดการเส้นทางในเครือข่ายมีจำนวนลดต่ำลง จึงทำให้ค่าดังกล่าวมีค่าลดลง เมื่อทำการพิจารณาพลังงานที่ใช้ในการรับ-ส่งในส่วนของข้อความการจัดการเส้นทางพบว่าโพรโทคอล AODV ที่พัฒนาการทำงานของ Local repair สามารถลดการใช้พลังงานได้ 16.00 เปอร์เซ็นต์ และ 11.31 เปอร์เซ็นต์ ตามลำดับสาเหตุเนื่องจากเมื่อมีการซ่อมแซมเส้นทางแบบ Local repair ที่พัฒนาขึ้น การส่งข้อความร้องขอเส้นทางจะเป็นแบบปลายทางเดียวแทนการส่งแบบแพร่กระจาย ซึ่งลักษณะดังกล่าวจะทำให้ปริมาณการรับ-ส่งต่อข้อความร้องขอเส้นทางมีจำนวนลดลงเมื่อเปรียบเทียบการทำงานแบบดั้งเดิม และเมื่อพิจารณาการใช้พลังงานของการรับ-ส่งที่ใช้ทั้งหมดพบว่ามีค่าลดลง 8.73 เปอร์เซ็นต์ และ 2.89 เปอร์เซ็นต์ ซึ่งสาเหตุที่ลดการใช้พลังงานลงไม่มากเนื่องจาก ในกระบวนการที่ได้ทำการพัฒนา Local repair จะส่งผลให้ความสำเร็จในการส่งข้อมูลดีขึ้น ซึ่งจะทำให้โหนดสามารถทำการรับ-ส่งข้อมูลที่ต้องการได้มากขึ้น จึงทำให้ต้องมีการใช้พลังงานเพื่อส่งข้อมูลที่ต้องการดังกล่าวส่งผลให้การใช้พลังงานในส่วนของรับ-ส่งข้อมูลที่ต้องการมีค่ามากขึ้น ไปด้วย และเมื่อพิจารณาจำนวนครั้งในการซ่อมแซมเส้นทางแบบ Local repair ระหว่างโพรโทคอล AODV แบบดั้งเดิมกับโพรโทคอล AODV ที่พัฒนา Local repair พบว่าไม่แตกต่างกันเนื่องจากเหตุการณ์ที่ทำให้เกิดการซ่อมแซมเส้นทางแบบ Local repair เกิดขึ้นคล้ายกัน แต่หากพิจารณาค่าจำนวนครั้งในการซ่อมแซมเส้นทางแบบ Source repair พบว่าโพรโทคอล AODV ที่พัฒนา Local repair จะมีการซ่อมแซมเส้นทางแบบ Source repair ลดลงถึง 27.19 เปอร์เซ็นต์ เนื่องจากโพรโทคอล AODV ที่พัฒนา Local repair มีโอกาสในการซ่อมแซมเส้นทางแบบ Local repair ได้สำเร็จสูงกว่าโพรโทคอล AODV แบบดั้งเดิม ซึ่งเกิดจากการส่งข้อความ RREQ แบบปลายทางเดียวแทนการส่งแบบแพร่กระจาย ส่งผลให้ลดจำนวนครั้งในการซ่อมแซมเส้นทางแบบ Source repair

#### 4.4 การเปรียบเทียบโพรโทคอล AODV แบบดั้งเดิมกับโพรโทคอล AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair และการเลือกใช้เส้นทางที่สั้นที่สุด

ในการทดสอบดังกล่าวจะใช้งานพารามิเตอร์ค่าเดียวกับตารางที่ 4-3 ผลลัพธ์ที่ได้จากการเปรียบเทียบแสดงได้ดังตารางที่ 4-5

ตารางที่ 4-5 แสดงการเปรียบเทียบผลที่ได้จากการทดสอบการทำงานของโพรโทคอล AODV แบบดั้งเดิมกับ AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair และการเลือกใช้เส้นทางที่สั้นที่สุด

| Value                                  | AODV แบบดั้งเดิม | AODV ที่พัฒนา Local repair และการเลือกใช้เส้นทางที่สั้นที่สุด | ผลการเปรียบเทียบ   |
|--|------------------|---|--------------------|
| PDR(%)                                 | 68.98            | 82.93   | (+) Better 13.95   |
| Delay(ms)                              | 77.86            | 33.19   | (+) Faster 44.67   |
| NRL                                    | 7.29             | 5.24  | (+) Better 2.05    |
| All Energy usage for Sending (mW)      | 11059.67         | 9070.11   | (+) Better 1989.56 |
| All Energy usage for Receiving (mW)    | 16498.46         | 13570.36  | (+) Better 2928.10 |
| Energy usage for header sending (mW)   | 1955.39          | 1685.94   | (+) Better 269.45  |
| Energy usage for header receiving (mW) | 7401.98          | 6109.15   | (+) Better 1292.83 |
| Number of Local Repair                 | 33               | 31.63   | (+) Better 1.37    |
| Number of Source Repair                | 36.88            | 25.75   | (+) Better 11.13   |
| Selected new less hop count            | 0                | 39.13   | (+) Better 39.13   |

จากตารางที่ 4-5 แสดงการเปรียบเทียบผลที่ได้จากการทดสอบการทำงานของโพรโทคอล AODV แบบดั้งเดิมกับ AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair และการเลือกใช้เส้นทางที่สั้นที่สุดที่น้อยกว่า แสดงให้เห็นว่าโพรโทคอล AODV ที่ทำการพัฒนามีค่าอัตราความสำเร็จในการส่งข้อมูลมีค่าที่สูงขึ้น 16.82 เปอร์เซ็นต์ และสามารถลดค่าหน่วยเวลาได้ถึง 57.37 เปอร์เซ็นต์ สาเหตุเนื่องจากในโพรโทคอล AODV แบบดั้งเดิม โหนดจะยังคงใช้งานเส้นทางเดิม แม้ว่าโหนดบางโหนดมีการเคลื่อนที่เข้าไปใกล้กับโหนดในเส้นทางที่สามารถใช้งานได้ ค่าจำนวน

โหนดที่ใช้ในการติดต่อสื่อสารจึงยังคงเท่าเดิม แต่หากเป็นโพรโทคอลที่ได้ทำการพัฒนา เมื่อโหนดเคลื่อนที่ตามเงื่อนไขที่กำหนด โหนดจะสามารถทำการร้องขอเส้นทางเพื่อใช้ในการส่งข้อมูลในเส้นทางที่มีค่าจำนวนโหนดที่ใช้ในการติดต่อสื่อสารที่สั้นลงจึงทำให้อัตราความสำเร็จในการส่งข้อมูลดีขึ้นและค่าหน่วยเวลาลดลงดังที่กล่าวไว้ข้างต้น ส่วนค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลโพรโทคอลที่ได้ทำการพัฒนามีค่าดีขึ้น 28.12 เปอร์เซ็นต์ เนื่องจากในกรณีที่โหนดเกิดการเชื่อมต่อเสียหายและเกิดการเลือกใช้การซ่อมแซมเส้นทางแบบ Local repair ซึ่งกระบวนการซ่อมแซม Local repair ที่ได้พัฒนาขึ้นจะใช้วิธีการส่งข้อความร้องขอเส้นทางแบบปลายทางเดียวแทนการส่งแบบแพร่กระจาย จึงทำให้ปริมาณข้อความการจัดการเส้นทางในเครือข่ายมีค่าลดลง ส่งผลให้ค่าปริมาณข้อความการจัดการเส้นทางต่อการรับข้อมูลมีค่าดีขึ้น ส่วนค่าพลังงานทั้งหมดที่ใช้ในการรับ-ส่งข้อมูลมีค่าลดลง 17.74 เปอร์เซ็นต์ และ 17.98 เปอร์เซ็นต์ เนื่องจากในขั้นตอนการปรับปรุงเส้นทางเพื่อใช้งานเส้นทางที่ใช้จำนวนโหนดในการส่งข้อมูลน้อยกว่า ทำให้จำนวนครั้งในการส่งต่อข้อมูลต่อหนึ่งข้อความมีค่าลดลง ประกอบกับการลดการส่งแบบแพร่กระจายด้วยการส่งแบบปลายทางเดียวในกรณีที่มีการซ่อมแซมเส้นทางแบบ Local repair จึงทำให้การใช้พลังงานลดลงดังกล่าว และเมื่อพิจารณาจำนวนครั้งในการซ่อมแซมเส้นทางแบบ Local repair พบว่าไม่แตกต่างกันเนื่องจากเหตุการณ์ที่ทำให้เกิดการซ่อมแซมเส้นทางแบบ Local repair เกิดขึ้นคล้ายกัน แต่หากพิจารณาค่าจำนวนครั้งในการซ่อมแซมเส้นทางแบบ Source repair พบว่าโพรโทคอล AODV ที่พัฒนา Local repair และการเลือกใช้เส้นทางที่สั้นที่สุดจะมีการซ่อมแซมเส้นทางแบบ Source repair ลดลงถึง 30.17 เปอร์เซ็นต์ เนื่องจากโพรโทคอล AODV ที่พัฒนา Local repair และการเลือกใช้เส้นทางที่สั้นที่สุดมีโอกาสในการซ่อมแซมเส้นทางแบบ Local repair ได้สำเร็จสูงกว่าโพรโทคอล AODV แบบดั้งเดิม เนื่องจากใช้งานการส่งข้อความ RREQ แบบปลายทางเดียวแทนการส่งแบบแพร่กระจาย ส่งผลให้ลดจำนวนครั้งในการซ่อมแซมเส้นทางแบบ Source repair และเมื่อพิจารณาจำนวนครั้งที่โหนดมีการเลือกใช้เส้นทางที่สั้นที่สุด พบว่ามีจำนวนถึง 39.13 ครั้ง ซึ่งกระบวนการดังกล่าวจะทำให้โหนดสามารถส่งข้อมูลด้วยเส้นทางที่สั้นลง และกระบวนการดังกล่าวจะไม่เกิดขึ้นในการทำงานของโพรโทคอล AODV แบบดั้งเดิม การทำงานดังกล่าวทำให้การใช้พลังงานของโหนดในระบบลดลงได้เป็นอย่างดี

## บทที่ 5

### บทสรุป ปัญหาและข้อเสนอแนะ

#### 5.1 บทนำ

ในบทนี้จะกล่าวถึงบทสรุปและข้อเสนอแนะของการดำเนินงานทำวิจัย ปัญหาและอุปสรรคต่างๆ ที่เกิดขึ้นในขณะที่ทำวิทยานิพนธ์ และท้ายที่สุดจะกล่าวถึงรายละเอียดและข้อเสนอแนะแก่ผู้ที่สนใจที่จะนำวิทยานิพนธ์ชุดนี้ไปพัฒนาต่อไป

#### 5.2 บทสรุปของการทำวิทยานิพนธ์

วิทยานิพนธ์ชุดนี้เป็นการนำเสนอวิธีการในการพัฒนาโพรโทคอลค้นหาเส้นทาง AODV โดยมีวัตถุประสงค์หลักคือเป็นการพัฒนาให้โพรโทคอล AODV ลดการใช้พลังงาน ทำงานได้รวดเร็ว และมีอัตราการส่งข้อมูลสำเร็จมากที่สุด เมื่อโหนดในระบบเกิดการเคลื่อนที่

โดยการพัฒนาดังกล่าวจะแบ่งออกเป็นสองส่วนหลักคือ การพัฒนา Local repair ซึ่งเกิดขึ้นในขั้นตอนการซ่อมแซมเส้นทางเมื่อเส้นทางเสียหาย และส่วนของการค้นหาเส้นทางที่ใช้จำนวน โหนดในการส่งข้อมูลที่น้อยลงเมื่อโหนดเคลื่อนที่เข้าใกล้กัน ซึ่งในขั้นตอนของการพัฒนา Local repair นั้นได้พัฒนาจากโพรโทคอลที่มีอยู่เดิมใน AODV ดั้งเดิม แต่กระบวนการที่ได้ทำการพัฒนาขึ้นจะช่วยลดจำนวนแพ็กเกจข้อมูลได้ด้วยการส่งแพ็กเกจร้องขอเส้นทางแบบปลายทางเดียว แทนการส่งข้อมูลแบบแพร่กระจาย ส่งผลให้ความสำเร็จในการรับส่งข้อมูลหรือค่า PDR มีค่าเพิ่มมากขึ้นประมาณ 9.85 เปอร์เซ็นต์ ลดระยะเวลาในการส่งข้อมูลหรือ Delay ประมาณ 28.24 เปอร์เซ็นต์ และลดการใช้พลังงานประมาณ 5.81 เปอร์เซ็นต์ ส่วนที่สองคือส่วนของการพัฒนาให้โหนดค้นหาเส้นทางที่ใช้จำนวนโหนดในการส่งข้อมูลที่น้อยลงเมื่อโหนดมีทิศทางในการเคลื่อนที่เข้าใกล้กัน กระบวนการดังกล่าวเป็นการเพิ่มความสามารถให้แก่โพรโทคอล AODV เดิม โดยเมื่อระบบมีกระบวนการทำงานดังกล่าว จะเพิ่มโอกาสให้ระบบสามารถทำการส่งข้อมูลที่มีจำนวนโหนดที่ใช้ในการส่งข้อมูลที่สั้นลง ซึ่งจะช่วยให้เพิ่มความสำเร็จในการส่งข้อมูล และช่วยลดอัตราการใช้พลังงานให้แก่ระบบ ผลลัพธ์ที่ได้คือสามารถเพิ่มค่า PDR ประมาณ 16.82 เปอร์เซ็นต์ ลดระยะเวลาในการส่งข้อมูลหรือ Delay ประมาณ 57.37 เปอร์เซ็นต์ ลดการใช้พลังงานประมาณ 17 เปอร์เซ็นต์

### 5.3 ปัญหาและอุปสรรคของการทำวิทยานิพนธ์

ปัญหาที่พบคือระยะเวลาในการจำลองระบบค่อนข้างนาน เนื่องจากคอมพิวเตอร์ที่นำมาใช้งานมีความเร็วซีพียูค่อนข้างน้อย

### 5.4 ข้อเสนอแนะ

งานวิจัยดังกล่าวเน้นไปที่การพัฒนา Local repair และการพัฒนาการค้นหาเส้นทางที่ใช้จำนวนโหนดในการส่งข้อมูลที่น้อยกว่าเมื่อโหนดมีทิศทางในการเคลื่อนที่เข้าใกล้เส้นทางที่ใช้งานเท่านั้น ในกรณีของ Local repair จะเกิดขึ้นต่อเมื่อเส้นทางที่ขาดอยู่ใกล้กับโหนดปลายทางมากกว่าโหนดต้นทาง



## บรรณานุกรม

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, Issue 4, pp. 393-422, March 2002.
- [2] P. Baronti, P. Pillai, V. Chook, A. Gotta, Y. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards," *Computer Communications* (2007), vol. 30, Issue 7, pp. 1655-1695, May 2007.
- [3] G. Tolle, D. Culler, "Design of an application-cooperative management system for wireless sensor networks," *Wireless Sensor Networks* (2005), Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN 2005), pp. 121-132, February 2005.
- [4] A. Milenkovic, C. Otto, E. Jovanov, "Wireless sensor networks for personal health monitoring: Issue and an implementation," *Computer Communications*, vol. 29, Issues 13-14, pp. 2521-2533, August 2006.
- [5] S. Olariu, Q. Xu, A. Y. Zomaya, "An energy-efficient self-organization protocol for wireless sensor networks," *Proceeding of the 2004 Intelligent Sensors on. Sensor Networks and Information Proceeding Conference*, pp. 55-60, December 2004.
- [6] F. Dressler, "Self-Organization in Ad Hoc Networks: Overview and Classification," *Technical Report Feb 2006*, University of Erlangen, Department of Computer Science 7, March 2006.
- [7] K. L. Mills, "A brief survey of self-organization in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 7, Issue 7, pp. 823-834, September 2007.
- [8] S. J. Lee, M. Gerla, "AODV-BR:Backup Routing in Ad hoc Networks," In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 3, pp. 1311-1316, September 2000.
- [9] J. Singh, P. Singh, S. Rani, "Enhanced Local Repair AODV (ELRAODV)," In *Proceedings of the 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies (ACT'09)*, pp. 787-791, December 2009.
- [10] H. P. Wang, L. Cui, "An enhanced AODV for mobile ad hoc network," In *Proceedings of the 7<sup>th</sup> International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 1135-1140, July 2008.

- [11] J. Broch, D. A. Maltz, D. B. Johnson, Yih-Chun Hu, J. Jetcheva, "A Performance comparison of multi-hop wireless ad hoc network routing protocols," In Proceedings of the 4<sup>th</sup> annual ACM/IEEE international conference on Mobile computing and networking (MobiCom '98), pp. 85-97, October 1998.
- [12] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy Conservation in Wireless Sensor Networks: a Survey," In Proceedings of Ad Hoc Networks, vol. 7, Issue 3, pp. 537-568, May 2009.
- [13] C. E. Perkins, E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," In Proceedings of Mobile Computing Systems and Applications 1990 (WMCSA '90), Second IEEE Workshop on, pp. 90-100, February 1999.
- [14] C. E. Perkins, E. M. Royer and S. Das, "Ad hoc on-demand distance vector (AODV) routing," IETF RFC 3561, July 2003.
- [15] C. E. Perkins, P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," In Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM), vol. 24, Issue 4, pp. 234-244, October 1994.

**ภาคผนวก**

ภาคผนวก ก  
วิธีการติดตั้ง NS-2

### วิธีการติดตั้งโปรแกรมจำลอง NS-2 (NS-2 Simulator version 2.34)

1. ก่อนทำการติดตั้งโปรแกรมจำลอง NS-2 ซึ่งทำงานอยู่ในระบบปฏิบัติการ Ubuntu ในที่นี่จะทำการติดตั้ง Ubuntu 9.10
2. หลังจากทำการติดตั้ง Ubuntu 9.10 เสร็จเรียบร้อยแล้ว จะต้องทำการอัปเดตแพ็คเกจที่มีความจำเป็นเพื่อใช้งานร่วมกับโปรแกรมจำลอง NS-2 โดยขั้นตอนการอัปเดตจะต้องทำการแก้ไขไฟล์ source.list ดังคำสั่งต่อไปนี้

```
sudo /ect/apt/source.list
```

3. ทำการใส่เพิ่ม list ในการดาวน์โหลดข้อมูลเพิ่มเติมในไฟล์ source.list ดังนี้

```
deb ftp://ftp.psu.ac.th/pub/ubuntu/ karmic main multiverse restricted universe
deb ftp://ftp.psu.ac.th/pub/ubuntu/ karmic-backports main multiverse restricted universe
deb ftp://ftp.psu.ac.th/pub/ubuntu/ karmic-proposed main multiverse restricted universe
deb ftp://ftp.psu.ac.th/pub/ubuntu/ karmic-security main multiverse restricted universe
deb ftp://ftp.psu.ac.th/pub/ubuntu/ karmic-updates main multiverse restricted universe
```

4. พิมพ์คำสั่ง update ดังนี้

```
sudo apt-get update
```

5. ติดตั้งคอมไพเลอร์ที่จำเป็นด้วยคำสั่งดังต่อไปนี้

```
sudo apt-get install build-essential autoconf automake libxmu-dev
```

6. ทำการดาวน์โหลดไฟล์จากเซิร์ฟเวอร์ที่ให้บริการ ในที่นี่ทำการดาวน์โหลดมาเก็บไว้ใน Desktop ดังคำสั่งต่อไปนี้

```
cd ~/Desktop/
wget http://www.paulson.in/tools/ns2/ns-allinone-withpath-2.34.tbz
```

7. หลังจากทำการดาวน์โหลดไฟล์เสร็จสิ้นให้ทำการคัดลอกไฟล์ไว้ใน /usr/local และ extract ด้วยคำสั่งต่อไปนี้

```
cd /usr/local
sudo tar -jxvf ns-allinone-withpath-2.34.tbz
```

8. ทำการแก้ไขไฟล์ใน /ns-allinone-2.34/otcl-1.13/configure ดังคำสั่งต่อไปนี้

```
vi ns-allinone-2.34/otcl-1.13/configure
```

9. ค้นหาโค้ดโปรแกรมที่มีรายละเอียดดังนี้

```
SHLIB_CFLAGS="-fpic"
SHLIB_LD="ld -shared"
SHLIB_SUFFIX=".so"
```

แก้ไขเป็น

```
SHLIB_CFLAGS="-fpic"
SHLIB_LD="gcc -shared"
SHLIB_SUFFIX=".so"
```

10. พิมพ์คำสั่งต่อไปนี้เพื่อทำการ install โปรแกรมและตั้งค่าเพื่อใช้งาน NS-2

```
sudo ./install
source /etc/profile.d/ns2.sh
```

11. เสร็จสิ้นการติดตั้งโปรแกรม

**ภาคผนวก ข**

**การใช้งาน NS-2 เพื่อจำลองการทำงานของโปรโตคอล AODV เบื้องต้น**

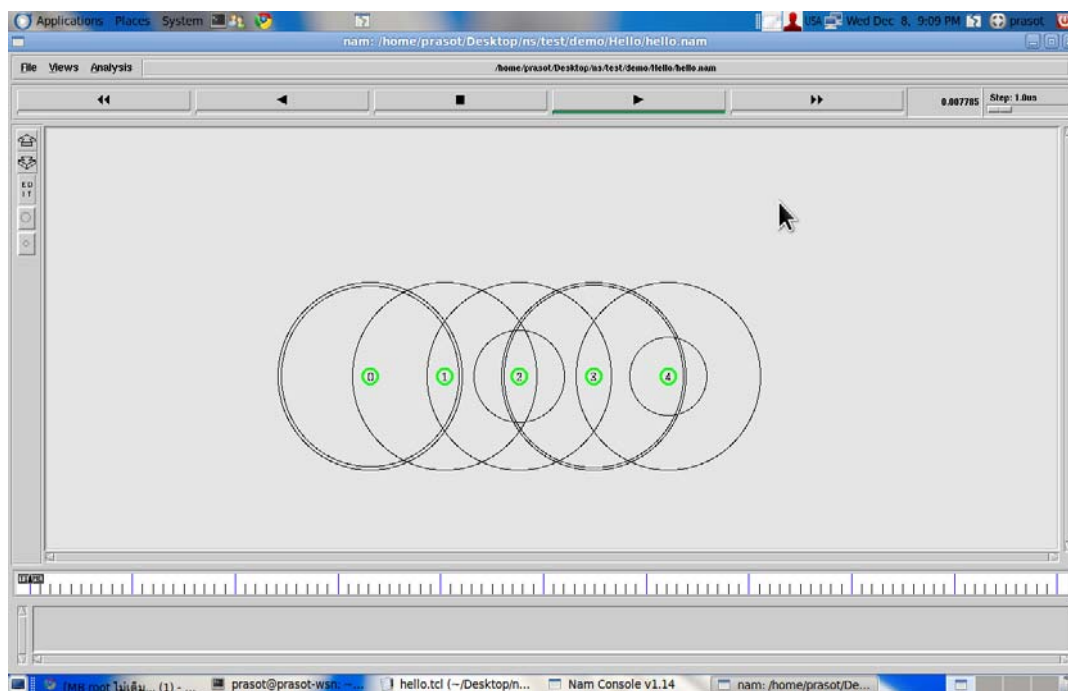
## พารามิเตอร์ที่ใช้งาน

ตารางที่ ข-1 พารามิเตอร์พื้นฐานที่มีการใช้งานในระบบ

|   |          |
|---|----------|
| จำนวนโหนด   | 5        |
| ขนาดของพื้นที่  | 50*50    |
| Routing protocol  | AODV     |
| ระยะทางที่สามารถติดต่อสื่อสาร   | 15m      |
| Packet type   | CBR      |
| CBR size  | 70 Bytes |
| ในที่นี้ให้ node_(0) ส่งไปให้ node_(4) โดยมี interval time 0.2 วินาที |          |

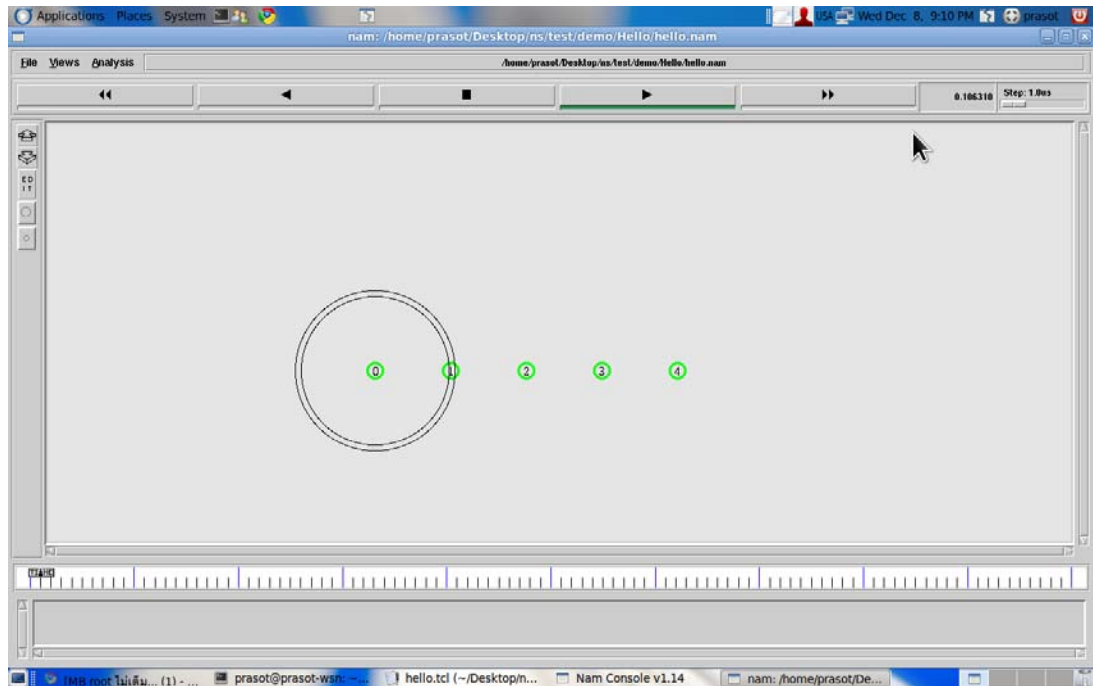
หลังจากทำการเซตค่าต่างๆ ของระบบแล้วทำการรัน โดยการรันจะรันไฟล์โค้ดนามสกุล .tcl ซึ่งจะได้ผลลัพธ์ออกมาเป็นไฟล์สกุล .tr และไฟล์สกุล .nam โดยที่ไฟล์ .tr คือ trace file ซึ่งเป็น log ที่เก็บข้อมูลลักษณะของการติดต่อสื่อสารของทั้งระบบ ส่วนไฟล์สกุล .nam จะถูกนำไปใช้ในการสร้าง animation เพื่อช่วยในการวิเคราะห์ผล ซึ่งหากทำการรันไฟล์สกุล .nam ตาม scenario ที่ได้ตั้งไว้ข้างต้นจะได้ลักษณะดังภาพประกอบ ข-1





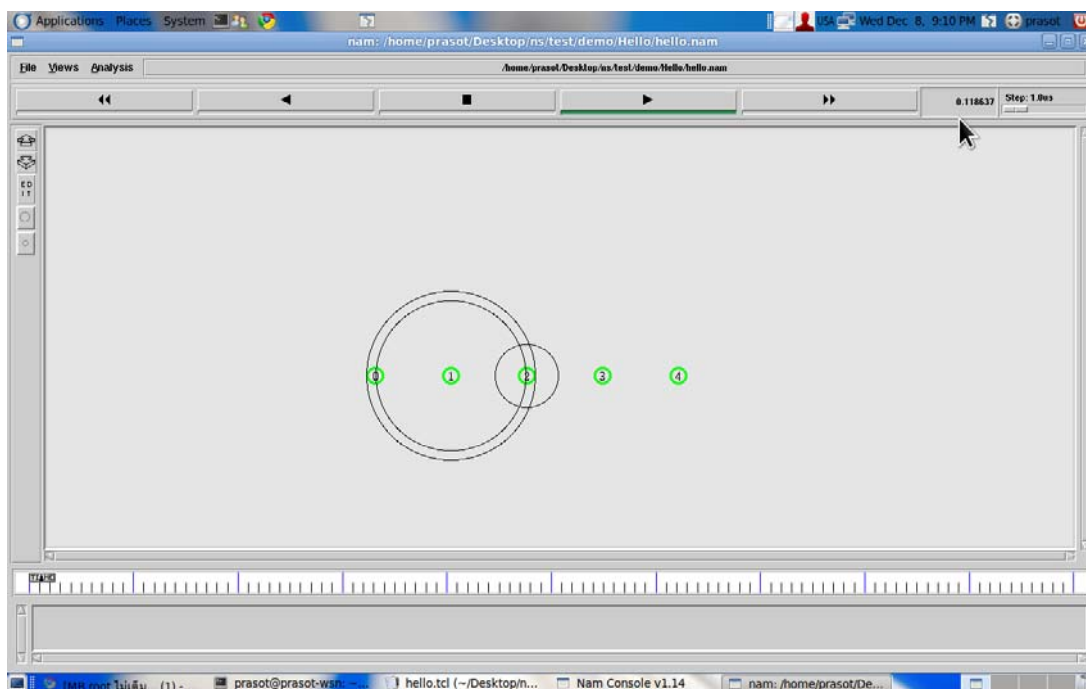
ภาพประกอบ ข-1 ขั้นตอนการส่งข้อความตรวจสอบ โหนดข้างเคียงแบบแพร่กระจาย

จากภาพประกอบ ข-1 ขั้นตอนการส่งข้อความตรวจสอบ โหนดข้างเคียงแบบแพร่กระจาย ซึ่งจะเกิดขึ้นในช่วงเวลาเริ่มต้นของระบบ (ในขั้นตอนดังกล่าวอาจเกิดการชนกันของข้อมูลอยู่มาก)



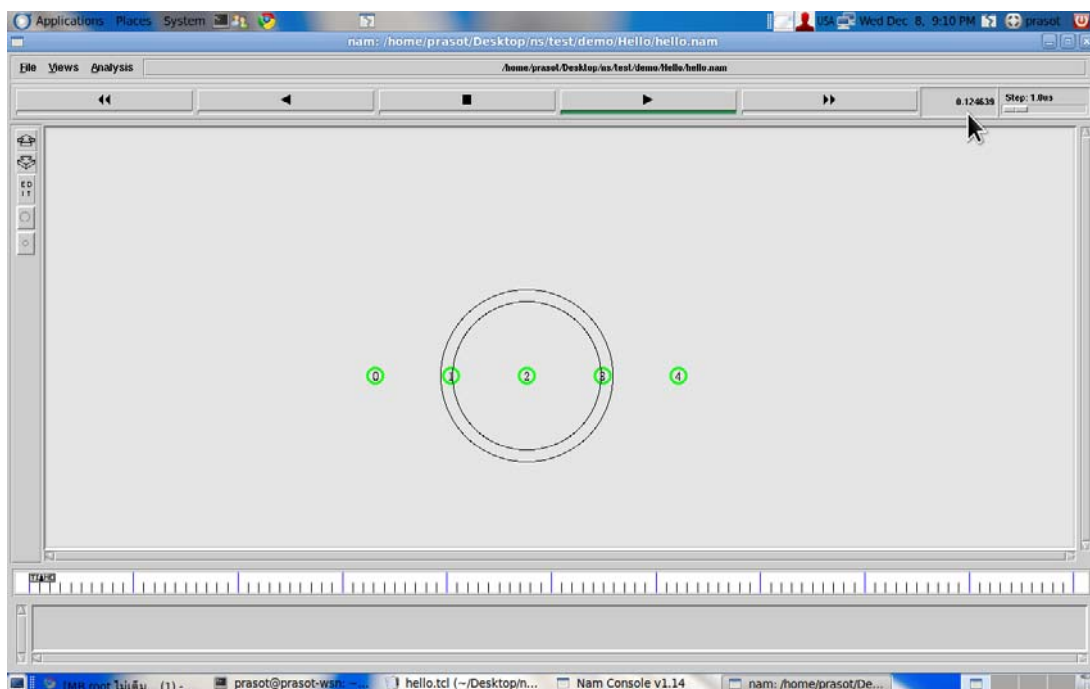
ภาพประกอบ ข-2 การส่งข้อความ RREQ จาก node\_ (0)

ภาพประกอบ ข-2 เป็นการส่งข้อความ RREQ จาก node\_(0) โดย node\_(0) ต้องการส่งข้อมูลไปยัง node\_(4) มันจะเริ่มทำการส่งข้อความ RREQ เพื่อทำการค้นหาเส้นทาง ซึ่งในที่ node\_(1) จะได้รับข้อความ RREQ เพียงโหนดเดียว



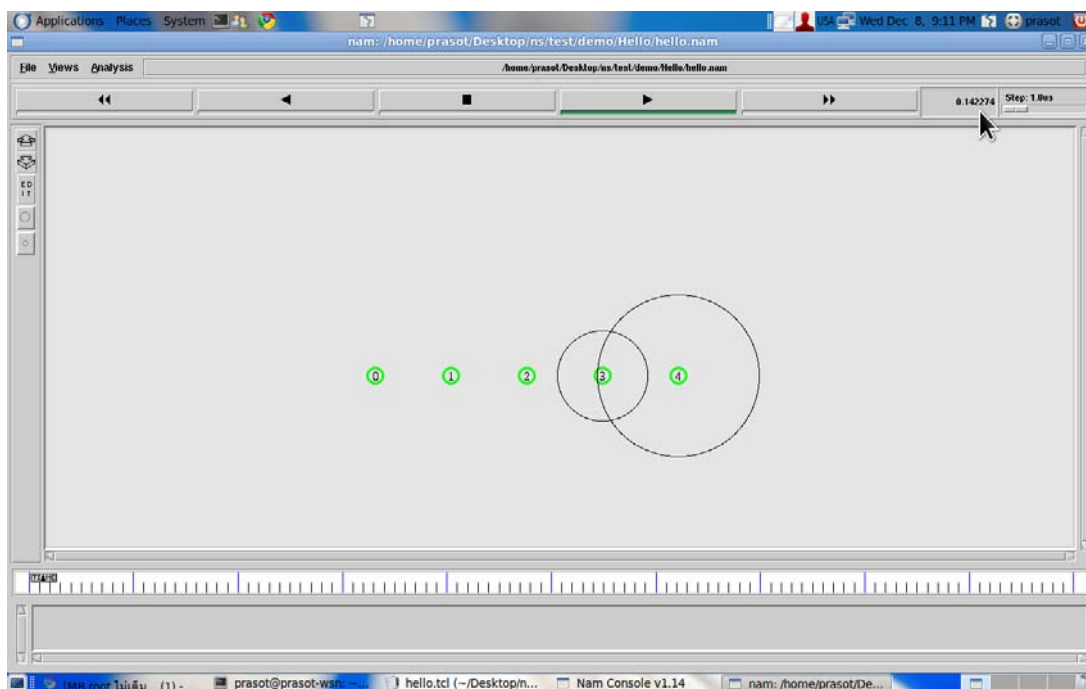
ภาพประกอบ ข-3 การส่งข้อความ RREQ จาก node<sub>(1)</sub>

ภาพประกอบ ข-3 การส่งข้อความ RREQ จาก node<sub>(1)</sub> โดยในขั้นตอนดังกล่าว node<sub>(2)</sub> และ node<sub>(0)</sub> จะได้รับข้อความ RREQ จาก node<sub>(1)</sub> แต่เนื่องจาก node<sub>(0)</sub> มีข้อความ RREQ ดังกล่าวอยู่แล้ว จึงเพิกเฉยต่อข้อมูลดังกล่าว ส่วน node<sub>(2)</sub> ซึ่งได้รับข้อมูลเหมือนกัน แต่พิจารณาแล้วว่าเป็นข้อมูลที่ยังไม่เคยได้รับ node<sub>(2)</sub> ก็จะทำการค้นหาเส้นทางต่อ โดยการส่งข้อความ RREQ ต่อดังภาพประกอบ ข-4



ภาพประกอบ ข-4 การส่งข้อความ RREQ จาก node\_(2)

การทำงานของ การส่งข้อความ RREQ จะเป็นลักษณะเช่นนี้ไปเรื่อยๆ จนกระทั่งพบปลายทางที่ต้องการส่งข้อมูล ในเหตุการณ์นี้คือ node\_(3) จะต้องทำการส่งข้อความ RREQ ไปยัง node\_(4) ซึ่งหลังจาก node\_(4) ได้รับข้อความ RREQ มันจะทำการส่งข้อความร้องขอ ARP กลับไปยัง node\_(3) เพื่อทำการยืนยันตัวตนว่ามีอยู่จริง โดยการส่งข้อความตอบกลับ ARP กลับไปยัง node\_(4) ดังภาพประกอบ ข-5



ภาพประกอบ ข-5 การรับ-ส่งข้อความร้องขอ ARP และตอบกลับ ARP

หลังจากทำการส่งข้อความ ARP เพื่อทำการยืนยันว่ามีตัวตน node\_(4) จะทำการส่งข้อความ RREP กลับมาตามเส้นทางย้อนกลับ กลับมายัง node\_(0) และหลังจากได้เส้นทางที่ใช้ในการติดต่อสื่อสาร node\_(0) จะสามารถส่งข้อความ CBR ไปยัง node\_(4) ได้ โดยผ่านทาง node\_(1), node\_(2), node\_(3) ตามลำดับ

**ภาคผนวก ค**  
**ผลการทดสอบระบบเพิ่มเติม**

### ผลการทดสอบระบบเพิ่มเติม

ในการทดสอบเพื่อเปรียบเทียบประสิทธิภาพการทำงานของโปรโตคอล AODV แบบดั้งเดิมกับ AODV ที่พัฒนา Local repair และการเลือกใช้เส้นทางที่สั้นที่สุดจะมีการจำลองระบบเป็นจำนวน 10 ครั้ง โดยได้ผลการทดสอบแสดงดังตารางที่ ค-1

ตารางที่ ค-1 แสดงผลที่ได้จากการทดสอบการทำงานของโปรโตคอล AODV แบบดั้งเดิมกับ AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair และการเลือกใช้เส้นทางที่สั้นที่สุด

| No. simulation | Defaulted AODV   |                   |                             | Improved AODV    |                   |                             |
|----------------|------------------|-------------------|-----------------------------|------------------|-------------------|-----------------------------|
|                | No. Local repair | No. Source repair | Selected new less hop count | No. Local repair | No. Source repair | Selected new less hop count |
| 1              | 32               | 33                | 0                           | 32               | 18                | 40                          |
| 2              | 33               | 30                | 0                           | 28               | 22                | 33                          |
| 3              | 35               | 40                | 0                           | 33               | 34                | 37                          |
| 4              | 41               | 40                | 0                           | 38               | 27                | 41                          |
| 5              | 42               | 41                | 0                           | 35               | 28                | 35                          |
| 6              | 31               | 38                | 0                           | 33               | 29                | 27                          |
| 7              | 30               | 32                | 0                           | 28               | 28                | 45                          |
| 8              | 34               | 50                | 0                           | 34               | 31                | 42                          |
| 9              | 28               | 38                | 0                           | 28               | 22                | 46                          |
| 10             | 23               | 33                | 0                           | 30               | 19                | 40                          |
| Avg.           | 33               | 36.88             | 0                           | 31.63            | 25.75             | 39.13                       |

จากตารางที่ ค-1 แสดงผลที่ได้จากการทดสอบการทำงานของโปรโตคอล AODV แบบดั้งเดิมกับ AODV ที่พัฒนาการซ่อมแซมเส้นทางแบบ Local repair และการเลือกใช้เส้นทางที่สั้นที่สุด โดยมีจำนวนครั้งในการทดสอบ 10 ครั้ง และจะทำการวิเคราะห์ค่าโดยการนำค่าทั้ง 10 ครั้ง มาทำการตัดค่ามากที่สุดและค่าน้อยสุดทิ้งไป แล้วนำค่าที่เหลือมาหาค่าเฉลี่ยเพื่อทำการวิเคราะห์ต่อไป



ภาคผนวก ง  
การตีพิมพ์เผยแพร่วิทยานิพนธ์

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



# ICESIT 2011

*International Conference  
on Embedded Systems  
and Intelligent Technology*

# Study of AODV Performance and Energy Consumption over IEEE802.15.4 in WSN

Parkpoom Manoyut, Anirooth Thongklin, Chaiyut Jandaeng and W.Suntiamorntut  
 Collaborative Research Unit on Wireless Sensor Network  
 Department of Computer Engineering, Faculty of Engineering,  
 Prince of Songkla University, Hat Yai, Songkhla 90112 Thailand  
 Email kgprasot@gmail.com

*Abstract- Wireless sensor network is widely used in many applications. A large number of sensor nodes are deployed as an unstructured network in an ad-hoc manner. In addition, the mobility is another research challenge applied in wireless sensor network. Therefore, we focus on an existing routing protocol, named AODV to use on wireless sensor network. This paper will show the possibility and the behavior of this protocol. The simulation results using NS-2 will show the performance of the network in term of dropped packet, PDR and NRL. The energy dissipation is also addressed in this paper.*

**Keyword- Wireless Sensor Network, AODV, IEEE802.15.4**

## I. INTRODUCTION

A large scale wireless sensor network uses an advantage of wireless communication technology. The wireless sensor network has been deployed in various applications [1] such as environmental monitoring [2], health care monitoring, automation control, disaster warning system or military domain. Wireless sensor network composes of a large number of tiny sensor nodes to sense the information and transmit back to base station. The standard of wireless sensor network is IEEE 802.15.4 MAC specification[3,4].

Wireless sensor network can be both unstructured and structured. Most practical usage, it will be an unstructured and deployed in an ad-hoc manner. There is no infrastructure between nodes and base station. Due to node is very small and portable therefore wireless sensor network has a high potential to apply in Mobile Ad Hoc Network (MANET) which each node can move freely in any direction.

With this reason, we turn the focus from a static ordinary sensor network protocol to an existing routing protocol, named AODV [IETF RFC 3561][5,6]. This paper is aiming at investigating the behavior of AODV in term of dropped packet, PDR, energy dissipation and NRL in each test case.

The rest of the paper is organized as follows. The ADOV routing protocol is explained in section 2. Our experiments using NS-2 simulation is described in section3. In section 4, we will discussion the results and draw the conclusion.

## II. AODV ROUTING PROTOCOL

AODV (Ad-hoc On-Demand Vector) is a reactive routing protocol. The route will be created when source node wants to send the data. AODV uses the traditional route tables. The main feature in AODV is time-based

states. The routing can be expired if it is not used for some period. The route discovery (source to sink node) is based on query. The intermediate node keeps the route information using route table. There are four types of control message as follows: Route Request Message (RREQ), Route Reply Message (RREP), Route Error Message (RERR) and HELLO Messages.

RREQ is flooded through the network when a route is not available. RREQ packet contains source id, request id, source sequence number, sink id, sink sequence number and hop count. RREP is used when they found sink node or having a valid route. Neighbor node will send RERR when the route is lost. Finally, HELLO messages are broadcasted to check the neighbor.

## III. NS-2 SIMULATION

We varied four parameters, the number of connection, buffer size, an interval time and the number of node. The number of connection is the number of source node which sent the packet to sink. The buffer size is the number of ifqlen indicated the size of queue. The interval time is the time of sending 1 packet. The final parameter is the number of node in the system.

We have set up a simulation experiment using NS-2 (version 2.34) order to understand the behavior of AODV in wireless sensor network environment in term of dropped packet (bytes), packet delivery ratio (PDR), energy consumption and NRL. The first indicator is the number of dropped packets in unit of byte. The main reasons of packet dropped are collision, duplicate, packet error, retry count exceed, invalid state, callback and busy in MAC. Moreover, there will be another cause such as route is not available, routing loop, TTL reaches zero, packet expired, no buffer space in IFQ, ARP full and link quality indicator (LQI).

The second behavior we observed is Packet Delivery Ratio (PDR). PDR can be calculated from the division of the number of received packets and the number of sent packets. Then later, we also explore the average energy remaining (mW) of all node in the network. The last indicator is NRL (normalized routing load) expressed as the following:

$$\text{NRL} = \text{number of routing packets sent} / \text{number of data packets received}$$

We can know the performance of routing protocol from NRL. The routing protocol suited with the nature of the network will give NRL low.

The simulation is based on IEEE 802.15.4 MAC specification. All hundred nodes works in non-beacon enabled mode with unslotted CSMA-CA algorithm, and they have the same transmission range at 15 meters (give the best PDR from our experiments), power dissipation rate and buffer size. The detail of network parameter is shown in table1.

TABLE I. SIMULATION PARAMETER SETTING (DEFAULT)

|                           |               |
|---------------------------|---------------|
| Node number               | 100           |
| Simulation Area           | 100 x 100     |
| Simulation Time           | 100 sec       |
| Transmission Range        | 15 meters     |
| Traffic type, Packet size | CBR, 70 bytes |
| Buffer Length             | 50            |
| Tx Power                  | 0.1 mW        |
| Rx Power                  | 0.1 mW        |
| Idle Power                | 0.1 mW        |
| Number of connection      | 20            |
| Number of source node     | 20            |
| Number of sink node       | 1             |

The following experimental results are averaged from 10 simulation runs. Note that, we did not use “sleep mode” we mean a device always keeps listening in the “idle mode”.

#### A. Vary number of connection

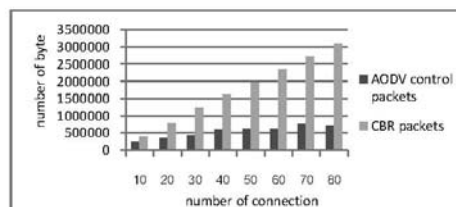


Figure 1 Number of dropped packet (bytes) – varied number of connection

From figure1, the number of dropped packet of AODV control packets is a slightly increased when the number of connection is grown up. This is because of the static network system tested in the simulation. Source node can send the data using the same route even the number of connection is increased. Meanwhile, the numbers of CBR dropped packet are grown up rapidly when the number of connection is increasing. The reason is the default of interval time used at 0.2 seconds. Therefore, the number of data packets in the network becomes larger when they have more connections.

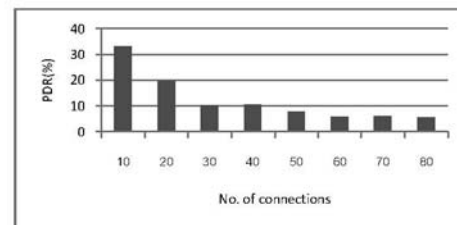


Figure 2 Packet delivery ratio (%) – varied number of connection

In figure2, when the number of connection increases, the PDR starts to decrease as we expected. This is because of the collision in the network.

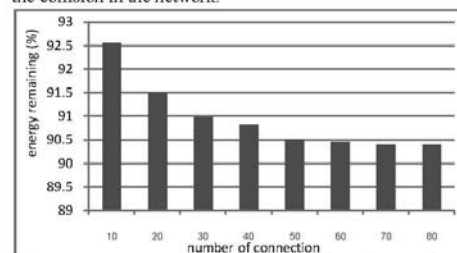


Figure 3 Energy remaining – varied number of connection

When the simulation time finished at 100 seconds, the percentage of energy remaining is varied from 92.5% to nearly 90.5% as the number of connection is step up by 10 as shown in figure3. However, the energy remaining is not lower than 90% at 50 connections even the number of connection is still growing up. This may be a short simulation times. The larger connection may require a longer period in the initial phase.

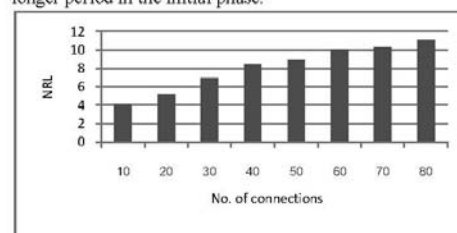


Figure 4 Normalized routing load – varied the number of connection

The figure 4 shows that the routing load will be bigger when we increased the number of connections. According to the growth of NRL, it indicates that the route packet sent is not increased but the number of received data is reduced (from %PDR in figure2) when the number of connection is growing up.

### B. Vary buffer size (ifqlen)

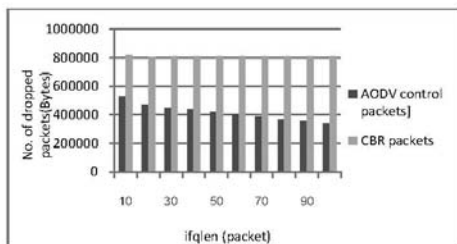


Figure 5 Number of dropped packet (bytes) – varied buffer size

We have learnt from figure1 that the number of dropped AODV control packet was increased when the number of connection grew up. Therefore we set the experiment to vary the buffer length and observe the effect of dropped AODV control packet. Figure5 shows the improvement of dropped AODV control packet when we increase the buffer length from 10 – 90 while the dropped CBR packets is stable at 800KBytes. Thus the number of buffer size does not affect to the number of dropped CBR packets.

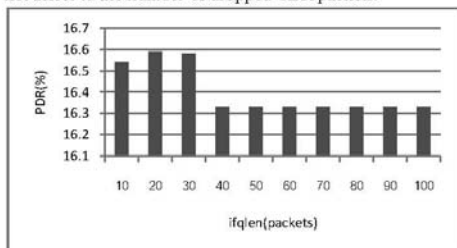


Figure 6 Packet delivery ratio (%) – varied buffer size

In the simulation, we used the default parameter as shown in table1. The interval time is 0.2 seconds with 20 connections. From figure6, we found that the buffer length at 10-30 gives the good feature of %PDR. Apart from that, the buffer size does not affect to %PDR.

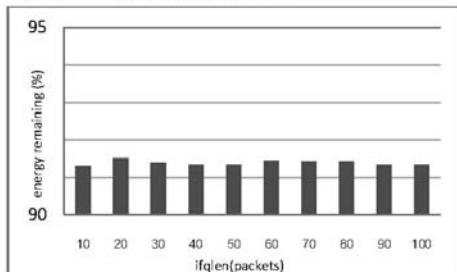


Figure 7 Energy remaining – varied buffer size

After the simulation reached 100 seconds, the percentage of energy remaining of each buffer length is not different. Thus changing the buffer size does not affect to the energy dissipation.

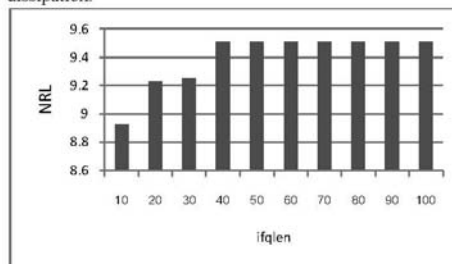


Figure 8 Normalized routing load – varied buffer size

NRL was not much affected by the change of buffer size as shown in figure8 because of the small number of connection (20) and interval time (0.2s).

### C. Vary interval time

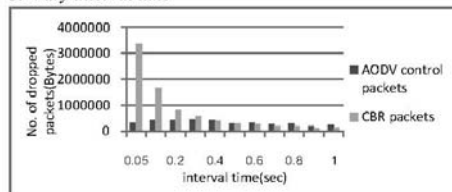


Figure 9 Number of dropped packet (bytes) – varied interval time

The internal time affects directly to the dropped packet, especially CBR packets (data) because each node sends out the data packet more frequently (small interval time). In contrast with the AODV control packet, it is rarely changed according to an interval time. We also found the main reason of dropped packets is QFull (queue is full). It is clearly that the routing protocol does not concern with the interval time.

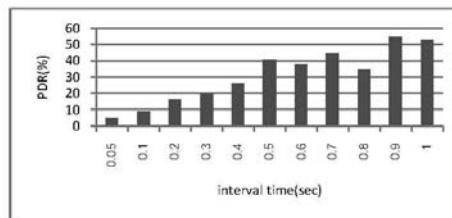


Figure 10 Packet delivery ratio (%) – varied interval time

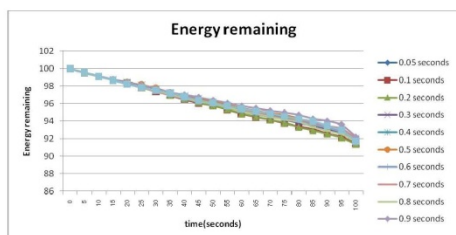


Figure 11 Energy remaining – varied interval time

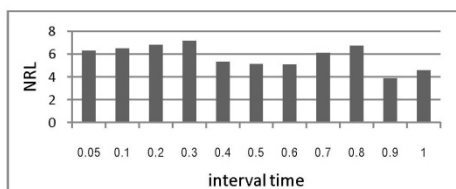


Figure 12 Normalized routing load – varied interval time

From figure 10, the success received packet is increased when we reduce the frequency of sending data. In figure 11 also shows the energy was dissipated very quickly during the time 0-90 seconds. There is a small gap of energy remaining at the time 60-95 seconds. That gap shows that the larger interval time has the energy remaining more than the smaller interval time. The normalized routing load is good when the interval time is high or frequency of sending data is low as can be seen in figure12. The best NRL uses the interval time at 0.9 seconds.

D. Vary number of node

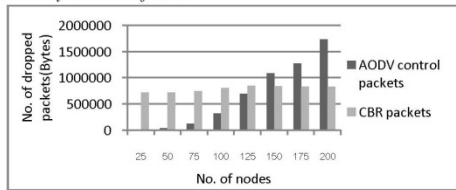


Figure 13 Number of dropped packet (bytes) – varied number of node

The number of AODV control packet is increased depends on the number of device in the system while the CBR packets is not involved as shown in figure13. In a larger system, they have to use a large control packet to discover the neighbor for a route setup.

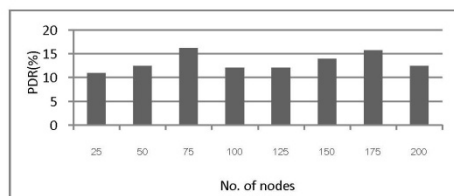


Figure 14 Packet delivery ratio (%) – varied number of node

The PDR does not directly depend on the number of node in the system. We measure the CBR packet at the application layer and the number of device cannot affect to the number of dropped CBR packet as shown in figure 13. Thus we can see not much different PDR when the number of node was varied as shown in figure 14.

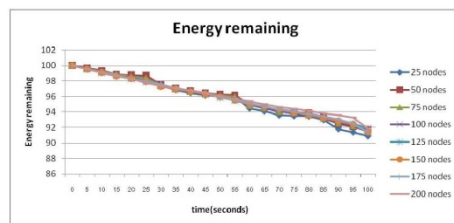


Figure 15 Energy remaining – varied number of node

At the end of simulation, the energy remaining of 25 nodes is lowest as shown in figure15 while the energy consumption in 200 nodes is the lowest dissipation. We have to explore the reason in the future work. However, it may be because of node density. With 200 nodes in the same area, each node can forward a data hop by hop in a short range. This may help the device to save the energy.

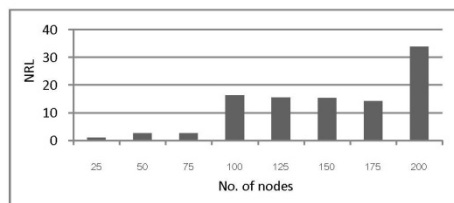


Figure 16 Normalized routing load – varied number of node

When the number of node is increased, we found a lot of RREQ packets in route discovery as shown in figure16.

IV. CONCLUSION AND FUTURE WORK

From our experiments varying the number of connection, we found that AODV routing protocol will use the same route even the number of connection is increasing and CBR

drop packet is very high. The energy dissipation becomes bigger when the connection is grown up. The interval time can affect to %PDR and CBR drop packet. The increased buffer length can improve the dropped packet. Finally the number of node affects to dropped AODV control packet, energy remaining and also NRL.

#### ACKNOWLEDGEMENT

The authors would like to thank TRIDI for their supports, NTC scholarship, grant no. 009/2553.

#### REFERENCES

- [1] I. F. Akyldiz, S. Weillian, S. Yogesh and C. Erdal, "A Survey on Sensor Networks", Vol: 40, Aug, pp. 102-114, 2002.
- [2] V. Rohit, "Application Of Wireless Sensor Networks for Environmental Monitoring & Development of an Energy Efficient Hierarchical Cluster based Routing," Department Of Electrical Engineering, National Institute Of Technology, Rourkela, 2008-2009.
- [3] Jianliang Zheng and Myung J. Lee, "A comprehensive performance study of IEEE 802.15.4," *Sensor Network Operations*, IEEE Press, Wiley Inter science, Chapter 4, pp. 218 - 237, 2006.
- [4] Gang Lu, B. Krishnamachari and C.S. Raghavendra, "Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks," *IEEE International Conference on Performance, Computing, and Communications 2004*, pp. 701 - 706, Phoenix, Arizona, April 15- 17, 2004.
- [5] Ian D. Chakeres and Luke Klein-Berndt, "AODVjr, AODV Simplified," *ACM SIGMOBILE Mobile Computing and Communications Review*, pp.100 - 101, Volume 6 , Issue 3, July 2002.
- [6] C. Gomez, P. Salvatella, O. Alonso and J. Paradells, "Adapting AODV for IEEE 802.15.4 Mesh Sensor Networks: Theoretical Discussion and Performance Evaluation in a Real Environment," *Proceedings of WoWMoM'06*, pp. 159 -170, Niagara Falls, USA, June 26 - 29, 2006.



ICUIMC(IMCOM) 2012

The 6<sup>th</sup> International Conference on Ubiquitous Information Management and Communication

February 20-22 2012, at Sunway Lagoon Resort Hotel Kuala Lumpur, Malaysia



---

***ACM ICUIMC 2012, February 20–22***  
***Kuala Lumpur, Malaysia***  
**Conference Program**

---

| Hosted by |

Association for Computing Machinery (ACM)  
Sungkyunkwan University (SKKU), Korea  
Universiti Kuala Lumpur, Malaysia

| Sponsored by |

ACM SIGAPP

# Modified Local Repair in AODV for LoWPANs

Parkpoom Manoyut

Centre of Excellence in Wireless Sensor Networks,  
Department of Computer Engineering,  
Faculty of Engineering  
Prince of Songkla University, Hat Yai, Songkhla 90112  
Thailand  
kgprasot@gmail.com

Wannarat Suntiamorntut

Centre of Excellence in Wireless Sensor Networks,  
Department of Computer Engineering,  
Faculty of Engineering  
Prince of Songkla University, Hat Yai, Songkhla 90112  
Thailand  
wannarat@coe.psu.ac.th

**Abstract**— Wireless sensor networks are widely used in many applications such as environment monitoring, intelligent transportation system (ITS) and health care monitoring. In wireless sensor networks, a large number of sensor nodes are deployed without infrastructure in an ad-hoc manner. Therefore, we focus on the existing ad-hoc routing protocol named AODV which is widely applied in wireless sensor networks. This paper proposes the method to improve the performance of the local repair mechanism in AODV. We found that our mechanism gives the performance improvement varying from 20 to 36%. In addition, the routing overhead in term of dropped packet is better than the original AODV. The number of CBR drop and the energy usage in the network are also reported and analyzed in this paper.

## Keywords

AODV, routing protocol, Local repair, LoWPANs, IEEE802.15.4

## Categories and Subject Descriptors

C.2.2 [Network Protocols] – Routing protocols.

## General Terms

Design, Performance, Experimentation

## 1. INTRODUCTION

A large scale wireless sensor networks have been introduced by the combination of the wireless communication, sensing and embedded system technologies. The wireless sensor networks based on IEEE 802.15.4 standard have been deployed in various applications [1] such as environmental monitoring [2], health care monitoring [3], automation control, disaster warning system or military domain. Wireless sensor network composes of a large number of tiny sensor nodes to sense the physical environment and transmit the data back to base station using single hop or multi hop.

---

The first author is a post graduate student supported by NBTC, TRIDI, NTC scholarship.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICUJMC '12, February 20–22, 2012, Kuala Lumpur, Malaysia.  
Copyright 2012 ACM 978-1-4503-1172-4...\$10.00.

The IEEE 802.15.4 standard [4] supports any network topologies such as star and mesh network. This standard is a Low-Rate Wireless Personal Area Network (LR-WPAN) contained a low-cost and low power sensor node with the limited memory storage and processing capability. Thus routing protocol has to consider these constraints.

In low-power wireless personal area networks (LoWPANs) for elderly assistive living system [5], there are two types of node, infrastructure node for backbone connectivity and mobile node. The self-configuration and self-healing are mostly required in this application due to the mobility and network topology changing. Therefore the repair route mechanism is needed for this purpose.

The most studied routing protocol likes Ad hoc On-demand Distance Vector (AODV) [6] has been adapted for LoWPANs. AODV is a reactive routing protocol which the route will be established on-demand basis. The Route Request (RREQ) message is broadcasted by the source node in order to find a route to destination. The route table entries for source and destination node are created by the intermediate node during the route discovering process. In addition, AODV also provides two types of route maintenance, source repair and local repair in order to fix the link failure.

This paper is aiming to investigate the behavior of AODV protocol in term of Packet Deliver Ratio (PDR), the number of dropped packet and energy dissipation when the number of moving node is varied. We also explore the local/source repair mechanisms and analyze the route overhead of local repair when the link failure happens. Finally, the modified local repair is proposed to improve the performance and reduce the route overhead in the networks.

The rest of the paper is organized as follows. The detail of AODV routing protocol such as route discovery, route maintenance and link failure detection are explained in Section 2. Section 3 proposes the AODV analysis. The performance

improvement is proposed in Section 4. The experiment results are shown in Section 5. Finally, we discuss and conclude our research work in Section 6.

## 2. AODV ROUTING PROTOCOL

AODV (Ad hoc On-demand Distance Vector) is a reactive routing protocol. The route will be searched only the source node wants to send the data. Moreover, the route will be maintained only if the route is used.

AODV uses the traditional route tables [7] which contains destination IP address, destination sequence number, valid destination sequence number flag, routing flag, hop count, next hop, list of precursors and lifetime. This information will be used in each routing process of AODV.

### A. Route Discovery

When the source node is required to send the data to the destination node and the route is not existed in the route table, the source node will broadcast the Route Request (RREQ) asking for the available route. The RREQ contains RREQ ID, Destination IP address, Destination sequence number, Originator IP address, Originator sequence number and hop count. The RREQ broadcasting is limited by expanding ring search in order to reduce the traffic in the network. The neighbor node will update the route table and increase the hop count when it receives the RREQ. This mechanism will be repeated until the RREQ can reach the destination node.

### B. Route Maintenance

The maintenance will occur when the link failure is found. Node will consider the causes of the failure before choosing the suitable method to fix the route. In AODV, there are two possible methods of the route maintenance. If the link is failed closed to the source node, the source repair will be activated. Node which found the link fail will send the Route Error (RERR) back to the source node using the reverse route. As soon as the source node receives the RERR, the route discovery will start.

In contrast, if the failure happens nearby the destination node, the local repair will be activated. Node which found the link fail will broadcast the RREQ by itself. In this case, the source node will not know the link fail in the network.

### C. Link Failure Detection

Normally, the link failure can be detected by two mechanisms, link layer feedback and Hello Message. Link layer feedback is a passive method and needed a support from MAC layer.

Another mechanism is the use of "Hello Message". This method is simple for implement and does not depend on MAC

layer. Therefore, we use this method for this work. The Hello Message will be sent out every period to check whether the link is still alive. This period can be calculated from  $\text{HELLO\_INTERVAL} * \text{ALLOWED\_HELLO\_LOSS}$ . The default value we used in the simulation of HELLO\_INTERVAL and ALLOWED\_HELLO\_LOSS are 1 and 2 second, respectively. Even this method takes longer time to check the link failure than the layer feedback, it is simple and light weight to implement on a sensor node.

### D. Local Repair in AODV

In AODV, the local repair has been used when the link failure is found nearby the destination node. The node will broadcast RREQ packet together with the destination IP address to the destination node. The local repair in AODV generates the traffic to repair the route. Therefore, a high traffic as broadcasting can occur when repairing more than one route happens at the same time.

## 3. AODV ANALYSIS

AODV has been widely applied in mobile applications. Route discovery will be initiated on demand of the source node by broadcasting a Route Request (RREQ) message through the network. RREQ message is broadcasted until it reaches the destination node. During this process, the intermediate node creates a temporary route table to be a reverse route back to the source node. The destination will send Route Reply (RREP) message to the source node via unicast along the reverse route. This route will be established and used to connect between the source and destination node.

However, node can move around or be interfered that causing to break the active route. In this situation, there is a large amount of the dropped packet in the network, especially when the speed of node is increased or the number of mobile node in the network is increased as shown in Figure 1. In Figure 1, we investigate the relation between the number of dropped CBR packet and the speed of mobile node. We also study how the number of mobile node can affect to the network.

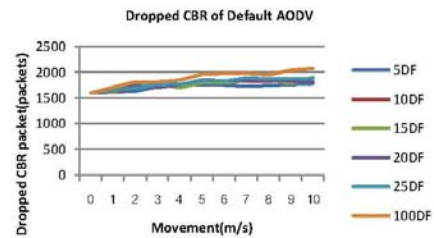


Figure 1. The number of dropped CBR packet in AODV

AODV has responsibility to repair and re-establish the route. There are two approaches which can repair the route [8]. First, the source node starts to discover the route when it is received a Route Error (RERR) message from the broken node. We call this method as *Source repair*.

Another approach is a *Local repair* which the broken node initiates the route discovery by broadcasting RREQ message. During this process, the data has to be buffered at the repairing node. Within a fixed amount of time, the initial repairing node has to receive a RREP message. This will indicate the success of the local repair process. In contrast, the repairing node will report back to the source node by sending a RERR message in order to start a source repair process.

At this point, we are interested in the number of dropped CBR packet when AODV has only the source or local repair happen. The speed of node and the number of mobile node have been varied to investigate the number of dropped CBR packet by setting to turn on only the source or local repair process as shown in Figure 2 and 3, respectively.

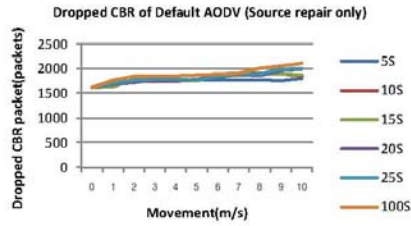


Figure 2. The number of dropped CBR packet in source repair process

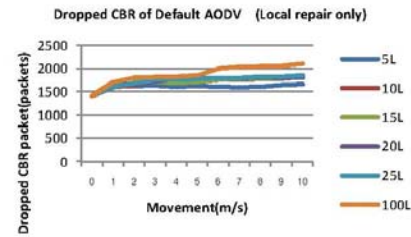


Figure 3. The number of dropped CBR packet in local repair process

From the experiment results in Figure 2 and 3, we found that the number of dropped CBR packet is not different even only one approach of repairing process or combing the two approaches in AODV has been used in the mobile network. The link failure cannot be solved by local repair. The new route cannot be established. Thus the large amount of packet has been broadcasted in the network. Moreover, this

inefficient route repair can also increase the end-to-end delay and decrease the packet delivery ratio (PDR) as well.

The modified local repair is therefore introduced to improve the dropped packet overhead which leading to save the waste energy consumption. In addition, the PDR is expected to be improved by using the modified local repair. The concept and detail of the modified local repair is given in the next section.

#### 4. MODIFIED LOCAL REPAIR

##### A. Modified Local Repair

The additional field in *Hello Message* packet has been introduced to keep the neighbor of neighbor information as shown in Figure 4. The *Route\_next\_Reply* value will be attached with RREP packet. Each node will have the information of next hop and the next of reply.

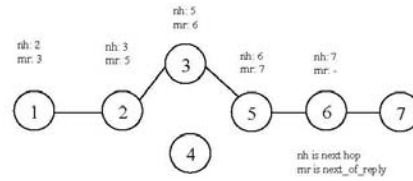


Figure 4. Next hop and next of reply information

When the link failure happens, the typical AODV will detect whether the link fails nearby the source node or the destination node. If the link fails nearby the source node, the RERR packet will be sent out to the source node in order to begin the route discovery. In contrast, the local repair will be activated when the link fails nearby the destination node.

In our modified local repair process, the neighbor of the neighbor information has been checked. If the *route\_next\_reply* is the neighbor of neighbor, the RREQ packet is sent directly to the node neighbor immediately. By using this technique, the broadcasting overhead can be reduced. The pseudo code of our modified local repair has been shown in Figure 5.

```

IF link failure occur near destination THEN
  WHILE neighbor of neighbor
    IF neighbor of neighbor equal route_next_reply
      Unicast RREQ neighbor
      success
    ENDIF
  ENDWHILE
  Broadcast RREQ (Default Local repair)
ELSE
  Source repair (Send RERR packet)
ENDIF

```

Figure 5. Pseudo code of the proposed modified local repair

### B. Modified Message

The Hello Message packet has to be modified by inserting the neighbor of neighbor information to support the modified local repair. This modified Hello Message is used to update the neighbor of neighbor when the node receives the last Hello Message. The period of receiving modified Hello Message will be two times of the link failure detecting time,  $HELLO\_INTERVAL * ALLOWED\_HELLO\_LOSS * 2$ .

The format of Route Reply Message (RREP) is modified by inserting the route\_next\_reply field as shown in Figure 6. The route table is updated after receiving a RREP packet. Meanwhile, the Route Request Message (RREQ) is also modified to carry the next hop of broken node and used only in route repair as shown in Figure 7.

| Type                           | Reserved | Numb of NB |
|--------------------------------|----------|------------|
| Sequence Number                |          |            |
| Neighbor IP Address(1)         |          |            |
| Neighbor IP Address(if needed) |          |            |

Figure 6. Modified Hello Message

| Type                        | R | A | Reserved | Pr S | Hop count |
|-----------------------------|---|---|----------|------|-----------|
| Destination IP Address      |   |   |          |      |           |
| Destination Sequence Number |   |   |          |      |           |
| Originator IP Address       |   |   |          |      |           |
| Originator Sequence Number  |   |   |          |      |           |
| Lifetime                    |   |   |          |      |           |
| Next of reply               |   |   |          |      |           |

Figure 7. Modified RREP Message

## 5. PERFORMANCE EVALUATION

The network simulator NS-2 has been used in this evaluation. We use the packet delivery ratio (PDR), the number of dropped CBR packet and the energy remaining as the performance metric. For each scenario, we run the simulation at 10 times and the average result is used. Table I shows the summary of simulation parameters. We varied the speed of node and observed the PDR, the number of dropped CBR packet and the energy remain.

TABLE I. GENERAL PARAMETERS

|                    |                       |
|--------------------|-----------------------|
| Physical Layer     | IEEE 802.15.4         |
| Mac Layer          | Mac/802.15.4          |
| Ifqlen             | 50                    |
| Number of nodes    | 100 nodes             |
| Routing protocol   | AODV routing protocol |
| Area of Simulation | 100 m * 100 m         |
| Mobility model     | Random waypoint       |
| Application packet | CBR packet (70Bytes)  |

|                                |                        |
|--------------------------------|------------------------|
| Simulation time                | 1000 seconds           |
| Sending range                  | 30 m                   |
| Receiving range                | 30 m                   |
| Initial Energy                 | 27540 Joules           |
| Rx Power                       | 47 mW.                 |
| Tx Power                       | 47 mW.                 |
| Idle Power                     | 0.03 mW.               |
| Number of movement nodes (m/s) | 5, 10, 15, 20, 25, 100 |
| Receiver                       | 1 node                 |
| Sender                         | 10 nodes               |
| Interval of sender send        | 5 second/ packet       |

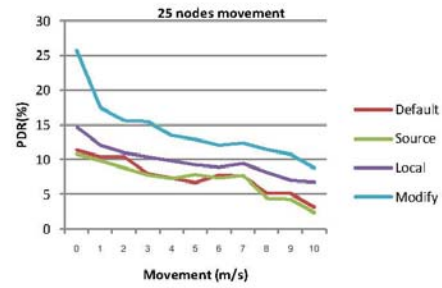


Figure 8. The comparison of PDR with 25 nodes movement

The experiments have been setup to four scenarios, default AODV (using source repair and local repair), AODV using only source repair, AODV using only local repair and the modified local repair. Figure 8 shows the comparison of PDR results when there is 25 nodes moving in the network. We found that the modified local repair gives the best PDR in every speed of node. With our local repair, the broken link can be re-established very quick even some nodes are moving around.

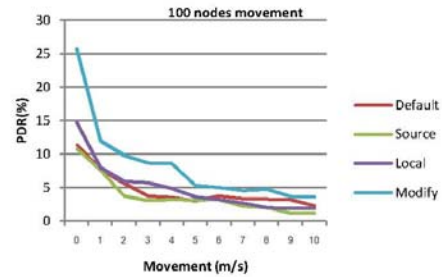


Figure 9. The comparison of PDR with 100 nodes movement

We also evaluate the modified local repair when every node in the network is moving. The result is shown in Figure 9. The modified local repair can also give the best PDR even the PDR is lower than the scenario of 25 nodes movement. However, the PDR of the original AODV (default) tends to have a better PDR compared to only source repair or only local repair approach. The reason is because of the frequent link failure when every node is moving. Neither source repair nor local repair can re-establish the route. While our local repair can handle this situation very well.

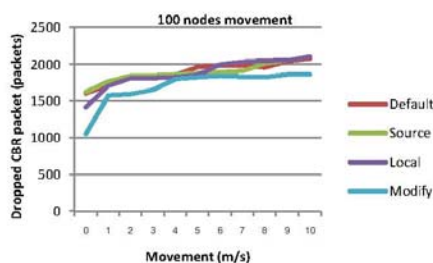


Figure 10. The number of dropped CBR packet with 100 nodes movement

By using the modified local repair, the number of dropped CBR packet is reduced when compared to other scenarios. The maximum dropped CBR packet is not more than 1,800 packets within 1000 seconds in the simulation time as shown in Figure 10. We found that the modified local repair can reduce the number of dropped CBR at about 26% which leading to save the waste energy consumption.

## 6. CONCLUSION

We propose the modified local repair of AODV routing protocol aiming to employ in health care monitoring system. The performance has been evaluated using NS2 and found that the modified local repair can give the best PDR compared to the original AODV at about 36%. Moreover, we can reduce the waste energy because of the number of CBR packet reduction.

## 7. ACKNOWLEDGMENT

The authors would like to thank financial support from the office of NBTC (the office of the national broadcasting and telecommunications commission) for their supports, NTC scholarship, grant no. 009/2553.

## 8. REFERENCES

- [1] I. F. Akyıldız, S. Weillian, S. Yogesh and C. Erdal, "A Survey on Sensor Networks", *IEEE Communications Magazine*, Vol.40, August, pp. 102-114, 2002.
- [2] V. Rohit, "Application Of Wireless Sensor Networks for Environmental Monitoring & Development of an Energy Efficient Hierarchical Cluster based Routing," *Ph.D. thesis in Department Of Electrical Engineering, National Institute Of Technology, Rourkela*, 2008-2009.
- [3] De Schatz, C.H.V., Medeiros, H.P., Schneider, F.K., Abatti, P.J., "Wireless protocols for ad-hoc medical sensor networks," *Proceedings of the IASTED International Symposia on Imaging and Signal Processing in Healthcare and Technology (ISPHT)*, pp. 135-142, May 2011.
- [4] Salman, N., Rasool, I., Kemp, A.H., "Overview of the IEEE 802.15.4 standards family for low rate wireless personal area networks," *Proceedings of the 7<sup>th</sup> International Symposium on Wireless Communication Systems (ISWCS)*, pp. 701-705, Sep 2010.
- [5] W. Santiamornit, S. Charoenpanyasak and J. Ruksachum, "An Elderly Assisted Living System with Wireless Sensor Networks," *Proceedings of 4<sup>th</sup> Joint IFIP Wireless and Mobile Networking Conference (WMNC)*, October 2011.
- [6] Perkins, C.E., Royer, E.M., "Ad-hoc on-demand distance vector routing," *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMC-SA)*, pp. 90-100, Feb 1999.
- [7] Perkins, C., Belding-Royer, E., Das, S., "Ad hoc on Demand Distance Vector routing (AODV)," RFC 3561, 2003.
- [8] S. R. Azzuhri, M. Portmann, W. L. Tan, "Evaluation of parameterised route repair in AODV," *Proceedings of the 4<sup>th</sup> Signal Processing and Communication Systems (ICSPCS)*, pp. 1-7, Dec 2010.

## ประวัติผู้เขียน

ชื่อ สกุล นายภาคภูมิ มโนยุทธ

รหัสประจำตัวนักศึกษา 5310120065

วุฒิการศึกษา

| วุฒิ   | ชื่อสถาบัน               | ปีที่สำเร็จการศึกษา |
|--|--------------------------|---------------------|
| วิศวกรรมศาสตรบัณฑิต<br>(วิศวกรรมคอมพิวเตอร์) | มหาวิทยาลัยสงขลานครินทร์ | 2552                |

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

- ทุนการศึกษา NTC Scholarship สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการ โทรทัศน์ และกิจการ โทรคมนาคมแห่งชาติ (NBTC, TRIDI) ประจำปีการศึกษา 2553

การตีพิมพ์เผยแพร่ผลงาน

- P. Manoyut, A. Thongklin, C. Jandaeng, and W. Suntiamorntut, "Study of AODV Performance and Energy Consumption over IEEE802.15.4 in WSN," In Proceedings of 4<sup>th</sup> International Conference on Embedded System and Intelligent Technology 2011 (ICESIT2011), Phuket, Thailand, 12 th – 14 th May 2010, pp. 154-158.
- P. Manoyut, W. Suntiamorntut, "Modified Local Repair in AODV for LoWPANs," In Proceedings of 6<sup>th</sup> International Conference on Ubiquitous Informatino Management and Communication 2012 (IMCOM 2012), Kula Lumpur, Malaysia, 20 th – 22 th February 2012.