รายงานวิจัยฉบับสมบูรณ์

การพัฒนาเครือข่ายเฉพาะกิจสมาร์ทโฟนเพื่อการสื่อสาร แบบไม่พึ่งพาโครงสร้างพื้นฐาน

A Study on MANET-Based Smart Phone Ad hoc Network for Infrastructureless Communications

ผศ.ดร. สุเมธ ประภาวัต

ได้รับทุนสนับสนุนงานวิจัยจาก เงินงบประมาณแผ่นดิน ประจำปีงบประมาณ 2559
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงานวิจัยฉบับสมบูรณ์

การพัฒนาเครือข่ายเฉพาะกิจสมาร์ทโฟนเพื่อการสื่อสาร แบบไม่พึ่งพาโครงสร้างพื้นฐาน

A Study on MANET-Based Smart Phone Ad hoc Network for Infrastructureless Communications

ผศ.ดร. สุเมธ ประภาวัต

ได้รับทุนสนับสนุนงานวิจัยจาก เงินงบประมาณแผ่นดิน ประจำปีงบประมาณ 2559
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อโครงการ การพัฒนาเครือข่ายเฉพาะกิจสมาร์ทโฟนเพื่อการสื่อสารแบบไม่พึ่งพาโครงสร้างพื้นฐาน

แหล่งเงิน เงินงบประมาณแผ่นดิน

ประจำปีงบประมาณ พ.ศ. 2559 จำนวนเงินที่ได้รับการสนับสนุน 500,000 บาท

ระยะเวลาทำการวิจัย 2 ปี ตั้งแต่ ตุลาคม 2558 ถึง กันยายน 2560

ชื่อ-สกุล หัวหน้าโครงการ และผู้ร่วมโครงการวิจัย พร้อมระบุ หน่วยงานต้นสังกัด

หัวหน้าโครงการ ผศ.ดร.สุเมธ ประภาวัต

หน่วยงานต้นสังกัด คณะเทคโนโลยีสารสนเทศ สจล.

บทคัดย่อ

เครือข่ายเฉพาะกิจเคลื่อนที่คือเครือข่ายสื่อสารรูปแบบหนึ่งซึ่งสามารถทำงานได้โดยไม่จำเป็นต้องได้รับการสนับสนุนจากโครงสร้างพื้นฐานต่าง ๆ โดยการให้โหนดในเครือข่ายทำหน้าที่เป็นตัวกลางในการส่งต่อข้อมูลจากต้นทางไปยังปลายทางซึ่งกันและกันในขณะที่โหนดทุกตัวสามารถเคลื่อนที่ได้อย่างอิสระ ด้วยคุณสมบัติดังกล่าวจึงมีแนวคิดที่จะนำเครือข่ายเฉพาะกิจเคลื่อนที่มาใช้ในฐานะเครือข่ายสำรองในกรณีฉุกเฉิน งานวิจัยนี้จึงได้นำแนวคิดของเครือข่ายเฉพาะกิจเคลื่อนที่มาพัฒนาแอพพลิเคชั่นเพื่อการติดต่อสื่อสารบนอุปกรณ์สมาร์ทโฟน ซึ่งมีการใช้งานอย่างแพร่หลายในปัจจุบัน และเพื่อให้เครือข่ายเฉพาะกิจเคลื่อนที่บนสมาร์ทโฟนมีโอกาสที่จะถูกนำมาใช้จริงมากขึ้น งานวิจัยนี้จึงได้ศึกษา วิจัย และพัฒนาโพรโทคอลบนเครือข่ายเฉพาะกิจเคลื่อนที่ให้มีประสิทธิภาพในด้านต่าง ๆ สูงขึ้น ซึ่งผลของการพัฒนาโพรโทคอลเหล่านี้ยังสามารถนำไปต่อยอดในงานวิจัยทางด้านเทคโนโลยีเครือข่ายเฉพาะกิจเคลื่อนที่ต่อไป

การศึก

Research Title: A Study on MANET-Based Smart Phone Ad hoc Network for Infrastructureless Communications
Researcher: Asst.Prof.Dr. Sumet Prabhavat
Faculty: Information Technology
Department: King Mongkut's Institute of Technology Ladkrabang

ABSTRACT

Mobile Ad Hoc Network (MANET) is an infrastructure-less self-organizing network, which can provide communication services by allowing every mobile node to act as a relay node to form the network. Because of this distinctive characteristic, MANET is suitable to operate while infrastructure network cannot be used. In this paper, we propose a communication application to connect smartphones together and establish MANET connection. Moreover, we further study conventional MANET routing protocols with the aim to improve the performance of MANET. As a result, we propose the novel approaches, which can achieve the better result in the routing process. The main contribution of this project is to increase the possibility of MANET technology and aim to deploy it as a reliable communication network on a smart phone.

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การศึก

กิตติกรรมประกาศ

โครงการวิจัยนี้ได้รับทุนสนับสนุนการวิจัยจากเงินงบประมาณแผ่นดิน ประจำปีงบประมาณ พ.ศ. 2559 คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คณะผู้วิจัย

การศึก

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VI
สารบัญภาพ VII
บทที่ 1 บทนำ 9
1.1 ความเป็นมาและความสำคัญ 9
1.2 ความมุ่งหมายและวัตถุประสงค์ 9
1.3 ขอบเขตของโครงงาน 10
1.4 ขั้นตอนการวิจัย 10
1.5 ประโยชน์ที่คาดว่าจะได้รับ 10
บทที่ 2 การทบทวนวรรณกรรมที่เกี่ยวข้อง 11
2.1 เครือข่ายเฉพาะกิจเคลื่อนที่ (Mobile Ad Hoc Network: MANET) 11
2.2 Wireless Fidelity (Wi-Fi) 11
2.3 Wi-Fi Direct (Wi-Fi Peer-to-Peer) 12
2.4 Generic Advertisement Service (GAS) 13
2.5 โพรโทคอลค้นหาเส้นทางในเครือข่ายเฉพาะกิจเคลื่อนที่ 14
2.6 โพรโทคอลสำหรับค้นหาเส้นทางไปยังเกตเวย์สำหรับเครือข่ายเฉพาะกิจเคลื่อนที่ 23
บทที่ 3 การออกแบบและพัฒนาแอพพลิเคชัน 26
3.1 การพัฒนาแอพพลิเคชันเครือข่ายเฉพาะกิจเพื่อการสื่อสารแบบไม่พึ่งพาโครงสร้างพื้นฐาน 26
3.2 ผลการทดสอบการทำงานเบื้องต้นของแอพพลิเคชัน 40
บทที่ 4 การวิเคราะห์และพัฒนาประสิทธิภาพ 47
4.1 การพัฒนาโพรโทคอลค้นหาเส้นทางที่มีโอเวอร์เฮดต่ำ 47
4.2 การพัฒนาโพรโทคอลค้นหาเส้นทางโดยการควบคุมการกระจายแพ็คเกตค้นหาเส้นทาง 55
4.3 การพัฒนากลไกค้นหาเกตเวย์ที่มีโอเวอร์เฮดต่ำ 66
บทที่ 5 สรุปการพัฒนา 76
บรรณานุกรม 77
ภาคผนวก ก. 80
แอพพลิเคชันเกตเวย์สำหรับการเชื่อมต่อเครือข่ายเฉพาะกิจสมาร์ทโฟนกับอินเทอร์เน็ต เผยแพร่ในวารสารวิชาการคณะเทคโนโลยีสารสนเทศ สจล.
ภาคผนวก ข. 90

สารบัญ (ต่อ)

รายงานผลการทดสอบระบบเครือข่ายไร้สายเฉพาะกิจแบบแมเน็ตบนสมาร์ทโฟนสำหรับโครงการวิจัยการพัฒนาเครือข่ายเฉพาะกิจสมาร์ทโฟนเพื่อการสื่อสารแบบไม่พึ่งพาโครงสร้างพื้นฐาน

ภาคผนวก ค. 110

เกตเวย์เพื่อการเชื่อมโยงเครือข่ายเฉพาะกิจสมาร์ทโฟนเข้ากับเครือข่ายอินเทอร์เน็ต เผยแพร่ในงานประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 13 (NCCIT 2017)

ภาคผนวก ง 117

การศึกษาประสิทธิภาพด้านพลังงานของกลไกค้นหาเส้นทางในเครือข่ายเฉพาะกิจเคลื่อนที่ เผยแพร่ในงานประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 12 (NCCIT 2016)

ภาคผนวก จ. 124

การศึกษาประสิทธิภาพกลไกค้นหาเกตเวย์บนเครือข่ายเฉพาะกิจเคลื่อนที่ เผยแพร่ในงานประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 12 (NCCIT 2016)

ภาคผนวก ฉ. 131

Overhead Reduction for Route Repair in Mobile Ad Hoc Networks เผยแพร่ในหนังสือ (Book Series) - Advances in Intelligent Systems and Computing: Recent Advances in Information and Communication Technology (vol. 566) สำนักพิมพ์ Springer

ภาคผนวก ช. 142

Modified BERS for Energy Efficiency in Mobile Ad Hoc Network เผยแพร่ในงานประชุมวิชาการระดับประเทศด้านเทคโนโลยีสารสนเทศ ครั้งที่ 8 (NCIT 2016)

ภาคผนวก ช. 151

Overhead reduction of location-aided gateway discovery protocols เผยแพร่ในงานประชุมทางวิชาการระดับนานาชาติ IEEE International Conference on Information Technology and Electrical Engineering (ICITEE 2016)

ภาคผนวก ฌ. 158

A Dynamic Routing for Load Distribution in Mobile Ad-Hoc Network เผยแพร่ในหนังสือ (Book Series) - Advances in Intelligent Systems and Computing: Recent Advances in Information and Communication Technology (vol. 566) สำนักพิมพ์ Springer

ข้อมูลประวัติคณะผู้วิจัย 169

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 3.1 คำอธิบายยูสเคส Login 28
ตารางที่ 3.2 คำอธิบายยูสเคส Send Message 28
ตารางที่ 3.3 คำอธิบายยูสเคส Receive Message 29
ตารางที่ 3.4 คำอธิบายยูสเคส Show Debug 29
ตารางที่ 4.1 แสดงข้อมูลสภาพแวดล้อมของการจำลองเครือข่าย 51
ตารางที่ 4.2 แสดงข้อมูลสภาพแวดล้อมของการจำลองเครือข่าย 58

สารบัญภาพ

ภาพที่ **หน้า**

ภาพที่ 2.1 แสดงตัวอย่างการสื่อสารในเครือข่าย MANET 11
ภาพที่ 2.2 เปรียบเทียบการทำงานของ Wi-Fi กับ Wi-Fi Direct 12
ภาพที่ 2.3 แสดงการติดต่อระหว่างอุปกรณ์ของ Wi-Fi Direct 13
ภาพที่ 2.4 แสดงหลักการทำงานของ GAS Service 13
ภาพที่ 2.5 แสดงรูปแบบการสื่อสารของโพรโทคอล GAS 14
ภาพที่ 2.6 แสดงรูปแบบการสื่อสารเมื่อข้อมูล Service มีขนาดใหญ่เกินไป 14
ภาพที่ 2.7 กลไกการทำงานของโพรโทคอล ERS ที่โหนดต้นทาง 17
ภาพที่ 2.8 กลไกการทำงานของโพรโทคอล ERS ที่โหนดระหว่างทาง 17
ภาพที่ 2.9 กลไกการทำงานของโพรโทคอล BERS ที่โหนดต้นทาง 19
ภาพที่ 2.10 กลไกการทำงานของโพรโทคอล BERS ที่โหนดระหว่างทาง 19
ภาพที่ 2.11 แสดงคู่ข้อมูลต้นทาง-ปลายทางที่เก็บในโหนดที่ได้รับแพ็คเกต RREP 21
ภาพที่ 2.12 แผนผังการทำงานของโพรโทคอล QL เมื่อโหนดได้รับแพ็คเกต RREQ 22
ภาพที่ 2.13 รูปแบบการกระจายแพ็คเกต GWADV ของโพรโทคอล LAGAD 25
ภาพที่ 3.1 แผนผังแสดงทอพอโลยีของระบบ 26
ภาพที่ 3.2 แผนภาพยูสเคส (Use Case Diagram) 27
ภาพที่ 3.3 แผนภาพคลาส (Class Diagram) 30
ภาพที่ 3.4 แผนภาพกิจกรรมการส่งข้อความ 31
ภาพที่ 3.5 แผนภาพกิจกรรมการรับข้อความ 32
ภาพที่ 3.6 แผนภาพกิจกรรมการร้องขอเส้นทาง 33
ภาพที่ 3.7 แผนภาพกิจกรรมการตอบกลับการร้องขอเส้นทาง 34
ภาพที่ 3.8 รูปแบบข้อความของ Route Request Message 35
ภาพที่ 3.9 รูปแบบข้อความของ Route Reply Message 35
ภาพที่ 3.10 รูปแบบข้อความของ Acknowledgement Message 36
ภาพที่ 3.11 รูปแบบข้อความของ Data Message 36
ภาพที่ 3.12 แสดงหน้าเข้าสู่ระบบของแอพพลิเคชัน 37
ภาพที่ 3.13 แสดงหน้ารับ-ส่งข้อความของแอพพลิเคชัน 38
ภาพที่ 3.14 แสดงหน้ารับ-ส่งข้อความเมื่อมีข้อความถูกส่งออกและรับเข้า 39
ภาพที่ 3.15 แสดงหน้า Debug Mode 40
ภาพที่ 3.16 แสดงสถานการณ์ในการทดลองแอพพลิเคชัน 40
ภาพที่ 3.17 แสดงการส่งข้อความจากเครื่อง c ใน Debug Mode 41
ภาพที่ 3.18 แสดงการรับและส่งต่อข้อความของเครื่อง b 42
ภาพที่ 3.19 แสดงหน้าจอเครื่อง a ที่ได้รับข้อความจากเครื่อง c 43
ภาพที่ 3.20 แสดงหน้า Debug Mode ของเครื่อง a เมื่อได้รับข้อความ 44
ภาพที่ 3.21 แสดงหน้า Debug Mode ของเครื่อง b เมื่อได้รับ ACK 45
ภาพที่ 4.1 ทอพอโลยีเหตุการณ์เมื่อเส้นทางขาด 48

สารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

สารบัญภาพ (ต่อ)

ภาพที่ 4.2 แสดงคู่ข้อมูลปลายทาง-จำนวนฮ็อปที่เก็บในโหนดที่ได้รับแพ็คเกต RREP 48
ภาพที่ 4.3 แสดงเหตุการณ์เมื่อโหนด D กระจายแพ็คเกต $RREQ_{QL}$ 49
ภาพที่ 4.4 แผนผังการทำงานเมื่อโหนดได้รับแพ็คเกต $RREQ_{QL}$ ของโพรโทคอลที่นำเสนอ 50
ภาพที่ 4.5 โอเวอร์เฮด 52
ภาพที่ 4.6 อัตราการชนกันของเฟรมข้อมูลในระดับ MAC 53
ภาพที่ 4.7 อัตราการส่งข้อมูลสำเร็จ 53
ภาพที่ 4.8 ความหน่วงในการส่งข้อมูล 54
ภาพที่ 4.9 เงื่อนไขการกระจายแพ็คเกต stop_instruction 56
ภาพที่ 4.10 แสดงโอเวอร์เฮดในเครือข่ายของแนวคิดที่ 1 59
ภาพที่ 4.11 แสดงโอเวอร์เฮดในเครือข่ายของแนวคิดที่ 2 60
ภาพที่ 4.12 แสดงการใช้พลังงานในเครือข่ายของแนวคิดที่ 1 61
ภาพที่ 4.13 แสดงการใช้พลังงานในเครือข่ายของแนวคิดที่ 2 61
ภาพที่ 4.14 แสดงเวลาที่โหนดแรกพลังงานหมดของแนวคิดที่ 1 62
ภาพที่ 4.15 แสดงเวลาที่โหนดแรกพลังงานหมดของแนวคิดที่ 2 62
ภาพที่ 4.16 แสดงจำนวนโหนดที่เหลืออยู่ในเครือข่ายของแนวคิดที่ 1 63
ภาพที่ 4.17 แสดงจำนวนโหนดที่เหลืออยู่ในเครือข่ายของแนวคิดที่ 2 63
ภาพที่ 4.18 แสดงความหน่วงในการส่งข้อมูลของแนวคิดที่ 1 64
ภาพที่ 4.19 แสดงความหน่วงในการส่งข้อมูลของแนวคิดที่ 2 64
ภาพที่ 4.20 แสดงอัตราการส่งข้อมูลสำเร็จของแนวคิดที่ 1 65
ภาพที่ 4.21 แสดงอัตราการส่งข้อมูลสำเร็จของแนวคิดที่ 2 65
ภาพที่ 4.22 แผนภาพการทำงานของกระบวนการค้นหาเกตเวย์ที่โหนดต้นทางบนเครือข่ายเฉพาะกิจแบบผสม 67
ภาพที่ 4.23 แผนภาพการทำงานของกระบวนการค้นหาเกตเวย์ที่โหนดระหว่างทางบนเครือข่ายเฉพาะกิจแบบผสม 68
ภาพที่ 4.24 รูปแบบการแพร่กระจายแพ็คเกต GWADV ของโพรโทคอลที่นำเสนอ 71
ภาพที่ 4.25 ปริมาณโอเวอร์เฮดที่เกิดขึ้นทั้งหมดในการจำลองกระบวนการค้นหาเกตเวย์ 72
ภาพที่ 4.26 อัตราการส่งสำเร็จของข้อมูลในการจำลองกระบวนการค้นหาเกตเวย์ 73
ภาพที่ 4.27 ความหน่วงในการส่งข้อมูลในการจำลองกระบวนการค้นหาเกตเวย์ 74
ภาพที่ 4.28 เวลาที่โหนดต้นทางใช้ในการค้นหาเกตเวย์ 74

การศึกษา
และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ทุกวันนี้มีเหตุการณ์ที่ไม่คาดฝันเกิดขึ้นมากมายอันเนื่องมาจากสถานการณ์ภัยพิบัติต่าง ๆ เช่น แผ่นดินไหว สึนามิ น้ำท่วม หรือสายเคเบิลที่เชื่อมต่อกับสถานีฐานขาด จนทำให้โครงสร้างพื้นฐานของระบบสื่อสารในบางพื้นที่ไม่สามารถใช้งานได้ ส่งผลต่ออุปกรณ์สื่อสารในพื้นที่ ทำให้ขาดการติดต่อ การให้ความช่วยเหลือและการขอความช่วยเหลือจึงเป็นไปอย่างยากลำบาก ซึ่งถือเป็นภัยคุกคามที่มีผลต่อความมั่นคงแห่งรัฐเพราะกระทบต่อคุณภาพชีวิตและความปลอดภัยของพลเมืองโดยตรง นอกจากนี้อีกสถานการณ์หนึ่งที่เกิดขึ้นบ่อย คือ การเกิดความคับคั่งในการใช้ระบบสื่อสารในบางพื้นที่เฉพาะ เช่น พื้นที่ที่มีการชุมนุม มีงานคอนเสิร์ต หรือมีงานรับพระราชทานปริญญาบัตร ซึ่งมีผู้คนจำนวนมากใช้งานโทรศัพท์เคลื่อนที่พร้อม ๆ กัน จนกระทั่งทำให้ช่องสัญญาณของสถานีฐานเต็ม ส่งผลให้ผู้คนไม่สามารถติดต่อสื่อสารกันได้ และก่อให้เกิดความเสียหายต่อภาคธุรกิจอย่างร้ายแรง

แม้แต่การติดต่อโดยใช้อุปกรณ์สื่อสาร เช่น สมาร์ทโฟน แท็บเล็ต ซึ่งเป็นอุปกรณ์สมัยใหม่ที่กำลังถูกใช้อย่างแพร่หลาย แต่ก็ยังต้องพึ่งพาโครงสร้างเครือข่ายพื้นฐาน เช่น เครือข่าย Cellular หรือเครือข่ายไร้สาย (Wi-Fi) ซึ่งระบบเหล่านี้โดยปกติแล้วจะต้องมีอุปกรณ์โครงข่ายพื้นฐาน (Infrastructure) เช่น เสาสถานีฐาน (Base Station) หรือ Access Point จึงจะทำให้สามารถติดต่อกันได้ ซึ่งหากเกิดภัยคุกคามดังที่กล่าวมาข้างต้น ก็จะทำให้ผู้ใช้ไม่สามารถสื่อสารกันได้

ดังนั้นการมีระบบสื่อสารที่เป็นช่องทางสำรองจึงเป็นเรื่องจำเป็นอย่างยิ่ง การสร้างเครือข่ายเคลื่อนที่เฉพาะกิจจึงเป็นแนวทางที่เป็นไปได้ โดยการพัฒนาเครือข่ายสื่อสารโดยใช้เทคโนโลยีที่มีอยู่แล้วในอุปกรณ์สมาร์ทโฟนหรือแท็บเล็ต เช่น Wi-Fi Bluetooth Wi-Fi Direct เพื่อสร้างการเชื่อมต่อระหว่างอุปกรณ์โดยตรง ให้สามารถติดต่อสื่อสารกันได้โดยไม่จำเป็นต้องพึ่งพาโครงสร้างพื้นฐาน (Infrastructureless) ผนวกกับการประยุกต์ใช้เทคโนโลยีเครือข่ายเฉพาะกิจเคลื่อนที่ MANET ซึ่งมีกลไกที่ช่วยให้การส่งข้อมูลมีความน่าเชื่อถือ เพื่อให้ผู้ใช้ยังคงติดต่อสื่อสารกันได้แม้ในสถานการณ์ที่โครงสร้างเครือข่ายพื้นฐานไม่สามารถทำงานได้

1.2 ความมุ่งหมายและวัตถุประสงค์

1.2.1 เพื่อศึกษาการพัฒนาเครือข่ายเฉพาะกิจบนสมาร์ทโฟน โดยใช้เทคโนโลยีเครือข่าย MANET ซึ่งรองรับการติดต่อสื่อสารในสถานการณ์ที่ระบบโครงข่ายพื้นฐานในพื้นที่นั้น ๆ ไม่สามารถให้บริการได้ตามปกติ

1.2.2 เพื่อเป็นแนวทางในการพัฒนาระบบสื่อสารสำรองที่มีความน่าเชื่อถือ สำหรับเป็นช่องทางการติดต่อสื่อสารเมื่อโครงสร้างเครือข่ายพื้นฐานไม่สามารถทำงานได้

1.3 ขอบเขตของโครงงาน

1.3.1 นำเทคโนโลยีเครือข่ายเฉพาะกิจเคลื่อนที่มาประยุกต์เข้ากับการพัฒนาซอฟท์แวร์ในสมาร์ทโฟน

1.3.2 พัฒนาระบบการเชื่อมต่อของสมาร์ทโฟนในลักษณะ Ad hoc โดยที่ไม่ต้องพึ่งพาสถานีฐาน

1.3.3 พัฒนากลไกการค้นหาเส้นทางของเครือข่าย MANET และกลไกการส่งข้อมูลในลักษณะ Multi-hop

1.3.4 พัฒนาต้นแบบเครือข่ายเฉพาะกิจบนสมาร์ทโฟน โดยใช้กลไกการส่งข้อมูลแบบ Multi-hop และมีโครงสร้างการเชื่อมต่อในลักษณะ MANET

1.3.5 ทดสอบ และประเมินประสิทธิภาพในการสื่อสารในเครือข่ายที่ได้จำลองขึ้น

1.4 ขั้นตอนการวิจัย

1.4.1 ศึกษางานวิจัยที่เกี่ยวข้องในเรื่องของการเชื่อมต่ออุปกรณ์ไร้สายในการสร้างเครือข่ายเฉพาะกิจเคลื่อนที่ (Mobile Ad hoc Network: MANET)

1.4.2 ศึกษาเทคโนโลยีไร้สายที่ใช้ในการติดต่อระหว่างสมาร์ทโฟน

1.4.3 พัฒนา และทดสอบการส่งข้อมูลระหว่างสมาร์ทโฟนในลักษณะ Single-Hop

1.4.4 ศึกษากลไกค้นหาเส้นทางสำหรับส่งข้อมูลในลักษณะ Multi-Hop สำหรับเครือข่าย MANET

1.4.5 พัฒนาระบบเครือข่ายเฉพาะกิจเคลื่อนที่ ซึ่งทำให้สมาร์ทโฟนสามารถส่งข้อมูลในลักษณะ Multi-Hop โดยใช้กลไกค้นหาเส้นทางสำหรับเครือข่าย MANET

1.4.6 ทดสอบ และประเมินประสิทธิภาพในการสื่อสาร โดยการใช้สมาร์ทโฟนสร้างเครือข่ายเฉพาะกิจเคลื่อนที่ในพื้นที่ที่จำลองขึ้นมาสำหรับการทดสอบ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

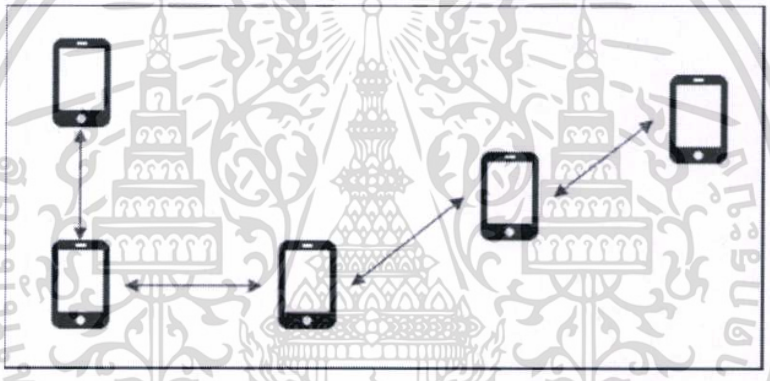
1.5.1 สามารถนำชุดต้นแบบเครือข่ายเฉพาะกิจเคลื่อนที่บนสมาร์ทโฟนไปประยุกต์ใช้เป็นระบบสื่อสารสำรองสำหรับใช้งานในสถานการณ์ที่ระบบสื่อสารหลัก ซึ่งใช้โครงสร้างพื้นฐาน ไม่สามารถใช้งานได้ปกติ รวมถึงการใช้งานในการส่งข้อมูลระยะใกล้ ซึ่งจะทำให้ไม่ต้องเสียค่าบริการในการแลกเปลี่ยนข้อมูล

1.5.2 เป็นการเพิ่มศักยภาพในการทำงานของสมาร์ทโฟน ซึ่งช่วยให้สมาร์ทโฟนสามารถแลกเปลี่ยนข้อมูลกันผ่านเทคโนโลยีเครือข่าย MANET โดยที่ไม่จำเป็นต้องใช้อุปกรณ์อื่นใด ๆ เพิ่มเติม

บทที่ 2
การทบทวนวรรณกรรมที่เกี่ยวข้อง

2.1 เครือข่ายเฉพาะกิจเคลื่อนที่ (Mobile Ad Hoc Network: MANET)

Mobile Ad hoc Network (MANET) [1] คือ เครือข่ายของโหนดแบบไร้สายที่เชื่อมต่อกันอย่างอิสระ โดยไม่จำเป็นต้องมีการพึ่งพาโครงสร้างพื้นฐาน เช่น สถานีฐาน (Base Station) โดยที่มักจะมีการเคลื่อนที่ซึ่งทำให้โครงสร้างการเชื่อมต่อนั้นมีการเปลี่ยนแปลงตลอดเวลา แต่ละโหนดจะมีความสามารถในการส่งต่อข้อมูลจากโหนดหนึ่งไปยังโหนดหนึ่ง ซึ่งในการส่งข้อมูลให้ไปถึงปลายทางนั้น โหนดจะต้องรู้เส้นทางไปยังปลายทาง และแต่ละโหนดจะทำหน้าที่ส่งต่อข้อมูลตามข้อมูลเส้นทางที่ทราบ โดยใช้โพรโทคอลที่ออกแบบมาสำหรับเครือข่ายเฉพาะกิจโดยเฉพาะ จึงทำให้การเชื่อมต่อแบบเครือข่ายเฉพาะกิจนั้นมีความยืดหยุ่นสูง



ภาพที่ 2.1 แสดงตัวอย่างการสื่อสารในเครือข่าย MANET

2.2 Wireless Fidelity (Wi-Fi)

ในปัจจุบันเทคโนโลยีการสื่อสารไร้สายได้พัฒนาขึ้นอย่างรวดเร็วอุปกรณ์การสื่อสารสามารถที่จะรองรับการเชื่อมต่อไร้สายได้ ในปี ค.ศ. 1999 ได้มีการจัดตั้งองค์กร Wi-Fi Alliance ที่เป็นองค์กรไม่แสวงหาผลกำไรขึ้น เพื่อกำหนดมาตรฐานของอุปกรณ์ Wireless Local Area Network (WLAN) โดยใช้เครื่องหมาย Wi-Fi

WLAN คืออุปกรณ์การเชื่อมต่อแบบไร้สายที่สื่อสารกันด้วยสัญญาณวิทยุ WLAN ถูกพัฒนาขึ้นโดย Institute of Electrical and Electronic Engineer (IEEE) ตามมาตรฐาน IEEE 802.11 โดยการรับรองของ Wi-Fi อุปกรณ์ใน WLAN ถูกแบ่งออกเป็นสองกลุ่มคือ

Stations (STA) คือ อุปกรณ์ที่รองรับมาตรฐาน IEEE 802.11 [1] ที่ถูกเชื่อมต่อแบบ Infrastructure และ Ad hoc mode โดย Stations อาจจะเป็นสมาร์ทโฟนหรือคอมพิวเตอร์

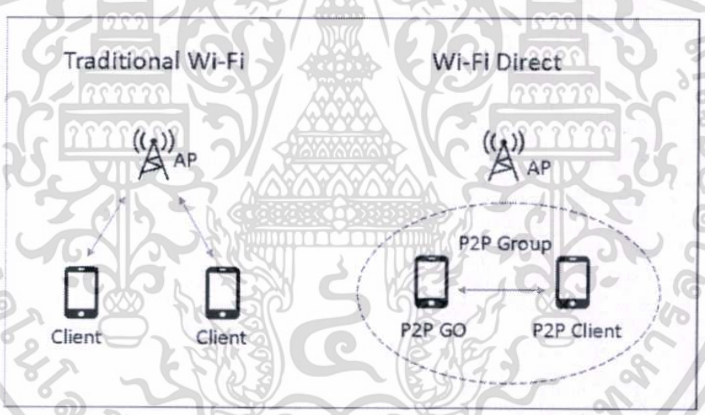
Access Point (AP) คือ อุปกรณ์ที่ถูกใช้ในการเชื่อมต่อแบบ Infrastructure โดยทุก STA จะถูกเชื่อมต่อกับ AP ในการเชื่อมต่อรูปแบบดาว โดย AP จะส่งข้อมูลสื่อสารกับ STA ซึ่ง AP สามารถทำงานเป็น Bride mode ซึ่งทำให้อุปกรณ์ที่เชื่อมต่อกับ AP สามารถที่จะสื่อสารไปยังเครือข่ายอื่นที่อยู่นอกเหนือจากที่เชื่อมต่ออยู่กับ AP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

การเชื่อมต่อแบบไร้สายมีรูปแบบการเชื่อมต่อสองแบบ [2] คือ Infrastructure Mode ที่มี AP เป็นศูนย์กลางการเชื่อมต่อ และ Ad hoc mode ที่มีรูปแบบการเชื่อมต่อแบบโหนดต่อโหนด (Peer-to-Peer) โดย STA แต่ละตัวสามารถเชื่อมต่อกันได้โดยตรงไม่ต้องอาศัย AP ดังภาพที่ 2.1

2.3 Wi-Fi Direct (Wi-Fi Peer-to-Peer)

Wi-Fi Direct หรือ Wi-Fi Peer-to Peer (P2P) มีการทำงานที่เหมือนกับเทคโนโลยี Wi-Fi แบบดั้งเดิม โดยมีส่วนที่แตกต่างคือ Wi-Fi Direct ไม่จำเป็นต้องเชื่อมต่อผ่าน AP ทางกายภาพ แต่จะใช้ซอฟท์แวร์ AP ในการเชื่อมต่อแทน โดยการเชื่อมต่อนั้นจะถูกจัดเป็นกลุ่ม ๆ ในแต่ละกลุ่มมีอุปกรณ์ที่ทำหน้าที่เป็น Group Owner (GO) ซึ่งใช้ซอฟต์แวร์มาทำให้ GO มีหน้าที่เหมือน AP ในการเชื่อมต่อแบบ Infrastructure Mode โดยจะมีการเชื่อมต่อเข้ามาจาก P2P Client เพื่อเข้าใช้งานกับ GO และ GO จะทำหน้าที่เป็นเครื่องแม่ข่ายที่มีการทำงานของ Dynamic Host Configuration Protocol (DHCP) เพื่อแจกจ่ายหมายเลขไอพีให้กับ P2P Client มาใช้ติดต่อกันตามรูปแบบของ OSI Network Layer หน้าที่การทำงานของ P2P Client จะทำงานเหมือนเทคโนโลยีการเชื่อมต่อแบบเก่าที่มีการใช้งาน AP ในการเชื่อมต่อ



ภาพที่ 2.2 เปรียบเทียบการทำงานของ Wi-Fi กับ Wi-Fi Direct

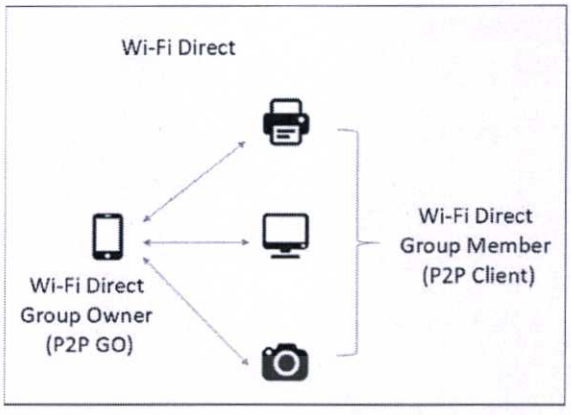
รูปแบบการค้นหาอุปกรณ์ มี 3 รูปแบบดังต่อไปนี้

2.3.1 Standard formation จะถูกใช้เมื่ออุปกรณ์ที่จะทำการเชื่อมต่อกันไม่ถูกเคยเชื่อมต่อเข้ากับ P2P Group เดียวกันมาก่อน จะต้องมีกระบวนการเพื่อหาว่าอุปกรณ์ใดจะทำหน้าที่เป็น GO โดยจะทำงานใน Discovery phase และ Find phase [3]

2.3.2 Autonomous formation จะถูกใช้เมื่อเลือกอุปกรณ์ที่จะทำหน้าที่เป็น GO ได้แล้ว โดยอุปกรณ์อื่นจะเชื่อมเข้ามาเป็น Client จะข้ามกระบวนการเลือก GO ไป

2.3.3 Persistent formation เมื่ออุปกรณ์มีการเชื่อมต่อกันอยู่ใน Group เดียวกันแล้วมีการขาดการเชื่อมต่อ เมื่อกลับมาเชื่อมต่อกันใหม่จะสามารถที่จะเชื่อมต่อกันได้เร็วขึ้น เนื่องจากมีการเก็บสถานะในการเชื่อมต่อเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

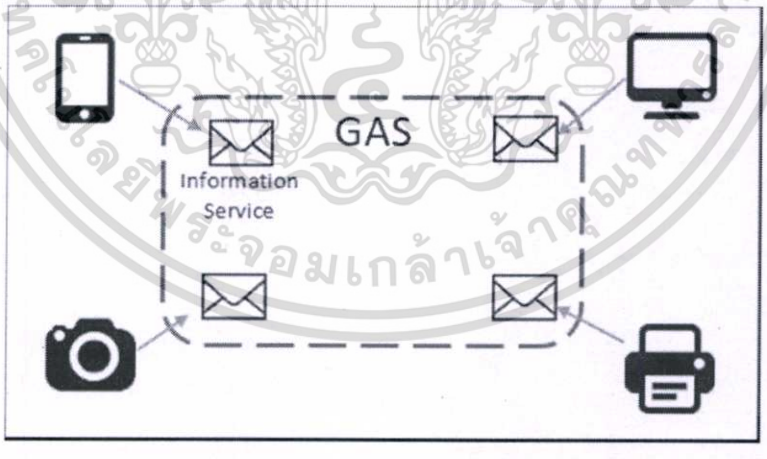


ภาพที่ 2.3 แสดงการติดต่อระหว่างอุปกรณ์ของ Wi-Fi Direct

2.4 Generic Advertisement Service (GAS)

GAS เป็นโพรโทคอลที่ใช้แลกเปลี่ยนข้อมูลเกี่ยวกับเซอร์วิสที่ถูกประกาศออกมาจากตัวอุปกรณ์ โดย GAS ถูกออกแบบมาให้ใช้งานบรรจุข้อมูลที่ต้องใช้ในการติดต่อสื่อสาร และถูกออกแบบมาให้ใช้ในการส่งข้อมูลได้ในอนาคต

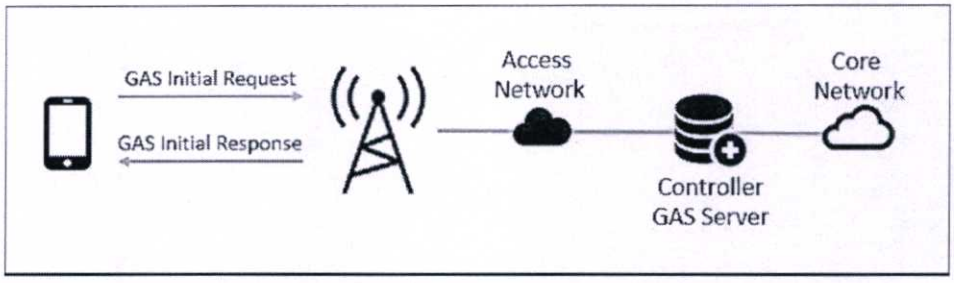
โดยโพรโทคอล GAS นั้น เปรียบเสมือนช่องทางสื่อสารที่แต่ละอุปกรณ์สามารถประกาศเซอร์วิสเข้ามาในช่องทางนี้ได้และอุปกรณ์อื่น ๆ ที่อยู่ในช่องทาง สามารถที่จะนำเซอร์วิสต่าง ๆ ที่ถูกประกาศไว้ไปใช้งานต่อได้ ซึ่งในหลักการทำงานนี้ของ GAS ผู้จัดทำจึงได้ทำการปรับเปลี่ยนชุดข้อมูลที่ใช้ในการสื่อสาร (Information Service) ให้เป็นชุดข้อความที่แอพพลิเคชันส่งออกไปในเครือข่าย ทำให้อุปกรณ์ที่อยู่ในช่องทางการสื่อสาร GAS สามารถติดต่อสื่อสารกันได้โดยตรง



ภาพที่ 2.4 แสดงหลักการทำงานของ GAS Service

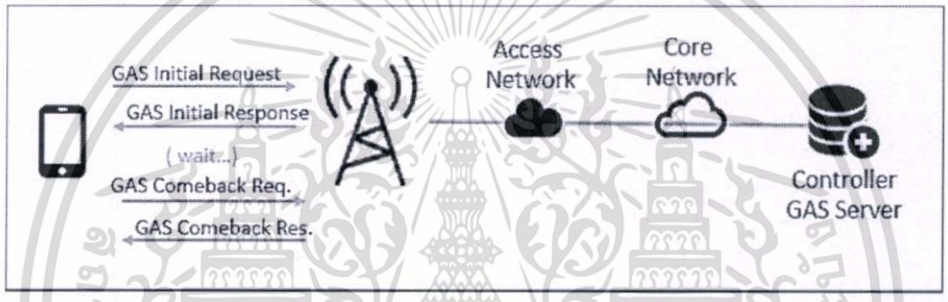
การทำงานของ GAS อยู่ในรูปแบบ Request และ Response ซึ่งรูปแบบการติดต่อแสดงดังภาพที่ 2.4 [4] กระบวนการการทำงานของ GAS ที่เกิดขึ้นระหว่าง 2 อุปกรณ์ โดยที่เครื่องหนึ่งจะทำการส่ง GAS Initial Request เพื่อทำการร้องขอข้อมูลเซอร์วิสจากอุปกรณ์อีกเครื่อง และอุปกรณ์ที่ได้รับการร้องขอจะตอบกลับโดยจะทำการส่งข้อมูลเซอร์วิสส่งไปในรูปแบบ GAS Initial Response

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



ภาพที่ 2.5 แสดงรูปแบบการสื่อสารของโพรโทคอล GAS

แต่ถ้าหากในกรณีที่ข้อมูลเซอร์วิสมีขนาดใหญ่เกินกว่าที่กำหนดจะทำการแบ่งส่งข้อมูล โดยให้ตัวอุปกรณ์ที่ทำการร้องขอมานั้นส่ง GAS Comeback Request กลับมาอีกครั้งเพื่อนร้องขอข้อมูลที่เหลือดังภาพที่ 2.5



ภาพที่ 2.6 แสดงรูปแบบการสื่อสารเมื่อข้อมูล Service มีขนาดใหญ่เกินไป

2.5 โพรโทคอลค้นหาเส้นทางในเครือข่ายเฉพาะกิจเคลื่อนที่

2.5.1 Ad Hoc On-demand Distance Vector (AODV)

Ad Hoc On-demand Distance Vector (AODV) [5] เป็นโพรโทคอลประเภท Distance Vector และมีการทำงานแบบเชิงรับ (Reactive) โหนดต้นทางจะค้นหาเส้นทางเมื่อต้องการส่งข้อมูลโดยการกระจายแพ็คเกต Route Request (RREQ) ออกไป

ในการส่งแพ็คเกต RREQ โหนดจะเก็บข้อมูลลงไปในตารางเส้นทาง (Routing Table) ของแต่ละโหนดที่มีแพ็คเกต RREQ ถูกส่งผ่านไปมา โหนดจะทำการเพิ่มค่าฮ็อป (Hop Count) ในแพ็คเกตเพิ่มขึ้นทีละ 1 และมีการบันทึกเส้นทางที่ชี้ไปยังโหนดก่อนหน้าที่ส่งต่อแพ็คเกตนี้เป็นเส้นทางย้อนกลับไปหาโหนดต้นทางในตารางเส้นทาง

AODV จะมีการระบุหมายเลขลำดับของโหนดปลายทาง (Destination Sequence Number) เพื่อใช้ในการตัดสินว่าข้อมูลเส้นทางจากแพ็คเกตนั้นมีความใหม่ของข้อมูล โดยเมื่อโหนดใด ๆ ได้รับแพ็คเกต RREQ และโหนดนั้นมีข้อมูลโหนดถัดไปที่จะส่งต่อไปยังปลายทางเดียวกันอยู่ในตารางเส้นทางแล้ว โหนดจะส่งแพ็คเกต Route Reply (RREP) กลับไปยังโหนดต้นทางก็ต่อเมื่อหมายเลขลำดับของโหนดปลายทางที่อยู่ในแพ็คเกต RREQ ที่ได้รับมานั้นมีค่าน้อยกว่า หรือเท่ากับหมายเลขลำดับของปลายทางที่อยู่ในตารางเส้นทางของแต่ถ้ามีค่ามากกว่า โหนดนั้นจะกระจายแพ็คเกต RREQ ต่อไป เมื่อแพ็คเกต RREQ ไปถึงโหนดปลายทาง โหนดปลายทางจะส่งแพ็คเกต RREP ที่มีหมายเลขลำดับใหม่ที่สุดกลับไปยัง

สารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

โหนดต้นทาง กลไกนี้จะช่วยป้องกันไม่ให้โหนดระหว่างทางใด ๆ สร้างแพ็คเกต RREP ที่มีข้อมูลเส้นทางเก่าที่ใช้งานไม่ได้ (Stale Route) กลับไปยังโหนดต้นทาง ทำให้ลดโอกาสที่โหนดต้นทางจะส่งข้อมูลโดยใช้เส้นทางที่ใช้งานไม่ได้

ในการส่งต่อแพ็คเกตข้อมูลไปยังโหนดถัดไปนั้น จะใช้กลไกของ Link layer มาตรวจสอบว่าสามารถส่งแพ็คเกตไปยังโหนดถัดไปได้หรือไม่ เช่น กรณีที่ไม่ได้รับ ACK (ใน Link layer) จนกระทั่งมีการส่งซ้ำจนถึงจำนวนที่กำหนด Link layer จะแจ้งต่อโพรโทคอล AODV ว่าไม่สามารถส่งแพ็คเกตข้อมูลไปยังโหนดถัดไปได้ นอกจากนี้โหนดที่เป็นตัวกลางในเส้นทางจะกระจายแพ็คเกต Hello ที่มีค่า TTL เป็น 1 เป็นระยะตามเวลาที่กำหนด (โดยประมาณ 1 วินาที) ถ้าโหนดไม่ได้รับแพ็คเกต Hello จากโหนดเพื่อนบ้านที่เป็นเส้นทางไปยังโหนดปลายทางเป็นจำนวน และในระยะเวลาที่กำหนด โหนดจะถือว่าโหนดเพื่อนบ้านนั้นไม่สามารถติดต่อได้เช่นเดียวกัน โหนดที่ไม่สามารถติดต่อโหนดถัดไปได้สามารถที่จะเริ่มใช้กลไก Local Repair โดยจะส่งแพ็คเกต RREQ แทนโหนดต้นทาง และจะทำได้ก็ต่อเมื่อโหนดนั้นมีจำนวนฮ็อปจากตัวเองไปยังปลายทางไม่เกินค่าที่กำหนดไว้ (แพ็คเกตข้อมูลที่ส่งไม่ได้เนื่องจากเส้นทางขาด จะถูกเก็บไว้ในบัฟเฟอร์) เมื่อโหนดได้รับแพ็คเกต Route Reply ก็จะนำจำนวนฮ็อปไปยังปลายทางมาเปรียบเทียบกับเส้นทางเก่า (ที่ใช้ไม่ได้) โดยแพ็คเกตข้อมูลจะถูกส่งต่อไปยังปลายทางด้วยเส้นทางใหม่ ก็ต่อเมื่อเส้นทางใหม่มีจำนวนฮ็อปไปยังปลายทางไม่มากกว่าเส้นทางเก่าเท่านั้น แต่ถ้าไม่เข้าเงื่อนไข โหนดนั้นจะส่งแพ็คเกต Route Error (RERR) กลับไปยังโหนดต้นทาง เพื่อแจ้งให้โหนดที่อยู่ระหว่างเส้นทางที่ขาดนั้นทราบและให้โหนดต้นทางทำการค้นหาเส้นทางใหม่อีกครั้ง

2.5.2 Dynamic Source Routing (DSR)

Dynamic Source Routing (DSR) เป็นโพรโทคอลค้นหาเส้นทางแบบเชิงรับ โดยโหนดต้นทางจะมีการกระจายแพ็คเกตชนิดพิเศษที่มีช่องในการเก็บข้อมูลของโหนดต่าง ๆ ภายในเครือข่าย เช่น ข้อมูลไอพีแอดเดรส ของโหนดที่ส่งผ่านเข้าไปในเฮดเดอร์ของแพ็คเกต และเมื่อพบโหนดปลายทางแล้วก็จะส่งแพ็คเกตค้นหาเส้นทางที่มีเส้นทางจากโหนดต้นทางไปยังปลายทางในเฮดเดอร์กลับไปยังโหนดต้นทาง จากนั้นก็จะเริ่มทำการสร้างเส้นทางเพื่อทำการส่งแพ็คเกตปกติตามเส้นทางที่ได้ทำการบันทึกไว้

กลไก Route Discovery หรือการค้นหาเส้นทางจะเกิดขึ้นเมื่อโหนดใด ๆ ต้องการส่งแพ็คเกตข้อมูลไปยังปลายทาง และยังไม่มีข้อมูลเส้นทางไปยังปลายทางที่ต้องการ โหนดจะทำการกระจายแพ็คเกต Route Request (RREQ) กระจายออกไปยังทุก ๆ โหนดข้างเคียง โดยที่อยู่ของในตัวแพ็คเกตจะมีช่องที่ถูกกำหนดเป็นหมายเลขไอพีของโหนดปลายทางเพื่อใช้สำหรับตรวจสอบว่าตนเองเป็นโหนดปลายทางสำหรับการส่งข้อมูลครั้งนั้นหรือไม่

เมื่อโหนดแต่ละตัวได้รับแพ็คเกต RREQ ของโพรโทคอล DSR จะตรวจสอบดังนี้

1. หากตนเองเป็นโหนดปลายทาง ก็จะทำการตอบกลับไปยังโหนดต้นทางผ่านทางโมบายล์โหนดที่ทำการกระจายมา ด้วยแพ็คเกต Route Reply (RREP) โดยส่งกลับในลักษณะยูนิแคสท์ (Unicast)
2. หากโหนดเคยได้รับแพ็คเกตนี้มาก่อนหน้าแล้ว โดยตรวจสอบจากหมายเลข ID ของแพ็คเกต และเส้นทางที่แพ็คเกตนี้เคยผ่านมา) ก็จะละทิ้งแพ็คเกตนั้น

3. หากโหนดมีข้อมูลเส้นทางไปยังโหนดปลายทางเก็บไว้ในแคช (Route Cache) อยู่แล้วก็จะทำการตอบกลับด้วยแพ็คเกต RREP ที่มีเส้นทางตั้งแต่โหนดต้นทางไปยังปลายทางในเฮดเดอร์กลับไปยังโหนดต้นทาง
4. นอกเหนือจากกรณีข้างต้น โหนดจะทำการเพิ่มหมายเลขระบุตัวตนของตนเอง (เช่น หมายเลขไอพี) ลงไปแพ็คเกต และทำการกระจายแพ็คเกตต่อไป

เมื่อโหนดต้นทางได้รับแพ็คเกต RREP แล้ว ก็จะนำข้อมูลเส้นทางที่ได้จากแพ็คเกต RREP มาใส่ลงในแพ็คเกตข้อมูล และทำการส่งแพ็คเกตข้อมูลออกไป ในกรณีถ้าโหนดต้นทางไม่ได้รับแพ็คเกต RREP ในระยะเวลาที่กำหนด โหนดต้นทางจะส่งแพ็คเกต RREQ ใหม่อีกครั้ง โดยมีการกำหนดหมายเลข ID ให้เพิ่มขึ้น

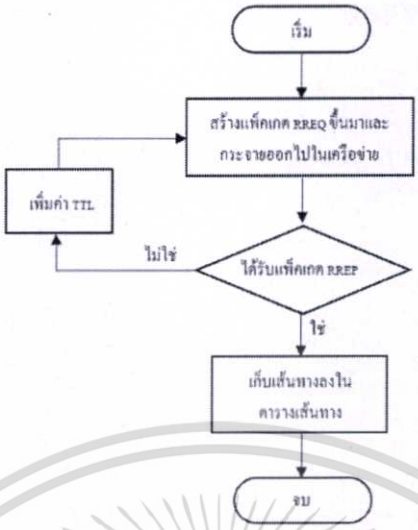
กลไก Route Maintenance เป็นกลไกที่ใช้จัดการกับกรณีที่เส้นทางขาด อันเนื่องมาจากการเคลื่อนที่ของโหนดภายในเครือข่าย โดยเมื่อโหนดใด ๆ กำลังส่งต่อข้อมูลไปยังโหนดไปแต่ไม่สามารถส่งไปยังโหนดถัดไปได้ โหนด (ที่ส่งต่อข้อมูลไม่ได้) จะส่งแพ็คเกต Route Error (RERR) แบบยูนิแคสท์กลับมายังโหนดต้นทางเพื่อแจ้งให้โหนดต้นทางทราบว่าเส้นทางดังกล่าวใช้ไม่ได้แล้ว และโหนดต้นทางจะเริ่มกระบวนการ Route Discovery ใหม่อีกครั้งหนึ่ง

โพรโทคอล DSR นั้นจะมีการกำหนดเส้นทางที่จะส่งแพ็คเกตข้อมูลนั้น ๆ โดยโหนดต้นทาง (Source Routing) กล่าวคือ ข้อมูลจะถูกส่งต่อไปยังโหนดถัดไปตามเส้นทางที่โหนดต้นทางได้กำหนดเอาไว้ในเฮดเดอร์ของแพ็คเกตข้อมูลเท่านั้น

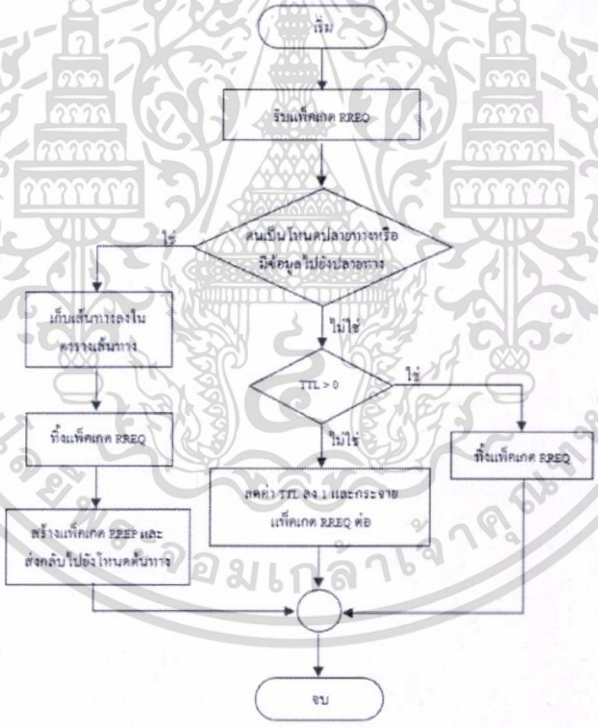
2.5.3 Expanding Ring Search (ERS)

Expanding Ring Search (ERS) [6] เป็นกระบวนการหนึ่งที่ใช้ในการค้นหาเส้นทางจากโหนดต้นทางไปยังปลายทาง โดยใช้ค่า Time to Live (TTL) เป็นเงื่อนไขในการส่งต่อแพ็คเกต RREQ ในการค้นหาเส้นทาง โหนดต้นทางจะทำการสร้างแพ็คเกต RREQ โดยเริ่มจาก TTL ค่าน้อย ๆ และกระจายแพ็คเกตไปยังโหนดเพื่อนบ้าน การส่งต่อแพ็คเกตแต่ละครั้งค่า TTL จะลดลงทีละ 1 และโหนดที่ได้รับแพ็คเกต RREQ จะทำการพิจารณาค่า TTL ว่าต้องทำการส่งต่อแพ็คเกตนี้หรือไม่ ถ้าค่า TTL มีค่ามากกว่า 0 โหนดจะกระจายแพ็คเกตต่อ แต่ถ้าค่า TTL มีค่าเท่ากับ 0 จะหยุดการกระจายแพ็คเกต หลังจากที่โหนดต้นทางทำการกระจายแพ็คเกต RREQ ไปแล้ว ถ้าไม่ได้รับแพ็คเกต RREP ภายในเวลาที่กำหนด โหนดจะสร้างแพ็คเกต RREQ ขึ้นมาใหม่ และทำการเพิ่มค่า TTL ขึ้น เพื่อเป็นการเพิ่มระยะทางในการกระจายแพ็คเกต เมื่อโหนดปลายทางได้รับแพ็คเกต RREQ แล้ว จะทำการสร้าง RREP ส่งกลับไปยังโหนดต้นทาง

ผลจากกลไก ERS คือทำให้ RREQ ไม่กระจายแพ็คเกตไปยังทุกโหนด บนเครือข่ายเพื่อค้นหาโหนดปลายทาง แต่จะค่อยๆเพิ่มระยะทาง (จำนวนฮ็อป) ในการค้นหา ทำให้ช่วยลดโอเวอร์เฮดของการค้นหาเส้นทาง และลดการใช้พลังงานในเครือข่ายได้



ภาพที่ 2.7 กลไกการทำงานของโพรโทคอล ERS ที่โหนดต้นทาง



ภาพที่ 2.8 กลไกการทำงานของโพรโทคอล ERS ที่โหนดระหว่างทาง

2.5.4 Blocking Expanding Ring Search (BERS)

Blocking Expanding Ring Search (BERS) [7] เป็นกระบวนการค้นหาเส้นทางที่พัฒนามาจาก ERS เพื่อปรับปรุงการแพร่กระจายของแพ็คเกต RREQ ให้มีประสิทธิภาพมากขึ้น โดยการใช้แพ็คเกต stop_instruction ในการควบคุมการกระจายแพ็คเกต RREQ แทนค่า TTL

ในการค้นหาเส้นทางโหนดต้นทางจะสร้างแพ็คเกต RREQ และกระจายแพ็คเกตออกไปเป็นวงแหวน (Ring) โดยเริ่มจากวงแหวนรอบที่ 1 ซึ่งวงแหวนรอบที่ 1 จะมีค่า Hop count เท่ากับ 1 ดังนั้น ทุกโหนดในวงแหวนที่ 1 จะได้รับแพ็คเกต RREQ เมื่อได้รับแพ็คเกตแล้วโหนดจะมีเวลารอ (Waiting time) ซึ่งหาค่าได้จาก

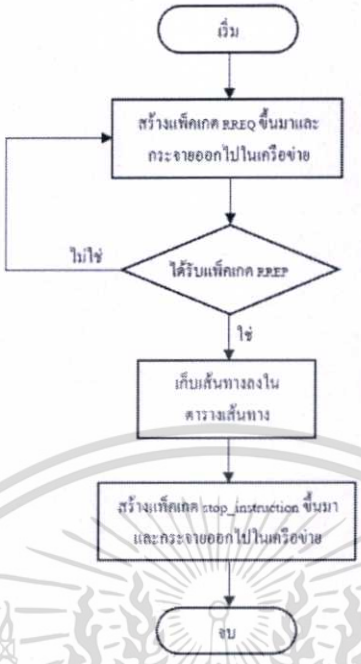
$$\text{WAITING TIME} = 2 * \text{Hop Count} * \text{TRAVERSAL TIME} \quad (2.1)$$

โดยที่ Waiting time คือ เวลารอ Hop Count คือ จำนวนฮ็อป และ TRAVERSAL TIME คือเวลาที่ใช้ในการส่งแพ็คเกตในระยะ 1 ฮ็อป

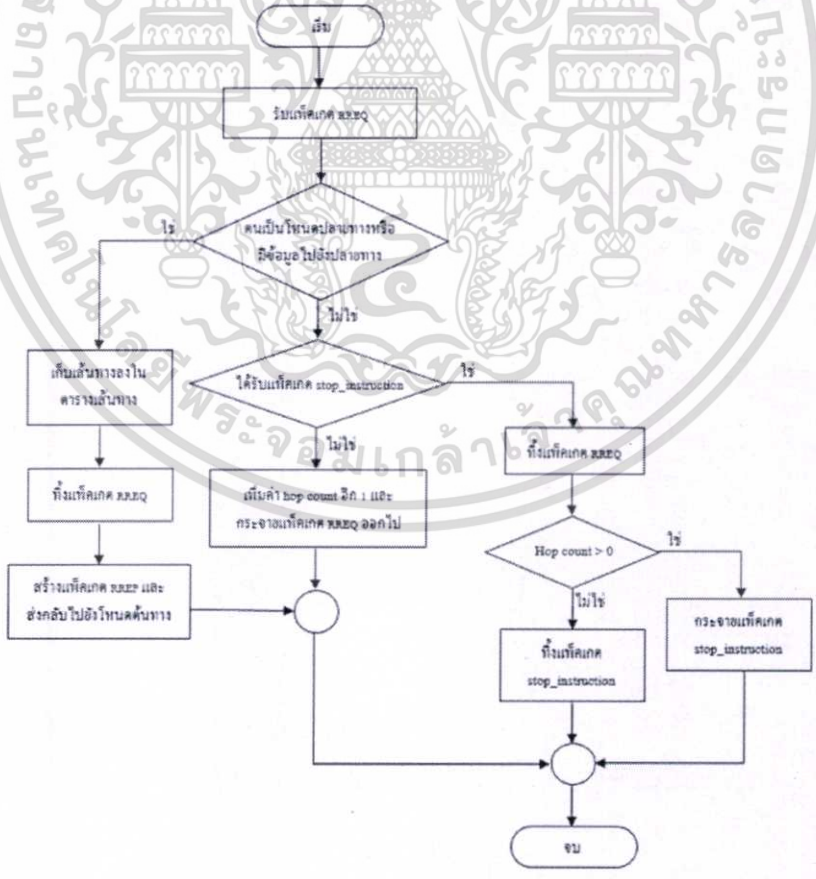
หากโหนดไม่ได้รับแพ็คเกต stop_instruction ภายในเวลารอนั้น หมายความว่า ไม่มีโหนดใดที่มีข้อมูลเส้นทางของโหนดปลายทาง และทุกโหนดในวงแหวนนั้นจะเพิ่มค่า Hop Count ในแพ็คเกต RREQ และกระจายแพ็คเกตไปยังวงแหวนถัดไป แต่ถ้าในเวลารอนั้นโหนดได้รับแพ็คเกต stop_instruction โหนดจะทำการละทิ้งแพ็คเกต RREQ ทันที และเมื่อแพ็คเกต RREQ ไปถึงโหนดปลายทางหรือโหนดที่มีข้อมูลไปยังโหนดปลายทาง โหนดจะทำการสร้างแพ็คเกต RREP และส่งกลับมายังโหนดต้นทาง เมื่อโหนดต้นทางได้รับ RREP แล้วโหนดจะสร้างแพ็คเกต stop_instruction และกระจายออกไปตาม Hop Count ที่ได้รับจากแพ็คเกต RREP เพื่อให้ทุกโหนดหยุดกระจายแพ็คเกต RREQ

BERS จะไม่ทำการกระจายแพ็คเกต RREQ จากโหนดต้นทางทุกครั้งเหมือน ERS แต่โหนดที่ได้รับแพ็คเกต RREQ จะทำการกระจายแพ็ตเกตแทน เพื่อลดความซ้ำซ้อนในการกระจายแพ็คเกต และใช้แพ็คเกต stop_instruction ในการควบคุมการกระจายของแพ็คเกตค้นหาเส้นทาง ทำให้ช่วยลดการใช้พลังงานในเครือข่ายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึก



ภาพที่ 2.9 กลไกการทำงานของโพรโทคอล BERS ที่โหนดต้นทาง



ภาพที่ 2.10 กลไกการทำงานของโพรโทคอล BERS ที่โหนดระหว่างทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.5 Blocking Expanding Ring Search Plus (BERS+)

Blocking Expanding Ring Search+ (BERS+) [8] เป็นโพรโทคอลที่พัฒนามาจากโพรโทคอล BERS ซึ่งจะปรับปรุงเรื่องอัตราการส่งข้อมูลสำเร็จ โดยจะมีเงื่อนไขการส่งต่อของแพ็คเกต stop_instruction แทนการกระจายแพ็คเกต stop_instruction ที่จะกระจายแพ็คเกตเท่ากับจำนวนฮ็อปของแพ็คเกต RREP ที่ได้รับมา ทำให้แพ็คเกต stop_instruction มีโอกาสหยุดการแพร่กระจายของแพ็คเกต RREQ ได้มากกว่า

เงื่อนไขการกระจายแพ็คเกต stop_instruction คือ หากโหนดได้รับแพ็คเกต stop_instruction ซ้ำ หรือได้รับแพ็คเกต stop_instruction แต่ยังไม่ได้รับแพ็คเกต RREQ ของแพ็คเกต stop_instruction นั้น โหนดจะละทิ้งแพ็คเกตทันที แต่ถ้าโหนดเคยได้รับ และส่งต่อแพ็คเกต RREQ ไปแล้ว โหนดจะส่งต่อแพ็คเกต stop_instruction ถ้าหากโหนดได้รับแพ็คเกต stop_instruction และยังมีแพ็คเกต RREQ ที่รอการส่งอยู่ในโหนดเดียวกัน โหนดตัวนั้นจะละทิ้งทั้งแพ็คเกต RREQ และแพ็คเกต stop_instruction เพื่อเป็นการหยุดการแพร่กระจายของแพ็คเกต RREQ

2.5.6 Query Localization (QL)

Query Localization (QL) [9] เป็นแนวคิดในการลดโอเวอร์เฮดสำหรับโพรโทคอลค้นหาเส้นทางเชิงรับ แนวคิดหลักของโพรโทคอลนี้คือ เส้นทางใหม่น่าจะอยู่บริเวณเส้นทางเก่าที่เคยใช้งานในครั้งก่อนหน้า ในโพรโทคอลนี้จะมีการจำกัดการแพร่กระจายแพ็คเกต RREQ ให้ถูกกระจายต่อโดยโหนดที่อยู่ในเส้นทางเก่า และโหนดที่อยู่ในละแวกใกล้เคียงเท่านั้น QL จะมีตัวแปร k เป็นตัวแปรสำคัญที่กำหนดให้แพ็คเกต RREQ นั้นจะแพร่กระจายออกไปจากโหนดที่เป็นเส้นทางเก่าได้ไม่เกินกี่ฮ็อป ค่านี้จะถูกกำหนดโดยผู้ใช้ และส่งผลต่อประสิทธิภาพของโพรโทคอลโดยตรง หากผู้ใช้กำหนดค่า k ให้มีค่าน้อย ๆ จะส่งผลให้มีจำนวนโหนดที่แพร่กระจายแพ็คเกต RREQ น้อยตาม ซึ่งจะช่วยลดโอเวอร์เฮดได้ แต่ก็อาจจะส่งผลให้กระบวนการค้นหาเส้นทางล้มเหลวได้ในบางสถานการณ์เนื่องจากมีการจำกัดจำนวนโหนดมากเกินไป ทั้งนี้ QL สามารถนำไปใช้กับโพรโทคอลค้นหาเส้นทางประเภท Source Routing เช่น DSR และแบบ Hop-by-hop Routing เช่น AODV ได้

การทำงานของ QL บนโพรโทคอล DSR ซึ่งเป็นโพรโทคอลประเภท Source Routing นั้นจะเริ่มต้นโดยโหนดต้นทางสร้างแพ็คเกต RREQ กระจายออกไป และโหนดปลายทางส่งแพ็คเกต RREP กลับมายังโหนดต้นทางตามปกติของโพรโทคอล DSR

เมื่อโหนดต้นทางเริ่มกระบวนการค้นหาเส้นทางใหม่อันเนื่องมาจากเส้นทางใช้งานไม่ได้ แพ็คเกต RREQ ที่โหนดต้นทางสร้างขึ้นจะมีการใส่ข้อมูลเพิ่มเติมลงไปในแพ็คเกตที่สร้างขึ้น (ผู้วิจัยขอเรียกว่าแพ็คเกต $RREQ_{QL}$) ประกอบไปด้วย

1. ค่า k ซึ่งเป็นค่าที่กำหนดโดยผู้ใช้
2. ตัวนับ (counter) ซึ่งจะถูกกำหนดให้เป็น 0
3. ข้อมูลไอพีของโหนดที่เป็นตัวกลางทั้งหมดในเส้นทางเดิมก่อนหน้า

เมื่อโหนดใด ๆ ได้รับแพ็คเกต $RREQ_{QL}$ ที่ยังไม่เคยได้รับมาก่อน โหนดจะส่งแพ็คเกต RREP กลับไปถ้าหากมีข้อมูลเส้นทางไปยังโหนดปลายทาง แต่ถ้าไม่มี โหนดนั้นจะตรวจสอบว่าตัวโหนดนั้นเคยเป็นตัวกลางในเส้นทางเก่าหรือไม่โดยการตรวจสอบว่ามีข้อมูลไอพีของตนเองปรากฏอยู่ในข้อมูลเส้นทางเดิมที่แนบมากับแพ็คเกต $RREQ_{QL}$ ที่ได้รับมาหรือไม่ ถ้ามี

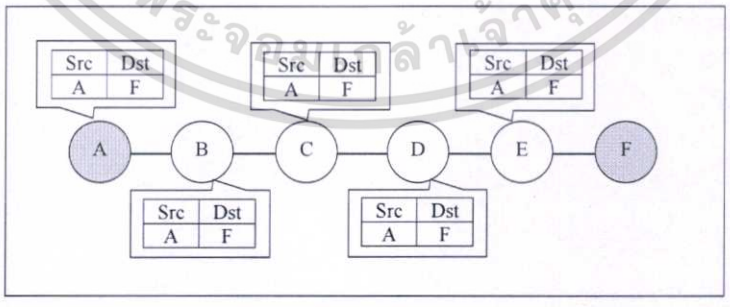
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

นั่นหมายความว่าโหนดนั้นเคยเป็นตัวกลางในเส้นทางระหว่างโหนดต้นทาง และโหนดปลายทางดังกล่าว โหนดนั้นจะทำการตั้งค่าตัวนับที่อยู่ในแพ็คเกต $RREQ_{QL}$ ใหม่ โดย QL ได้เสนอวิธีการ 2 แบบ โดยแบบแรกคือ Node Locality ซึ่งโหนดจะตั้งค่าตัวนับเป็น 0 ในขณะที่แบบที่ 2 คือ Path Locality โหนดจะไม่เปลี่ยนแปลงค่าของตัวนับโดยเมื่อโหนดแก้ไขค่าตัวนับเรียบร้อยแล้ว ก็จะทำการกระจายแพ็คเกต $RREQ_{QL}$ ต่อไปตามปกติ

ถ้าหากโหนดไม่พบข้อมูลไอพีของตนเองในแพ็คเกต $RREQ_{QL}$ นั่นหมายความว่าโหนดนั้นไม่เคยเป็นตัวกลางในเส้นทางการส่งข้อมูลเดิม โหนดจะแก้ไขค่าตัวนับในแพ็คเกต $RREQ_{QL}$ ให้บวกเพิ่มขึ้นอีก 1 จากนั้นโหนดจะทำการตรวจสอบว่าค่าตัวนับในแพ็คเกต $RREQ_{QL}$ โดยถ้าค่าตัวนับน้อยกว่าหรือเท่ากับค่า k โหนดจะกระจายแพ็คเกต $RREQ_{QL}$ ต่อไป แต่ถ้าตัวนับมีค่ามากกว่า ค่า k โหนดจะละทิ้งแพ็คเกต $RREQ_{QL}$ ทันที

การทำงานของ QL บนโพรโทคอล DSR ซึ่งเป็นโพรโทคอลประเภท Source Routing นั้นมีข้อเสียคือ แพ็คเกต $RREQ_{QL}$ นั้นนอกจากจะต้องเก็บข้อมูลไอพีของโหนดที่เป็นเส้นทางปัจจุบันตามกลไกของโพรโทคอล DSR แล้ว จะต้องเก็บข้อมูลไอพีของโหนดที่อยู่ในเส้นทางเก่าของ QL ด้วย ซึ่งจะทำให้แพ็คเกต $RREQ_{QL}$ มีขนาดใหญ่ขึ้นมาก ส่งผลด้านลบต่อประสิทธิภาพในกระบวนการค้นหาเส้นทาง

การทำงานของ QL บนโพรโทคอลประเภท Hop-by-hop Routing เช่น AODV นั้นจะช่วยลดจุดอ่อนของโพรโทคอลประเภท Source Routing โดยโหนดระหว่างทางจะทำการเก็บข้อมูลว่าตัวโหนดนั้นเป็นตัวกลางในการติดต่อสื่อสารระหว่างโหนดต้นทาง และโหนดปลายทางตัวใดแทน ซึ่งช่วยให้แพ็คเกต $RREQ_{QL}$ นั้นมีขนาดไม่ใหญ่มากเกินไป กระบวนการทำงานของ QL บนโพรโทคอล AODV จะเริ่มต้นโดยโหนดต้นทางสร้างแพ็คเกต RREQ กระจายออกไปทั้งเครือข่ายตามปกติ เมื่อโหนดปลายทางได้รับแพ็คเกต RREQ ก็จะส่งแพ็คเกต RREP กลับไปหาต้นทาง โดยเมื่อโหนดระหว่างทางได้รับแพ็คเกต RREP จะทำการบันทึกข้อมูลที่อยู่ในแพ็คเกต RREP ที่ได้รับลงในตารางพิเศษซึ่งประกอบไปด้วยข้อมูลหมายเลขไอพีของต้นทาง จับคู่กับหมายเลขไอพีของปลายทาง และกำหนดเวลาหมดอายุสำหรับคู่ของข้อมูลดังกล่าว

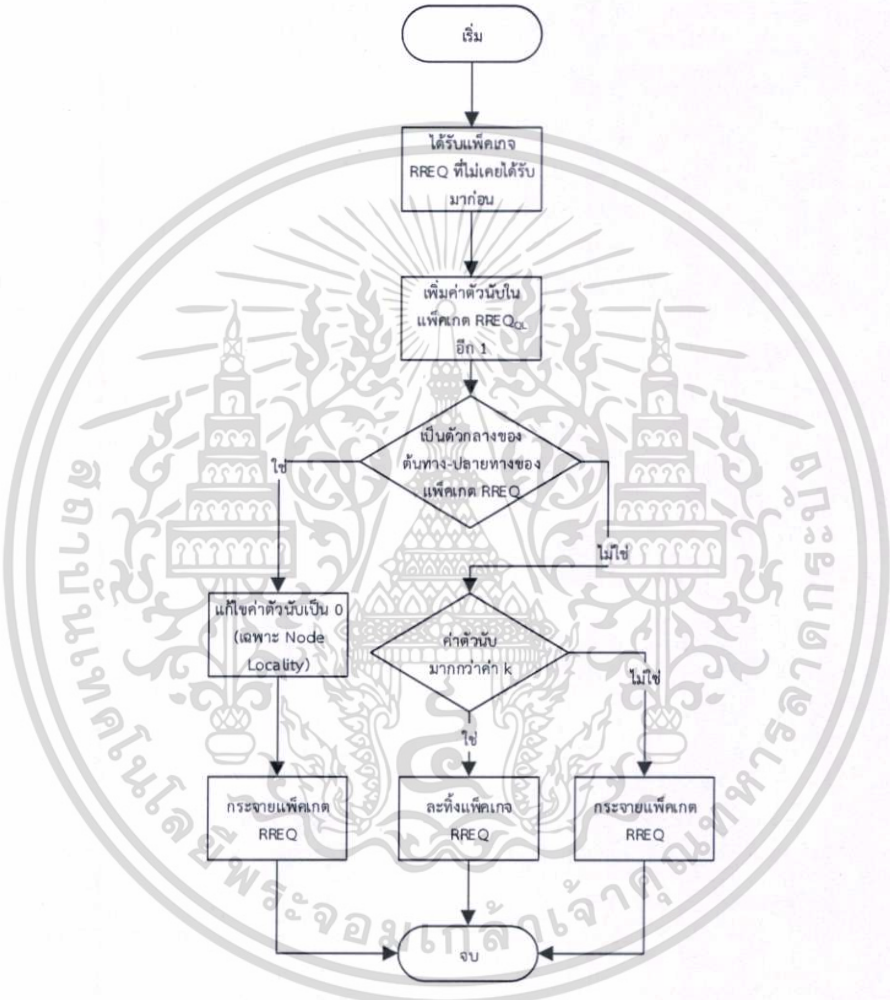


ภาพที่ 2.11 แสดงคู่ข้อมูลต้นทาง-ปลายทางที่เก็บในโหนดที่ได้รับแพ็คเกต RREP

ภาพที่ 2.11 แสดงตัวอย่างทอพอโลยีของเครือข่าย โดยเป็นการติดต่อระหว่างโหนดต้นทาง A ไปยังโหนดปลายทาง F เมื่อโหนด E ได้รับแพ็คเกต RREP จากโหนด F ก็จะบันทึกข้อมูลไอพีของโหนดต้นทาง A คู่กับโหนดปลายทาง F

เสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เมื่อโหนดต้นทางพบว่าเส้นทางขาด และจำเป็นที่จะต้องเริ่มกระบวนการค้นหาเส้นทางใหม่ โหนดต้นทางจะเริ่มกระจายแพ็คเกต $RREQ_{QL}$ โดยที่เมื่อโหนดระหว่างทางได้รับแพ็คเกต $RREQ_{QL}$ ก็จะทำตามกระบวนการของโพรโทคอล QL ตามปกติ แต่ในการตรวจสอบว่าโหนดนั้นเป็นตัวกลางของเส้นทางเดิมหรือไม่นั้นจะนำหมายเลขไอพีของโหนดต้นทาง และโหนดปลายทางที่ต้องการค้นหานั้นมาเทียบกับข้อมูลต้นทาง-ปลายทางที่ได้จัดเก็บไว้จากการได้รับแพ็คเกต RREP ในครั้งก่อนหน้านี้แทน



ภาพที่ 2.12 แผนผังการทำงานของโพรโทคอล QL เมื่อโหนดได้รับแพ็คเกต RREQ

กลไกของโพรโทคอล QL สามารถสรุปเป็นแผนผังการทำงานได้ตามภาพที่ 2.12 (แสดงกระบวนการเฉพาะในส่วนของการตัดสินใจกระจายแพ็คเกต $RREQ_{QL}$) วิธีการนี้จะทำให้แพ็คเกต RREQ ไม่จำเป็นต้องเก็บข้อมูลโหนดตัวกลางของเส้นทางเดิมไว้ และช่วยลดขนาดของแพ็คเกต RREQ อย่างไรก็ตาม วิธีการนี้จะไม่สามารถใช้ร่วมกับกลไก Local Repair ของโพรโทคอล AODV ได้อย่างสมบูรณ์ เนื่องจากหากโหนดระหว่างทางส่งแพ็คเกต RREQ แทนต้นทาง จะทำให้หมายเลขไอพีต้นทางของแพ็คเกต RREQ นั้นกลายเป็นของโหนดระหว่างทาง ซึ่งส่งผลให้โหนดระหว่างจะต้องกระจายแพ็คเกต RREQ แบบปกติใน AODV แทน โดยจากตัวอย่างภาพที่ 2.11 หากโหนด D เริ่มกระบวนการ Local Repair โดยใช้แพ็คเกต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

$RREQ_{QL}$ จะทำให้หมายเลขไอพีต้นทางของแพ็คเกตเป็นโหนด D ซึ่งทำให้เมื่อโหนด E ได้รับแพ็คเกตนี้จะทำให้ตัวโหนด E เข้าใจว่าตัวมันเองไม่ใช่ตัวกลางของเส้นทางเดิม ซึ่งจะทำให้มีการเพิ่มค่าตัวนับในแพ็คเกต และส่งผลให้ลดโอกาสที่แพ็คเกต $RREQ_{QL}$ จะแพร่กระจายไปถึงโหนดปลายทาง

2.6 โพรโทคอลสำหรับค้นหาเส้นทางไปยังเกตเวย์สำหรับเครือข่ายเฉพาะกิจเคลื่อนที่

2.6.1 โพรโทคอลค้นหาเกตเวย์แบบเชิงรุก (Proactive) และเชิงรับ (Reactive)

การทำงานของเกตเวย์ภายใต้โพรโทคอลแบบเชิงรุกนั้น [10] เกตเวย์จะแพร่กระจายแพ็คเกต Gateway Advertisement (GWADV) ออกมาเป็นระยะ ๆ เมื่อใดก็ตามที่โหนดได้รับแพ็คเกต GWADV โหนดจะทำการบันทึกข้อมูลของเส้นทางเอาไว้ในตารางเส้นทางในส่วนของ Default Route ซึ่งรูปแบบของการบันทึกเส้นทางนั้นจะขึ้นอยู่กับโพรโทคอลค้นหาเส้นทางในเครือข่ายเฉพาะกิจเคลื่อนที่ เมื่อโหนดต้นทางต้องการจะส่งข้อมูลไปหาปลายทาง โหนดต้นทางจะทำการค้นหาเส้นทางโดยคาดว่าโหนดปลายทางอยู่ในเครือข่ายเฉพาะกิจเคลื่อนที่เสียก่อน ในกรณีที่โหนดต้นทางไม่ได้รับแพ็คเกตตอบกลับจากโหนดใด ๆ เลย โหนดต้นทางจะคาดว่าโหนดปลายทางนั้นไม่ได้อยู่ในเครือข่ายเฉพาะกิจเคลื่อนที่ และจะเลือกใช้เส้นทางที่เก็บเอาไว้ใน Default Route ในการส่งข้อความไปยังเกตเวย์ และส่งต่อออกไปยังเครือข่ายภายนอก

ในทางกลับกันเกตเวย์ภายใต้โพรโทคอลแบบเชิงรับ [11] นั้นจะไม่ทำการแพร่กระจายแพ็คเกต GWADV ออกมา แต่จะรอจนกว่าจะได้รับแพ็คเกต Gateway Request (GWREQ) ที่ถูกสร้างขึ้นจากโหนดต้นทางที่มีความต้องการใช้งานเกตเวย์ หรือแม้แต่ได้รับแพ็คเกต RREQ จากกระบวนการค้นหาเส้นทางแบบปกติ เมื่อเกตเวย์ได้รับแพ็คเกต GWREQ หรือ RREQ เกตเวย์จะตอบกลับด้วยแพ็คเกต Gateway Reply (GWREP) ด้วยเส้นทางย้อนกลับที่ได้รับจากแพ็คเกต โหนดเพื่อนบ้านทุก ๆ โหนดที่ทำหน้าที่ส่งต่อแพ็คเกต GWREP จะทำการบันทึกเส้นทางไปยังเกตเวย์ลงไปในตารางเส้นทางในส่วนของ Default Route ทันที การสำรองข้อมูลเส้นทางไปยังเกตเวย์ในลักษณะนี้จะช่วยให้โหนดเพื่อนบ้านไม่จำเป็นต้องร้องขอเส้นทางไปยังเกตเวย์อีกถ้าโหนดเหล่านี้ต้องการจะส่งข้อความไปยังเกตเวย์ จนกระทั่งแพ็คเกต GWREP ถูกส่งกลับมายังโหนดต้นทาง โหนดต้นทางก็จะทำการบันทึกเส้นทางลงไปในลักษณะเดียวกัน

2.6.2 โพรโทคอลค้นหาเกตเวย์แบบผสม (Hybrid)

โพรโทคอลค้นหาเกตเวย์แบบผสม [12] เป็นการนำเอาข้อดีของโพรโทคอลค้นหาเกตเวย์เชิงรุกและเชิงรับมารวมเข้าด้วยกัน โดยเริ่มต้นเกตเวย์จะทำการแพร่กระจายแพ็คเกต GWADV ออกมาเป็นระยะ ๆ แต่แทนที่จะกระจายแพ็คเกตดังกล่าวออกไปทั้งเครือข่าย เกตเวย์จะจำกัดขอบเขตในการกระจายแพ็คเกตโดยการกำหนดค่า TTL เพื่อให้แพ็คเกตถูกกระจายออกไปในระยะฮ็อปที่กำหนด โหนดทุกตัวที่ได้รับแพ็คเกต GWADV จะสำรองเส้นทางเอาไว้ในส่วนของ Default Route เช่นเดียวกับโพรโทคอลค้นหาเกตเวย์เชิงรุก เมื่อโหนดต้องการส่งข้อความไปหาเกตเวย์ โหนดนั้น ๆ จะทำการตรวจสอบว่ามีข้อมูลเส้นทางไปยังเกตเวย์อยู่ใน Default Route หรือไม่ ถ้ามีโหนดนั้นก็จะสามารถส่งข้อความได้ทันที แต่ถ้าไม่มีข้อมูลดังกล่าว (โหนดอาจจะไม่ได้อยู่ในระยะการแพร่กระจายแพ็คเกต GWADV) โหนด

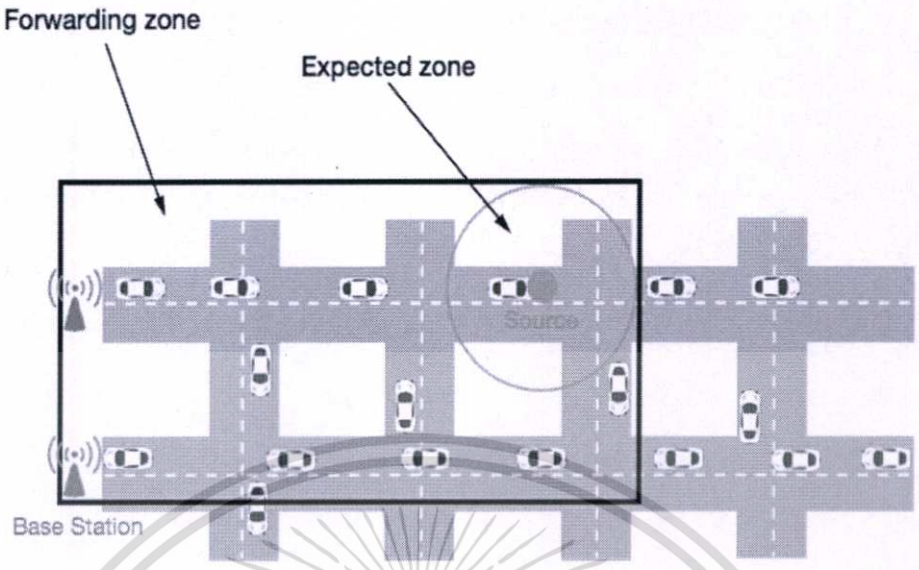
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

จะทำการแพร่กระจายแพ็คเกต GWREQ ออกไปเพื่อร้องขอเส้นทาง เมื่อเกตเวย์ หรือโหนดเพื่อนบ้านที่การสำรองข้อมูลเส้นทางไปยังเกตเวย์ได้รับแพ็คเกต GWREQ โหนดนั้น ๆ ก็จะตอบกลับด้วยแพ็คเกต GWREP เพื่อแจ้งเส้นทางที่สามารถใช้ส่งไปหาเกตเวย์ได้ โดยประสิทธิภาพของโพรโทคอลค้นหาเกตเวย์แบบผสมขึ้นอยู่กับการกำหนดค่า TTL ให้เหมาะสมกับขนาดของเครือข่าย

2.6.3 Location-Aided Gateway Advertisement and Discovery Protocol for VANets (LAGAD)

Location-Aided Gateway Advertisement and Discovery Protocol for VANets : LAGAD [13] เป็นโพรโทคอลค้นหาเกตเวย์ที่ถูกออกแบบมาให้ใช้งานกับเครือข่ายเฉพาะกิจเคลื่อนที่บนยานพาหนะโดยจะแบ่งการทำงานออกเป็น 2 ส่วนหลัก ๆ คือ กระบวนการค้นหาเส้นทาง และกระบวนการค้นหาเกตเวย์ ถึงแม้กระบวนการค้นหาเส้นทางนั้นถูกออกแบบให้เหมาะสมกับรูปแบบเครือข่ายเฉพาะกิจเคลื่อนที่บนยานพาหนะ แต่กระบวนการค้นหาเกตเวย์นั้นได้นำแนวคิดของโพรโทคอล LAR [14] มาใช้งาน เมื่อโหนดต้นทางต้องการจะส่งข้อมูลไปยังเกตเวย์ โหนดต้นทางจะเริ่มกระบวนการค้นหาเกตเวย์โดยการส่งแพ็คเกต GWREQ ออกไปพร้อมกับเพิ่มข้อมูลพิกัดจีพีเอส และความเร็วในขณะนั้นลงไปบนเฮดเดอร์ของแพ็คเกต เมื่อเกตเวย์ได้รับแพ็คเกต GWREQ เกตเวย์จะทำการบันทึกข้อมูลความเร็วและพิกัดจีพีเอสของโหนดต้นทางลงไปบนตารางผู้ใช้งานเกตเวย์ และตอบกลับด้วยแพ็คเกต GWREP ที่ระบุเส้นทางที่สามารถใช้ส่งกลับมาหาเกตเวย์ได้

เนื่องจากโหนดในเครือข่ายเฉพาะกิจเคลื่อนที่บนยานพาหนะนั้นสามารถเคลื่อนที่ เปลี่ยนความเร็ว และตำแหน่งได้ตลอดเวลา จนอาจทำให้เส้นทางจากโหนดต้นทางไปยังเกตเวย์นั้นไม่สามารถใช้งานได้ และในกรณีนั้นโหนดต้นทางจะเริ่มกระบวนการการค้นหาเกตเวย์ใหม่ซึ่งจะเป็นการเพิ่มปริมาณโอเวอร์เฮดเข้าไปในระบบเครือข่ายโดยไม่จำเป็น เพื่อแก้ไขปัญหาดังกล่าวเกตเวย์จะทำการสร้าง Expected zone และ Forwarding zone ตามภาพที่ 2.13 ซึ่งจะสังเกตได้ว่ามีลักษณะเช่นเดียวกับโพรโทคอล LAR แต่จุดประสงค์ในการใช้งานนั้นต่างออกไป เกตเวย์ถูกออกแบบให้ประกาศแพ็คเกต GWADV ที่ระบุถึงเส้นทางที่สามารถใช้ตอบกลับมาหามันได้ออกมาเป็นระยะ ๆ ซึ่งอนุญาตให้เฉพาะโหนดระหว่างทางที่อยู่ใน Expected zone และ Forwarding zone เท่านั้นที่จะสามารถกระจายแพ็คเกต GWADV ต่อไปได้ เมื่อโหนดต้นทางได้รับแพ็คเกต GWADV โหนดต้นทางจะทำการปรับปรุงข้อมูลเส้นทางที่จะใช้ส่งไปหาเกตเวย์ วิธีการนี้จะทำให้โหนดต้นทางได้รับเส้นทางใหม่ตลอดเวลาซึ่งสามารถช่วยลดความเสี่ยงที่เส้นทางเดิมจะไม่สามารถใช้งานได้ ส่งผลให้โอกาสที่โหนดต้นทางจะต้องเริ่มกระบวนการค้นหาเกตเวย์ใหม่นั้นลดลง



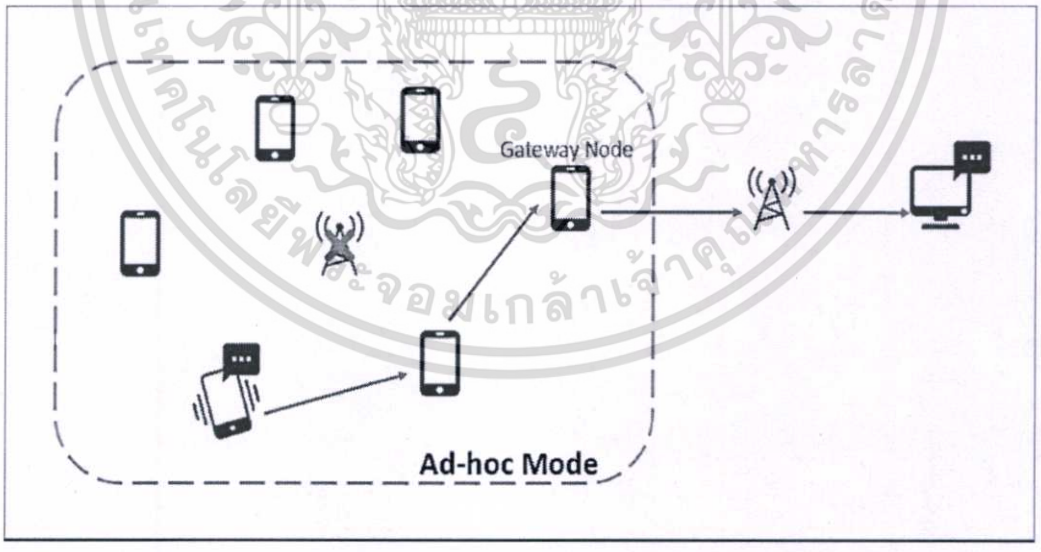
ภาพที่ 2.13 รูปแบบการกระจายแพ็คเกต GWADV ของโพรโทคอล LAGAD

บทที่ 3
การออกแบบและพัฒนาแอพพลิเคชัน

ในบทนี้จะกล่าวถึงแนวคิด และการดำเนินงานโครงการวิจัยในส่วนของการพัฒนาแอพพลิเคชันเครือข่ายเฉพาะกิจเพื่อการสื่อสารแบบไม่พึ่งพาโครงสร้างพื้นฐาน ซึ่งจะช่วยให้สามารถนำสมาร์ทโฟนมารวมกลุ่มเพื่อสร้างกลุ่มของเครือข่ายเฉพาะกิจเคลื่อนที่

3.1 การพัฒนาแอพพลิเคชันเครือข่ายเฉพาะกิจเพื่อการสื่อสารแบบไม่พึ่งพาโครงสร้างพื้นฐาน

ในสถานการณ์ที่โครงสร้างพื้นฐานของเครือข่ายไม่สามารถใช้งานได้ สมาร์ทโฟนที่ติดตั้งแอพพลิเคชันจะทำงานเพื่อสร้างกลุ่มของเครือข่ายเฉพาะกิจเคลื่อนที่ (MANET) โดยเครื่องที่จะส่งข้อความนั้นจะเริ่มกระบวนการค้นหาเส้นทางตามกลไกของโพรโทคอล AODV โดยการกระจายแพ็คเกต RREQ ออกไปในเครือข่ายที่เชื่อมต่อกันแบบไร้สายผ่านเทคโนโลยี Wi-Fi Direct สมาร์ทโฟนที่ได้รับแพ็คเกต RREQ จะช่วยกระจายแพ็คเกตต่อไปจนกระทั่งไปถึงปลายทาง เมื่อเครื่องปลายทางได้รับแพ็คเกต RREQ ก็จะส่งแพ็คเกต RREP กลับมายังต้นทาง สมาร์ทโฟนที่ได้รับแพ็คเกต RREP จะสร้างตารางเส้นทางไปยังปลายทาง เมื่อเครื่องต้นทางได้รับแพ็คเกต RREP ก็จะส่งแพ็คเกตข้อมูลไปยังเครื่องถัดไปตามข้อมูลในตารางเส้นทางจนกระทั่งแพ็คเกตข้อความนั้นไปถึงปลายทาง โดยผู้วิจัยจะพัฒนาแอพพลิเคชันให้รองรับการพัฒนาต่อเพิ่มเติมในส่วนของการค้นหา และใช้งานเกตเวย์ซึ่งเป็นสมาร์ทโฟนที่สามารถเชื่อมต่อกับโครงสร้างพื้นฐานได้



ภาพที่ 3.1 แผนผังแสดงทอพอโลยีของระบบ

3.1.1 ความต้องการที่เป็นหน้าที่หลักของระบบ (Functional Requirement)

- ระบบสามารถที่จะระบุผู้รับได้
- ระบบสามารถที่จะระบุผู้ส่งได้

3.1.2 ความต้องการที่ไม่ใช่หน้าที่หลักของระบบ (Non-Functional Requirement)

- ระบบมีส่วนติดต่อกับผู้ใช้ (User Interface) ที่ง่ายต่อการใช้งาน

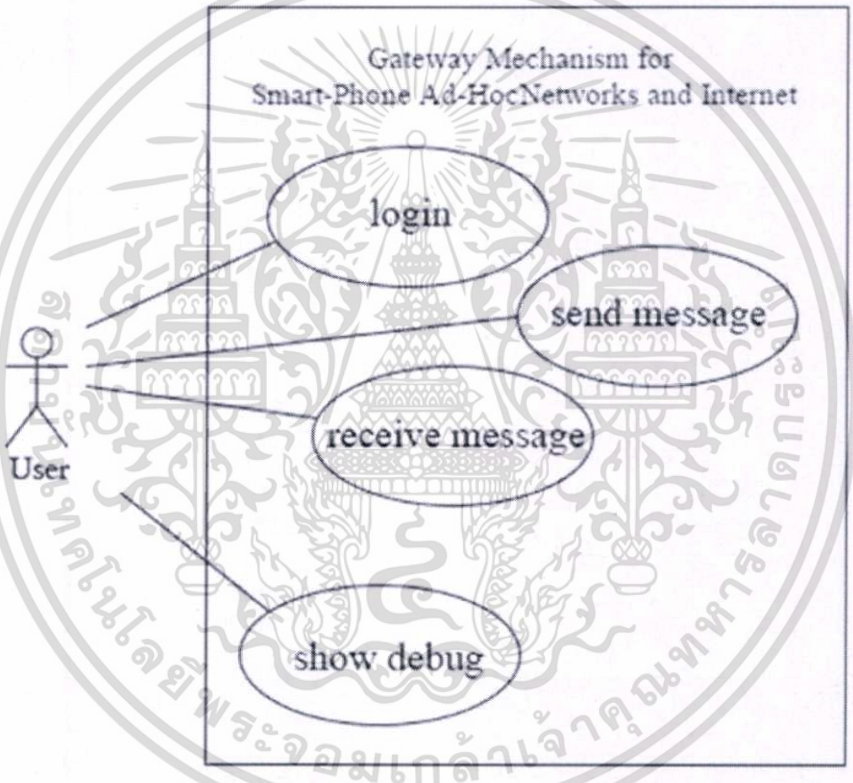
3.1.3 แผนภาพยูสเคส (Use Case Diagram)

ผู้ใช้ในระบบ (Actor) ประกอบไปด้วย

- ผู้ใช้งานทั่วไป (User)

Use Case ของระบบ ประกอบด้วย

- เข้าสู่ระบบ (Login)
- ส่งข้อความ (Send Message)
- รับข้อความ (Receive Message)
- แสดงข้อความที่อยู่ในระบบ (Show Debug)



ภาพที่ 3.2 แผนภาพยูสเคส (Use Case Diagram)

ตารางที่ 3.1 คำอธิบายยูสเคส Login

Use Case Name:	Login	
Actor:	User	
Description:	เข้าสู่ระบบ	
Preconditions:	กรอกชื่อ Username ที่ผู้ใช้ต้องการ	
Post conditions:	แสดงหน้าส่งข้อความ พร้อมแสดง Username ของผู้ใช้	
Flow of events:	Actor	System
	1. เปิดแอพพลิเคชัน 2. กรอก Username ที่ต้องการ 3. กดปุ่ม Login	4. บันทึก Username ของผู้ใช้ 5. แสดงหน้าส่งข้อความ 6. แสดง Username ของผู้ใช้
Exceptions Conditions:	-	

ตารางที่ 3.2 คำอธิบายยูสเคส Send Message

Use Case Name:	Send Message	
Actor:	User	
Description:	ส่งข้อความ	
Preconditions:	กรอกชื่อผู้รับ	
Post conditions:	ส่งข้อความให้ผู้รับและแสดงข้อความที่ส่ง	
Flow of events:	Actor	System
	1. พิมพ์ข้อความที่ต้องการส่ง 2. กรอกชื่อผู้รับที่ต้องการส่งข้อความให้ 3. กดปุ่ม Send	4. ส่งข้อความไปให้ผู้รับ 5. แสดงข้อความที่ส่งออกไป
Exceptions Conditions:	ไม่กรอกชื่อผู้รับ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

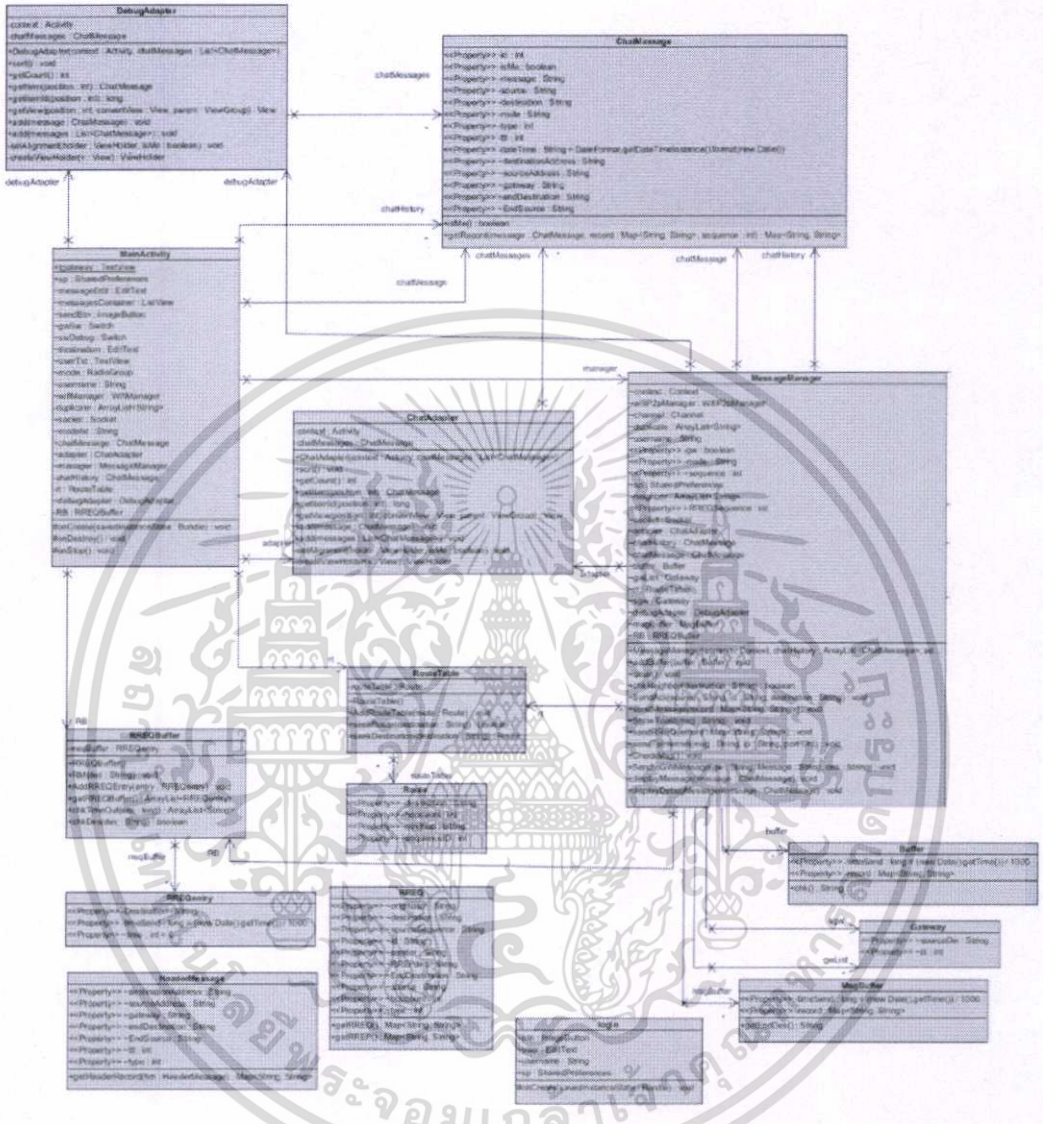
ตารางที่ 3.3 คำอธิบายยูสเคส Receive Message

Use Case Name:	Receive Message	
Actor:	User	
Description:	รับข้อความส่งข้อความ	
Preconditions:	-	
Post conditions:	แสดงข้อความที่ได้รับ	
Flow of events:	Actor	System
		1. แสดงข้อความที่ได้รับ พร้อมชื่อผู้ส่ง
Exceptions Conditions:	-	

ตารางที่ 3.4 คำอธิบายยูสเคส Show Debug

Use Case Name:	Show Debug	
Actor:	User	
Description:	แสดงข้อความทั้งหมดของระบบ	
Preconditions:	-	
Post conditions:	แสดงข้อความที่ถูกส่งภายในระบบ	
Flow of events:	Actor	System
	1. กดเปิดปุ่ม Debug	2. แสดงข้อความที่มีอยู่ในระบบ
Exceptions Conditions:	-	

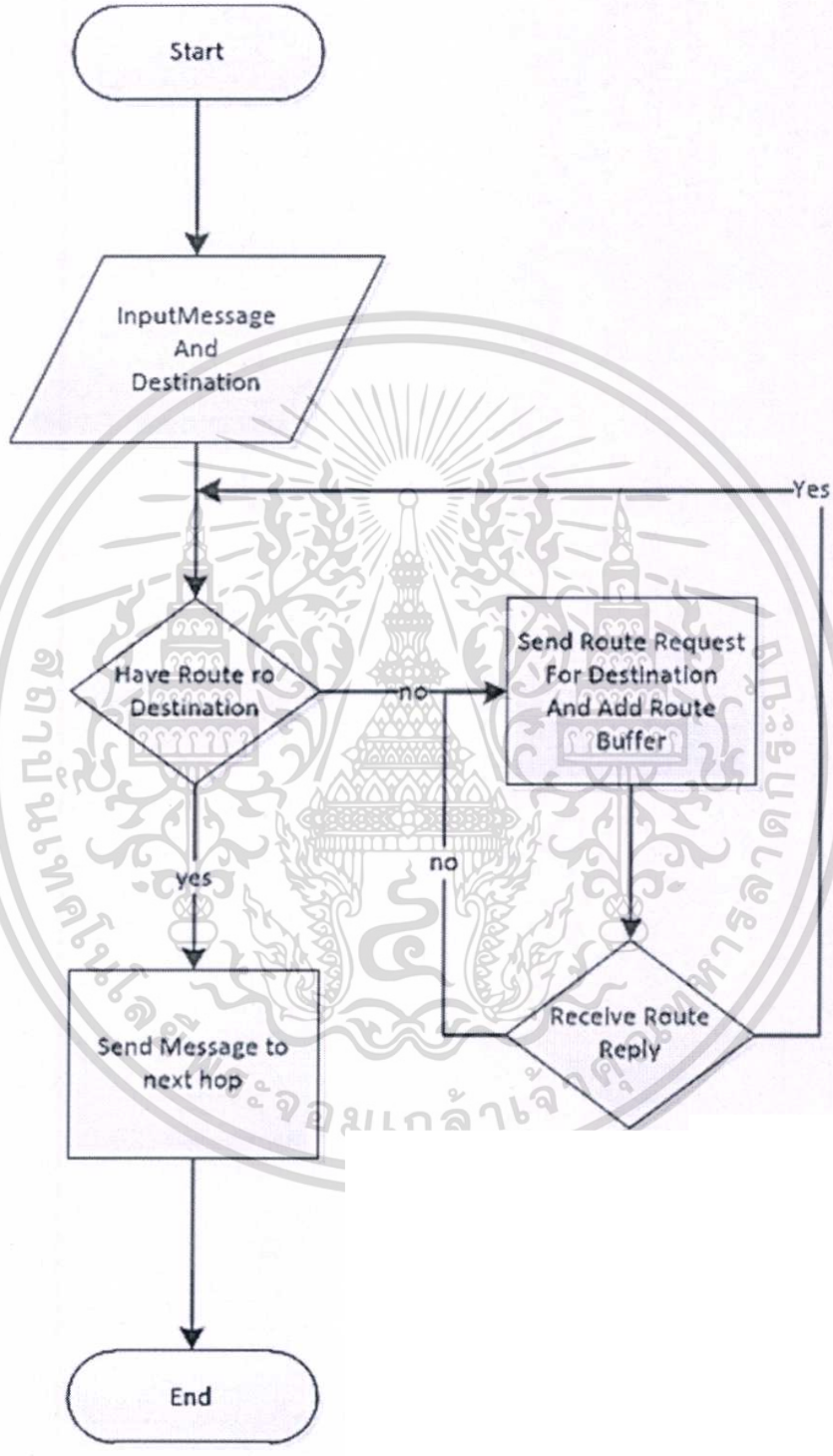
3.1.4 แผนภาพคลาส (Class Diagram)



ภาพที่ 3.3 แผนภาพคลาส (Class Diagram)

3.1.5 แผนภาพกิจกรรม (Activity Diagram)

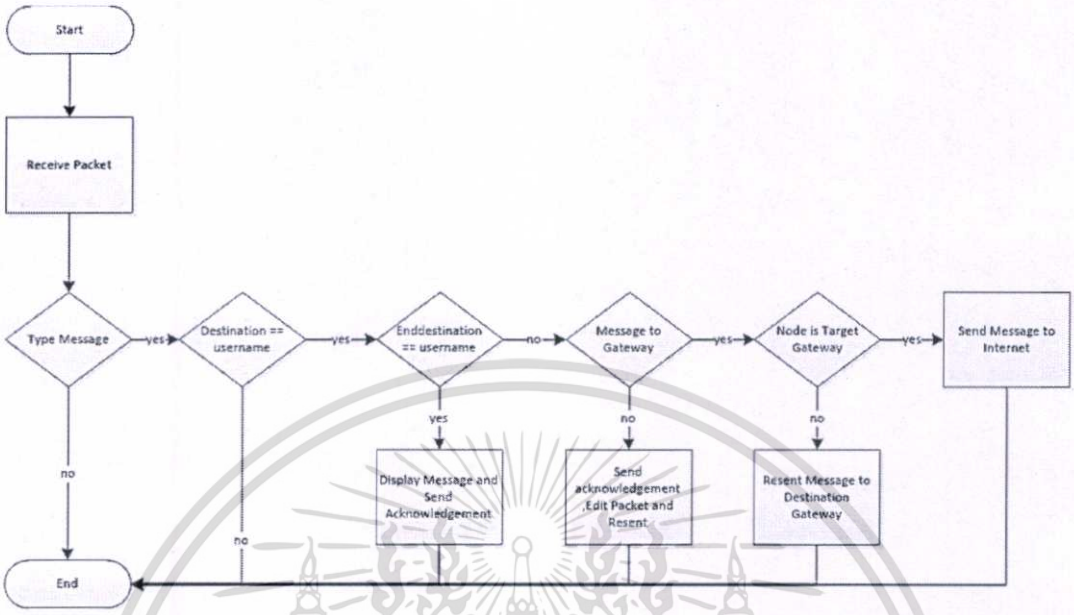
1. การส่งข้อความ



ภาพที่ 3.4 แผนภาพกิจกรรมการส่งข้อความ

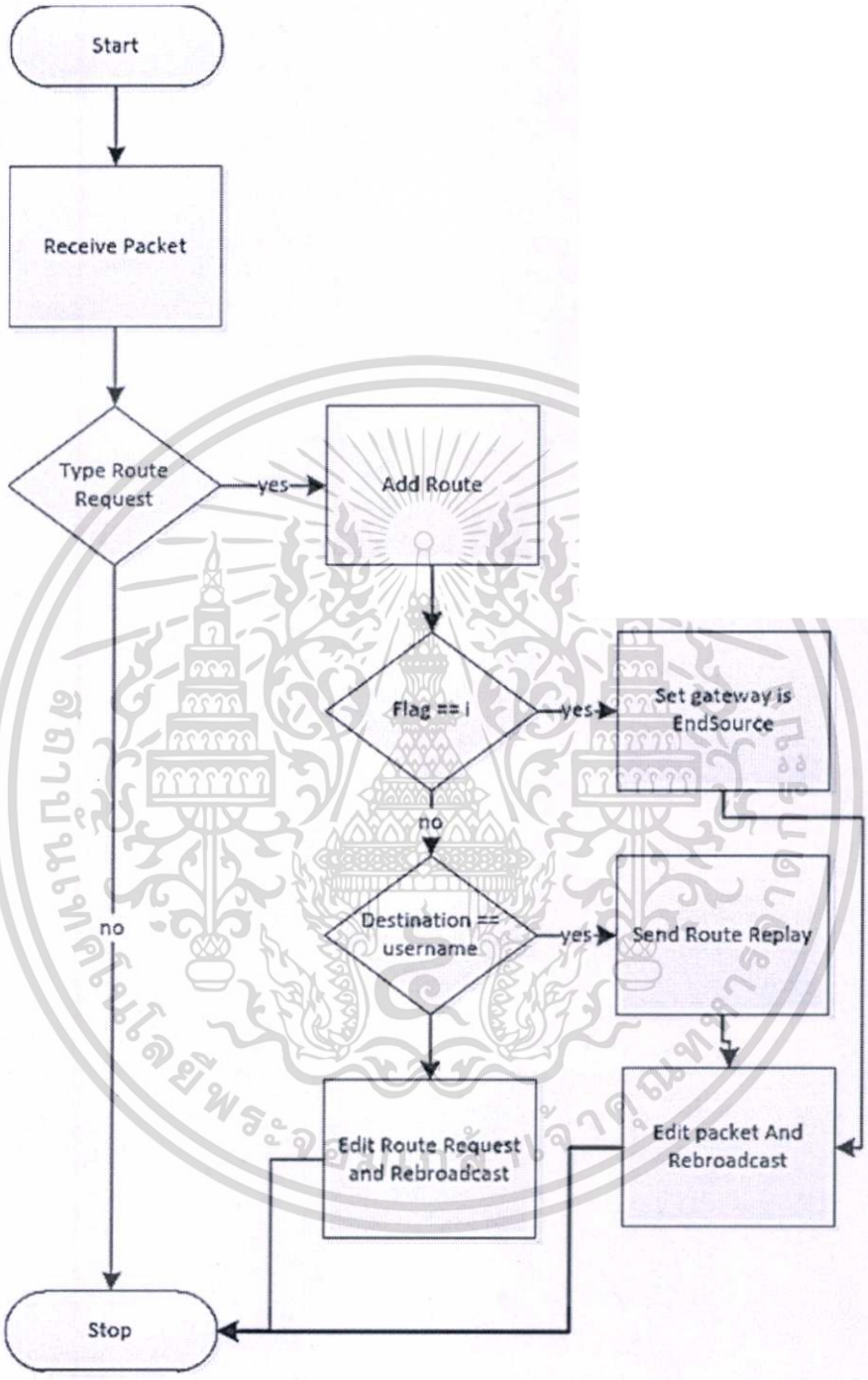
การศึกษา

2. การรับข้อความ



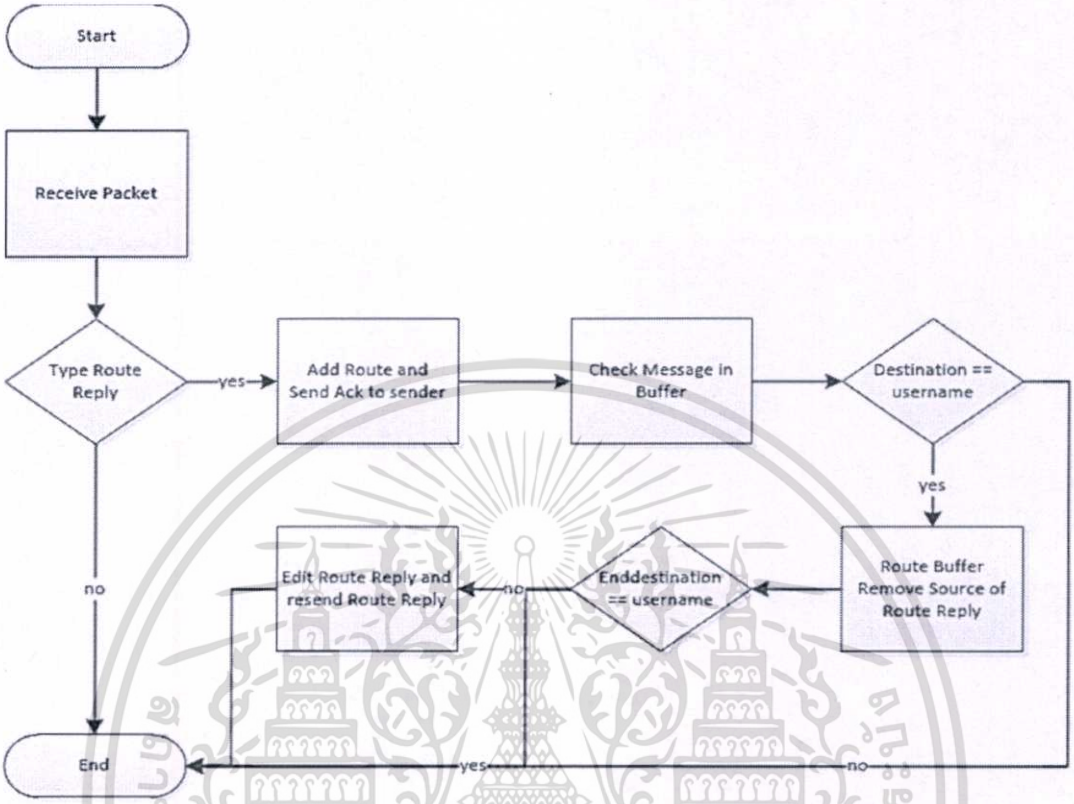
ภาพที่ 3.5 แผนภาพกิจกรรมการรับข้อความ

3. การร้องขอเส้นทาง



ภาพที่ 3.6 แผนภาพกิจกรรมการร้องขอเส้นทาง

4. การตอบกลับการร้องขอเส้นทาง



ภาพที่ 3.7 แผนภาพกิจกรรมการตอบกลับการร้องขอเส้นทาง

3.1.6 การออกแบบชุดข้อมูลที่ใช้ในแอพพลิเคชัน

ในแอพพลิเคชันได้มีการใช้ชุดข้อความ 4 ชนิด คือ

1. Route Request Message คือ ชุดข้อความที่ถูกใช้ในการร้องขอเส้นทางภายในเครือข่ายเฉพาะกิจ โดยภายในชุดข้อความมีรูปแบบข้อความดังต่อไปนี้

Hop-to-hop Source Address	Hop-to-hop Destination Address	Originator Address	Source Sequence	RREQ. Message ID	Hop-count	Type

ภาพที่ 3.8 รูปแบบข้อความของ Route Request Message

- Hop-to-hop Source Address คือ ชื่อผู้ส่งในการส่งแบบโหนดต่อโหนด
- Hop-to-hop Destination Address คือ ชื่อผู้รับในการส่งแบบโหนดต่อโหนด
- Originator Address คือ ชื่อโหนดที่ทำการเริ่มส่งข้อความ
- Source Sequence คือ หมายเลขลำดับของต้นทาง เพื่อใช้ตรวจสอบ Routing Update
- RREQ. Message ID คือ หมายเลขกำกับข้อความ Route Request Message
- Hop-count คือ จำนวนฮ็อป
- Type คือ ชนิดของข้อความ ใน Route Request Message มี Type = 1

2. Route Reply Message คือ ชุดข้อความที่ถูกใช้ในการตอบกลับการร้องขอเส้นทางภายในเครือข่ายเฉพาะกิจ โดยภายในชุดข้อความมีรูปแบบข้อความดังต่อไปนี้

Hop-to-hop Source Address	Hop-to-hop Destination Address	Originator Address	Destination Address	Source Sequence	RREP. Message ID	Hop-count	Type

ภาพที่ 3.9 รูปแบบข้อความของ Route Reply Message

- Hop-to-hop Source Address คือ ชื่อผู้ส่งในการส่งแบบโหนดต่อโหนด
- Hop-to-hop Destination Address คือ ชื่อผู้รับในการส่งแบบโหนดต่อโหนด
- Originator Address คือ ชื่อโหนดที่ทำการเริ่มส่งข้อความ
- Destination Address คือ ชื่อโหนดปลายทางที่ต้องการส่งข้อความไปหา

- Source Sequence คือ หมายเลขลำดับของต้นทาง เพื่อใช้ตรวจสอบ Routing Update
- RREP. Message ID คือ หมายเลขกำกับข้อความ Route Reply
- Hop-count คือ จำนวนฮ็อป
- Type คือ ชนิดของข้อความ ใน Route Reply Message มี Type = 2

3. Acknowledgement Message คือ ชุดข้อความที่ถูกส่งเพื่อยืนยันการได้รับข้อความต่าง ๆ โดยภายในชุดข้อความมีรูปแบบข้อความดังต่อไปนี้

Hop-to-hop Source Address	Hop-to-hop Destination Address	ACK. Message ID	Type

ภาพที่ 3.10 รูปแบบข้อความของ Acknowledgement Message

- Hop-to-hop Source Address คือ ชื่อผู้ส่งในการส่งแบบโหนดต่อโหนด
- Hop-to-hop Destination Address คือ ชื่อผู้รับในการส่งแบบโหนดต่อโหนด
- ACK. Message ID คือ หมายเลขกำกับข้อความ Acknowledgement
- Type คือ ชนิดของข้อความ ใน Acknowledgement Message มี Type = 4

4. Data Message คือ ชุดข้อความที่บรรจุข้อความที่ต้องการส่ง โดยภายในชุดข้อความมีรูปแบบข้อความดังต่อไปนี้

Hop-to-hop Source Address	Hop-to-hop Destination Address	Gateway Address	End-to-end Source Address	End-to-end Destination Address	Data Message ID	Type	TTL
Data							

ภาพที่ 3.11 รูปแบบข้อความของ Data Message

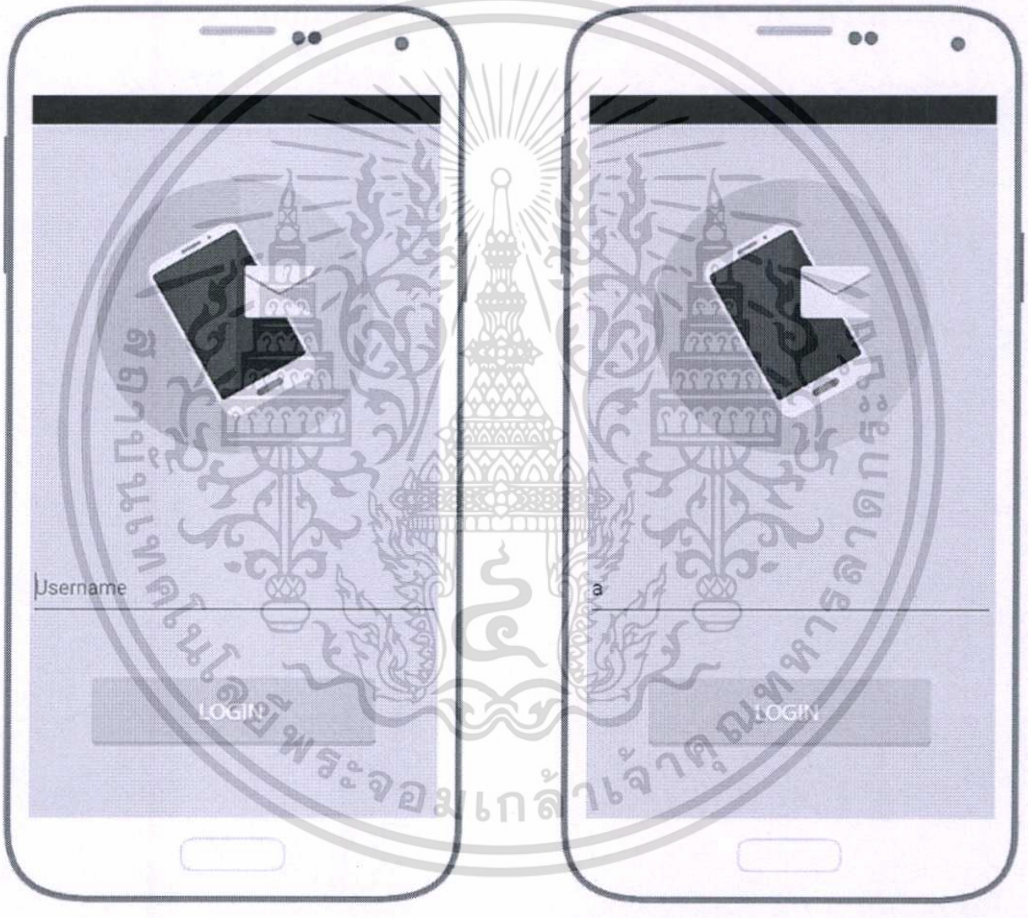
- Hop-to-hop Destination Address คือ ชื่อผู้รับในการส่งแบบโหนดต่อโหนด
- Hop-to-hop Source Address คือ ชื่อผู้ส่งในการส่งแบบโหนดต่อโหนด
- Gateway Address คือ ชื่อโหนดที่เป็นเกตเวย์
- End-to-End Source Address คือ ชื่อผู้ส่งต้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

- End-to-End Destination Address คือ ชื่อผู้รับปลายทาง
- Data Message ID คือ หมายเลขกำกับข้อความ Data
- Time to Live (TTL) คือ ค่าที่ใช้กำหนดให้ข้อความสามารถส่งผ่านโหนดได้สูงสุดเท่าไหร่ เพื่อป้องกันการเกิดลูปและเพื่อลด Traffic ในเครือข่าย
- Type คือ ชนิดของข้อความ ใน Data Message มี Type = 10
- Data คือ ส่วนที่บรรจุข้อความที่ต้องการส่ง

3.1.7 การออกแบบส่วนติดต่อผู้ใช้ของแอพพลิเคชัน

1. หน้าเข้าสู่ระบบ

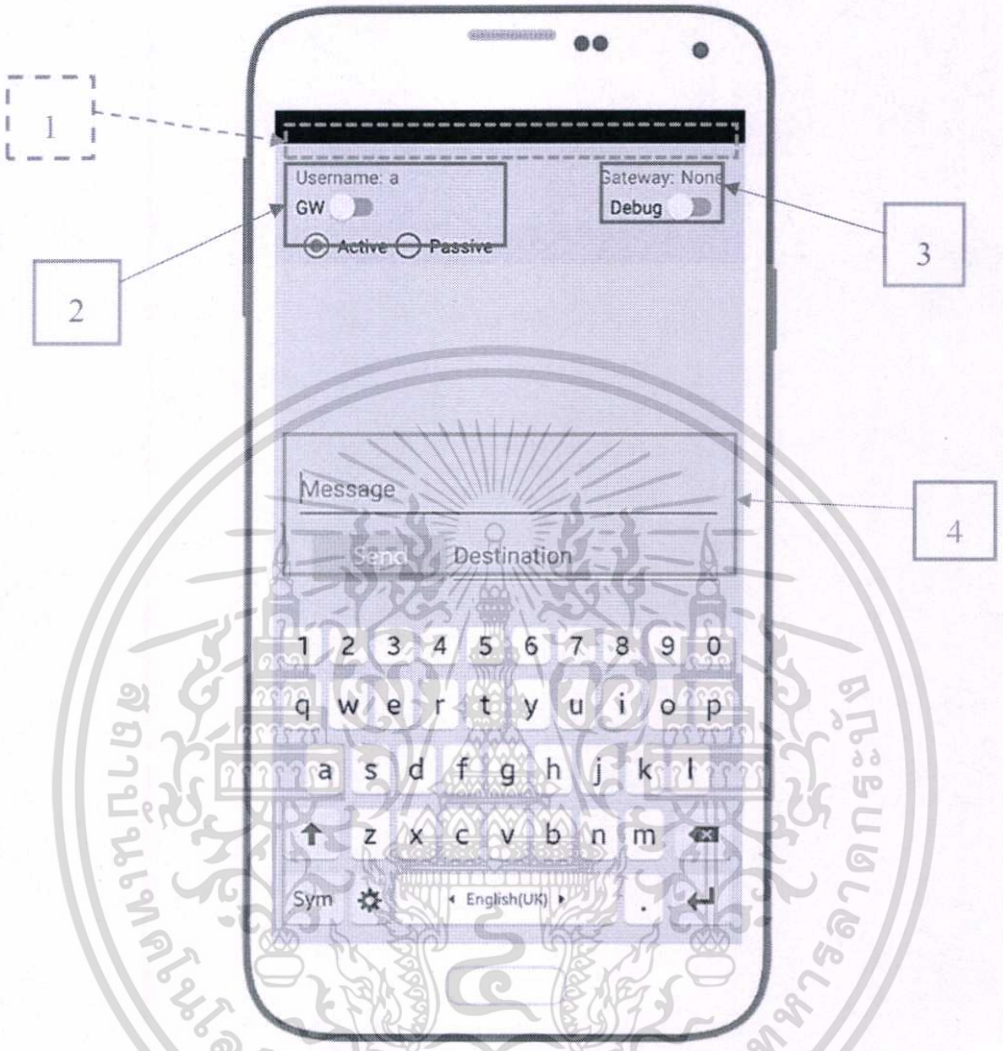


ภาพที่ 3.12 แสดงหน้าเข้าสู่ระบบของแอพพลิเคชัน

ภาพที่ 3.12 ในหน้าแรกของแอพพลิเคชันจะเป็นหน้าเข้าสู่ระบบ ที่ให้ผู้ใช้ทำการกรอกชื่อ (Username) ที่ต้องการ จากนั้นให้ทำการกดปุ่ม LOGIN เพื่อเข้าสู่ระบบของแอพพลิเคชัน โดยแอพพลิเคชันจะใช้ Username ที่ผู้ใช้ตั้งเพื่อเป็นการระบุตัวตนของแต่ละเครื่อง เมื่อผู้ใช้กรอก หลังจากนั้นแอพพลิเคชันจะแสดงหน้ารับและส่งข้อความให้ และจะแสดงชื่อ Username ไว้ด้านบนของจอภาพดังภาพที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

2. หน้าแสดงการส่งและการรับข้อความ



ภาพที่ 3.13 แสดงหน้ารับ-ส่งข้อความของแอพพลิเคชัน

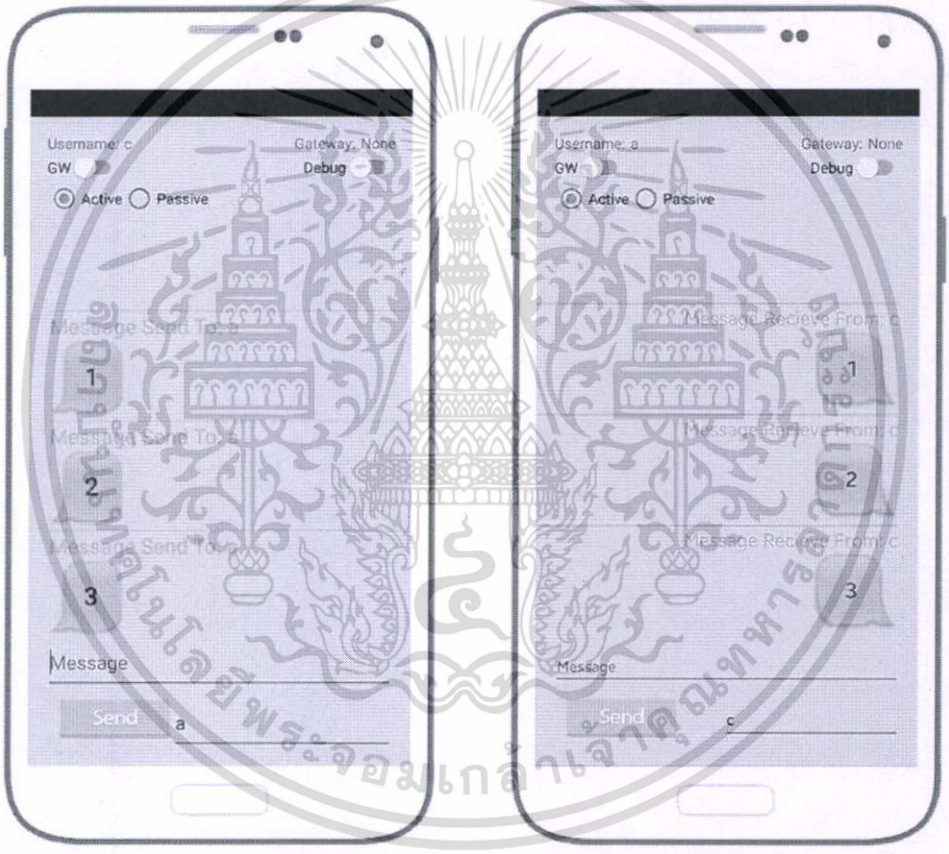
ภาพที่ 3.13 หน้าแสดงการรับและส่งข้อความ หลังจากที่ผู้ใช้เข้าสู่ระบบแล้วจะเข้ามาสู่หน้านี้ ซึ่งหน้านี้จะแสดงการส่งและรับข้อความทั้งหมด ภายในหน้านี้ประกอบด้วย

- ส่วนที่ 1 คือ ส่วนที่แสดงชื่อ Username ของผู้ใช้ และแสดงชื่อเครื่องที่เป็นเกตเวย์
- ส่วนที่ 2 คือ ปุ่มที่แสดงสถานะความเป็นเกตเวย์ของเครื่อง ถ้าหากเครื่องนี้เป็นเกตเวย์ที่สามารถเชื่อมต่อไปยังอินเทอร์เน็ตได้ ผู้ใช้ต้องเปลี่ยนค่าให้เป็น ON โดยในกลไกเกตเวย์นั้นผู้ใช้สามารถเลือกโหมดในการค้นหาเกตเวย์ได้ 2 โหมด ดังนี้
- Active Mode คือ โหมดที่เครื่องเกตเวย์จะประกาศตนเองให้เครื่องในเครือข่าย
- Passive Mode คือโหมดที่เครื่องที่ต้องการค้นหาเกตเวย์จะต้องส่งข้อความออกไปค้นหาเกตเวย์ด้วยตนเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

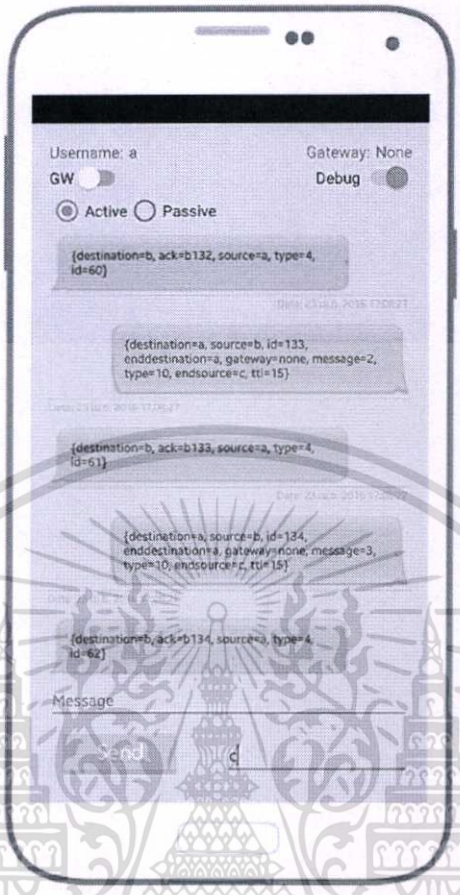
- ส่วนที่ 3 คือ ปุ่มแสดง debug ของแอพพลิเคชัน เพื่อแสดงข้อความทั้งหมดในระบบทั้งที่แสดงในส่วนรับส่งข้อความและข้อความที่ไม่แสดงในส่วนดังกล่าว
- ส่วนที่ 4 คือ ส่วนที่ผู้ใช้สามารถส่งข้อความ โดยต้องกำหนด Destination เป็นชื่อ (Username) ของเครื่องปลายทาง และ Message คือข้อความที่ผู้ใช้ต้องการส่งไปให้เครื่องปลายทาง จากนั้นกดปุ่ม SEND ข้อความจะถูกส่งออกไปให้เครื่องที่ระบุเป็น Destination

ในภาพที่ 3.14 เป็นการส่งข้อความระหว่างเครื่อง c กับเครื่อง a โดยแอพพลิเคชันจะแสดงให้เห็นว่าข้อความนั้นถูกส่งไปที่เครื่องใด และเมื่อเครื่อง a ได้รับข้อความจะแสดงข้อความบนหน้าจอ พร้อมกับแสดงว่าข้อความนั้นถูกส่งมาจากเครื่องใด



ภาพที่ 3.14 แสดงหน้ารับ-ส่งข้อความเมื่อมีข้อความถูกส่งออกและรับเข้า

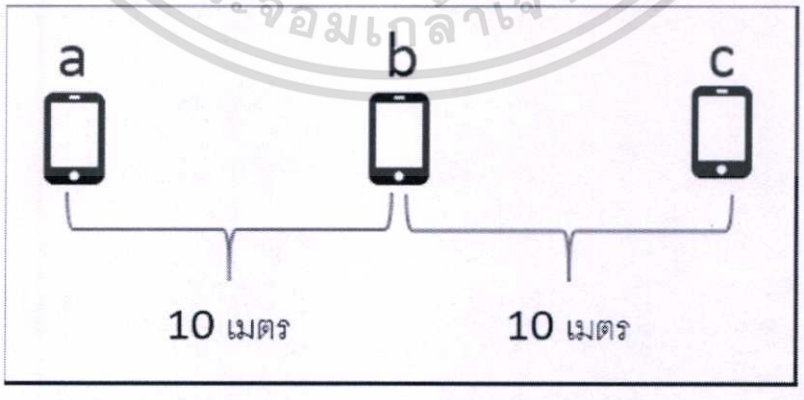
ในส่วนของหน้า Debug จะทำการแสดงข้อความทั้งหมดที่ถูกส่งและรับเข้ามาในเครื่องของผู้ใช้ พร้อมทั้งแสดงข้อมูลทั้งหมดในแพ็คเกตของข้อความเหล่านั้น เช่น ข้อความที่เป็นข้อความที่เป็น Acknowledgement หรือข้อความที่เป็น Data Message เป็นต้น ดังภาพที่ 3.15



ภาพที่ 3.15 แสดงหน้า Debug Mode

3.2 ผลการทดสอบการทำงานเบื้องต้นของแอพพลิเคชัน

ในการทดลองนี้จะใช้เครื่องทั้งหมด 3 เครื่องด้วยกัน ซึ่งได้ตั้งค่า Username ของทั้งสามเครื่องว่า a, b และ c ตามลำดับ โดยจะให้แต่ละเครื่องตั้งเรียงกันดังภาพที่ 3.16



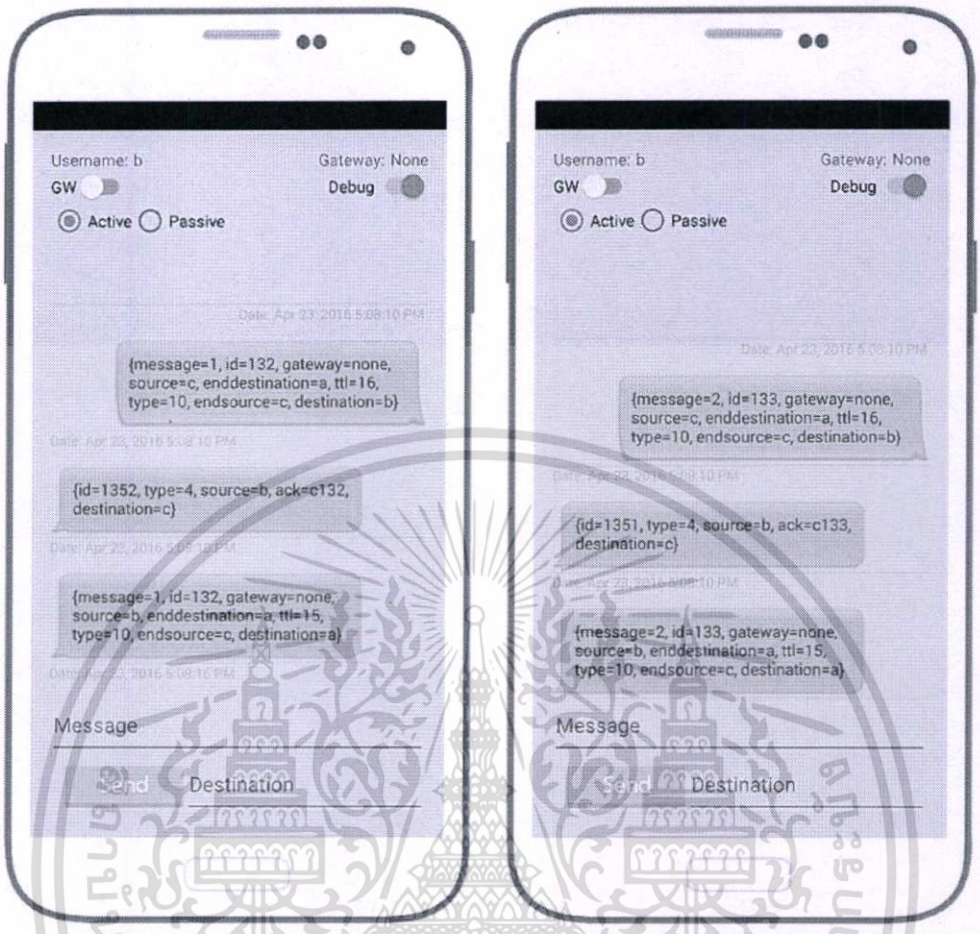
ภาพที่ 3.16 แสดงสถานการณ์ในการทดลองแอพพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

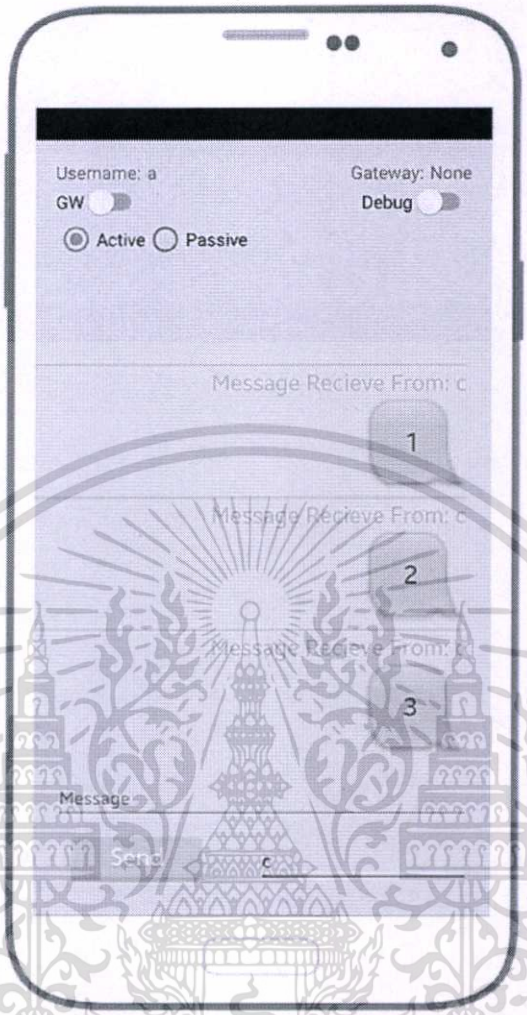
การทดลองนี้เป็นการส่งข้อความในเครือข่ายเฉพาะกิจแบบหลายฮ็อป โดยให้เครื่อง c ทำการส่งข้อความหาเครื่อง a และให้เครื่อง b เป็นเครื่องที่ช่วยส่งต่อข้อความระหว่างทั้งสองเครื่อง เครื่อง c ส่งข้อความ 1, 2, 3 ไปให้เครื่อง a เมื่อเปิด Debug Mode ของแอพพลิเคชันจะเห็นแพ็คเกตที่เครื่อง c ทำการส่งไปหาเครื่อง a ดังภาพที่ 3.17

ภาพที่ 3.17 แสดงการส่งข้อความจากเครื่อง c ใน Debug Mode

ในการส่งข้อความแบบหลายฮ็อปในเครือข่ายเฉพาะกิจ เครื่อง c จะไม่สามารถส่งหาเครื่อง a ได้โดยตรง แต่จะอาศัยเครื่อง b ในการส่งต่อข้อความ ซึ่งในภาพที่ 3.18 แสดงให้เห็นว่าเครื่อง b ได้รับข้อความจากเครื่อง c และจะตรวจสอบว่าใช่ข้อความที่ส่งถึงของตนเองหรือไม่ ถ้าไม่ใช่เครื่อง b ก็จะส่งต่อข้อความออกไป โดยก่อนส่งข้อความออก เครื่อง b จะส่ง Acknowledgement Message (ACK) กลับไปเครื่อง c ด้วยเพื่อยืนยันการได้รับข้อความ เครื่อง c ก็จะได้รับ ACK ดังในภาพที่ 3.18



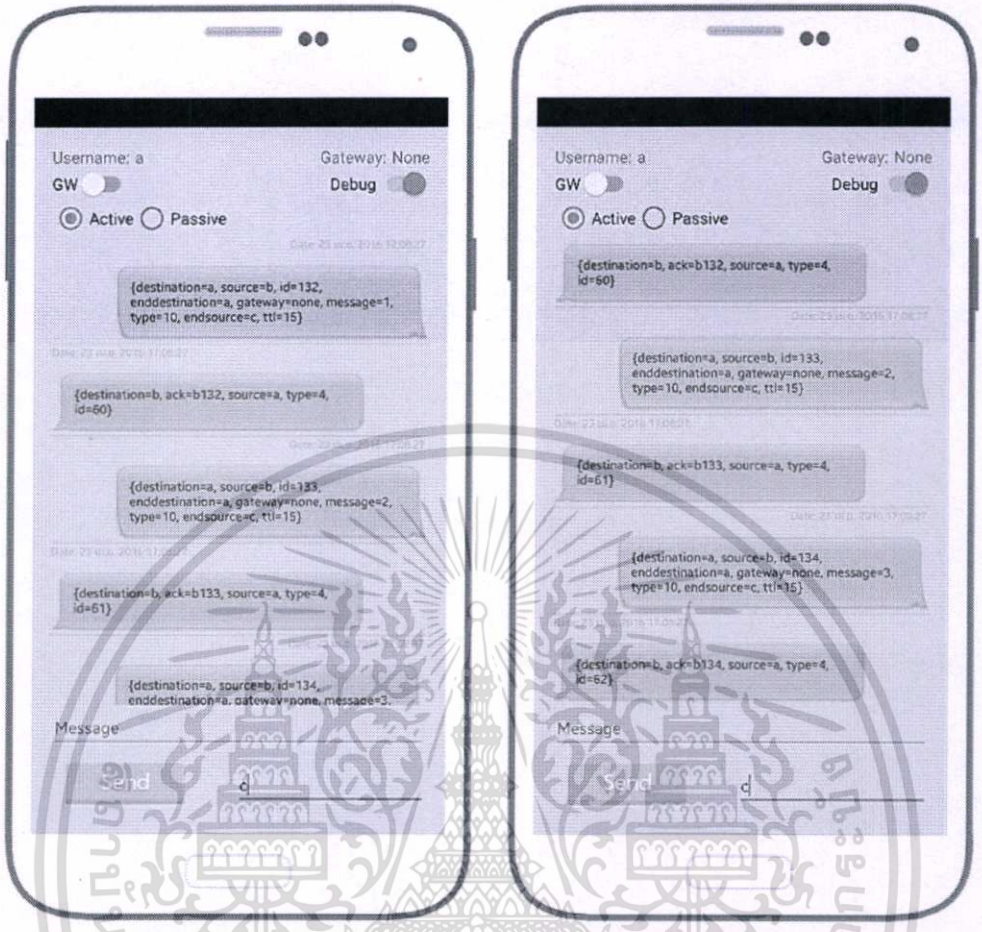
ภาพที่ 3.18 แสดงการรับและส่งต่อข้อความของเครื่อง b



ภาพที่ 3.19 แสดงหน้าจอเครื่อง a ที่ได้รับข้อความจากเครื่อง c

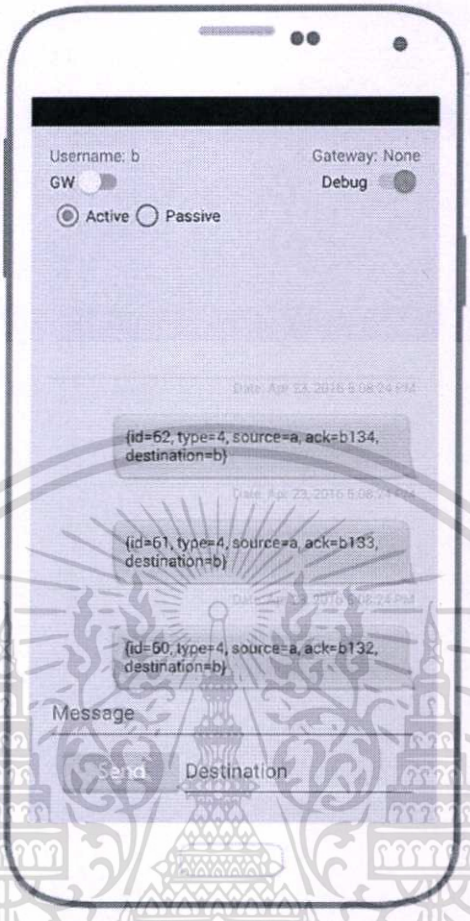
เมื่อเครื่อง b ทำการส่งต่อข้อความไปให้เครื่อง a แล้ว เครื่อง a เมื่อได้รับข้อความก็จะแสดงผลในหน้ารับส่งข้อความ (ภาพที่ 3.19) ถึงแม้ว่าเครื่อง a จะได้รับข้อความที่เครื่อง b ส่งต่อมา แต่แท้จริงแล้วข้อความถูกส่งมาจากต้นทางที่เป็นเครื่อง c ดังนั้นในหน้ารับส่งข้อความจึงแสดงว่าข้อความส่งมาจากเครื่อง c ที่เป็นต้นทางนั่นเอง

แต่เมื่อกดเปิด Debug mode จะเห็นว่า a รับข้อความมาจาก b จริง โดยดูจาก Source=b แต่ endSource=c ดังนั้นยืนยันได้ว่าข้อความถูกส่งมาจากเครื่อง c แต่เครื่อง b เป็นแค่ตัวกลางในการส่งต่อเท่านั้น



ภาพที่ 3.20 แสดงหน้า Debug Mode ของเครื่อง a เมื่อได้รับข้อความ

เมื่อ a ได้รับข้อความแล้วก็จะส่ง ACK กลับไปให้เครื่อง b โดยอัตโนมัติ ซึ่งเครื่อง b ก็จะได้รับ ACK ที่เครื่อง a ส่งกลับมาดังภาพที่ 3.21



ภาพที่ 3.21 แสดงหน้า Debug Mode ของเครื่อง b เมื่อได้รับ ACK

จากการทดลองแสดงให้เห็นว่าแอพพลิเคชันสามารถส่งข้อความแบบหลายฮ็อปได้จริง โดยสามารถแสดงรายละเอียดทั้งหมดเพื่อยืนยันการส่งแบบหลายฮ็อปได้

ยิ่งไปกว่านั้นยังมีการทดลองการส่งข้อความภายในอาคารคณะเทคโนโลยีสารสนเทศจากชั้นที่ 1 ถึงชั้นที่ 5 โดยใช้ผู้ร่วมทดสอบจำนวนทั้งหมด 5 คน หลังจากทำการติดตั้งแอพพลิเคชันลงบนเครื่องที่ใช้ทดสอบแล้ว ผู้ร่วมทดสอบจะถูกกระจายไปยังส่วนต่าง ๆ ภายในอาคาร โดยได้มีการทดสอบและคำนวนระยะการส่งสัญญาณของไวไฟภายใต้สภาพแวดล้อมแบบปิดเอาไว้ล่วงหน้า ซึ่งได้คำนึงถึงสิ่งกีดขวาง เช่น กำแพงในพื้นที่ต่าง ๆ ที่มีผลต่อการลดทอนสัญญาณ เพื่อให้ข้อความทั้งหมดต้องถูกส่งผ่านเครื่องทุกเครื่องในเครือข่ายเพื่อให้ไปถึงยังปลายทางได้

เพื่อความแม่นยำในการตรวจสอบประสิทธิภาพของแอพพลิเคชัน ได้มีการติดตั้งเครื่องคอมพิวเตอร์ที่มีการลงโปรแกรม Wireshark และติดตั้งอุปกรณ์ในการดักจับสัญญาณ AirPcap เสริมเข้าไปในการทดลอง เพื่อให้โปรแกรมสามารถตรวจสอบการแพร่กระจายของข้อความได้ในทุกช่องสัญญาณ จากการทดลองพบว่า แอพพลิเคชันสามารถส่งข้อความจากต้นทางที่ชั้น 1 ขึ้นไปถึงเครื่องปลายทางที่ชั้น 5 ได้สำเร็จ แต่อย่างไรก็ตามแอพพลิเคชั่นยังคงมีข้อจำกัดดังต่อไปนี้

- แอพพลิเคชันใช้เวลาในการสร้างเส้นทาง และส่งข้อความค่อนข้างนาน โดยใช้เวลาในการส่งข้อความหลังจากสร้างเส้นทางสำเร็จถึง 3 นาที สาเหตุนั้นเกิดมาจากปริมาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

แพ็คเกตจำนวนมากที่ถูกสร้าง และส่งออกมาตลอดเวลาจากการทำงานของโพรโทคอล Wi-Fi Direct ส่งผลให้เกิดความคับคั่งในเครือข่าย

- แอพพลิเคชันยังมีปัญหาในการส่งแพ็คเกต RERR ที่จะกระตุ้นการค้นหาเส้นทางใหม่ในกรณีที่เครื่องต้นทางไม่สามารถค้นหาเส้นทางไปยังเครื่องปลายทางได้ ซึ่งปัญหาในส่วนนี้ต้องการการทดสอบเพิ่มเติม

ผลงานการพัฒนาแอพพลิเคชันเครือข่ายเฉพาะกิจเพื่อการสื่อสารแบบไม่พึ่งพาโครงสร้างพื้นฐานได้รับการตอบรับเพื่อเผยแพร่ในวารสารวิชาการคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ดังแสดงในภาคผนวก ก. รายละเอียด และผลการทดสอบการทำงานของแอพพลิเคชันแสดงในเอกสารรายงานการทดสอบระบบแสดงในภาคผนวก ข. และวีดิทัศน์ [15]

นอกจากนี้ ผู้วิจัยได้ศึกษาเพิ่มเติมในส่วนของการพัฒนาแอพพลิเคชันให้สามารถเชื่อมต่อไปยังเครือข่ายต่างประเภท คือ เครือข่ายอินเทอร์เน็ต โดยอาศัยกลไกการค้นหาเกตเวย์เพื่อส่งข้อความไปยังเครื่องแม่ข่ายที่อยู่ในเครือข่ายอินเทอร์เน็ต ซึ่งจะช่วยเพิ่มอรรถประโยชน์ และสามารถนำไปประยุกต์ใช้งานในสถานการณ์จริงได้ ผลงานนี้ [16] ได้รับการเผยแพร่ในงานประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 13 ดังแสดงในภาคผนวก ค.

ลิขสิทธิ์

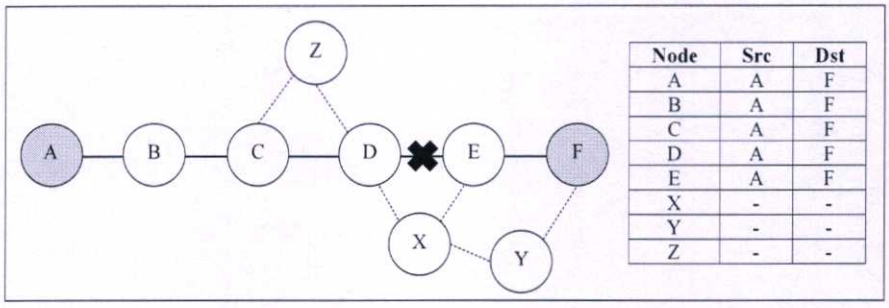
บทที่ 4
การวิเคราะห์และพัฒนาประสิทธิภาพ

จากการพัฒนา และทดสอบแอพพลิเคชันเพื่อการติดต่อสื่อสารในเครือข่ายเฉพาะกิจเคลื่อนที่ โดยการใช้แนวคิดของโพรโทคอล AODV นั้น พบว่าแอพพลิเคชันสามารถทำงานได้ และส่งข้อความจากโหนดต้นทางไปยังโหนดปลายทางได้สำเร็จ อย่างไรก็ตาม โพรโทคอล AODV นั้นยังมีจุดอ่อนอยู่หลายด้าน ผู้วิจัยได้ดำเนินการทดสอบประสิทธิภาพของโพรโทคอลค้นหาเส้นทางในเครือข่ายเฉพาะกิจเคลื่อนที่ ดังที่แสดงในภาคผนวก ง. พบว่า โพรโทคอล AODV มีโอเวอร์เฮดที่สูงมาก เนื่องจากจะต้องมีการกระจายแพ็คเกตค้นหาเส้นทางออกมาเป็นจำนวนมาก ส่งผลให้การส่งข้อความมีความล่าช้า และอาจเกิดการสูญหายได้ นอกจากนี้ จากการทดสอบประสิทธิภาพกลไกค้นหาเกตเวย์ดังที่แสดงในภาคผนวก จ. ผู้วิจัยพบว่ากลไกค้นหาเกตเวย์ที่ใช้แนวคิดของโพรโทคอล AODV นั้นยังมีปัญหาด้านโอเวอร์เฮด และความล่าช้าในการส่งข้อมูลเช่นกัน ดังนั้นในบทนี้ ผู้วิจัยจึงนำเสนองานวิจัยที่มุ่งเน้นในการปรับปรุงประสิทธิภาพด้านโอเวอร์เฮดของโพรโทคอลค้นหาเส้นทาง และกลไกค้นหาเกตเวย์ของเครือข่าย ผลงานวิจัยที่จะกล่าวถึงต่อไปนี้ในหัวข้อ 4.1–4.3 ซึ่งแสดงในภาคผนวก ฉ.–ช. และได้รับการเผยแพร่ใน [17]–[19] ตามลำดับ

4.1 การพัฒนาโพรโทคอลค้นหาเส้นทางที่มีโอเวอร์เฮดต่ำ

ในงานวิจัยส่วนนี้ ผู้วิจัยนำเสนอแนวทางในการลดโอเวอร์เฮดที่เกิดขึ้นจากการซ่อมแซมเส้นทางของกระบวนการรักษาสภาพเส้นทาง (Route Maintenance) จากการศึกษาวรรณกรรมที่เกี่ยวข้อง จะพบว่าเมื่อเส้นทางในโพรโทคอล AODV ไม่สามารถใช้งานได้ และโหนดระหว่างทางมีการส่งแพ็คเกต RERR กลับมายังต้นทาง โหนดต้นทางจะเริ่มกระบวนการค้นหาเส้นทาง (Route Discovery) ใหม่อีกครั้งโดยการกระจายแพ็คเกต RREQ ที่มีค่า TTL เป็นค่าจำนวนฮ็อปไปยังปลายทางบวกด้วยค่า TTL_INCREMENT เพื่อจำกัดการแพร่กระจายของแพ็คเกต RREQ ให้อยู่ในระยะจำนวนฮ็อปที่โหนดปลายทางของเส้นทางเก่า (โดยเผื่อระยะฮ็อปเพิ่มเติมด้วยค่า TTL_INCREMENT)

เนื่องจากแพ็คเกต RREQ จากโหนดต้นทางถูกกระจายออกไปยังทุกทิศทาง จะทำให้เกิดโอเวอร์เฮดสูงโดยเฉพาะในเครือข่ายที่มีความหนาแน่นของโหนดสูง ส่งผลให้สิ้นเปลืองแบนด์วิดท์และพลังงานของโหนดที่มีอยู่อย่างจำกัดโดยไม่จำเป็น การกระจายแพ็คเกต RREQ ที่มีประสิทธิภาพ และประหยัดทรัพยากรของเครือข่ายนั้นจึงควรจะมีเฉพาะโหนดที่อยู่ในทิศทางไปยังปลายทางช่วยกระจายแพ็คเกตเท่านั้น ซึ่งในโพรโทคอล Query Localization (QL) นั้น ได้มีการจำกัดการแพร่กระจายแพ็คเกตโดยอ้างอิงจากโหนดที่เป็นตัวกลางในเส้นทางเดิมมาเป็นตัวควบคุมให้แพ็คเกต RREQ แพร่กระจายไปในทิศทางไปยังโหนดปลายทาง

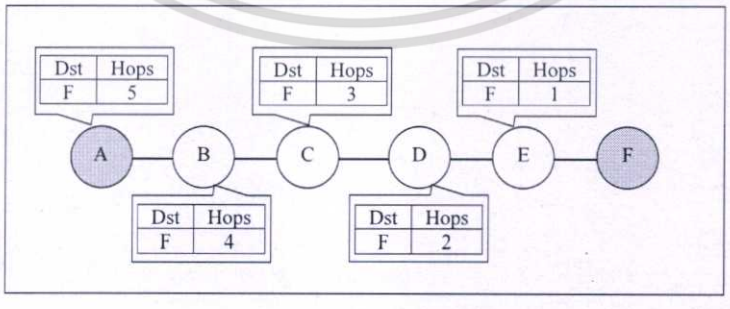


Node	Src	Dst
A	A	F
B	A	F
C	A	F
D	A	F
E	A	F
X	-	-
Y	-	-
Z	-	-

ภาพที่ 4.1 ทอพอโลยีเหตุการณ์เมื่อเส้นทางขาด

อย่างไรก็ตาม เมื่อนำแนวคิดของโพรโทคอล QL มาใช้กับโพรโทคอล AODV กลไกในการเก็บคู่ข้อมูลต้นทาง-ปลายทางนั้นจะทำให้โพรโทคอล QL ทำงานได้ไม่สมบูรณ์เมื่อมีการเรียกใช้กลไก Local Repair ของ AODV โดยจากภาพที่ 4.1 ซึ่งแสดงเส้นทางจากโหนดต้นทาง A ไปยังโหนดปลายทาง F เมื่อโหนด D ซึ่งเป็นโหนดตัวกลางพบว่าไม่สามารถติดต่อโหนด E ซึ่งเป็นโหนดถัดไปได้ ถ้าหากมีการใช้กลไก Local Repair โหนด D จะสร้างแพ็คเกต RREQ ซึ่งมีปลายทางเป็นโหนด F และต้นทางเป็นตนเอง (โหนด D) เนื่องจากโหนดระหว่างทางจากโหนด A ไปยังโหนด F นั้นมีข้อมูลคู่ต้นทาง-ปลายทางเป็นโหนด A และโหนด F ทำให้โหนด D จะต้องกระจายแพ็คเกต RREQ แบบปกติซึ่งมีการกระจายแบบรอบทิศทางแทน เนื่องจากโหนดถัดไปนั้นจะไม่มีคู่ข้อมูลต้นทาง-ปลายทางของโหนด D และ โหนด F ส่งผลให้ยังไม่สามารถลดการแพร่กระจายของแพ็คเกต RREQ ซึ่งเป็นส่วนหลักของโอเวอร์เฮดในระบบเครือข่ายได้มากนัก

เพื่อแก้ไขปัญหาโอเวอร์เฮดที่เกิดขึ้นในกลไก Local Repair ของโพรโทคอล AODV ที่มีการนำโพรโทคอล QL มาใช้นั้น ในโพรโทคอลที่ผู้วิจัยนำเสนอนั้น กระบวนการส่วนใหญ่จะอ้างอิงถึงโพรโทคอล QL ที่ใช้งานบนโพรโทคอล AODV โดยในกระบวนการค้นหาเส้นทางในตอนแรกเริ่มนั้น โหนดต้นทางจะกระจายแพ็คเกต RREQ ออกไป และโหนดปลายทางส่งแพ็คเกต RREP ตามปกติ เมื่อโหนดใด ๆ ได้รับแพ็คเกต RREP โหนดจะบันทึกคู่ของข้อมูลลงในตารางพิเศษประกอบด้วยหมายเลขไอพีของโหนดปลายทาง จำนวนฮ็อป และกำหนดเวลาหมดอายุของคู่ข้อมูล (ในโพรโทคอล QL เดิมจะบันทึกหมายเลขไอพีของโหนดต้นทางจับคู่กับหมายเลขไอพีของโหนดปลายทาง)



ภาพที่ 4.2 แสดงคู่ข้อมูลปลายทาง-จำนวนฮ็อปที่เก็บในโหนดที่ได้รับแพ็คเกต RREP

ภาพที่ 4.2 แสดงข้อมูลที่จัดเก็บในตารางพิเศษในแต่ละโหนดที่ได้รับแพ็คเกต RREP ประกอบด้วยหมายเลขไอพีของโหนดปลายทาง และจำนวนฮ็อปไปยังโหนดปลายทาง เมื่อโหนด

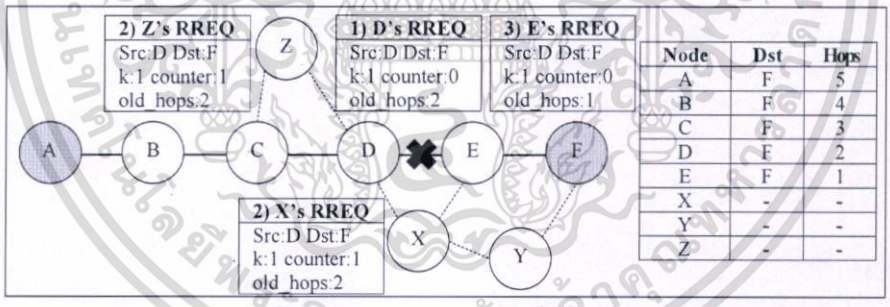
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ต้นทาง (โหนด A) เริ่มกระบวนการค้นหาเส้นทางใหม่โดยการกระจายแพ็คเกต $RREQ_{QL}$ จะมีการใส่ค่าต่าง ๆ ประกอบไปด้วย

1. ค่า k ซึ่งเป็นค่าที่กำหนดโดยผู้ใช้
2. ตัวนับ (counter) ซึ่งจะถูกกำหนดให้เป็น 0
3. ค่าจำนวนฮ็อปต่ำสุดของเส้นทางเก่า (old_hops) ซึ่งจะถูกใส่ค่าเป็นจำนวนฮ็อปไปยังโหนดปลายทางของเส้นทางเก่าโดยโหนดต้นทาง

เมื่อโหนดใด ๆ ได้รับแพ็คเกต $RREQ_{QL}$ ถ้าหากโหนดไม่สามารถสร้างแพ็คเกต RREP ได้เนื่องจากไม่มีข้อมูลเส้นทางล่าสุดไปยังโหนดปลายทาง โหนดจะเข้าสู่กระบวนการตัดสินใจว่าจะกระจายแพ็คเกต $RREQ_{QL}$ ต่อไป หรือจะละทิ้งแพ็คเกต โดยถ้าหากโหนดมีข้อมูลหมายเลขไอพีของโหนดปลายทางในตารางพิเศษ (ที่ยังไม่หมดอายุ) และจำนวนฮ็อปในตารางพิเศษมีค่าน้อยกว่าค่า old_hops ในแพ็คเกต $RREQ_{QL}$ ที่ได้รับ โหนดจะนำค่าจำนวนฮ็อปในตารางพิเศษไปใส่แทนที่ค่า old_hops เดิมของแพ็คเกต $RREQ_{QL}$ และแก้ไขค่าตัวนับเป็น 0 (สำหรับ Node Locality เท่านั้น หากเป็น Path Locality จะไม่มีการแก้ไขค่าตัวนับตามกลไกปกติของโพรโทคอล QL) แล้วกระจายออกไป แต่ค่าถ้าค่าจำนวนฮ็อปในตารางพิเศษมีค่าเท่ากับ หรือมากกว่าโหนดจะละทิ้งแพ็คเกตทันทีเนื่องจากโหนดนั้นเป็นตัวกลางในเส้นทางเดิมแต่มีความเป็นไปได้สูงที่จะอยู่ใกล้โหนดต้นทางมากกว่าโหนดปลายทาง

ในกรณีที่โหนดไม่มีคู่ข้อมูลในตารางพิเศษ (หรือมีแต่หมดอายุแล้ว) โหนดจะเพิ่มค่าตัวนับในแพ็คเกตอีก 1 ถ้าหากค่าตัวนับยังน้อยกว่า หรือเท่ากับค่า k โหนดจะกระจายแพ็คเกต $RREQ_{QL}$ แต่ถ้ามีค่ามากกว่า โหนดจะละทิ้งแพ็คเกตทันที



Node	Dst	Hops
A	F	5
B	F	4
C	F	3
D	F	2
E	F	1
X	-	-
Y	-	-
Z	-	-

ภาพที่ 4.3 แสดงเหตุการณ์เมื่อโหนด D กระจายแพ็คเกต $RREQ_{QL}$

กลไกที่นำเสนอนั้นยังสามารถช่วยให้กลไก Local Repair นั้นทำงานได้อย่างสมบูรณ์มากขึ้น โดยจากตัวอย่างในภาพที่ 4.3 เป็นเหตุการณ์ที่เส้นทางจากโหนดต้นทาง A ไปยังโหนดปลายทาง F ขาดที่ระหว่างโหนด D และ โหนด E โดยแสดงเป็นขั้นตอนตามรูปได้ดังนี้

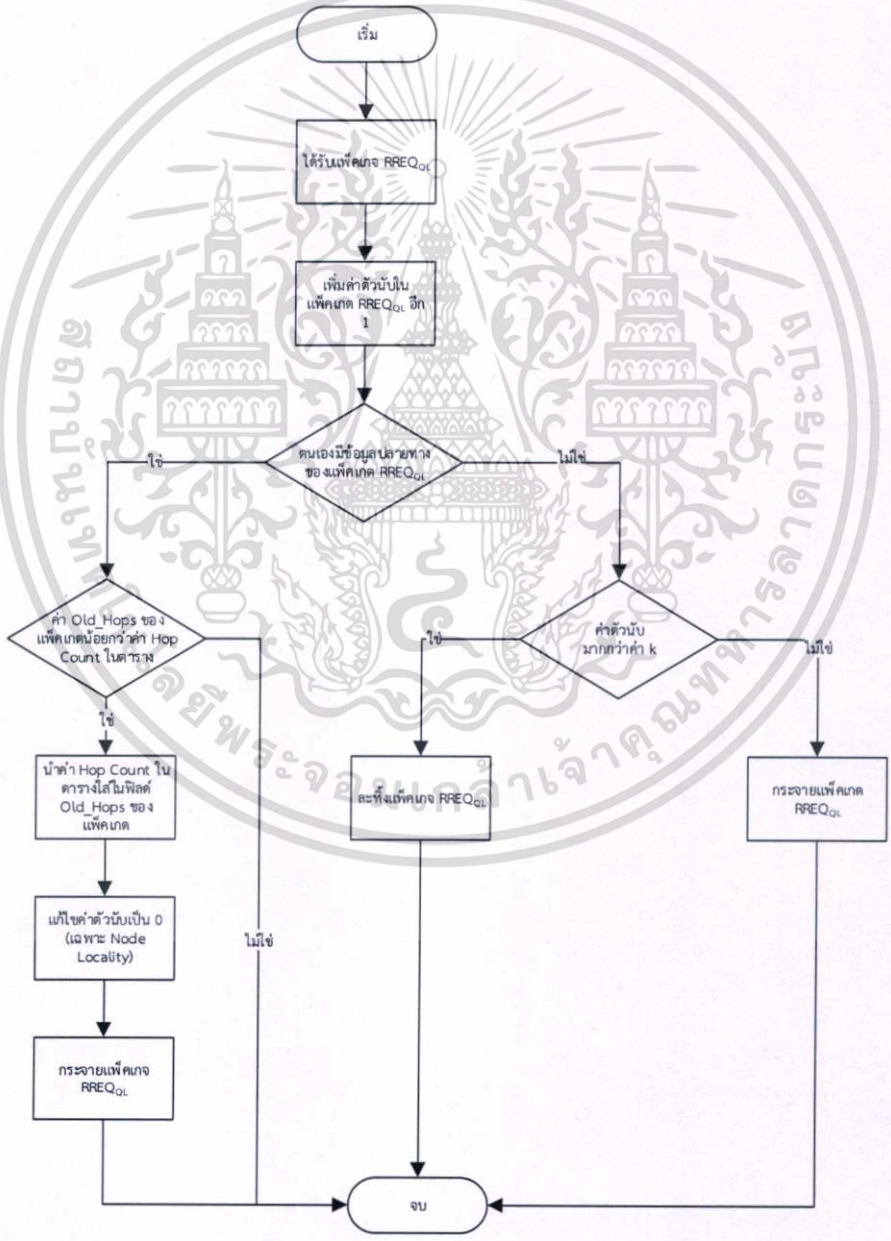
1. โหนด D จะสร้างแพ็คเกต $RREQ_{QL}$ และกระจายออกไป โดยกำหนดให้ค่า k เป็น 1 และตัวแปร old_hops เป็น 2 (จำนวนฮ็อปของเส้นทางเก่าไปยังโหนดปลายทาง F) เมื่อเทียบกับกลไกของโพรโทคอล QL โหนด D จะไม่สามารถสร้างแพ็คเกต $RREQ_{QL}$ ได้เนื่องจากหากสร้างแพ็คเกต $RREQ_{QL}$ ซึ่งจะมีหมายเลขไอพีของต้นทางเป็นตนเอง (โหนด D) จะทำให้โหนดที่เป็นตัวกลางของเส้นทางจากโหนดต้นทาง A ไปยังโหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปลายทาง F มีพฤติกรรมเหมือนโหนดทั่วไปที่ไม่ได้เป็นตัวกลางของเส้นทางเดิม ทำให้ต้องสร้างแพ็คเกต RREQ ปกติแทน

2. โหนด X และโหนด Z ได้รับแพ็คเกต $RREQ_{QL}$ และไม่มีข้อมูลปลายทางในตารางพิเศษ โหนด X จึงเพิ่มค่าตัวนับอีก 1 และเนื่องจากค่าตัวนับยังมีค่าไม่เกินกว่าค่า k โหนด X และโหนด Z จึงกระจายแพ็คเกตต่อไปด้วยกันทั้งคู่
3. โหนด E ได้รับแพ็คเกต $RREQ_{QL}$ และเนื่องจากโหนด E มีข้อมูลเส้นทางไปยังโหนดปลายทาง F และมีค่า Hop count น้อยกว่าในตัวแปร old_hops ของแพ็คเกต โหนด X จึงแก้ไขค่า old_hops เป็น 1 แล้วกระจายแพ็คเกตต่อไปจนถึงโหนด F ได้

กลไกของโพรโทคอลที่นำเสนอนั้นสามารถสรุปได้เป็นแผนผังในภาพที่ 4.4



ภาพที่ 4.4 แผนผังการทำงานเมื่อโหนดได้รับแพ็คเกต $RREQ_{QL}$ ของโพรโทคอลที่นำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ผู้วิจัยได้ทดลองประสิทธิภาพของกลไกที่นำเสนอด้วยการจำลองเครือข่ายโดยใช้ซอฟท์แวร์ Network Simulator 2 (NS-2) โดยกำหนดให้โหนดในเครือข่ายมีรัศมีการรับส่งสัญญาณเท่ากับ 100 เมตร มีอัตราเร็วในการส่งข้อมูล (Transmission rate) เท่ากับ 2 Mbps และใช้ IEEE802.11 DCF เป็นชั้น MAC (MAC Layer) ในเครือข่ายจะมีโหนดที่ทำหน้าที่ส่งข้อมูลเป็นจำนวนทั้งหมด 20 โหนด โดยรายละเอียดอื่น ๆ ได้แสดงในตารางที่ 4.1

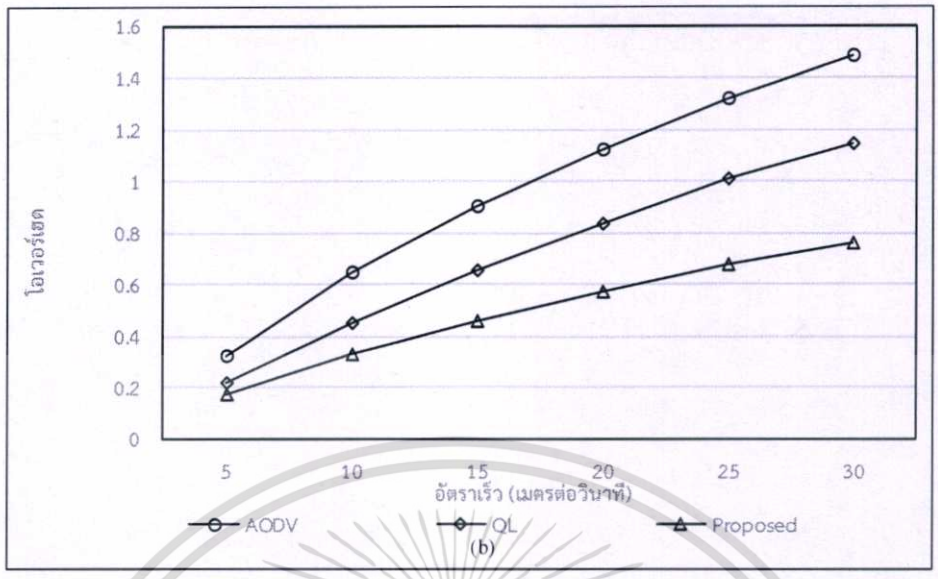
ตารางที่ 4.1 แสดงข้อมูลสภาพแวดล้อมของการจำลองเครือข่าย

รูปแบบการเคลื่อนที่	Random waypoint
ความเร็วเฉลี่ยในการเคลื่อนที่	5 10 15 20 25 30 เมตรต่อวินาที
เวลาหยุดการเคลื่อนที่	ไม่มี
จำนวนโหนดในเครือข่าย	50
เวลาที่ใช้ในการจำลอง	1000 วินาที
ขนาดเครือข่าย	300 เมตร x 600 เมตร
ประเภทของข้อมูล	UDP/CBR
ขนาดของข้อมูล	512 ไบต์
อัตราการส่งข้อมูล	4 แพ็คเกตต่อวินาที

ผู้วิจัยได้พัฒนาโพรโทคอลที่นำเสนอ (Proposed) โดยปรับปรุงมาจากโพรโทคอล QL และวัดประสิทธิภาพเปรียบเทียบกับโพรโทคอล QL เดิมที่พัฒนาบนพื้นฐานของโพรโทคอล AODV และโพรโทคอล AODV เดิมที่มีกลไก Expanding Ring Search (ERS) สำหรับโพรโทคอลที่นำเสนอ และโพรโทคอล QL นั้น ผู้วิจัยได้แสดงเฉพาะในส่วนของกลไกแบบ Node Locality ที่มีการกำหนดค่า k เริ่มต้นเป็น 1 และจะเพิ่มค่า k ขึ้นทีละ 1 ถ้าหากการค้นหาเส้นทางล้มเหลวแล้วต้องค้นหาเส้นทางใหม่ การทดลองมีตัวชี้วัดประสิทธิภาพดังต่อไปนี้

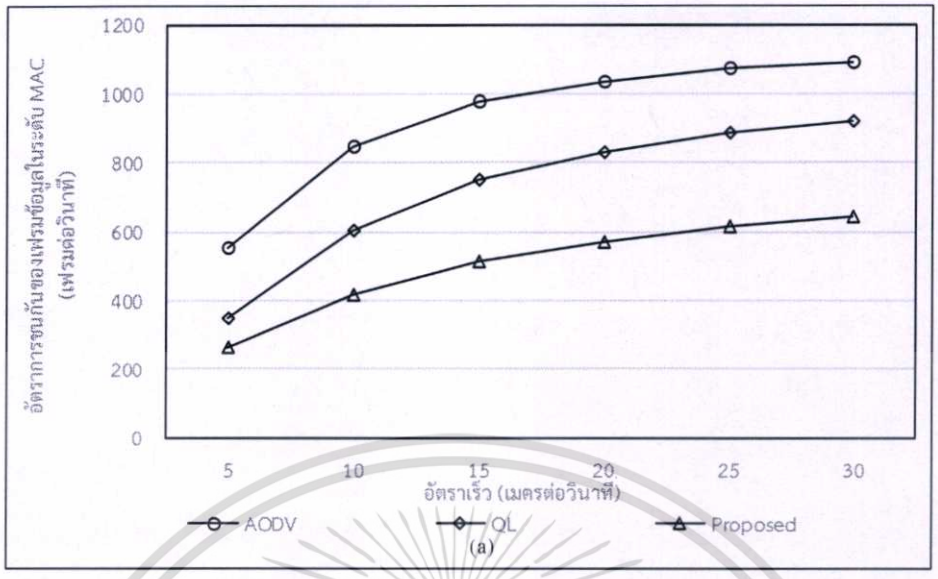
1. โอเวอร์เฮด คือ อัตราส่วนของขนาดของแพ็คเกตค้นหาเส้นทางที่ถูกส่งออกมาจากแต่ละโหนดต่อขนาดของแพ็คเกตข้อมูลทั้งหมดที่ส่งถึงปลายทาง การใช้ขนาดของแพ็คเกตจะทำให้สามารถวัดปริมาณโอเวอร์เฮดที่เกิดขึ้นได้ถูกต้องมากขึ้น เนื่องจากแพ็คเกตค้นหาเส้นทางของแต่ละโพรโทคอลนั้นมีขนาดไม่เท่ากัน
2. อัตราการชนกันของเฟรมข้อมูลในระดับ MAC คือ อัตราการชนกันของเฟรมต่าง ๆ ที่เกิดขึ้นในชั้น MAC การส่งแพ็คเกตออกมาเป็นจำนวนมากจะส่งผลให้เกิดอัตราการชนกันของเฟรมสูงมากขึ้น และเกิดการสูญหายของเฟรม
3. อัตราการส่งข้อมูลสำเร็จ คือ อัตราส่วนของจำนวนแพ็คเกตข้อมูลที่ถึงปลายทางต่อจำนวนแพ็คเกตข้อมูลที่ส่งจากต้นทาง ตัวชี้วัดนี้แสดงให้เห็นถึงความสำเร็จในการส่งข้อมูล
4. ความหน่วงในการส่งข้อมูล คือ เวลาที่ใช้ในการส่งแพ็คเกตข้อมูลจากต้นทางถึงปลายทาง โดยนับตั้งแต่เวลาที่แพ็คเกตถูกสร้างจากโหนดต้นทางจนไปถึงเวลาที่โหนดปลายทางได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



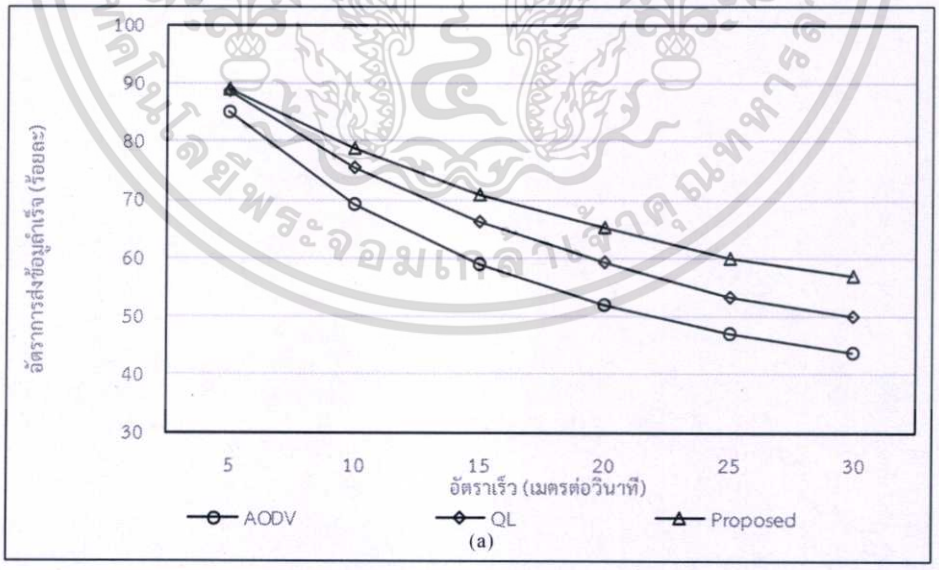
ภาพที่ 4.5 โอเวอร์เฮด

ภาพที่ 4.5 แสดงโอเวอร์เฮดที่เกิดขึ้นในเครือข่าย เมื่อความเร็วของโหนดเพิ่มมากขึ้น โอเวอร์เฮดจะสูงขึ้นตาม เนื่องจากเส้นทางในการส่งข้อมูลจะขาดบ่อยมากขึ้น ส่งผลให้มีการแพร่กระจายแพ็คเกตค้นหาเส้นทางมากขึ้น โพรโทคอล AODV นั้นมีโอเวอร์เฮดสูงที่สุด เนื่องจากมีการจำกัดการแพร่กระจายของแพ็คเกต RREQ โดยการกำหนดค่า TTL เท่านั้น แต่แพ็คเกต RREQ ยังมีการกระจายในลักษณะออกจากโหนดต้นทางในแบบรอบทิศทาง ในส่วนของโพรโทคอล QL นั้นมีการจำกัดจำนวนโหนดที่มีการกระจายแพ็คเกตโดยจำกัดขอบเขตให้อยู่ในบริเวณโหนดตัวกลางของเส้นทางเดิม ทำให้สามารถลดโอเวอร์เฮดลงอย่างเห็นได้ชัด ในขณะที่โพรโทคอลที่ผู้วิจัยนำเสนอนั้นมีโอเวอร์เฮดต่ำที่สุด เนื่องจากมีการปรับปรุงเงื่อนไขในการกระจายแพ็คเกต RREQ ซึ่งทำให้กลไกของโพรโทคอล QL นั้นสามารถทำงานได้ทันทีเมื่อมีการทำกระบวนการ Local Repair ทำให้ลดการกระจายแพ็คเกต RREQ แบบรอบทิศทางทั้งการค้นหาเส้นทางแบบปกติ และในกระบวนการ Local Repair ทำให้สามารถลดโอเวอร์เฮดได้อย่างมีประสิทธิภาพ



ภาพที่ 4.6 อัตราการชนกันของเฟรมข้อมูลในระดับ MAC

ภาพที่ 4.6 แสดงอัตราการชนกันของเฟรมข้อมูลในระดับ MAC ซึ่งเป็นผลสืบเนื่องส่วนหนึ่งจากโอเวอร์เฮดที่เกิดขึ้นในเครือข่าย เมื่อโหนดมีความเร็วมากขึ้นจะมีมีอัตราการชนกันของเฟรมสูงขึ้นเนื่องจากมีโอเวอร์เฮดสูงขึ้น โดยโพรโทคอล AODV มีการอัตราการชนกันของเฟรมสูงสุดเนื่องจากมีการแพร่กระจายแพ็คเกตที่ทำให้เกิดโอเวอร์เฮดเป็นจำนวนมาก ตามด้วยโพรโทคอล QL ซึ่งมีโอเวอร์เฮดต่ำกว่ารองลงมา ในส่วนของโพรโทคอลที่ผู้วิจัยนำเสนอนั้นมีอัตราการชนกันของเฟรมต่ำที่สุด ซึ่งเป็นผลมาจากการลดโอเวอร์เฮดที่มีประสิทธิภาพ

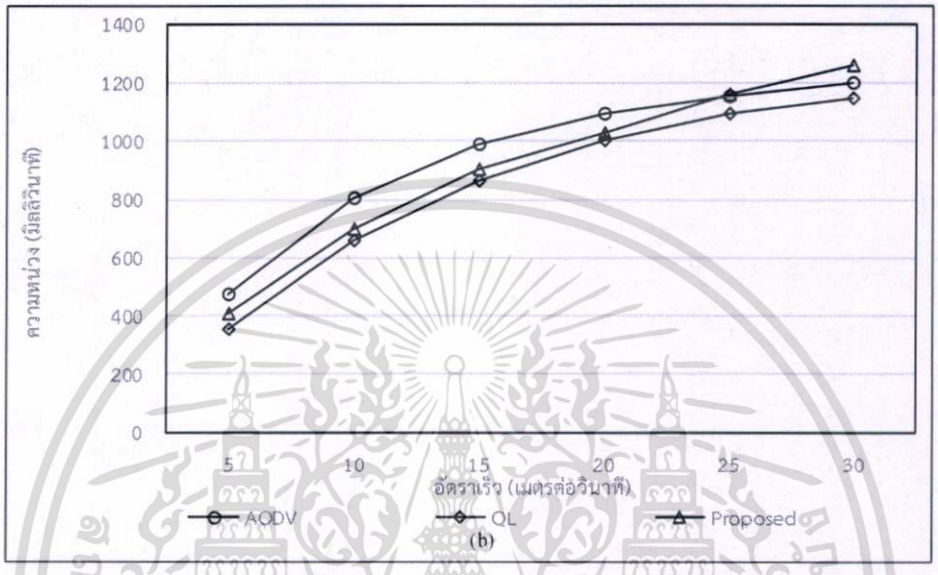


ภาพที่ 4.7 อัตราการส่งข้อมูลสำเร็จ

ภาพที่ 4.7 แสดงอัตราการส่งข้อมูลสำเร็จเมื่อโหนดมีความเร็วสูงขึ้น อัตราการส่งข้อมูลสำเร็จนั้นลดลงเนื่องจากเครือข่ายมีอัตราการชนกันของเฟรมสูงขึ้น ซึ่งส่งผลให้แพ็คเกตต่าง ๆ มีการสูญหายระหว่างทาง โดยโพรโทคอลที่ผู้วิจัยนำเสนอนั้นมีอัตราการส่งข้อมูลสำเร็จสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เนื่องจากมีประสิทธิภาพในการลดโอเวอร์เฮด และอัตราการชนกันของเฟรม ทำให้มีผลกระทบต่อการสูญหายของแพ็คเกตข้อมูลต่ำที่สุด ในขณะที่โพรโทคอล QL และ AODV นั้นมีประสิทธิภาพรองลงมา โพรโทคอล AODV นั้นมีอัตราการส่งข้อมูลโดยรวมสำเร็จต่ำที่สุด เนื่องจากมีโอเวอร์เฮดที่สูงที่สุด และมีอัตราอัตราการชนกันของเฟรมที่สูงมาก ซึ่งส่งผลให้เกิดการสูญหายของแพ็คเกตข้อมูลเป็นจำนวนมากในที่สุด



ภาพที่ 4.8 ความหน่วงในการส่งข้อมูล

ภาพที่ 4.8 แสดงความหน่วงในการส่งข้อมูล เมื่อโหนดมีความเร็วสูงขึ้นจะมีความหน่วงที่สูงขึ้นตาม เนื่องจากเส้นทางในการส่งข้อมูลจะใช้งานได้ไม่นาน และขาดบ่อย โหนดจะต้องรอนานขึ้นในการค้นหาเส้นทางที่สามารถใช้งานได้ โดยจากผลการทดลองแสดงให้เห็นว่าแต่ละโพรโทคอลนั้นมีความหน่วงที่ไม่ต่างกันอย่างเห็นได้ชัดนัก โดยโพรโทคอล AODV มีความหน่วงโดยรวมสูงที่สุด ในขณะที่โพรโทคอลของผู้นำเสนอนั้นมีความหน่วงต่ำเมื่อมีความเร็วของโหนดไม่สูงมาก แต่เมื่อความเร็วของโหนดสูงมากจะมีความหน่วงที่สูงขึ้นเนื่องจากการจำกัดจำนวนโหนดที่แพร่กระจายแพ็คเกตนั้นเข้มงวดเกินไปจนอาจจะทำให้การค้นหาเส้นทางล้มเหลว และต้องมีการค้นหาเส้นทางใหม่ ทำให้มีความหน่วงสูงขึ้น

โดยสรุปแล้ว โพรโทคอลที่ถูกปรับปรุง และนำเสนอนั้นจะลดโอเวอร์เฮดจากการเกิดการแลกเปลี่ยนแพ็คเกตค้นหาเส้นทางที่ไม่จำเป็นเมื่อเส้นทางขาด ซึ่งจากผลการทดลองที่ได้นำเสนอไปนั้น โพรโทคอลที่นำเสนอยังสามารถรักษาประสิทธิภาพทางด้านอัตราการส่งข้อมูลสำเร็จ และความหน่วงในการส่งข้อมูลได้ ทำให้มีความคุ้มค่า และประสิทธิภาพต่อค่าใช้จ่ายในการใช้งานนั้นสูงขึ้น เหมาะสำหรับการนำไปประยุกต์ใช้ในสถานการณ์จริงได้เป็นอย่างดี

4.2 การพัฒนาโพรโทคอลค้นหาเส้นทางโดยการควบคุมการกระจายแพ็คเกตค้นหาเส้นทาง

จากการศึกษาโพรโทคอลค้นหาเส้นทางแบบต่าง ๆ จะเห็นว่าการเพิ่มขึ้นของโอเวอร์เฮดในเครือข่าย เกิดได้จากหลายสาเหตุ เช่น ความหนาแน่นในเครือข่ายที่มีมากขึ้น ทำให้มีจำนวนโหนดที่จะต้องกระจายแพ็คเกตค้นหาเส้นทางมากขึ้นตามไปด้วย อีกทั้งโหนดสามารถเคลื่อนที่ได้ตลอดเวลา ทำให้รูปแบบของทอพอโลยีเปลี่ยนแปลงไป ไม่สามารถใช้เส้นทางเดิมในการส่งข้อมูลได้ โหนดต้นทางจึงต้องค้นหาเส้นทางใหม่ทำให้เกิดโอเวอร์เฮดในการค้นหาเส้นทาง ซึ่งโอเวอร์เฮดที่เพิ่มขึ้นจะส่งผลต่อประสิทธิภาพด้านต่าง ๆ ของโพรโทคอล

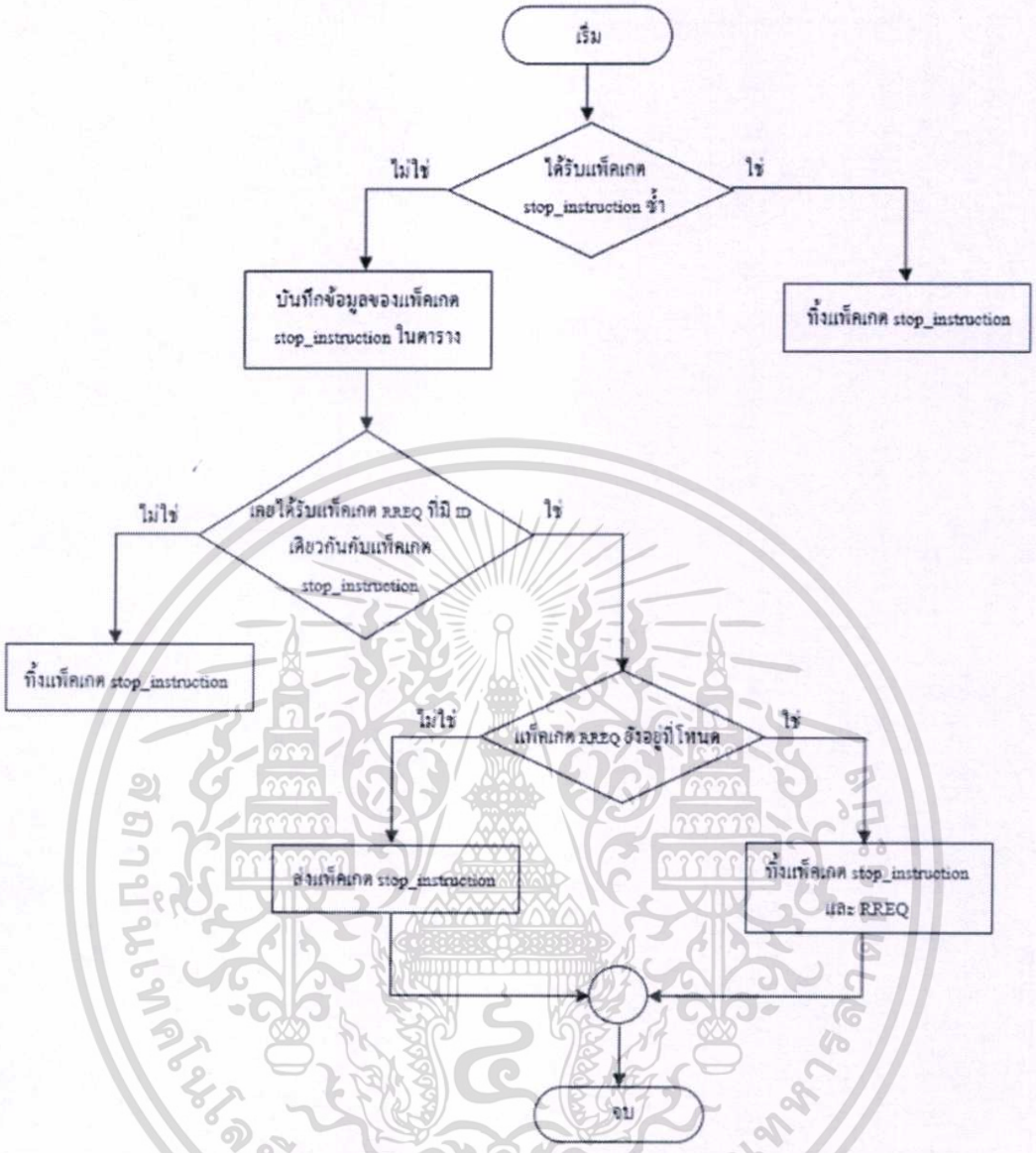
ในหลายบทความได้พัฒนาและปรับปรุงกระบวนการค้นหาเส้นทางโดยอ้างอิงจากโพรโทคอล AODV เนื่องจากเป็นโพรโทคอลแบบเชิงรับ จะค้นหาเส้นทางเมื่อโหนดต้นทางต้องการจะส่งข้อมูลถึงโหนดปลายทาง ซึ่งช่วยลดโอเวอร์เฮดได้ในระดับหนึ่ง แต่การกระจายแพ็คเกตค้นหาเส้นทาง ของ AODV จะกระจายแพ็คเกตไปทั่วทั้งเครือข่ายซึ่งมากเกินความจำเป็น จึงได้มีการศึกษา และปรับปรุงกลไกการค้นหาเส้นทางโดยอ้างอิงจากโพรโทคอล BERS ให้มีประสิทธิภาพที่ดีขึ้น เนื่องจากในการกระจายของแพ็คเกต stop_instruction ที่กระจายเท่ากับจำนวนฮ็อปจากโหนดต้นทางถึงโหนดปลายทางอาจจะไม่สามารถหยุดแพ็คเกต RREQ ในเครือข่ายได้ทั้งหมด

4.2.1 สร้างเงื่อนไขการกระจายแพ็คเกต stop_instruction

ในส่วนของแพ็คเกต stop_instruction ของกระบวนการค้นหาเส้นทางแบบ Blocking Expanding Ring Search (BERS) แพ็คเกตจะถูกสร้างขึ้นจากโหนดต้นทาง เมื่อโหนดต้นทางได้รับแพ็คเกต RREP จากโหนดปลายทางแล้ว จะกระจายแพ็คเกต stop_instruction ออกไปเพื่อใช้ในการหยุดการส่งแพ็คเกต RREQ ที่ยังคงกระจายอยู่ในเครือข่าย ซึ่งแพ็คเกตนี้จะกระจายไปตามจำนวน Hop Count ที่ได้รับมาจากแพ็คเกต RREP และเมื่อกระจายไปจนถึงค่า Hop Count แล้ว แพ็คเกตนี้จะถูกละทิ้งทันที

ผู้วิจัยจึงได้นำงานส่วนหนึ่งของ [8] หรือ BERS+ มาใช้ เนื่องจากแพ็คเกต stop_instruction ของ BERS ใช้จำนวน Hop Count ในการกระจาย จึงอาจทำให้ไม่เพียงพอต่อการหยุดแพ็คเกต RREQ ในเครือข่าย เพราะโหนดที่ถือแพ็คเกต RREQ อยู่อาจจะกระจายแพ็คเกตออกไปไกลมากกว่าจำนวน Hop Count ที่แพ็คเกต stop_instruction จะไปถึง

BERS+ จึงได้เปลี่ยนจากการใช้ Hop Count เป็นการใช้เงื่อนไขในการส่งต่อแพ็คเกตนี้แทน ซึ่งเงื่อนไขดังภาพที่ 4.9



ภาพที่ 4.9 เงื่อนไขการกระจายแพ็คเกต stop_instruction

4.2.2 สร้างเงื่อนไขเวลารอแพ็คเกต stop_instruction

จากการศึกษากระบวนการค้นหาเส้นทาง Blocking Expanding Ring Search (BERS) พบว่าโหนดระหว่างทางที่ได้รับแพ็คเกต RREQ จะต้องรอรับแพ็คเกต stop_instruction จากโหนดต้นทางตามเวลารอ คือ 2 * จำนวนฮ็อปไปยังโหนดต้นทาง * เวลาที่ใช้ในการส่งแพ็คเกตในระยะ 1 ฮ็อป ยิ่งโหนดอยู่ห่างจากโหนดต้นทางมาก จำนวนฮ็อปก็จะยิ่งเพิ่มขึ้น หมายความว่าเวลารอของโหนดก็จะเพิ่มขึ้นตามไปด้วย ส่งผลต่อเวลาที่ใช้ในการค้นหาเส้นทาง รวมไปถึงเวลาที่ใช้ส่งแพ็คเกตข้อมูลก็จะช้าลงไปด้วย

ทางผู้วิจัยจึงได้คิดแนวทางที่จะลดเวลาในการรอแพ็คเกต stop_instruction เพื่อลดเวลาที่ใช้ในการค้นหาเส้นทาง ซึ่งได้คิดไว้ 2 แนวคิด ประกอบด้วย

แนวคิดที่ 1 กำหนดให้โหนดที่อยู่ห่างจากโหนดต้นทางไม่เกิน k ฮ็อป ใช้เวลารอแบบ BERS คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

$$\text{WAITING TIME} = 2 \times \text{Hop Count} \times \text{TRAVERSAL TIME} \quad (4.1)$$

โดยที่ Hop Count คือ จำนวนฮ็อป

TRAVERSAL TIME คือ เวลาที่ใช้ในการส่งแพ็คเกตในระยะ 1 ฮ็อป

และสำหรับโหนดที่อยู่ห่างจากโหนดต้นทางเกินระยะ k ฮ็อป จะใช้เวลารอของโหนดที่อยู่ห่างจากโหนดต้นทางเป็นจำนวน k ฮ็อป คือ

$$\text{WAITING TIME} = 2 \times k \times \text{TRAVERSAL TIME} \quad (4.2)$$

แนวคิดที่ 2 จากการศึกษากลไกการทำงานของ BERS+ หลักการในการกระจายแพ็คเกต RREQ ภายใน k ฮ็อป จะกระจายแพ็คเกต RREQ โดยไม่มีเวลารอ กล่าวคือกระจายแพ็คเกตเหมือนกับ AODV ส่วนภายนอก k ฮ็อป จะกระจายแพ็คเกตโดยมีเวลารอเหมือนกับ BERS

ผู้วิจัยได้มีการกำหนดค่า k เพื่อให้มีเวลารอภายใน k ฮ็อป และภายนอก k ฮ็อปที่แตกต่างกัน เนื่องจากถ้าหากโหนดในเครือข่ายไม่มีเวลารอในการส่งแพ็คเกต RREQ จะทำให้มีดีเลย์ในการค้นหาเส้นทางน้อย แต่จะทำให้เกิดโอเวอร์เฮดจากการกระจายแพ็คเกต RREQ มาก ทำให้เกิดความคับคั่งภายในเครือข่าย และเกิดการชนกันของข้อมูลได้ แต่ถ้าโหนดมีเวลารอเพิ่มขึ้นตามจำนวนฮ็อปแล้ว แม้จะมีโอเวอร์เฮดในเครือข่ายน้อย แต่ก็จะทำให้มีดีเลย์ในการค้นหาเส้นทางมาก ส่งผลต่อเวลาในการส่งข้อมูลก็จะช้าลงตามไปด้วย

และได้ทำการประมาณค่า k ที่เหมาะสมจากการใช้สมการ Poisson โดยตั้งสมมติฐานไว้ว่าในระยะรัศมีของสัญญาณไร้สายของโหนดเพื่อจะมีโอกาสโหนดที่เป็นโหนดปลายทางดังนี้

$$k = \sqrt{\frac{n_k \times A}{N_T \times \pi r^2}} \quad (4.3)$$

โดยที่ k คือ จำนวนฮ็อปที่แบ่งระหว่างบริเวณระยะไม่เกิน k ฮ็อป และเกินระยะ k ฮ็อป

N_k คือ จำนวนโหนดที่อยู่ในพื้นที่รัศมี k ฮ็อปนับจากโหนดต้นทาง

A คือ พื้นที่การจำลอง

N_T คือ จำนวนโหนดทั้งหมดในการจำลอง

πr^2 คือ พื้นที่ภายใต้ระยะสัญญาณของโหนด

ส่วนค่า n_k นั้นหาได้จากสมการ Binomial โดยตั้งสมมติฐานไว้ว่า ในทอพอโลยีที่มีโหนดอยู่ทั้งหมด N_T โหนด จะมีโหนดปลายทางอยู่ m โหนด และจะทำการเลือกมา N_k โหนดจากโหนดทั้งหมด เพื่อให้เจอโหนดปลายทางทั้งหมดที่ความน่าจะเป็น P ซึ่งสามารถสรุปเป็นสูตรได้ดังนี้

$$P = \frac{(N_T - m)! n_k!}{N_T! (n_k - m)!} \quad (4.4)$$

และถ้าโหนดปลายทางมีอยู่ 1 โหนด (m = 1) จะได้ผลลัพธ์เป็น

$$P = \frac{n_k}{N_T} \tag{4.5}$$

จากสมมติฐานที่ตั้งไว้ว่าจะต้องเจอโหนดปลายทางด้วยความน่าจะเป็น 50% ค่า N_k ก็จะมีค่าดังนี้

$$n_k = 0.5 \times N_T \tag{4.6}$$

เพราะฉะนั้น สมการที่ใช้ประมาณค่า k ก็มีค่าเป็น

$$k = \sqrt{\frac{A}{2\pi r^2}} \tag{4.7}$$

เมื่อได้ค่า k มาแล้วจึงนำไปใช้กับแนวคิดที่นำเสนอไว้ข้างต้น

จากแนวคิดในการปรับปรุงโพรโทคอล BERS ผู้วิจัยได้ทำการทดลองประสิทธิภาพของโพรโทคอลด้วยจำลองเครือข่ายโดยใช้ซอฟท์แวร์ Network Simulator 2 (NS-2) โดยกำหนดให้โหนดในเครือข่ายมีรัศมีการรับส่งสัญญาณเท่ากับ 250 เมตร มีอัตราเร็วในการส่งข้อมูล (Transmission rate) เท่ากับ 2 Mbps และใช้ IEEE802.11 DCF เป็นชั้น MAC (MAC Layer) ในเครือข่ายจะมีโหนดที่ทำหน้าที่ส่งข้อมูลเป็นจำนวนทั้งหมด 20 โหนด

ตารางที่ 4.2 แสดงข้อมูลสภาพแวดล้อมของการจำลองเครือข่าย

รูปแบบการเคลื่อนที่	Random waypoint
ความเร็วในการเคลื่อนที่	แบบสุ่มยูนิฟอร์มในช่วง 1-20 เมตร/วินาที
เวลาหยุดการเคลื่อนที่	ไม่มี
จำนวนโหนดในเครือข่าย	50 60 70 80 90 100
เวลาที่ใช้ในการจำลอง	1000 วินาที
พลังงานเริ่มต้น	500 จูล
พลังงานที่ใช้ในสถานะ Transmit	1.34616 วัตต์
พลังงานที่ใช้ในสถานะ Receive	0.9006 วัตต์
พลังงานที่ใช้ในสถานะ Idle	0.74 วัตต์
พลังงานที่ใช้ในสถานะ Sleep	0.0474 วัตต์
ขนาดเครือข่าย	1000 เมตร x 1000 เมตร
ประเภทของข้อมูล	UDP/CBR
ขนาดของข้อมูล	512 ไบต์
อัตราการส่งข้อมูล	4 แพ็คเกตต่อวินาที

หมายเหตุ – ใช้ค่าการใช้พลังงานในสถานะต่าง ๆ ของโมดูลไร้สายจาก [20]

โดยจะเป็นการเปรียบเทียบประสิทธิภาพของ BERS แนวคิดที่ 1 ที่มีค่า k = 1 k = 2 และ k = 5 แนวคิดที่ 2 ที่มีค่า k = 1 k = 2 และ k = 5 ซึ่งค่า k = 1 เป็นค่าจำนวนฮ็อปที่น้อยที่สุด k

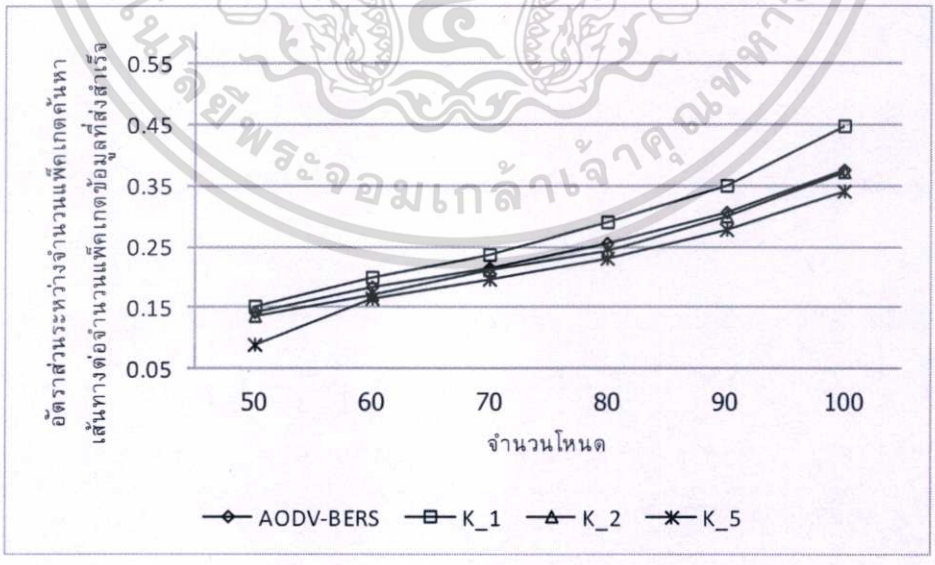
= 2 ได้จากการคำนวณจากสมการ และค่า k = 5 เป็นค่าจำนวนฮ็อปที่มากที่สุดที่เป็นไปได้ในขนาดเครือข่ายแบบนี้ ผู้วิจัยได้พัฒนาโพรโทคอลเหล่านี้บนพื้นฐานของโพรโทคอล AODV

จากกราฟแสดงสัญลักษณ์ K_1 คือ แนวคิดที่ 1 ที่ตั้งค่า k = 2 K_2 คือ แนวคิดที่ 1 ที่ตั้งค่า k = 2 ในขณะที่ K_5 คือ แนวคิดที่ 1 ที่ตั้งค่า k = 5 ในส่วนของ Plus_k_1 คือ แนวคิดที่ 2 ที่ตั้งค่า k = 1 Plus_k_2 คือ แนวคิดที่ 2 ที่ตั้งค่า k = 2 และ Plus_k_5 คือ แนวคิดที่ 2 ที่ตั้งค่า k = 5

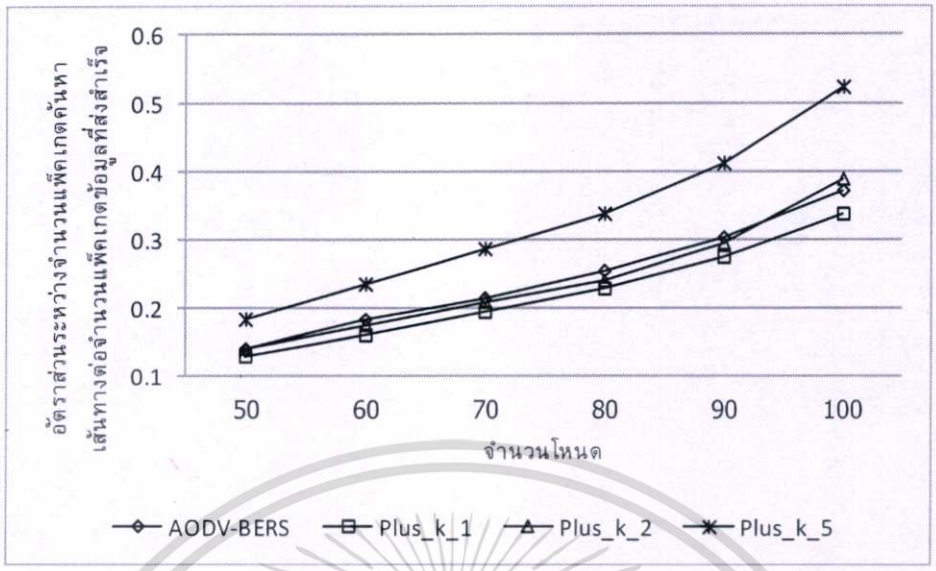
ในการเปรียบเทียบประสิทธิภาพการทำงานของแต่ละกระบวนการที่นำมาจำลอง ผู้วิจัยได้เลือกตัวชี้วัดที่ใช้ในการเปรียบเทียบประสิทธิภาพประกอบด้วยโอเวอร์เฮด ความหน่วงในการส่งข้อมูล และอัตราการส่งข้อมูลสำเร็จ นอกจากนี้ ผู้วิจัยได้เพิ่มตัวชี้วัดด้านพลังงาน และความยั่งยืนของเครือข่ายดังต่อไปนี้

1. การใช้พลังงานของเครือข่าย คือ ค่าพลังงานที่ถูกใช้ไปในระหว่างการสื่อสารกัน เป็นตัวชี้วัดว่าในระบบมีการใช้พลังงานมากหรือน้อยเพียงใด
2. อายุขัยของเครือข่าย คือ เวลาที่เครือข่ายเริ่มมีโหนดที่พลังงานหมด
3. จำนวนโหนดที่เหลือ คือ จำนวนโหนดที่เหลืออยู่ในเครือข่ายทั้งหมดเมื่อเวลาในการจำลองสิ้นสุดลง

ตัวชี้วัดเหล่านี้บ่งบอกถึงประสิทธิภาพของเครือข่าย โดยตัวชี้วัดแต่ละตัวจะมีความสอดคล้องและสัมพันธ์กัน โดยการใช้พลังงานของเครือข่าย จะขึ้นอยู่กับโอเวอร์เฮดของเครือข่าย ส่งผลกระทบต่ออายุขัยของเครือข่าย อายุขัยของเครือข่าย ขึ้นอยู่กับการใช้พลังงานของเครือข่าย และโอเวอร์เฮด ส่งผลต่ออัตราการส่งข้อมูลสำเร็จ อัตราการส่งข้อมูลสำเร็จ ขึ้นอยู่กับโอเวอร์เฮด และความหน่วงในการส่งข้อมูล ความหน่วงในการส่งข้อมูล ขึ้นอยู่กับโอเวอร์เฮด ส่งผลต่ออัตราการส่งข้อมูลสำเร็จ และโอเวอร์เฮด ส่งผลต่อความหน่วงในการส่งข้อมูล อัตราการส่งข้อมูลสำเร็จ การใช้พลังงานของเครือข่าย และอายุขัยของเครือข่าย

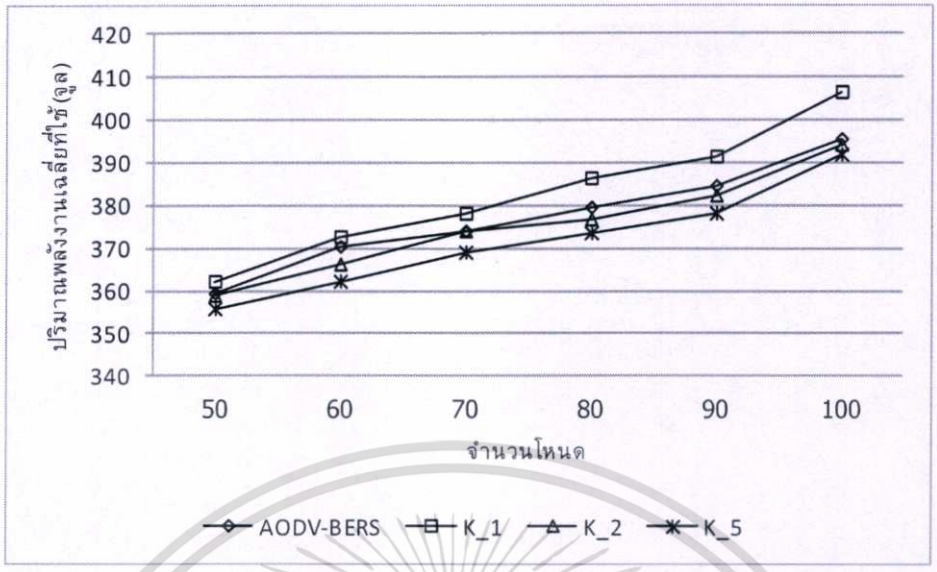


ภาพที่ 4.10 แสดงโอเวอร์เฮดในเครือข่ายของแนวคิดที่ 1

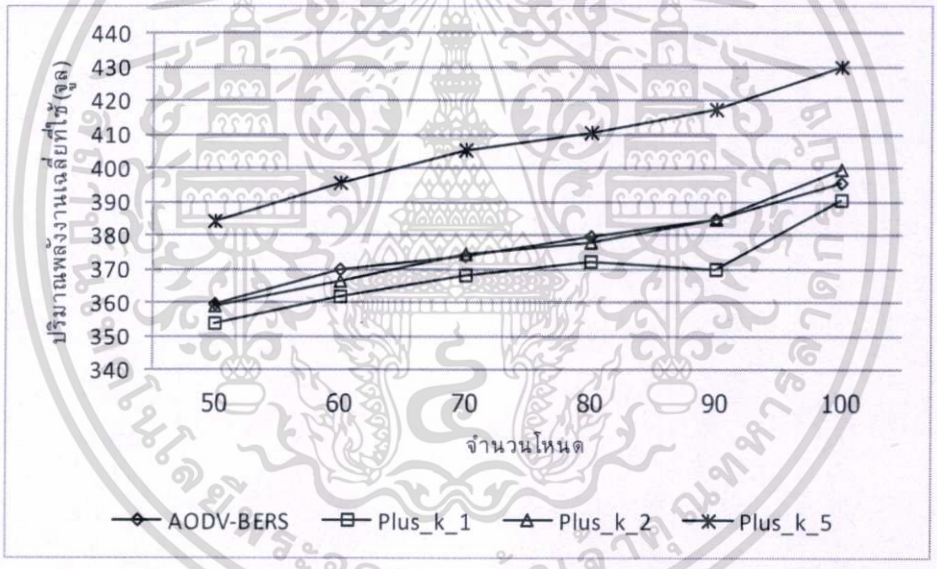


ภาพที่ 4.11 แสดงโอเวอร์เฮดในเครือข่ายของแนวคิดที่ 2

ภาพที่ 4.10 และ ภาพที่ 4.11 แสดงโอเวอร์เฮดที่เกิดขึ้นในเครือข่าย จะเห็นได้ว่า ภาพที่ 4.10 แนวคิดที่ 1 (k = 1) มีโอเวอร์เฮดมากที่สุด เนื่องจากมีเวลารอในการส่งแพ็คเกตค้นหาเส้นทางคงที่หลังจากฮ็อปที่ 1 ไปแล้ว ทำให้ใช้เวลารอในการกระจายแพ็คเกตค้นหาเส้นทางน้อยกว่า k = 2 และ k = 5 ส่งผลให้มีการกระจายของแพ็คเกตค้นหาเส้นทางเร็วขึ้น และมีแพ็คเกตที่กระจายอยู่เป็นจำนวนมาก ซึ่งตรงข้ามกับ k = 5 จะมีโอเวอร์เฮดน้อยที่สุด เนื่องจากมีเวลารอมากขึ้นเรื่อย ๆ ตามจำนวนฮ็อป ทำให้มีแพ็คเกตที่ถูกกระจายน้อยลง โอเวอร์เฮดในเครือข่ายจึงน้อยลงตามไปด้วย ส่วนภาพที่ 4.11 แนวคิดที่ 2 (k = 5) มีโอเวอร์เฮดมากที่สุด เนื่องจากภายในจำนวนฮ็อป 5 ฮ็อป จะมีการกระจายแพ็คเกตค้นหาเส้นทางออกไปโดยไม่มีเวลอรอ ทำให้มีการกระจายแพ็คเกตมาก และ AODV-BERS มีโอเวอร์เฮดมากกว่า k = 2 และ k = 5 เนื่องจากการกระจายแพ็คเกต stop_instruction เท่ากับจำนวนฮ็อปที่ได้รับมาจาก RREP ทำให้ไม่สามารถหยุดการกระจายของแพ็คเกต RREQ ที่อยู่นอกจำนวนฮ็อปนั้นได้ จึงทำให้มีโอเวอร์เฮดสูงกว่า

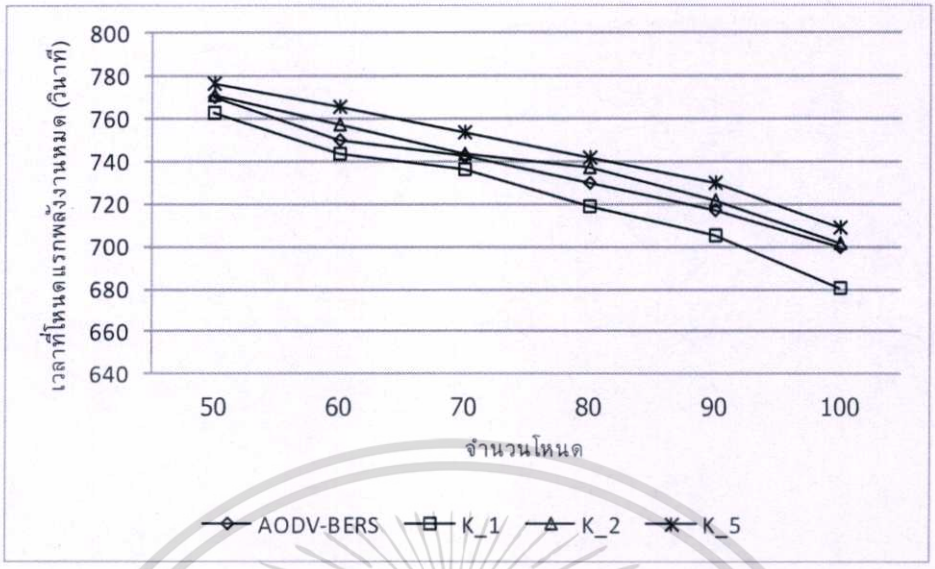


ภาพที่ 4.12 แสดงการใช้พลังงานในเครือข่ายของแนวคิดที่ 1

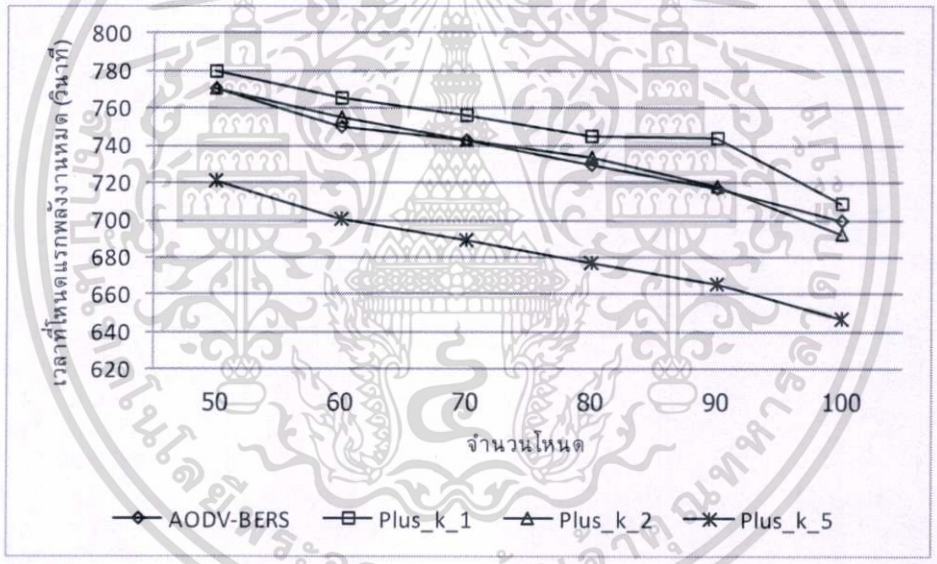


ภาพที่ 4.13 แสดงการใช้พลังงานในเครือข่ายของแนวคิดที่ 2

ภาพที่ 4.12 และ ภาพที่ 4.13 แสดงการใช้พลังงานในเครือข่าย จะเห็นได้ว่า กราฟพลังงานจะแปรผันตามกราฟโอเวอร์เฮด โดยภาพที่ 4.12 แนวคิดที่ 1 (k = 1) มีระดับการใช้พลังงานมากที่สุด เนื่องจากเกิดโอเวอร์เฮดในการกระจายแพ็คเกตค้นหาเส้นทางมากกว่า ส่วน k = 2 มีระดับการใช้พลังงานน้อยกว่า AODV-BERS เนื่องจาก k = 2 แพ็คเกต stop_instruction จะใช้เงื่อนไขในการกระจายแพ็คเกตแทนการกระจายเท่ากับจำนวนฮ็อปโหนดต้นทางถึงโหนดปลายทาง ทำให้สามารถหยุดการกระจายของแพ็คเกต RREQ ได้มากกว่า ในภาพที่ 4.13 แนวคิดที่ 2 (k = 5) มีระดับการใช้พลังงานมากที่สุด เนื่องจากมีความหน่วงในการส่งแพ็คเกต RREQ น้อย ทำให้การกระจายแพ็คเกตเป็นไปอย่างรวดเร็ว เกิดโอเวอร์เฮดในเครือข่ายมาก ส่งผลให้ใช้พลังงานมากขึ้นด้วย



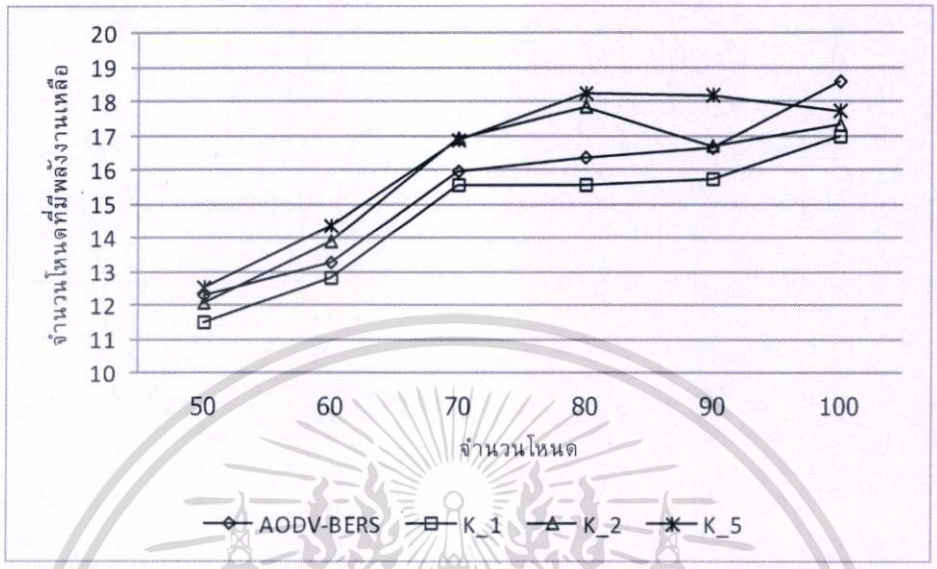
ภาพที่ 4.14 แสดงเวลาที่โหนดแรกพลังงานหมดของแนวคิดที่ 1



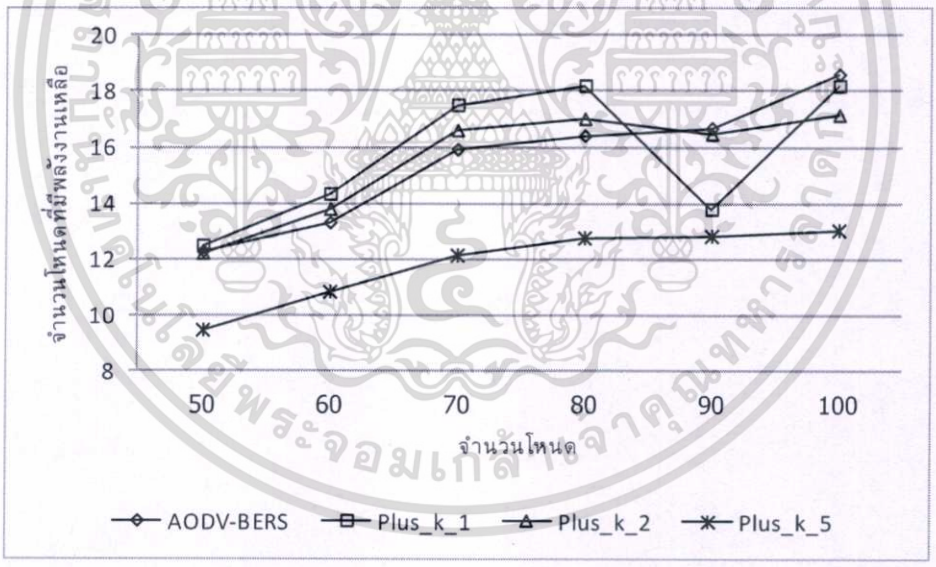
ภาพที่ 4.15 แสดงเวลาที่โหนดแรกพลังงานหมดของแนวคิดที่ 2

ภาพที่ 4.14 และ ภาพที่ 4.15 แสดงเวลาที่โหนดแรกพลังงานหมด จะเห็นได้ว่า กราฟเวลาที่โหนดแรกพลังงานหมดจะแปรผกผันกับกราฟพลังงาน โดยภาพที่ 4.14 เวลาที่โหนดแรกพลังงานหมดของแนวคิดที่ 1 (k = 5) จะช้าที่สุด เนื่องจากมีการใช้พลังงานน้อยที่สุด และโหนดแรกพลังงานหมดช้ากว่า k = 2 เนื่องจากเวลารอการส่งแพ็คเกต RREQ มากกว่า ทำให้มีแพ็คเกตกระจายอยู่ในเครือข่ายน้อยกว่าภาพที่ 4.15 แนวคิดที่ 2 (k = 5) โหนดแรกจะพลังงานหมดรวดเร็วกว่า k = 2 เนื่องจากภายใน 5 ฮ็อปจะกระจายแพ็คเกต RREQ โดยไม่มีเวลารอ ทำให้มีแพ็คเกตกระจายอยู่ในเครือข่ายมาก ทำให้ใช้พลังงานมากกว่า และ k = 1 เวลาที่โหนดแรกพลังงานหมดช้าที่สุด เนื่องจากมีเวลารอในการ

จึงเกิดโอเวอร์เฮดในการกระจายแพ็คเกต RREQ ในเครือข่ายน้อยกว่า ทำให้ใช้พลังงานน้อยกว่าแบบอื่น ๆ



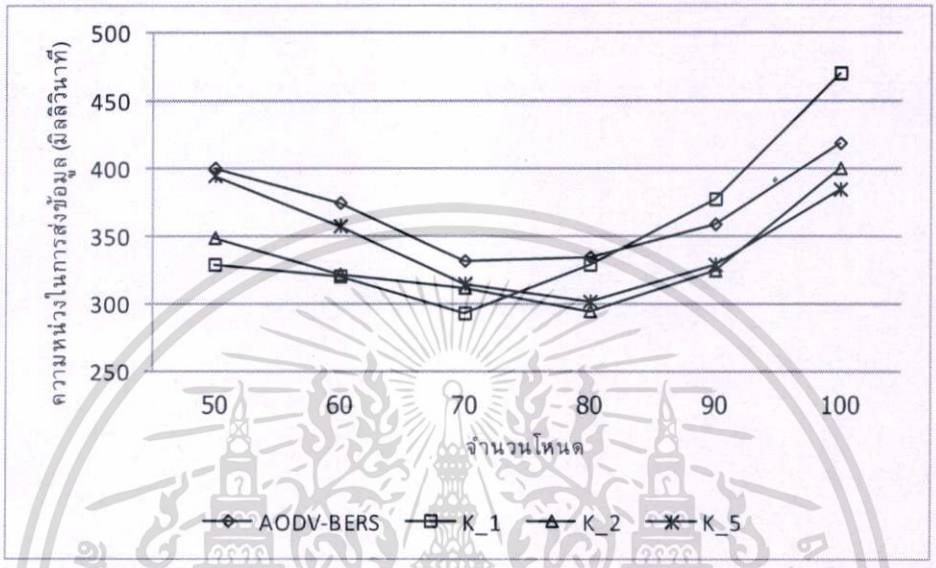
ภาพที่ 4.16 แสดงจำนวนโหนดที่เหลืออยู่ในเครือข่ายของแนวคิดที่ 1



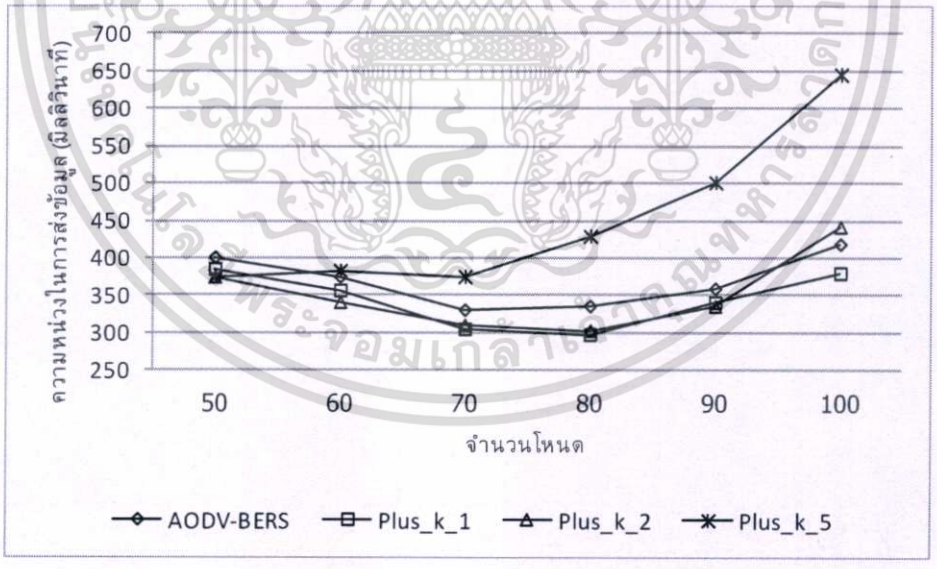
ภาพที่ 4.17 แสดงจำนวนโหนดที่เหลืออยู่ในเครือข่ายของแนวคิดที่ 2

ภาพที่ 4.16 และ ภาพที่ 4.17 แสดงจำนวนโหนดที่เหลืออยู่ในเครือข่าย จะเห็นได้ว่าทุกโพรโทคอลจะมีจำนวนโหนดเหลือในเครือข่ายมากขึ้น เมื่อมีจำนวนโหนดเพิ่มขึ้น และกราฟจะแปรผันตามกับกราฟเวลาที่โหนดแรกตาย โดยภาพที่ 4.16 แนวคิดที่ 1 (k = 5) จะมีโหนดเหลืออยู่ในเครือข่ายมากที่สุด เนื่องจากมีการกระจายแพ็คเกตค้นหาเส้นทางช้า ทำให้แพ็คเกต stop_instruction สามารถหยุดการแพร่กระจายของ RREQ ได้ ทำให้ใช้พลังงานน้อยลง ส่งผลให้สามารถทำการติดต่อสื่อสารในเครือข่ายได้นานขึ้น รองลงมาคือ k = 2 เนื่องจากมีเวลารอในการส่งแพ็คเกตเร็วกว่า k = 5 ทำให้มีการกระจายแพ็คเกต RREQ มากกว่า จึงใช้พลังงาน

มากกว่า ภาพที่ 4.17 แนวคิดที่ 2 (k =5) จะเหลือโหนดที่อยู่ในเครือข่ายน้อยที่สุด เนื่องจากมีการใช้พลังงานมากกว่าแบบอื่น ๆ ส่วน k = 2 มีจำนวนโหนดมากกว่า AODV-BERS เนื่องจากแพ็คเกต stop_instruction จะกระจายตามเงื่อนไขโดยดูจากแพ็คเกต RREQ แทนการกระจายตามจำนวนฮ็อป ทำให้มีโอกาสที่จะหยุดการแพร่กระจาย RREQ ได้มากกว่า



ภาพที่ 4.18 แสดงความหน่วงในการส่งข้อมูลของแนวคิดที่ 1

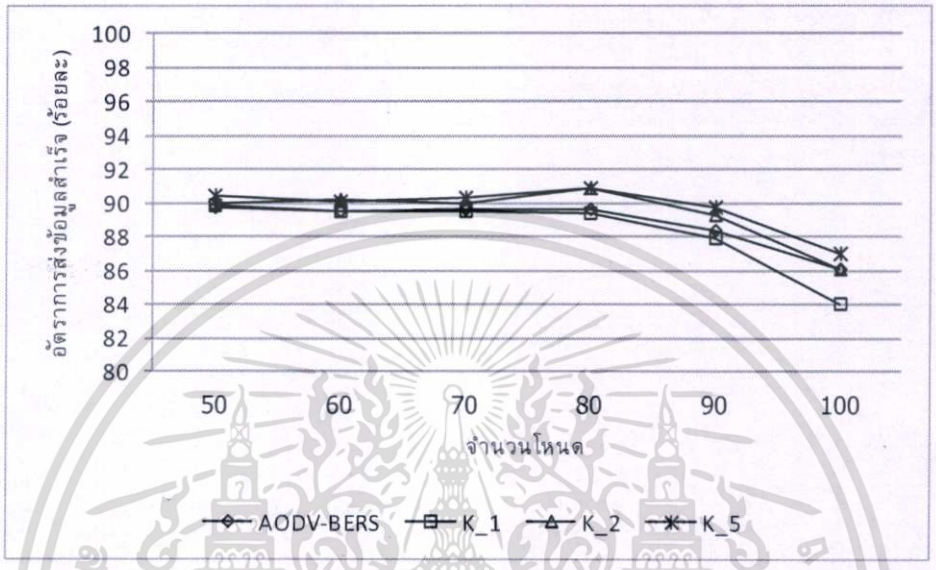


ภาพที่ 4.19 แสดงความหน่วงในการส่งข้อมูลของแนวคิดที่ 2

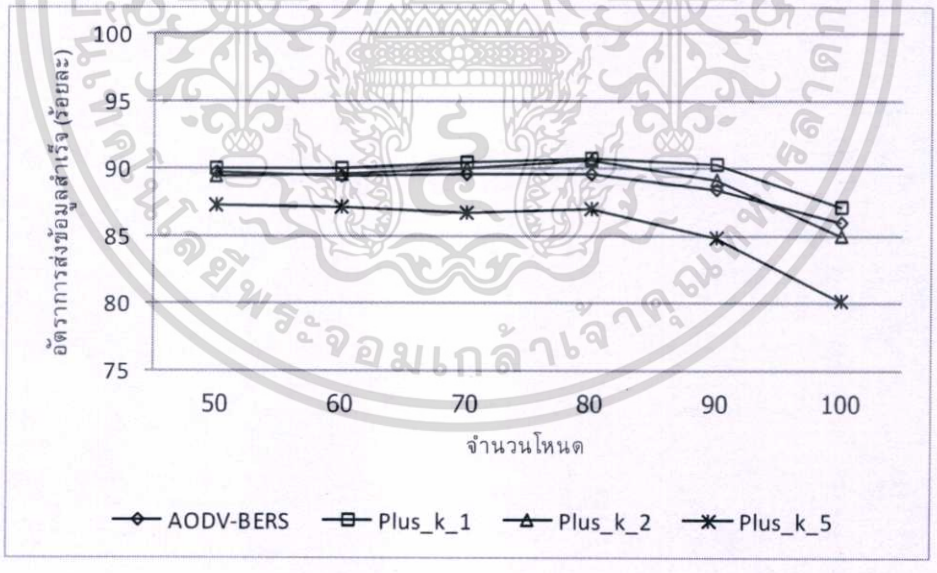
ภาพที่ 4.18 และ ภาพที่ 4.19 แสดงความหน่วงในการส่งข้อมูล จะเห็นได้ว่า เมื่อความหนาแน่นในเครือข่ายน้อย ในภาพที่ 4.18 AODV-BERS จะมีความหน่วงสูงกว่ารูปแบบอื่น ๆ เนื่องจากมีเวลารอในการกระจายแพ็คเกต RREQ มากกว่า แต่เมื่อความหนาแน่นของโหนดเพิ่มมากขึ้น แนวคิดที่ 1 (k = 1) จะมีความคับคั่งในเครือข่ายจากการกระจายแพ็คเกตค้นหาเส้นทางมากขึ้น เวลาหน่วงในการส่งข้อมูลจึงเพิ่มมากขึ้นตามไปด้วย ภาพที่ 4.19 แนวคิดที่ 2 (k = 5) มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ความหน่วงมากกว่า k = 2 เนื่องจากภายในจำนวน 5 ฮ็อปไม่มีเวลารอในการกระจายแพ็คเกต RREQ ทำให้เกิดความคับคั่งในเครือข่ายจากการกระจายแพ็คเกตมากกว่า ส่วน k = 1 และ k = 2 จะมีความหน่วงในการส่งข้อมูลน้อยกว่า AODV-BERS เนื่องจาก AODV-BERS มีเวลารอในการกระจายแพ็คเกต RREQ เพิ่มขึ้นตามจำนวนฮ็อป ทำให้ใช้เวลาในการค้นหาเส้นทางนานขึ้น



ภาพที่ 4.20 แสดงอัตราการส่งข้อมูลสำเร็จของแนวคิดที่ 1



ภาพที่ 4.21 แสดงอัตราการส่งข้อมูลสำเร็จของแนวคิดที่ 2

ภาพที่ 4.20 และภาพที่ 4.21 แสดงอัตราการส่งข้อมูลสำเร็จ จะเห็นได้ว่า กราฟแสดงอัตราการส่งข้อมูลสำเร็จแปรผันตามกราฟแสดงโอเวอร์เฮดในเครือข่าย และแปรผกผันกับกราฟแสดงความหน่วงในการส่งข้อมูล โดยภาพที่ 4.20 แนวคิดที่ 1 (k = 1) มีอัตราการส่งสำเร็จน้อยที่สุด เนื่องจากเกิดความคับคั่งในเครือข่ายมาก ส่งผลให้เกิดการชนกันของแพ็คเกต และสูญหายในที่สุด ภาพที่ 4.21 แนวคิดที่ 2 (k = 5) มีอัตราการส่งสำเร็จน้อยที่สุด เนื่องจากการกระจาย

แพ็คเกตค้นหาเส้นทางโดยไม่ต้องมีเวลารอภายใน k ฮ็อป ทำให้เกิดความคับคั่งในเครือข่ายมาก ในแนวคิดที่ 1 (k = 5) มีอัตราการส่งสำเร็จมากกว่าแนวคิดที่ 2 (k = 5) เนื่องจากแนวคิดที่ 2 ไม่มีเวลารอในการกระจายแพ็คเกต RREQ ในระยะ 5 ฮ็อป จึงมีการกระจายแพ็คเกตมากกว่า ทำให้เกิดความคับคั่งในเครือข่ายมากกว่า

เนื่องจากผู้วิจัยได้เล็งเห็นถึงปัญหาในการกระจายแพ็คเกต stop_instruction ตามจำนวนฮ็อปของโหนดปลายทาง ซึ่งอาจจะไม่สามารถหยุดการกระจายของแพ็คเกต RREQ ได้ จึงได้เสนอแนวคิดในการปรับปรุง คือ การสร้างเงื่อนไขในการกระจายแพ็คเกต stop_instruction แทนการกระจายแพ็คเกตตามจำนวนฮ็อปจากโหนดต้นทางไปยังโหนดปลายทาง อีกทั้งยังมีการกำหนดความหน่วงในการกระจายแพ็คเกต RREQ ให้แตกต่างกัน

จากการพัฒนาแนวคิดที่ 1 ภายในระยะ k ฮ็อปมีการหน่วงเวลาของการกระจายแพ็คเกต RREQ แปรผันตามจำนวนโหนดที่ห่างจากต้นทาง ทำให้แพ็คเกต stop_instruction มีโอกาสสูงที่แพ็คเกตสามารถหยุดการกระจายแพ็คเกต RREQ ได้ภายในระยะฮ็อปที่น้อย ส่วนแนวคิดที่ 2 ภายในระยะ k ฮ็อปจะไม่มีการหน่วงเวลาของการกระจายแพ็คเกต RREQ ทำให้ระยะที่แพ็คเกต stop_instruction มีระยะฮ็อปที่ไกลขึ้น ซึ่งทั้งสองแนวคิดนี้แสดงให้เห็นว่า การจำกัดขอบเขตในการหน่วงเวลาด้วยค่า k ที่เหมาะสม มีผลต่อประสิทธิภาพในด้านต่าง ๆ ทั้งโอเวอร์เฮด พลังงาน ความหน่วงในการส่งข้อมูล และอัตราการส่งสำเร็จ

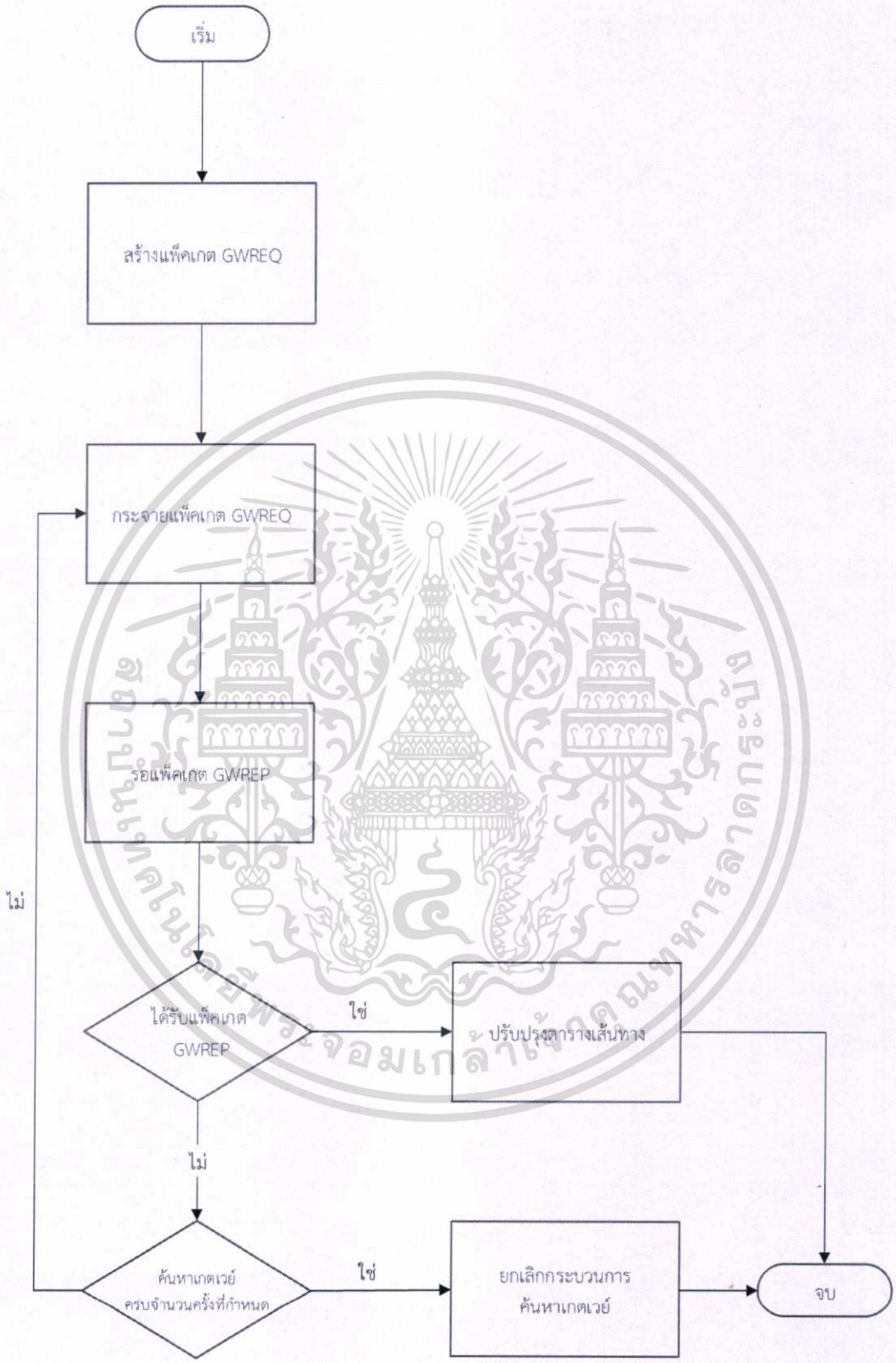
4.3 การพัฒนากลไกค้นหาเกตเวย์ที่มีโอเวอร์เฮดต่ำ

ในงานวิจัยส่วนนี้ ผู้วิจัยมีแนวคิดที่จะลดปริมาณโอเวอร์เฮดที่เกิดขึ้นจากกระบวนการค้นหาเกตเวย์เพราะในสถานการณ์จริง กลไกค้นหาเส้นทาง และกลไกค้นหาเกตเวย์จะต้องทำงานไปพร้อม ๆ กันเพื่อสร้างช่องทางการสื่อสารแบบผสมที่จะทำให้โหนดในเครือข่ายเฉพาะกิจเคลื่อนที่สามารถติดต่อสื่อสารออกไปยังระบบเครือข่ายอินเตอร์เน็ตได้ ในกรณีที่โหนดต้นทางไม่สามารถค้นหาเส้นทางไปยังโหนดปลายทางในเครือข่ายเฉพาะกิจเคลื่อนที่ได้สำเร็จ โหนดต้นทางจะเริ่มกระบวนการค้นหาเกตเวย์เพื่อจะส่งข้อความออกไปยังเครือข่ายภายนอก เนื่องจากมีความเป็นไปได้ว่าโหนดปลายทางนั้นไม่ได้อยู่ในส่วนของเครือข่ายเฉพาะกิจเคลื่อนที่ แต่อย่างไรก็ตามกระบวนการค้นหาเกตเวย์นั้นก็เป็นอีกส่วนหนึ่งที่ทำให้ปริมาณโอเวอร์เฮดในเครือข่ายเฉพาะกิจแบบผสมเพิ่มขึ้น เพื่อให้การติดต่อสื่อสารสามารถทำงานได้อย่างมีประสิทธิภาพ การลดปริมาณโอเวอร์เฮดบนกลไกการค้นหาเกตเวย์จึงเป็นเรื่องสำคัญ

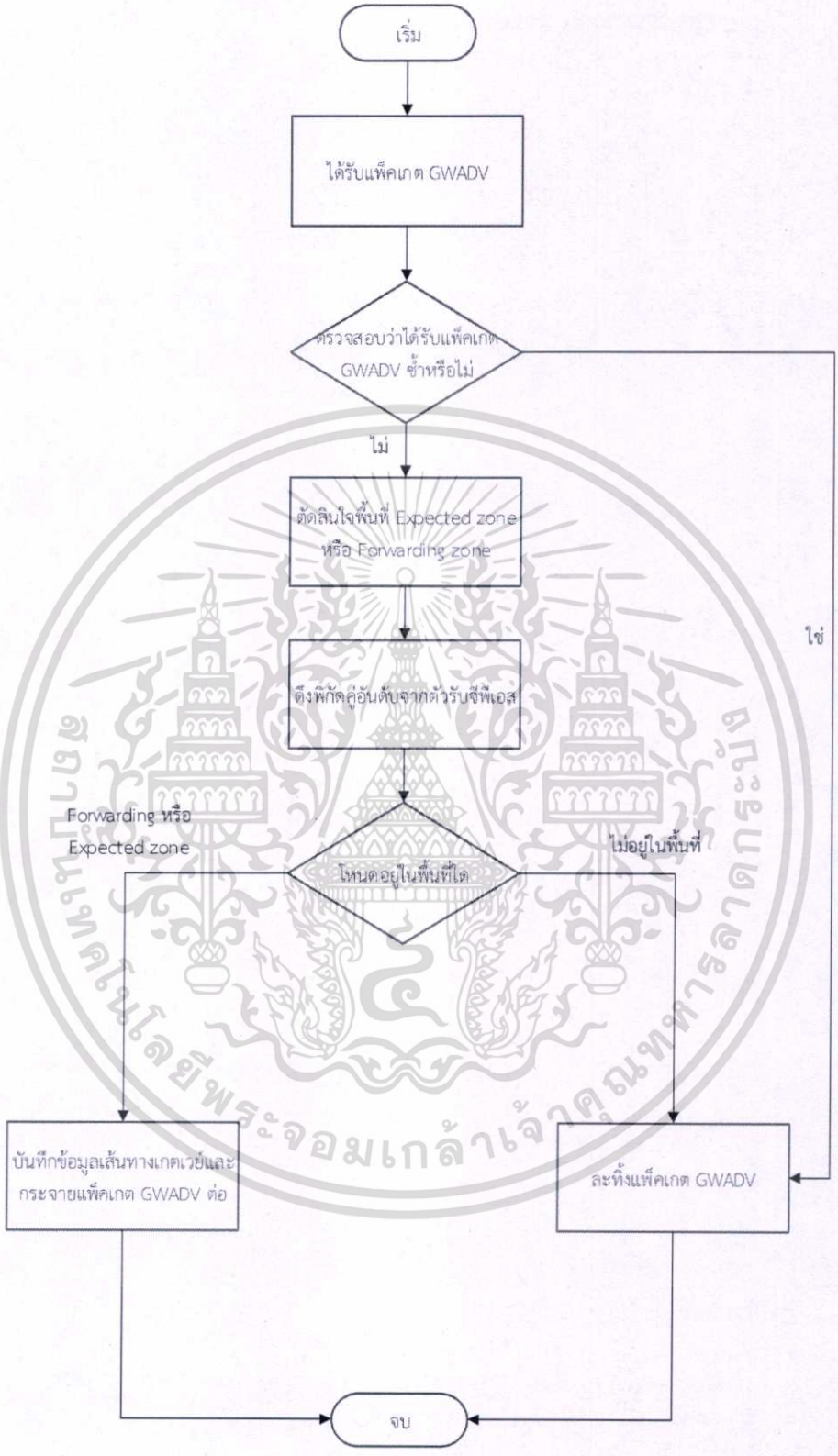
เพื่อจำลองสถานการณ์ดังกล่าวลงไปบนโปรแกรมจำลองระบบเครือข่าย ผู้วิจัยจำเป็นต้องคัดเลือกโพรโทคอลค้นหาเส้นทางสำหรับเครือข่ายเฉพาะกิจเคลื่อนที่เข้ามาเป็นส่วนหนึ่งของการจำลองเหตุการณ์ ผู้วิจัยจึงเลือกโพรโทคอล AODV เป็นโพรโทคอลค้นหาเส้นทางที่จะใช้ทดสอบร่วมกับการศึกษาการลดโอเวอร์เฮดบนโพรโทคอลค้นหาเกตเวย์ซึ่งสอดคล้องกับงานวิจัยก่อนหน้า

การศึ

และติ



ภาพที่ 4.22 แผนภาพการทำงานของกระบวนการค้นหาเกตเวย์ที่โหนดต้นทางบนเครือข่ายเฉพาะกิจแบบผสม



ภาพที่ 4.23 แผนภาพการทำงานของกระบวนการค้นหาเกตเวย์ที่โหนดระหว่างทางบนเครือข่ายเฉพาะกิจแบบผสม

ในการพัฒนากระบวนการค้นหาเกตเวย์นั้นผู้วิจัยมีแนวคิดที่จะปรับปรุงโพรโทคอลโดยนำตำแหน่งทางกายภาพของโหนดปลายทางเข้ามาช่วยจำกัดปริมาณโหนดระหว่างทางที่แพร่กระจายแพ็คเกต RREQ ที่ไม่จำเป็นเพื่อช่วยลดจำนวนโอเวอร์เฮดที่เกิดขึ้น จากการสืบค้นพบว่าโพรโทคอล LAGAD ได้ถูกนำเสนอขึ้นเพื่อลดโอเวอร์เฮดในกระบวนการค้นหาเกตเวย์โดยใช้แนวคิดพื้นฐานที่ต่อยอดมาจากโพรโทคอล LAR ดังนั้นผู้วิจัยจึงตั้งสมมติฐานว่า รูปแบบการจำกัดปริมาณโหนดระหว่างทางที่ผู้วิจัยเคยนำเสนอไว้ [21] นั้นสามารถนำมาช่วยลดโอเวอร์เฮดบนกลไกค้นหาเกตเวย์ได้เนื่องจากรูปแบบดังกล่าวมีการต่อยอดมาจากโพรโทคอล LAR เช่นเดียวกัน

รูปแบบการค้นหา และแพร่กระจายข้อมูลเกตเวย์ที่ผู้วิจัยนำเสนอนั้นมีหลักการทำงานแบบผสม โดยเริ่มต้นจากเกตเวย์ หรือเสาสัญญาณจะทำการแพร่กระจายข้อมูลเกตเวย์ (GWADV) ออกมาเป็นระยะ ๆ โดยมีค่า TTL = 1 เมื่อโหนดใด ๆ ได้รับแพ็คเกต GWADV ก็จะทำการบันทึกข้อมูลหมายเลขไอพีของเกตเวย์เอาไว้ในตารางเส้นทาง เมื่อโหนดที่ไม่ได้รับการประกาศแพ็คเกต GWADV ต้องการส่งข้อมูลหาเกตเวย์ โหนดเหล่านั้นจะมีวิธีการได้รับข้อมูลเส้นทางเพื่อติดต่อไปยังเกตเวย์ได้ดังภาพที่ 4.22

เมื่อโหนดต้นทางได้รับเส้นทางที่จะใช้ส่งไปยังเกตเวย์แล้ว ในทุก ๆ ครั้งที่โหนดต้นทางจะส่งข้อมูลไปยังเกตเวย์จะมีการเพิ่มข้อมูลความเร็วของตนเองในขณะนั้นไปพร้อมกับพิกัดจีพีเอสของตนเองเพื่อแจ้งตำแหน่งของตนเองให้เกตเวย์ทราบตลอดเวลา เมื่อเกตเวย์ได้รับข้อมูลดังกล่าวก็จะนำไปเก็บไว้ในตาราง Active node เปรียบเสมือนการลงทะเบียนใช้งานเกตเวย์ จากนั้นเกตเวย์จะเปลี่ยนการแพร่กระจายแพ็คเกต GWADV ด้วย TTL = 1 เป็นการแพร่กระจายแพ็คเกต GWADV ออกไปทั่วทั้งเครือข่ายพร้อมกับเพิ่มข้อมูลพิกัดจีพีเอสลงไปบนข้อความ โดยรายละเอียดการทำงานของโหนดระหว่างทาง และเงื่อนไขการแพร่กระจายแพ็คเกต GWADV นั้นถูกแสดงไว้ตามภาพที่ 4.23

จากภาพที่ 4.23 เกตเวย์จะสร้าง Expected zone และ Forwarding zone เพื่อใช้ในการแพร่กระจายแพ็คเกต GWADV เป็นระยะ ๆ ในส่วนของพื้นที่ Expected zone นั้นสามารถคำนวณได้จากสมการที่ 4.8 โดยพิกัด (x,y) นั้นแสดงถึงตำแหน่งของโหนดระหว่างทาง ส่วนพิกัด (x_d,y_d) แสดงถึงตำแหน่งของโหนดปลายทาง ตัวแปร r แสดงถึงรัศมีของ Expected zone ซึ่งค่ารัศมีนั้นสามารถคำนวณได้จากสมการที่ 4.9 โดยการเปลี่ยนค่าพิกัดใด ๆ ให้กลายเป็นพิกัดในรูปแบบของ (x",y") นั้นจะเกิดจากการคำนวณเปรียบเทียบกับพิกัดของโหนดต้นทางในกรณีที่แทนค่าสมการที่ 4.8 และ 4.10 มีค่าน้อยกว่า หรือเท่ากับศูนย์หมายความว่าโหนดระหว่างทางใด ๆ มีตำแหน่งอยู่ในพื้นที่ดังกล่าว ในกรณีที่โหนดระหว่างทางพบว่าตนเองอยู่ภายใน Forwarding zone หรือ Expected zone ก็จะทำการแพร่กระจายแพ็คเกต GWADV ต่อไปจนไปถึงปลายทาง แต่ถ้าไม่ใช่โหนดนั้นก็จะทำการละทิ้งแพ็คเกตนั้นไปเพื่อหยุดการแพร่กระจายแพ็คเกต GWADV ที่ไม่จำเป็น

$$g(x",y") = \left(\frac{y_d"}{(x_d"+r)^2}\right) x"^2 - y" \tag{4.8}$$

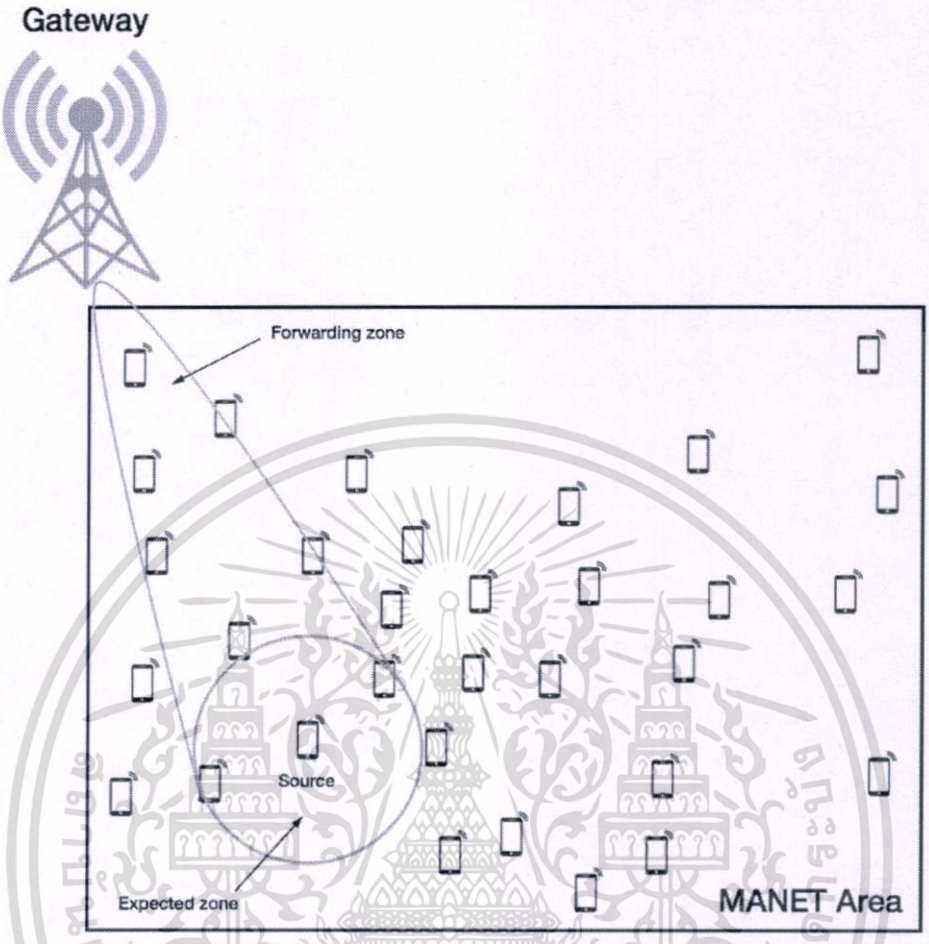
เมื่อ $f(x,y) \neq 0$

$$r = V_{avg}(t_1 - t_0) \tag{4.9}$$

$$f(x",y") = (x" - x_d")^2 + (y" - y_d")^2 - r^2 \tag{4.10}$$

เมื่อโหนดต้นทางได้รับแพ็คเกต GWADV โหนดต้นทางจะตรวจสอบหมายเลขลำดับของแพ็คเกตเปรียบเทียบกับหมายเลขลำดับของเส้นทางภายในตารางเส้นทาง ถ้าหมายเลขลำดับของแพ็คเกต GWADV มีค่ามากกว่าโหนดต้นทางจะทำการปรับปรุงเส้นทางโดยบันทึกเส้นทางที่ได้รับจากแพ็คเกต GWADV ลงไปแทนที่ ในกรณีที่มีการเปรียบเทียบค่าหมายเลขลำดับที่มีค่าเท่ากันโหนดต้นทางจะเลือกเส้นทางที่มีจำนวนฮ็อปน้อยกว่า การทำเช่นนี้เป็นการช่วยรักษาเส้นทางที่ใช้ติดต่อไปยังเกตเวย์ให้ใหม่อยู่เสมอ โดยรูปแบบของการกระจายแพ็คเกต GWADV นั้นถูกแสดงไว้ดังภาพที่ 4.24 อย่างไรก็ตามโพรโทคอลที่ผู้วิจัยนำเสนอยังคงเกิดปัญหาในกรณีที่โหนดต้นทางใด ๆ มีการเปลี่ยนแปลงความเร็ว และตำแหน่งอย่างฉับพลันจนทำให้เกตเวย์คำนวณ Expected zone ผิดพลาด ส่งผลให้การแพร่กระจายแพ็คเกต GWADV ส่งไปไม่ถึงโหนดต้นทาง และการเปลี่ยนแปลงตำแหน่งรูปแบบดังกล่าวยังทำให้เส้นทางที่ใช้ส่งไปหาเกตเวย์ไม่สามารถใช้งานได้ โหนดต้นทางจึงจำเป็นต้องเริ่มกระบวนการค้นหาเกตเวย์ใหม่อีกครั้ง ซึ่งส่งผลต่อการเพิ่มขึ้นของโอเวอร์เฮดในระบบเครือข่ายเฉพาะกิจเคลื่อนที่

เพื่อแก้ไขปัญหานี้ ผู้วิจัยจึงได้นำแนวคิด Solicitation ที่ถูกนำเสนอไว้ที่งานวิจัย [22] เข้ามาใช้งานร่วมกับโพรโทคอลที่พัฒนาขึ้น เมื่อใดก็ตามที่โหนดใด ๆ ได้รับแพ็คเกต GWREQ โหนดนั้น ๆ จะทำการตรวจสอบว่าตนเองเคยสำรองข้อมูลเส้นทางที่มีปลายทางเป็นเกตเวย์หรือไม่ โดยหลักการคัดเลือกเส้นทางสำรองของโหนดระหว่างทางยังคงใช้หลักการเดียวกับโหนดต้นทางที่ได้มีการกล่าวไว้ข้างต้น ถ้าโหนดดังกล่าวมีการสำรองเส้นทางเอาไว้ก็จะตอบกลับข้อมูลเส้นทางในรูปแบบของแพ็คเกต GWREP ไปยังโหนดที่ร้องขอได้ทันที วิธีการดังกล่าวนอกจากจะช่วยย่นระยะเวลาในการค้นหาเกตเวย์แล้ว ยังช่วยลดปริมาณโอเวอร์เฮดที่เกิดขึ้นในระบบเครือข่ายได้อย่างมาก ยิ่งไปกว่านั้นเมื่อโหนดต้นทางที่ใช้งานเส้นทางที่ถูกสำรองไว้สามารถส่งข้อมูลไปให้เกตเวย์ได้สำเร็จ เกตเวย์ก็จะทำการปรับปรุงข้อมูลพิกัดจีพีเอสของโหนดต้นทาง และคำนวณพื้นที่ Expected zone และ Forwarding zone ใหม่เพื่อแพร่กระจายแพ็คเกต GWADV ไปยังโหนดต้นทางทำให้เกิดการปรับปรุงเส้นทางอย่างต่อเนื่อง การกระทำเช่นนี้ทำให้โหนดต้นทางไม่จำเป็นต้องพึ่งพาเส้นทางที่ถูกสำรองไว้โดยไม่มีทางเลือกส่งผลให้โอกาสที่จะส่งข้อมูลไม่สำเร็จลดลงอีกด้วย



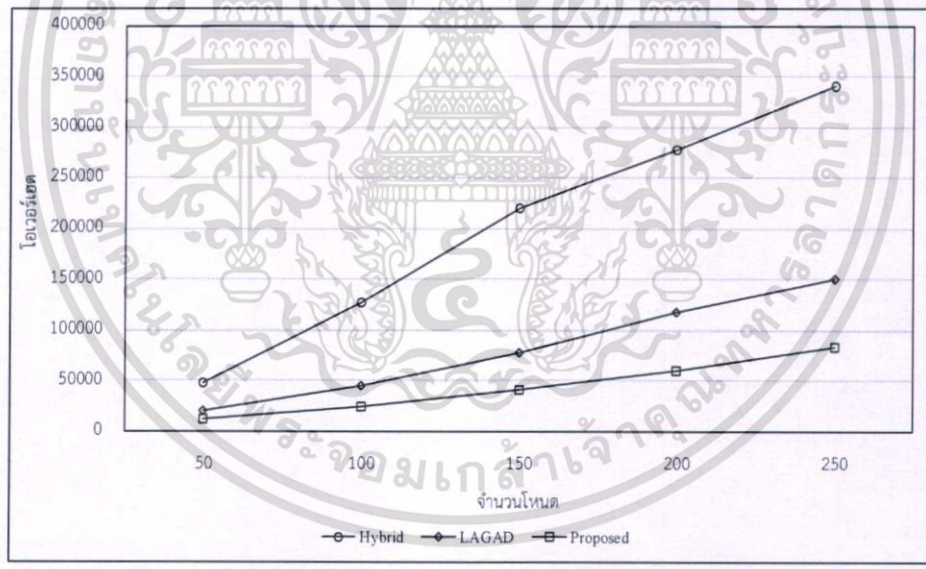
ภาพที่ 4.24 รูปแบบการแพร่กระจายแพ็คเกต GWADV ของโพรโทคอลที่นำเสนอ

เพื่อทำให้เทคนิค Solicitation สามารถทำงานได้อย่างมีประสิทธิภาพยิ่งขึ้น เมื่อเกตเวย์ไม่ได้รับข้อความจากโหนดต้นทางใด ๆ ในช่วงเวลาหนึ่ง เกตเวย์จะลบข้อมูลของโหนดนั้น ๆ ออกจากตาราง Active node และหยุดแพร่กระจายแพ็คเกต GWADV ไปยังโหนดดังกล่าว และเมื่อเวลาผ่านไปข้อมูลเส้นทางของเกตเวย์ในโหนดระหว่างทางที่ไม่ได้ถูกใช้งานจะถูกลบออกไปโดยอัตโนมัติ วิธีการดังกล่าวจะทำให้ข้อมูลเส้นทางไปยังเกตเวย์ถูกจำกัดเอาไว้เฉพาะโหนดที่กำลังใช้งานเกตเวย์เท่านั้น ส่งผลให้การตอบกลับแพ็คเกต GWREP เพื่อแจ้งเส้นทางที่ถูกสำรองเอาไว้จากโหนดระหว่างทางมีโอกาสที่จะได้รับเส้นทางที่ยังคงถูกใช้งาน และสามารถส่งข้อมูลไปหาเกตเวย์ได้ทันที

ผู้วิจัยได้มีการใช้ส่วนขยายของซอฟท์แวร์ Network Simulator 2 (NS-2) ที่มีชื่อว่า AODV+ [23] เข้ามาช่วยในการจำลองเหตุการณ์ระบบเครือข่ายเฉพาะกิจแบบผสม โดยจะนำโพรโทคอลที่ผู้วิจัยนำเสนอ (Proposed) เปรียบเทียบประสิทธิภาพกับโพรโทคอลค้นหาเกตเวย์แบบผสม (Hybrid) และโพรโทคอล LAGAD กำหนดขนาดของพื้นที่ในการจำลองเท่ากับ 1200 x 500 ตารางเมตร โดยมีเสาสัญญาณซึ่งทำหน้าที่เป็นเกตเวย์ถูกกำหนดตำแหน่งเอาไว้ที่จุดกึ่งกลางของพื้นที่ในการจำลอง ขนาดของข้อความเท่ากับ 512 ไบท์ จำนวนโหนดต้นทางทั้งหมดเท่ากับ 10 โหนด โดยจะไม่มีโหนดปลายทางใด ๆ ที่อยู่ในส่วนของเครือข่ายเฉพาะกิจเคลื่อนที่ซึ่งทำให้โหนด

ต้นทางต้องติดต่อสื่อสารผ่านเกตเวย์เท่านั้น ระยะสัญญาณของโหนดเท่ากับ 250 เมตร มีการส่งข้อมูลแบบ Constant bit rate (CBR) ด้วยอัตรา 5 แพ็คเกตต่อวินาที ความเร็วของโหนดทุกสุ่มแบบยูนิฟอร์มในช่วง 1 ถึง 20 เมตรต่อวินาที ช่วงเวลาที่โหนดหยุดเคลื่อนที่ก่อนที่จะเกิดการสุ่มความเร็ว และทิศทางใหม่ (Pause time) ถูกสุ่มแบบยูนิฟอร์มในช่วง 1 ถึง 10 วินาที จำนวนของโหนดทั้งหมดที่ใช้ในการจำลองคือ 50 ถึง 250 โหนด ระยะเวลาการจำลองเหตุการณ์เท่ากับ 900 วินาที ผลของกราฟจะถูกประมวลผลจากผลการจำลองเหตุการณ์ เหตุการณ์ละ 20 ครั้ง โดยตัวชี้วัดประสิทธิภาพของโพรโทคอลที่ทำการทดสอบมีดังต่อไปนี้

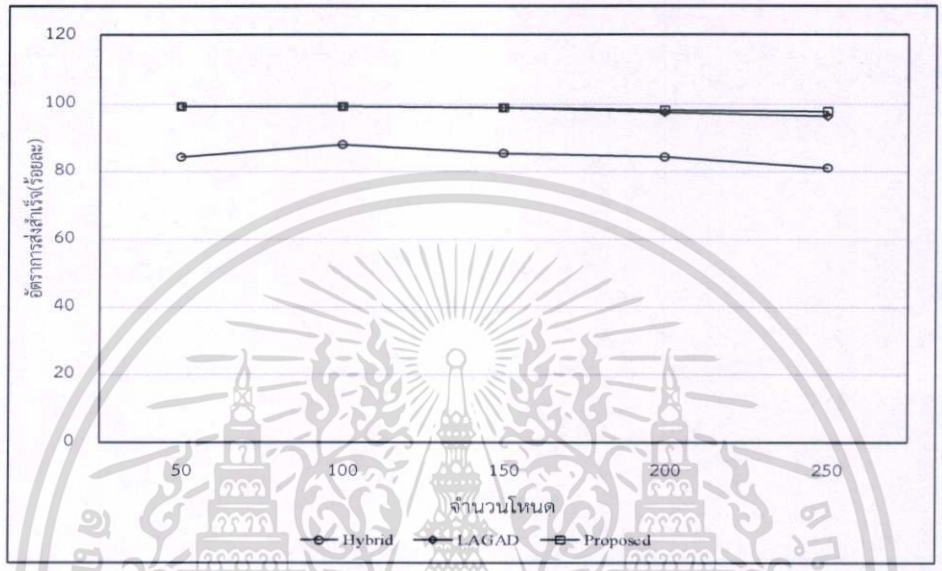
1. โอเวอร์เฮด โดยคำนวณจากปริมาณแพ็คเกตทั้งหมดที่ใช้ในการส่งข้อมูลทั้งหมดไม่ว่าจะมาจากกระบวนการค้นหาเส้นทาง หรือกระบวนการค้นหาเกตเวย์
2. อัตราการส่งสำเร็จของข้อมูลสามารถคำนวณได้จากปริมาณข้อมูลทั้งหมดที่ถูกส่งไปถึงปลายทางได้สำเร็จ
3. ความหน่วงของการส่งข้อมูลสามารถคำนวณได้จากระยะเวลาทั้งหมดที่ใช้ในการส่งข้อมูลจากโหนดต้นทางไปสู่โหนดปลายทาง
4. เวลาที่โหนดต้นทางใช้ในการค้นหาเกตเวย์หลังจากโหนดต้นทางไม่สามารถส่งข้อความไปยังเกตเวย์ด้วยเส้นทางที่มีอยู่เดิมได้ ซึ่งสามารถคำนวณได้จากระยะเวลาตั้งแต่โหนดต้นทางเริ่มต้นกระบวนการค้นหาเกตเวย์จนกระทั่งได้รับแพ็คเกต GWADV



ภาพที่ 4.25 ปริมาณโอเวอร์เฮดที่เกิดขึ้นทั้งหมดในการจำลองกระบวนการค้นหาเกตเวย์

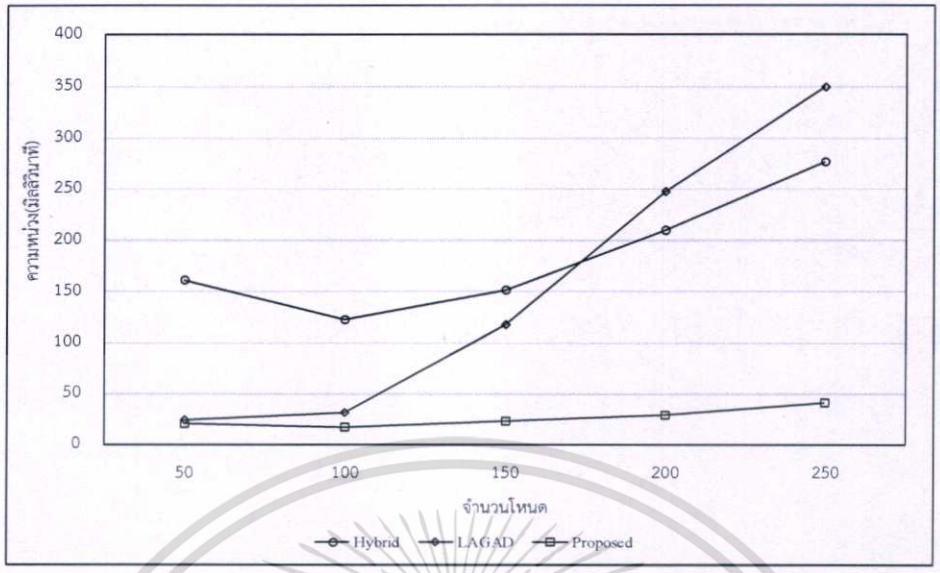
จากภาพที่ 4.25 แสดงให้เห็นถึงปริมาณโอเวอร์เฮดที่เกิดขึ้นในระบบเครือข่ายเฉพาะกิจแบบผสม โพรโทคอลค้นหาเกตเวย์แบบผสมนั้นมีปริมาณโอเวอร์เฮดมากที่สุด เนื่องจากวิธีการแพร่กระจายแพ็คเกต GWADV ในรูปแบบเชิงรุกยังไม่มีการควบคุมที่ดีพอ ในส่วนของโหนดที่ไม่ได้รับการประกาศแพ็คเกต GWADV ก็จะต้องค้นหาเส้นทางไปยังเกตเวย์ด้วยตนเอง และเมื่อเส้นทางขาด โพรโทคอลค้นหาเกตเวย์แบบผสมมีเพียงเทคนิค Solicitation เท่านั้นที่สามารถช่วยลดการแพร่กระจายแพ็คเกต GWREQ ได้ นอกจากนี้โพรโทคอลค้นหาเกตเวย์แบบผสมไม่มีกลไกรักษาหรือปรับปรุงเส้นทางใด ๆ ส่งผลให้เกิดการเรียกใช้งานกลไกค้นหาเกตเวย์บ่อยครั้ง

ในส่วนของโพรโทคอลที่นำเสนอมีค่าโอเวอร์เฮดต่ำที่สุดเนื่องจากมีการสร้างเงื่อนไขในการจำกัดปริมาณของโหนดระหว่างทางที่ทำหน้าที่แพร่กระจายแพ็คเกต GWADV ในส่วนของกระบวนการรักษาเส้นทางได้ดีกว่าโพรโทคอล LAGAD ถึงแม้เส้นทางที่ใช้ติดต่อไปยังเกตเวย์จะไม่สามารถใช้งานได้ ก็ยังมีเทคนิค Solicitation เข้ามาช่วยย่นระยะเวลา และลดอัตราการแพร่กระจายแพ็คเกต GWREQ ได้อีกด้วย



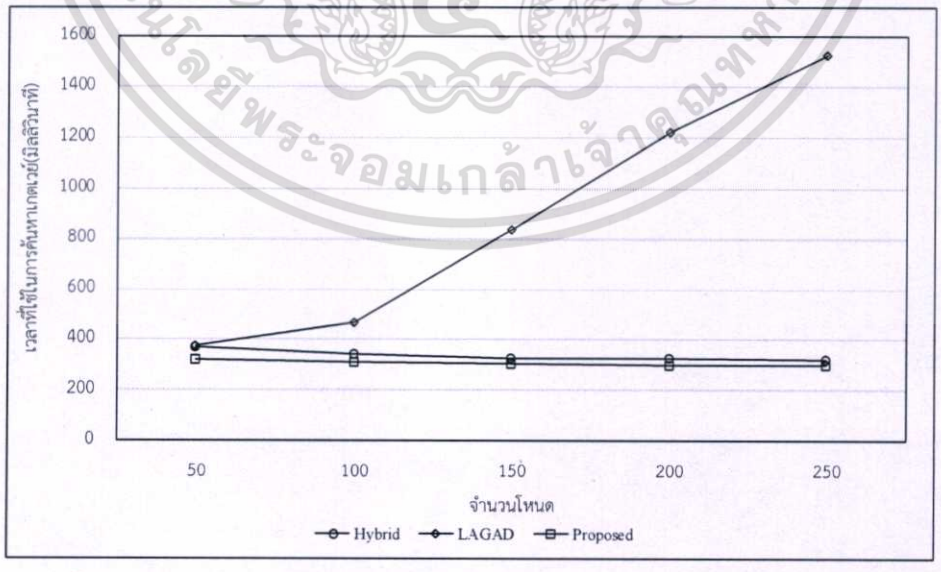
ภาพที่ 4.26 อัตราการส่งสำเร็จของข้อมูลในการจำลองกระบวนการค้นหาเกตเวย์

จากภาพที่ 4.26 แสดงให้เห็นถึงอัตราการส่งสำเร็จของข้อมูล จะสังเกตได้ว่าอัตราการส่งสำเร็จของโพรโทคอล LAGAD กับโพรโทคอลที่นำเสนอมีอัตราความสำเร็จใกล้เคียงกันเนื่องจากทั้ง 2 กลไกมีกระบวนการรักษาเส้นทางที่เหมาะสมทำให้การส่งข้อมูลไปยังเกตเวย์เป็นไปได้อย่างต่อเนื่อง ถึงแม้โพรโทคอลค้นหาเกตเวย์แบบผสมที่มีการใช้งานเทคนิค Solicitation เข้าไปช่วยลดระยะเวลาในการค้นหาเกตเวย์ใหม่ แต่เนื่องจากโหนดที่ได้รับการตอบกลับเส้นทางที่ถูกสำรองไว้จากโหนดระหว่างทางจำเป็นต้องใช้เส้นทางดังกล่าวจนกว่าเส้นทางนั้นจะไม่สามารถใช้งานได้ ซึ่งมีความเป็นไปได้ที่ข้อมูลเส้นทางที่ถูกสำรองไว้อาจจะไม่สามารถใช้งานได้ตั้งแต่ต้นหรือใช้งานได้แค่เพียงช่วงเวลาหนึ่งขึ้นอยู่กับการเคลื่อนไหว และความเร็วของโหนดอื่น ๆ ทำให้โหนดต้นทางต้องเริ่มต้นกระบวนการค้นหาเส้นทางใหม่ซ้ำ ๆ จนกระทั่งข้อมูลแพ็คเกตนั้น ๆ หมดระยะเวลาในการส่ง (Timeout) จนต้องถูกละทิ้งไปในที่สุด



ภาพที่ 4.27 ความหน่วงในการส่งข้อมูลในการจำลองกระบวนการค้นหาเกตเวย์

จากภาพที่ 4.27 แสดงให้เห็นถึงความหน่วงในการส่งข้อมูลระหว่างโหนดต้นทางถึงโหนดปลายทางซึ่งเป็นผลสืบเนื่องมาจากปริมาณโอเวอร์เฮดที่เพิ่มขึ้น ส่งผลให้การจราจรในระบบเครือข่ายเฉพาะกิจแบบผสมเกิดความคับคั่ง โพรโทคอลที่ถูกนำเสนอมีค่าโอเวอร์เฮดน้อยที่สุดทำให้มีค่าความหน่วงในการส่งข้อมูลต่ำที่สุดเช่นกัน ส่วนที่น่าสนใจคือโพรโทคอล LAGAD ที่มีอัตราการเพิ่มขึ้นของความหน่วงสูงขึ้นมากเมื่อความหนาแน่นของโหนดเพิ่มขึ้น เนื่องจากโพรโทคอล LAGAD ไม่ได้มีการนำเทคนิค Solicitation ไปใช้ส่งผลให้เวลาที่เส้นทางจากโหนดต้นทางไปยังเกตเวย์ไม่สามารถใช้งานได้ โหนดต้นทางต้องแพร่กระจายแพ็คเกต GWREQ ไปจนกว่าจะถึงเกตเวย์โดยตรง ทำให้โหนดต้นทางต้องเสียเวลามากขึ้นในส่วนนี้ส่งผลให้ค่าความหน่วงเพิ่มขึ้น



ภาพที่ 4.28 เวลาที่โหนดต้นทางใช้ในการค้นหาเกตเวย์

จากภาพที่ 4.28 แสดงถึงเวลาที่ใช้ในการค้นหาเกตเวย์ ซึ่งแสดงให้เห็นว่าโพรโทคอล LAGAD ต้องใช้เวลาในส่วนนี้มากกว่าโพรโทคอลอื่นซึ่งสอดคล้องกับเหตุผลที่ได้กล่าวมาข้างต้น ถึงแม้โพรโทคอล LAGAD จะมีความพยายามในการปรับปรุงข้อมูลเส้นทางจากการประกาศแพ็คเกต GWADV อยู่เสมอ แต่ในกรณีที่เส้นทางขาดโพรโทคอล LAGAD กลับไม่มีกระบวนการใด ๆ มารับมือกับปัญหาที่เกิดขึ้น ในส่วนของโพรโทคอลที่นำเสนอจะใช้เวลาในการค้นหาเกตเวย์ใหม่น้อยที่สุด เนื่องจากมีปริมาณโอเวอร์เฮดที่น้อยที่สุดประกอบกับการใช้เทคนิค Solicitation ได้อย่างมีประสิทธิภาพ

ผลลัพธ์จากการจำลองระบบเครือข่ายโดยรวมแสดงให้เห็นว่ากลไกดั้งเดิมที่ถูกใช้ในการค้นหาเกตเวย์บนเครือข่ายเฉพาะกิจแบบผสมนั้นมีการสร้างแพ็คเกตที่ไม่จำเป็นต่อการสื่อสารระหว่างโหนดต้นทาง และโหนดปลายทาง หรือเรียกอีกอย่างหนึ่งว่าโอเวอร์เฮดมากเกินไป ส่งผลให้ทรัพยากรพลังงาน และแบนด์วิดท์ที่มีอยู่อย่างจำกัดถูกใช้งานไปอย่างไม่มีประสิทธิภาพ เพื่อแก้ไขปัญหาดังกล่าว ผู้วิจัยได้นำเสนอโพรโทคอลใหม่ที่ประยุกต์การใช้งานเซนเซอร์จีพีเอสบนสมาร์ทโฟนเข้ามาช่วยระบุตำแหน่งทางกายภาพของโหนด เพื่อจำกัดพื้นที่ และกำหนดทิศทางในการแพร่กระจายแพ็คเกตที่ใช้ในการค้นหาเส้นทาง และแพ็คเกตที่ใช้ในการค้นหาเกตเวย์ เมื่อนำวิธีการดังกล่าวไปทดสอบ และเปรียบเทียบกับโพรโทคอลรูปแบบดั้งเดิม ผู้วิจัยพบว่าโพรโทคอลที่ผู้วิจัยนำเสนอสามารถช่วยลดปริมาณโอเวอร์เฮดในระบบเครือข่ายได้อย่างมีประสิทธิภาพ ยิ่งไปกว่านั้นโพรโทคอลที่ผู้วิจัยนำเสนอยังคงไว้ซึ่งอัตราการส่งข้อมูลสำเร็จระหว่างโหนดต้นทางและปลายทาง ช่วยยืนระยะเวลาที่ใช้ในการค้นหาเส้นทาง และลดค่าความหน่วงในการติดต่อสื่อสารได้อีกด้วย

บทที่ 5
สรุปการพัฒนา

เครือข่ายเฉพาะกิจเคลื่อนที่เป็นระบบเครือข่ายที่เชื่อมต่อกลุ่มของอุปกรณ์สื่อสารไร้สายเพื่อให้สามารถใช้ในการติดต่อสื่อสารกัน โดยที่ไม่จำเป็นต้องพึ่งพาโครงสร้างพื้นฐาน และผู้ใช้สามารถเคลื่อนที่ไปมาได้อย่างอิสระ คุณลักษณะเหล่านี้เหมาะสมที่จะนำมาใช้เป็นเครือข่ายสำรองในสถานการณ์ที่โครงสร้างพื้นฐานไม่สามารถให้บริการได้ ผู้วิจัยได้ศึกษาในส่วนของเทคโนโลยีที่เกี่ยวข้องในการพัฒนาแอพพลิเคชันสำหรับการสื่อสารบนเครือข่ายเฉพาะกิจเคลื่อนที่ และในส่วนของกลไกการทำงานของโพรโทคอลสำหรับเครือข่ายเฉพาะกิจเคลื่อนที่ ซึ่งองค์ความรู้เหล่านี้จะสามารถนำไปพัฒนาระบบสื่อสารที่ไม่พึ่งพาโครงสร้างพื้นฐานโดยใช้สมาร์ทโฟนได้

จากการศึกษาวรรณกรรมที่เกี่ยวข้อง ทำให้เกิดแนวคิดในการพัฒนาแอพพลิเคชันเพื่อสร้างเครือข่ายเฉพาะกิจสำรองสำหรับการติดต่อสื่อสารชั่วคราวที่สามารถนำไปใช้งานได้จริง โดยผู้วิจัยได้ดำเนินการพัฒนาแอพพลิเคชันต้นแบบที่สามารถสร้างการเชื่อมต่อโดยใช้โพรโทคอล Wi-Fi Direct ในการแลกเปลี่ยนข้อความระหว่างสมาร์ทโฟนในแต่ละฮ็อป และใช้แนวคิดของโพรโทคอล AODV เป็นแกนหลักในการค้นหาเส้นทางระหว่างโหนดต้นทาง และโหนดปลายทาง

ผลจากการพัฒนาแอพพลิเคชันเครือข่ายเฉพาะกิจเคลื่อนที่ และการทดสอบระบบที่ได้นำเสนอไปนั้น ทำให้ผู้วิจัยพบปัญหาด้านการกระจายแพ็คเกตค้นหาเส้นทางในปริมาณมากซึ่งเป็นโอเวอร์เฮดของระบบเครือข่าย ส่งผลให้มีความล่าช้า และเกิดการสูญหายของข้อมูล ผู้วิจัยจึงได้ศึกษา และพัฒนางานวิจัยที่มีวัตถุประสงค์เพื่อลดโอเวอร์เฮดที่เกิดขึ้นในระหว่างกระบวนการค้นหาเส้นทาง ผู้วิจัยได้นำเสนอวิธีการลดโอเวอร์เฮดโดยการปรับปรุงกระบวนการ Local Repair ของโพรโทคอล QL และวิธีการระงับการแพร่กระจายแพ็คเกตค้นหาเส้นทางที่ไม่จำเป็น ซึ่งงานวิจัยเหล่านี้สามารถลดโอเวอร์เฮดที่เกิดขึ้น ทำให้เครือข่ายมีประสิทธิภาพในด้านต่าง ๆ โดยรวมไม่ว่าจะเป็นอัตราการส่งข้อมูลสำเร็จ และความหน่วงในการส่งข้อมูลดีขึ้น

เพื่อให้งานวิจัยนี้สามารถนำไปใช้ได้จริงในปัจจุบัน ผู้วิจัยได้พัฒนากลไกการค้นหาเกตเวย์เพื่อเชื่อมโยงเครือข่ายเฉพาะกิจเคลื่อนที่เข้ากับเครือข่ายต่างประเภท คือ เครือข่ายแบบมีสาย และเครือข่ายอินเทอร์เน็ต ผลการทดสอบแสดงให้เห็นว่าแอพพลิเคชันสามารถส่งข้อความจากเครือข่ายเฉพาะกิจเคลื่อนที่ไปยังเครื่องแม่ข่ายที่เชื่อมต่อในเครือข่ายอินเทอร์เน็ตได้ ทำให้แอพพลิเคชันซึ่งเป็นผลิตภัณฑ์จากโครงการวิจัยนี้สามารถนำไปประยุกต์ใช้งานได้ทั้งการแลกเปลี่ยนข้อความปกติในสถานการณ์ทั่วไป และการติดต่อสื่อสารในสถานการณ์ภัยพิบัติสำหรับเหตุฉุกเฉิน

นอกจากนี้ ผู้วิจัยยังได้นำเสนอกลไกในการปรับปรุงประสิทธิภาพในการใช้พลังงาน เพื่อต่อยอดโครงงานวิจัยนี้ โดยได้ใช้แนวคิดในการกระจายการเลือกใช้เส้นทางในการส่งข้อมูลอย่างสมดุล เมื่อโหนดตัวกลางในเส้นทางใดพลังงานใกล้ถึงจุดวิกฤต โหนดนั้นจะส่งแพ็คเกตพิเศษกลับมาแจ้งโหนดต้นทางเพื่อให้หยุดใช้เส้นทางนั้น ซึ่งแนวคิดนี้จะมีจุดประสงค์เพื่อให้มีการกระจายการใช้พลังงานอย่างทั่วถึง ไม่ได้กระจุกตัวที่กลุ่มของโหนดใดโหนดหนึ่ง จากผลการทดลองเบื้องต้นพบว่า แนวคิดที่ได้นำเสนอนั้นสามารถกระจายการใช้พลังงานของโหนดได้ดีมากกว่าโพรโทคอลอื่น ๆ ดังแสดงในภาคผนวก ฌ. และได้รับการเผยแพร่ใน [24] ซึ่งจะช่วยให้แนวคิดเครือข่ายเฉพาะกิจเคลื่อนที่สามารถนำไปประยุกต์ใช้ในสถานการณ์จริงได้ดีมากขึ้น

บรรณานุกรม

[1] Ning Zhang. "Wi-Fi Direct based Smart Set-up (WDSS) in Lighting Systems." Master's Thesis Department of Telecommunications, Faculty of Engineering, Mathematics and Computer Science, Delft University of Technology. 2012

[2] Wi-Fi Alliance. "Mobile Ad-Hoc Networking: Wi-Fi CERTIFIED™ IBSS with Wi-Fi Protected Setup™" 2012

[3] Jimmy Tieu and Sihan Ye. "Wi-Fi Direct Services" Master's Thesis Department of Electrical and Information Technology, Faculty of Engineering, LTH, Lund University. June 2014.

[4] ARUBA network (2011). Wi-Fi Certified Passpoint Architecture for Public Access Aruba

[5] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, DOI 10.17487/RFC3561, July 2003.

[6] Sung-Ju Lee, Elizabeth M. Belding-Royer, and Charles E. Perkins. 2003. Scalability study of the ad hoc on-demand distance vector routing protocol. Int. J. Netw. Manag. 13, 2 (March 2003), 97-114.

[7] Park, J. Kim, and I. Pu, "Blocking Expanding Ring Search Algorithm for Efficient Energy Consumption in Mobile Ad Hoc Networks", Proceedings of the WONS, Les Menuires, France, pp. 185-190, 2006.

[8] Al-Rodhaan, M., Mackenzie, L., Ould-Khaoua, M. "Improvement to Blocking Expanding Ring Search for MANETs", Technical Report, University of Glasgow, 2008.

[9] R. Castenada, S. R. Das, M. K. Marina, "Query localization techniques for on-demand routing protocols for mobile ad hoc networks", ACM/Kluwer Wireless Networks (WINET) Journal, vol. 8, no. 2, pp. 137-151, Mar. 2002.

[10] P. M. Ruiz, F. J. Ros and A. Gomez-Skarmeta, "Internet connectivity for mobile ad hoc networks: solutions and challenges," in IEEE Communications Magazine, vol. 43, no. 10, pp. 118-125, Oct. 2005.

[11] J. Broch, D. A. Maltz and D. B. Johnson, "Supporting hierarchy and heterogeneous interfaces in multi-hop wireless ad hoc networks," Parallel Architectures, Algorithms, and Networks, 1999. (I-SPAN '99) Proceedings. Fourth InternationalSymposium on, Perth/Fremantle, WA, 1999, pp. 370-375.

[12] P. Ratanchandani and R. Kravets, "A hybrid approach to Internet connectivity for mobile ad hoc networks," Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE, New Orleans, LA, USA, 2003, pp. 1522-1527 vol.3.

บรรณานุกรม (ต่อ)

[13] K. Abrougui, A. Boukerche and R. W. N. Pazzi, "Location-Aided Gateway Advertisement and Discovery Protocol for VANets," in IEEE Transactions on Vehicular Technology, vol. 59, no. 8, pp. 3843-3858, Oct. 2010.

[14] Y. Ko and N.H. Vaidya. "Location-Aided Routing (LAR) in mobile Ad Hoc Networks", Journal Wireless network, vol. 6, pp. 307-321, July. 2000.

[15] วรวัชร ณรงคะชวนะ, ธนานพ ทองถาวร, และ สุเมธ ประภาวัต, "การพัฒนาเครือข่ายเฉพาะกิจสมาร์ทโฟนเพื่อการสื่อสารแบบไม่พึ่งพาโครงสร้างพื้นฐาน," ในรายงานผลการทดสอบระบบเครือข่ายไร้สายเฉพาะกิจแบบแมเน็ตบนสมาร์ทโฟนสำหรับโครงการวิจัยการพัฒนาเครือข่ายเฉพาะกิจสมาร์ทโฟนเพื่อการสื่อสารแบบไม่พึ่งพาโครงสร้างพื้นฐาน, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
Available: https://www.youtube.com/watch?v=yqKhY4jHgQ4

[16] วัชรธร หลายคุณากร, ธวัชชัย โกวิชสกุล, ธนานพ ทองถาวร, และ สุเมธ ประภาวัต, "เกตเวย์เพื่อการเชื่อมโยงเครือข่ายเฉพาะกิจสมาร์ทโฟนเข้ากับเครือข่ายอินเทอร์เน็ต," งานประชุมวิชาการระดับประเทศด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ (National Conference on Computing and Information Technology: NCCIT) ครั้งที่ 13 , กรุงเทพมหานคร, ประเทศไทย, ก.ค. 2560

[17] W. Narongkhachavana and S. Prabhavat, "Overhead Reduction for Route Repair in Mobile Ad Hoc Networks," Advances in Intelligent Systems and Computing: Recent Advances in Information and Communication Technology (IC2IT 2017), vol. 566, Springer, 2017, pp. 222-231.

[18] ชนากานต์ พันธ์แก้ว, วรวัชร ณรงคะชวนะ, และ สุเมธ ประภาวัต, "Modified BERS for Energy Efficiency in Mobile Ad Hoc Network," รายงานการประชุมวิชาการระดับประเทศด้านเทคโนโลยีสารสนเทศ (National Conference on Information Technology: NCIT) ครั้งที่ 8, กระบี่, ประเทศไทย, ต.ค. 2559.

[19] T. Thongthavorn, W. Narongkhachavana and S. Prabhavat, "Overhead reduction of location-aided gateway discovery protocols," 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 2016, pp. 1-6.

[20] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Anchorage, AK, 2001, pp. 1548-1557 vol.3.

[21] T. Thongthavorn, W. Narongkhachavana and S. Prabhavat, "A study on overhead reduction for GPS-assisted mobile ad-hoc networks," TENCON 2014 - 2014 IEEE Region 10 Conference, Bangkok, 2014, pp. 1-5.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านกา

บรรณานุกรม (ต่อ)

[22] P. Ratanchandani and R. Kravets, "A hybrid approach to Internet connectivity for mobile ad hoc networks," Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE, New Orleans, LA, USA, 2003, pp. 1522-1527 vol.3.

[23] Hamidian, Ali, Ulf Körner, and Anders Nilsson. "Performance of internet access solutions in mobile ad hoc networks." International Workshop of the EuroNGI Network of Excellence. Springer Berlin Heidelberg, 2004.

[24] M. Rungtaveesak, N. Chartkajekaew, T. Thongthavorn, W. Narongkhachavana and S. Prabhavat, "A Dynamic Routing for Load Distribution in Mobile Ad-Hoc Network," Advances in Intelligent Systems and Computing: Recent Advances in Information and Communication Technology (IC^2IT 2017), vol. 566, Springer, 2017, pp. 232–241.

ภาคผนวก ก.

ผลงานส่งเผยแพร่ในวารสารวิชาการคณะเทคโนโลยีสารสนเทศ สจล.

แอพพลิเคชันเกตเวย์สำหรับการเชื่อมต่อเครือข่ายเฉพาะกิจสมาร์ทโฟนกับอินเทอร์เน็ต

ธเนศ พุ่มไพจิตร, วรรณิศา กำมะหยี่, ธนานพ ทองถาวร,
วรวัชร ณรงคะชวนะ และ สุเมธ ประภาวัต
คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง
Emails: thanetpod@gmail.com, kim.wannisa@gmail.com, thananop@it.kmitl.ac.th,
57606007@kmitl.ac.th, sumet@it.kmitl.ac.th

บทคัดย่อ

ในปัจจุบันระบบการสื่อสารมีบทบาทสำคัญมากในการดำเนินชีวิตประจำวันของมนุษย์ หากเมื่อมีเหตุการณ์ภัยพิบัติที่ไม่คาดฝันเกิดขึ้น ทำให้โครงข่ายพื้นฐานที่เป็นส่วนสำคัญในการติดต่อสื่อสารผ่านระบบการสื่อสารไร้สายนั้นไม่สามารถใช้งานได้ จะส่งผลให้การติดต่อสื่อสารต่าง ๆ หยุดชะงักลง และไม่สามารถรับส่งข่าวสารได้ทันท่วงที โดยในปัจจุบันมีแนวคิดที่จะนำเครือข่ายเฉพาะกิจเคลื่อนที่เข้ามาช่วยแก้ไขปัญหาด้านการติดต่อสื่อสารภายใต้สถานการณ์ดังกล่าวในฐานะเครือข่ายสื่อสารสำรอง แต่ในปัจจุบันยังคงไม่มีแอพพลิเคชันใดที่สามารถสร้างเครือข่ายเฉพาะกิจเคลื่อนที่ที่สามารถติดต่อสื่อสารออกไปยังเครือข่ายอินเตอร์เน็ตได้ จากปัญหาดังกล่าวงานวิจัยนี้จึงได้มีการพัฒนาแอพพลิเคชันที่สามารถสื่อสารได้ในสถานการณ์ที่โครงข่ายพื้นฐานไม่สามารถทำงานได้ โดยมีการนำเครือข่ายเฉพาะกิจบนสมาร์ทโฟน (Smartphone Ad-Hoc Network) มาใช้สร้างเครือข่ายการสื่อสารที่ไม่จำเป็นต้องพึ่งพาโครงข่ายพื้นฐาน และพัฒนาให้สมาร์ทโฟนที่อยู่ในเครือข่ายเฉพาะกิจสามารถทำหน้าที่เป็นเกตเวย์เพื่อรับส่งข้อมูลออกไปยังเครือข่ายอื่น ๆ เช่น อินเทอร์เน็ต ทำให้สมาร์ทโฟนสามารถติดต่อสื่อสารกันได้แม้ภายในสถานการณ์ที่โครงข่ายพื้นฐานไม่สามารถใช้งานได้ ทั้งยังสามารถขยายขอบเขตการสื่อสารให้กว้างขึ้นได้อย่างมีประสิทธิภาพอีกด้วย

คำสำคัญ – เกตเวย์; เครือข่ายเฉพาะกิจ; สมาร์ทโฟน; Wi-Fi Direct

1. บทนำ

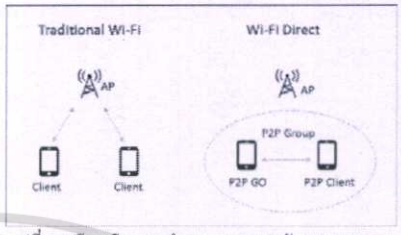
ในโลกยุคปัจจุบันที่เทคโนโลยีสารสนเทศด้านการสื่อสารแบบไร้สายเข้ามามีบทบาทในการดำเนินชีวิตของผู้คน ยกตัวอย่าง เช่น การใช้งานสมาร์ทโฟน หรือแท็บเล็ตกันอย่างแพร่หลาย โดยระบบสื่อสารแบบไร้สายจะสามารถติดต่อสื่อสารกันได้จำเป็นต้องพึ่งพาโครงข่ายพื้นฐาน (Infrastructure) ทั้งนี้ หากมีภัยพิบัติต่าง ๆ เกิดขึ้น ทำให้โครงข่ายพื้นฐานไม่สามารถใช้งานได้ ส่งผลกระทบต่อการสื่อสารของสมาร์ทโฟน และทำให้ไม่สามารถที่จะขอความช่วยเหลือจากบุคคลภายนอก ดังนั้น ระบบสื่อสารสำรองจึงเป็นสิ่งที่จำเป็นเพื่อเข้ามารองรับในสภาวะที่ระบบสื่อสารหลักไม่สามารถใช้งานได้ ผู้วิจัยจึงสนใจระบบเครือข่ายเฉพาะกิจเคลื่อนที่ (Mobile Ad-hoc Network : MANET) [1] ซึ่งการสื่อสารในระบบเครือข่ายเฉพาะกิจนั้นจะติดต่อสื่อสารกันแบบโหนดต่อโหนด

(Peer-to-Peer) โดยแต่ละโหนดจะส่งข้อความไปให้โหนดข้างเคียงที่อยู่ถัดไปเรื่อย ๆ จนถึงปลายทาง แต่การสื่อสารในเครือข่ายเฉพาะกิจนั้นมีข้อจำกัดด้านขอบเขตในการสื่อสาร โดยไม่สามารถที่จะติดต่อสื่อสารออกไปยังเครือข่ายอื่น ๆ เช่นเครือข่ายอินเทอร์เน็ตได้ จากปัญหาดังกล่าว ผู้วิจัยจึงสนใจเครือข่ายแบบ Heterogeneous Wireless Networks (HWNs) หรือที่เรียกว่าเครือข่ายไร้สายแบบผสม ซึ่งเป็นการเชื่อมต่อระหว่างเครือข่ายสองชนิดเข้าด้วยกัน ซึ่งในที่นี้หมายถึงเครือข่ายเฉพาะกิจกับเครือข่ายอินเทอร์เน็ต เพื่อให้แต่ละโหนดที่อยู่บนเครือข่ายเฉพาะกิจสามารถส่งข้อความออกไปหาโหนดที่อยู่บนอินเทอร์เน็ตได้ โดยการเชื่อมต่อของทั้งสองเครือข่ายจำเป็นต้องอาศัยกลไกเกตเวย์ที่ทำหน้าที่เปรียบเสมือนประตูเชื่อมต่อระหว่างสองเครือข่าย

ในปัจจุบันถึงแม้จะมีงานวิจัยจำนวนมากที่ให้ความสนใจและพัฒนาโพรโทคอลสำหรับการเชื่อมต่อเครือข่ายไร้สายเฉพาะกิจเข้ากับเครือข่ายอินเตอร์เน็ต แต่ยังคงไม่มีการกล่าวถึงการที่จะเชื่อมต่อเครือข่ายทั้ง 2 ชนิดเข้าด้วยกันภายใต้ข้อจำกัดที่มีอยู่ในปัจจุบันยกตัวอย่างเช่น อุปกรณ์สมาร์ทโฟน และระบบปฏิบัติการที่มีการใช้งานอยู่ในปัจจุบัน รวมไปถึงข้อจำกัดในการดัดแปลงและแก้ไขบนระบบปฏิบัติการดังกล่าว

งานวิจัยนี้จึงนำเสนอแอพพลิเคชันที่สามารถทำให้โหนดในเครือข่ายเฉพาะกิจสามารถติดต่อกับโหนดในเครือข่ายอินเทอร์เน็ตผ่านทางกลไกโหนดเกตเวย์ โดยให้โหนดที่ยังสามารถเข้าถึงโครงข่ายพื้นฐานทำหน้าที่เป็นเกตเวย์ในการเชื่อมต่อทั้งสองเครือข่ายไว้ด้วยกัน ทำให้อุปกรณ์สื่อสารที่อยู่ภายในเครือข่ายเฉพาะกิจสามารถที่จะติดต่อกันเองได้โดยไม่ต้องพึ่งพาโครงข่ายพื้นฐาน และสามารถเชื่อมต่อกับอินเทอร์เน็ตได้ผ่านทางโหนดเกตเวย์ เพื่อเพิ่มขีดความสามารถในการติดต่อสื่อสารให้มากยิ่งขึ้น โดยแอพพลิเคชั่นที่ถูกพัฒนาขึ้นสามารถใช้งานได้จริงโดยที่ไม่ต้องมีการดัดแปลงระบบปฏิบัติการบนสมาร์ทโฟน

2. เทคโนโลยีและงานที่เกี่ยวข้อง



รูปที่ 1 เปรียบเทียบการทำงานของ Wi-Fi กับ Wi-Fi Direct

2.1 Wireless Fidelity (Wi-Fi)

ในปัจจุบันเทคโนโลยีการสื่อสารไร้สายได้พัฒนาขึ้นอย่างรวดเร็ว อุปกรณ์การสื่อสารต่าง ๆ สามารถที่จะรองรับการเชื่อมต่อแบบไร้สายได้อย่างแพร่หลาย ในปี ค.ศ. 1999 ได้มีการจัดตั้งองค์กร Wi-Fi Alliance ที่เป็นองค์กรไม่แสวงหาผลกำไรขึ้น เพื่อกำหนดมาตรฐานของอุปกรณ์ที่เรียกว่า Wireless Local Area Network (WLAN) โดยใช้เครื่องหมาย Wi-Fi

WLAN คืออุปกรณ์การเชื่อมต่อแบบไร้สายที่สื่อสารกันด้วยสัญญาณวิทยุ WLAN ถูกพัฒนาขึ้นโดย Institute of Electrical and Electronic Engineer (IEEE)

Station (STA) คือ อุปกรณ์ที่รองรับมาตรฐาน IEEE 802.11 [2] เป็นเครื่องลูกข่ายที่จะเชื่อมต่อกับ Infrastructure network และ อุปกรณ์บน Ad hoc mode

Access Point (AP) คือ อุปกรณ์โครงสร้างพื้นฐานที่ถูกใช้ในการเชื่อมต่อ โดยทุก STA จะถูกเชื่อมต่อกับ AP ในการเชื่อมต่อรูปแบบ Star โดย AP จะส่งข้อมูลสื่อสารกับ STA ซึ่ง AP สามารถทำงานเป็น Bridge mode ซึ่งทำให้อุปกรณ์ที่เชื่อมต่อกับ AP สามารถที่จะสื่อสารไปยังเครือข่ายอื่นที่อยู่นอกเหนือจากที่เชื่อมต่ออยู่กับ AP

จากรูปที่ 1 การเชื่อมต่อแบบไร้สายมีรูปแบบการเชื่อมต่อสองแบบ [3] คือ Infrastructure Mode ที่มี AP

เป็นศูนย์กลางการเชื่อมต่อ และ Ad hoc mode ที่มีรูปแบบการเชื่อมต่อแบบโหนดต่อโหนด (Peer-to-Peer) โดย STA แต่ละตัวสามารถเชื่อมต่อกันได้โดยตรงไม่ต้องอาศัย AP

2.2 Wi-Fi Direct (Wi-Fi Peer-to-Peer)

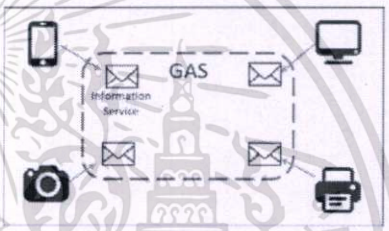
Wi-Fi Direct หรือ Wi-Fi Peer-to-Peer (P2P) มีการทำงานที่เหมือนกับเทคโนโลยี Wi-Fi แบบดั้งเดิม โดยมีส่วนที่แตกต่างคือ Wi-Fi Direct ไม่จำเป็นต้องเชื่อมต่อผ่าน AP โดยตรงแต่จะใช้ซอฟท์แวร์ AP ในการเชื่อมต่อแทน จากรูปที่ 2 การเชื่อมต่อของอุปกรณ์นั้นจะถูกจัดเป็นกลุ่ม ๆ ในแต่ละกลุ่มมีอุปกรณ์ที่ทำหน้าที่เป็น Group Owner (GO) ซึ่งใช้ซอฟต์แวร์มาทำให้ GO มีหน้าที่เหมือน AP ในการเชื่อมต่อแบบ Infrastructure Mode โดยจะมีการเชื่อมต่อเข้ามาจาก P2P Client เพื่อเข้าใช้งานกับ GO และ GO จะทำหน้าที่เป็น DHCP เพื่อแจกจ่าย IP ให้กับ P2P Client มาใช้ติดต่อกันตามรูปแบบของ OSI Network Layer หน้าที่การทำงานของ P2P Client จะทำงานเหมือนเทคโนโลยีการเชื่อมต่อแบบเก่าที่มีการใช้งาน AP ในการเชื่อมต่อ [4]

รูปที่ 2 แสดงการเชื่อมต่อระหว่างอุปกรณ์ Wi-Fi Direct

2.3 Generic Advertisement Service (GAS)

GAS [5] เป็นโพรโทคอลที่ใช้แลกเปลี่ยนข้อมูลเกี่ยวกับเซอร์วิสที่ถูกประกาศออกมาจากตัวอุปกรณ์ โดย GAS ถูกออกแบบมาให้ใช้งานบรรจุข้อมูลที่ต้องใช้ในการติดต่อสื่อสาร

โดยโพรโทคอล GAS นั้นเปรียบเสมือนช่องทางสื่อสารที่แต่ละอุปกรณ์สามารถประกาศเซอร์วิสเข้ามาในช่องทางนี้ได้ และอุปกรณ์อื่น ๆ ที่อยู่ในช่องทางสามารถที่จะนำเซอร์วิสต่าง ๆ ที่ถูกประกาศไว้ไปใช้งานต่อได้ ดังนั้นผู้วิจัยจึงได้ทำการเพิ่มชุดข้อความที่แอพพลิเคชันส่งออกไปในเครือข่ายลงไปบนชุดข้อมูลที่ใช้ในการติดต่อสื่อสาร (Information Service) ทำให้อุปกรณ์ที่อยู่ในช่องทางการสื่อสารของโพรโทคอล GAS นั้นสามารถติดต่อสื่อสารกันได้โดยตรง ดังที่แสดงแผนภาพอย่างง่ายในรูปที่ 3

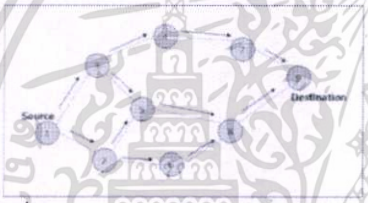


รูปที่ 3 แสดงหลักการทำงานพื้นฐานของ GAS Service

2.4 Ad hoc On-Demand Distance Vector Routing (AODV)

AODV [6] เป็นโพรโทคอลค้นหาเส้นทางในเครือข่ายเฉพาะกิจแบบรีแอกทีฟ โดยจะทำการค้นหาเส้นทางจากต้นทางไปถึงปลายทางก็ต่อเมื่อโหนดต้นทางต้องการที่จะทำการส่งข้อมูลเท่านั้น เมื่อโหนดต้นทางต้องการส่งข้อความจะทำการค้นหาเส้นทาง โดยจะส่งข้อความร้องขอเส้นทาง (Route Request Message: RREQ) ออกไปดังรูปที่ 4 และรอข้อความตอบรับการขอเส้นทาง (Route Reply Message: RREP) เมื่อโหนดที่ถูกระบุเป็นโหนดปลายทางได้รับข้อความ RREQ จะส่งข้อความ RREP ตอบกลับไปให้โหนดต้นทาง โหนดต้นทางก็จะสามารถรับรู้เส้นทางไปยังโหนดปลายทางว่าจะต้องส่งข้อมูลไปยัง

โหนดถัดไปตัวใด ที่จะทำให้สามารถส่งข้อความไปยังโหนดปลายทางได้ แต่ในกรณีที่โหนดต้นทางไม่ได้รับข้อความ RREP ตอบกลับมาภายในเวลาที่กำหนด โหนดต้นทางจะทำการกระจายข้อความ RREQ ออกไปใหม่ นอกจากนี้ยังมีกลไกการบำรุงรักษาเส้นทาง (Route Maintenance) โดยแต่ละโหนดจะทำการตรวจสอบเส้นทางที่จะส่งต่อข้อความ เมื่อพบว่าเส้นทางไม่สามารถใช้งานได้ โหนดที่ทราบจะส่งข้อความ Route Error Message (RERR) กลับไปให้โหนดต้นทางเพื่อแจ้งว่าเส้นทางนี้ไม่สามารถใช้งานได้อีก ซึ่งข้อดีของโพรโทคอล AODV คือ เป็นโพรโทคอลค้นหาเส้นทางโดยไม่เกิดลูป (Loop Free) และมีการใช้หมายเลขลำดับ (Sequence Number) เพื่อกำหนดความใหม่ของข้อมูลเส้นทาง ซึ่งหมายเลขลำดับที่มีค่าสูงกว่าคือข้อมูลเส้นทางที่ใหม่กว่า



รูปที่ 4 แสดงการกระจายข้อความ RREQ บน AODV

2.5 Gateway Discovery Protocol

ในการที่จะส่งข้อความผ่านทางเกตเวย์นั้น จะมีโหนดที่ต้องการใช้งานเกตเวย์ติดต่อไปหาโหนดเกตเวย์ก่อน โดยโหนดดังกล่าวจะต้องมีข้อมูลเส้นทางไปยังเกตเวย์ถึงจะสามารถติดต่อกับโหนดเกตเวย์ได้ ซึ่งกลไกการค้นหาเส้นทางเพื่อติดต่อไปยังโหนดเกตเวย์นั้นมีอยู่หลายวิธีการดังต่อไปนี้

2.5.1. การค้นหาเกตเวย์แบบรีแอกทีฟ

การค้นหาเกตเวย์แบบรีแอกทีฟ [7][8] เป็นการค้นหาโหนดเกตเวย์โดยเริ่มต้นจากโหนดที่ต้องการใช้งานเกตเวย์ในเครือข่ายเฉพาะกิจ โหนดดังกล่าวจะทำการสร้างข้อความ RREQ_I ซึ่งเป็นการปรับแก้ข้อความ RREQ ที่ใช้ในการร้องขอเส้นทางของโพรโทคอลค้นหาเส้นทางแบบรีแอกทีฟก่อนจะกระจายส่งออกไปให้โหนดข้างเคียง โหนดอื่น ๆ ที่ไม่ใช่โหนดเกตเวย์เมื่อได้รับข้อความ RREQ_I มาแล้วจะทำการส่งต่อไปให้โหนดข้างเคียงอีกเรื่อย ๆ จนข้อความนั้นถึงโหนดเกตเวย์ เมื่อโหนดเกตเวย์ได้รับข้อความ RREQ_I จะทำการตอบกลับไปยังโหนดต้นทางโดยใช้ข้อความ RREP_I โดยในข้อความ RREP_I จะระบุเส้นทางไปยังโหนดเกตเวย์ไว้ด้วย เมื่อโหนดต้นทางได้รับข้อความ RREP_I ก็จะสามารถทราบเส้นทางไปยังเกตเวย์ได้เหมือนโพรโทคอลค้นหาเส้นทางแบบรีแอกทีฟทั่วไป

2.5.2 การค้นหาเกตเวย์แบบโพรแอกทีฟ

การค้นหาเกตเวย์แบบโพรแอกทีฟ [7] จะเริ่มต้นที่โหนดเกตเวย์ทำการกระจายข้อความบอกข้อมูลเส้นทางไปหาตนเอง ที่เรียกว่า Gateway Advertisement (GWADV) ซึ่งเป็นข้อความที่ระบุเส้นทางไปยังโหนดเกตเวย์ให้กับโหนดที่อยู่ข้างเคียง เมื่อโหนดได้รับข้อความ GWADV มาแล้วจะทำการตัดสินใจว่าจะรับข้อมูลนั้นไว้หรือไม่ ซึ่งถ้าโหนดยังไม่รู้เส้นทางไปยังโหนดเกตเวย์จะทำการรับข้อมูลนั้นไว้แล้วนำไปสร้างข้อมูลเส้นทาง (Route entry) เพื่อเก็บลงในตารางเส้นทาง (Route table) หรือกรณีที่โหนดรู้เส้นทางไปยังโหนดเกตเวย์แล้ว แต่ข้อความที่ได้รับเข้ามามีจำนวนฮ็อปที่สั้นกว่าข้อมูลที่ตนมีอยู่ โหนดจะทำการรับข้อมูลนั้นมาปรับปรุงข้อมูลในตารางเส้นทาง หลังจากนั้นโหนดจะทำการส่งต่อข้อความไปให้กับโหนดข้างเคียงแบบนี้ไปเรื่อย ๆ ไปทั่วเครือข่าย

2.5.3 การค้นหาเกตเวย์แบบไฮบริด

การค้นหาเกตเวย์แบบไฮบริด [9] เป็นการผสมผสานระหว่างการค้นหาเกตเวย์แบบโพรแอกทีฟกับแบบรีแอก

ทีฟ โดยโหนดเกตเวย์จะทำการกระจายข้อความ GWADV ให้โหนดอื่น ๆ ที่อยู่ภายในจำนวนฮอปที่จำกัด ซึ่งสามารถตั้งค่าจำนวนฮอปได้ใน TTL-field ของ IP header เพื่อเป็นการลดโอเวอร์เฮดจากการกระจายข้อความร้องขอเส้นทางมายังโหนดเกตเวย์ เมื่อโหนดได้รับข้อความ GWADV มาแล้วจะทำการบันทึกข้อมูลเส้นทางไปยังโหนดเกตเวย์ ในกรณีที่โหนดอยู่นอกพื้นที่การกระจายข้อความ GWADV ของโหนดเกตเวย์ โหนดดังกล่าวจะทำการกระจายข้อความร้องขอเส้นทาง RREQ_I ออกไปให้โหนดข้างเคียง เมื่อโหนดข้างเคียงได้รับข้อความนั้นจะทำการตรวจสอบข้อมูลของโหนดเกตเวย์ที่ตัวมันมีอยู่ ถ้ามีข้อมูลอยู่แล้วจะทำการตอบกลับไปให้โหนดที่ร้องขอ แต่ถ้าไม่มีข้อมูลอยู่จะส่งต่อไปให้โหนดข้างเคียงไปเรื่อย ๆ เมื่อโหนดที่ร้องขอได้รับข้อมูลเกตเวย์มาแล้วก็จะสามารถหาบเส้นทางไปยังโหนดเกตเวย์ และสามารถส่งข้อความไปยังเกตเวย์ได้

3. การวิเคราะห์และออกแบบระบบ

3.1 แนวคิดการพัฒนาแอพพลิเคชัน

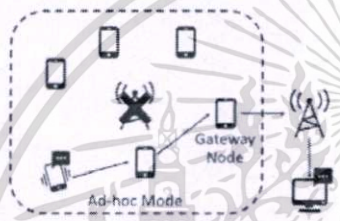
ในการที่จะใช้งานเทคโนโลยี Wi-Fi Direct ในการเชื่อมต่อสมาร์ทโฟนเข้าด้วยกันนั้น สมาร์ทโฟนสองเครื่องจำเป็นต้องผ่านกระบวนการจับคู่ หรือ Pairing เสียก่อน โดยขั้นตอนดังกล่าวจำเป็นต้องมีการตอบรับจากสมาร์ทโฟนทั้งสองเครื่องก่อนการเชื่อมต่อ ซึ่งส่งผลให้เกิดข้อจำกัดในการสร้างเครือข่ายเฉพาะกิจเคลื่อนที่ที่มีจำนวนโหนดมากกว่า 2 โหนดขึ้นไป เพื่อที่จะแก้ไขปัญหาดังกล่าวผู้วิจัยจึงได้นำเทคโนโลยี GAS ที่ได้มีการกล่าวถึงเอาไว้ข้างต้น เข้ามาเป็นช่องทางในการติดต่อสื่อสาร และส่งข้อความร่วมกับการใช้งาน Wi-Fi Direct

เมื่อโครงข่ายพื้นฐาน (Infrastructure) ไม่สามารถใช้งานได้ เครื่องที่ติดตั้งแอพพลิเคชันจะสามารถเปลี่ยนมาใช้งานบนเครือข่ายเฉพาะกิจ เมื่อสมาร์ทโฟนจะส่งข้อความจะมีการกระจายข้อความ Route request (RREQ) เพื่อตรวจสอบว่า เครื่องปลายทางอยู่ภายในเครือข่ายเฉพาะกิจเดียวกันหรือไม่ ในกรณีที่เครื่องปลายทางได้รับข้อความดังกล่าวก็จะตอบกลับด้วยข้อความ Route Reply (RREP) เพื่อแจ้งเส้นทางที่จะใช้ในการส่งข้อความ Data message แต่ถ้าเครื่องต้นทางไม่ได้รับข้อความตอบกลับภายในเวลาที่กำหนด อาจมีความเป็นไปได้ว่าเครื่องปลายทางไม่ได้อยู่ในเครือข่ายเดียวกัน เครื่องต้นทางจะทำการค้นหาเครื่องที่ทำหน้าที่เป็นโหนดเกตเวย์ ซึ่งเป็นเครื่องที่จะสามารถติดต่อสื่อสารกับโครงข่ายพื้นฐานข้างเคียงที่ยังสามารถใช้งานได้อยู่ เครื่องเกตเวย์จึงเปรียบเสมือนประตูเชื่อมต่อระหว่างสองเครือข่ายเข้าด้วยกัน โดยเครื่องเกตเวย์จะทำหน้าที่ในการส่งต่อข้อความภายในเครือข่ายเฉพาะกิจออกไปยังเครือข่ายภายนอก ในกระบวนการค้นหาเกตเวย์นั้น เครื่องต้นทางจะกระจายข้อความ Gateway Request ออกไปเพื่อค้นหาโหนดเกตเวย์ เมื่อโหนดเกตเวย์ได้รับข้อความดังกล่าว ก็จะทำการตอบกลับด้วยข้อความ Gateway Reply เพื่อแจ้งเส้นทางจากเครื่องต้นทางมายังเกตเวย์ จากกระบวนการการทำงานดังกล่าวทำให้เครื่องที่อยู่ภายในเครือข่ายเฉพาะกิจสามารถที่จะติดต่อสื่อสารกับเครื่องภายนอกเครือข่ายได้ แผนภาพสรุปการทำงานอย่างง่ายของแอพพลิเคชันที่ถูกแสดงไว้ดังรูปที่ 5

ในระหว่างการส่งข้อความ Data Message นั้น เครื่องระหว่างทางที่ทำหน้าที่ส่งต่อข้อความจะมีการตอบกลับข้อความ Acknowledgment ไปให้โหนดก่อนหน้าทุกครั้งเพื่อยืนยันว่าการส่งข้อความสำเร็จ ถ้าโหนดก่อนหน้าไม่ได้รับข้อความ Acknowledgement โหนดดังกล่าวจะพยายามส่งข้อมูลใหม่อีกครั้งหนึ่ง ซึ่งถ้าการส่งข้อมูลยังคงไม่สำเร็จภายในช่วงเวลาหนึ่ง เครื่องดังกล่าวจะทราบว่าเส้นทางในการส่งข้อมูลไม่สามารถใช้งานได้ และจะทำการค้นหาเส้นทางใหม่อีกครั้ง ในกรณีที่ไม่

สามารถหาเส้นทางใหม่ได้ในเวลาที่กำหนดโหนดดังกล่าวจะละทิ้งข้อความนั้น และหยุดกระบวนการทำงาน

ในส่วนของการจัดการและปรับปรุงเส้นทางนั้น โดยปกติโพรโทคอล AODV จะมีกลไกในการจัดการเมื่อเส้นทางที่มีอยู่เดิมไม่สามารถใช้งานได้ ซึ่งอาจเกิดขึ้นเมื่อมีการเปลี่ยนแปลงของเครือข่าย แต่ในแอพพลิเคชันรุ่นปัจจุบันจะยังไม่คำนึงเหตุการณ์ดังกล่าว เนื่องด้วยข้อจำกัดของปริมาณอุปกรณ์ที่ใช้ในการทดลองที่มีอยู่ไม่เพียงพอ

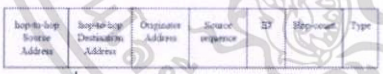


รูปที่ 5 หลักการทำงานโดยรวมของแอพพลิเคชันที่นำเสนอ

3.2 การออกแบบชุดข้อมูลที่ใช้ในแอพพลิเคชัน

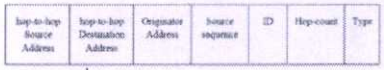
ผู้วิจัยได้ออกแบบรูปแบบข้อความที่ใช้ในแอพพลิเคชันโดยอิงพื้นฐานจากโพรโทคอล AODV โดยมีรูปแบบข้อความต่าง ๆ ดังต่อไปนี้

Route Request Message คือชุดข้อความที่ถูกใช้ในการร้องขอเส้นทางภายในเครือข่ายเฉพาะกิจ มีรูปแบบข้อความดังรูปที่ 6 โดย



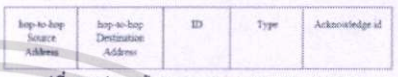
รูปที่ 6 รูปแบบข้อความของ Route Request

Route Reply Message คือชุดข้อความที่ถูกใช้ในการตอบกลับการร้องขอเส้นทางภายในเครือข่ายเฉพาะกิจ มีรูปแบบข้อความดังรูปที่ 7



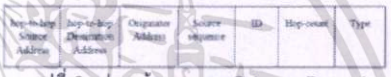
รูปที่ 7 รูปแบบข้อความของ Route Reply

Acknowledgement Message คือชุดข้อความที่ถูกส่งเพื่อยืนยันการได้รับข้อความต่าง ๆ มีรูปแบบข้อความดังรูปที่ 8



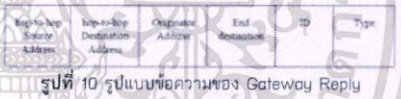
รูปที่ 8 รูปแบบข้อความของ Acknowledgement

Gateway Request Message คือ ชุดข้อความที่ถูกใช้ในการร้องขอเส้นทางไปยังเกตเวย์ มีรูปแบบข้อความดังรูปที่ 9



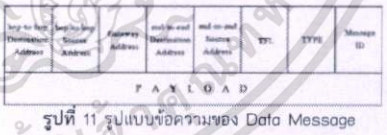
รูปที่ 9 รูปแบบข้อความของ Gateway Request

Gateway Reply Message คือ ชุดข้อความที่ถูกใช้ในการตอบกลับร้องขอเส้นทางไปยังเกตเวย์ มีรูปแบบข้อความดังรูปที่ 10



รูปที่ 10 รูปแบบข้อความของ Gateway Reply

Data Message คือ ชุดข้อความที่บรรจุข้อความที่ต้องการส่ง มีรูปแบบข้อความดังรูปที่ 11



รูปที่ 11 รูปแบบข้อความของ Data Message

โดยฟิลด์ "Type" จะถูกใช้ในการระบุประเภทของข้อความ โดยระบุเป็นหมายเลข ยกตัวอย่างเช่น Route Request Message มีค่าในฟิลด์ "Type" เท่ากับ 1 เป็นต้น

4. ผลการพัฒนาแอพพลิเคชัน

4.1 การออกแบบส่วนติดต่อผู้ใช้ของแอพพลิเคชัน

ในรูปที่ 12 เป็นหน้าแรกของแอพพลิเคชันที่ให้ผู้ใช้ทำการกรอกชื่อที่ต้องการ จากนั้นให้ทำการกดปุ่ม Login เพื่อเข้าสู่ระบบของแอพพลิเคชัน โดยแอพพลิเคชันจะใช้ชื่อที่ผู้ใช้ตั้งเพื่อเป็นการระบุตัวตนของแต่ละเครื่อง หลังจากนั้นแอพพลิเคชันจะแสดงหน้ารับส่งข้อความโดยจะแสดงชื่อ Username ไว้ด้านบนของจอภาพดังรูปที่ 13

รูปที่ 12 หน้าจอเข้าสู่ระบบของแอพพลิเคชัน

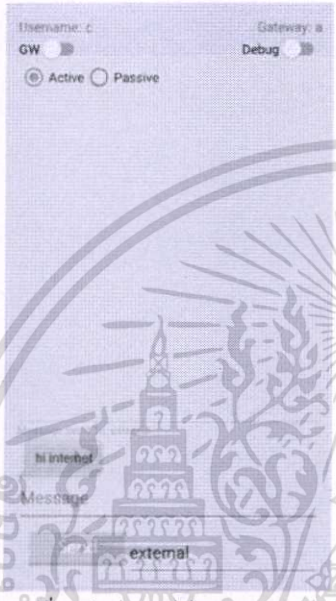
รูปที่ 13 หน้าหลักของแอพพลิเคชัน

รูปที่ 14 แสดงถึงหน้าต่างการรับ และส่งข้อความหลังจากที่ผู้ใช้เข้าสู่ระบบแล้ว ระบบจะนำผู้ใช้เข้ามาสู่หน้านี้ ซึ่งจะแสดงการส่งและรับข้อความทั้งหมดที่เกิดขึ้น ซึ่งประกอบด้วย ส่วนที่แสดงชื่อ Username ของผู้ใช้และแสดงชื่อเครื่องที่เป็นโหนดเกตเวย์ ส่วนที่อยู่ด้านบนสุดทางด้านซ้ายมือคือปุ่มที่แสดงความเป็นเกตเวย์ของเครื่อง ถ้าหากเครื่องนี้เป็นเกตเวย์ที่สามารถเชื่อมต่อไปยังอินเทอร์เน็ตได้ ผู้ใช้ต้องเปลี่ยนค่าให้เป็น ON ซึ่งจะมีให้เลือก 2 รูปแบบคือ Active ซึ่งโหนดจะทำการกระจายข้อความ GWADV โดยดัดแปลงข้อความ Route Request โดยการเปลี่ยนฟิลด์ "Type" เป็นหมายเลขอื่นเมื่อใดก็ตามที่โหนดได้รับข้อความ Route Request พิเศษดังกล่าว โหนดจะทราบว่าข้อความนั้นคือ GWADV และทำการปรับปรุงข้อมูลเกตเวย์ของตนเองเป็นระยะตามกลไกแบบโพรแอกทีฟ ในขณะที่รูปแบบ Passive จะเป็นการค้นหาเกตเวย์ในรูปแบบรีแอกทีฟซึ่งโหนดที่เป็นเกตเวย์จะรอตอบกลับข้อความ Gateway Request จากโหนดอื่น ๆ ก่อนจะตอบกลับด้วย Gateway Reply

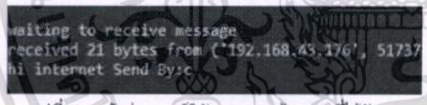
ปุ่มด้านบนสุดทางขวามือแสดงถึงปุ่มเปิดปิดการใช้งาน Debug mode ของแอพพลิเคชัน บริเวณกล่องข้อความด้านล่างคือส่วนที่ผู้ใช้สามารถส่งข้อความ โดยต้องกำหนด Destination เป็นชื่อ (Username) ของเครื่องปลายทาง และ Message คือข้อความที่ผู้ใช้ต้องการส่งไปให้เครื่องปลายทาง จากนั้นกดปุ่ม Send ข้อความจะถูกส่งออกไปให้เครื่องที่ระบุเป็น Destination

ในรูปที่ 14 และ 15 เป็นการส่งข้อความระหว่างเครื่อง c ที่ใช้งานเครือข่ายเฉพาะกิจ กับเครื่องคอมพิวเตอร์ที่มีการเชื่อมต่อกับอินเตอร์เน็ต โดยมีเครื่อง

a เป็นตัวกลางที่ทำหน้าที่เป็นเกตเวย์ เมื่อเครื่อง a ได้รับข้อความจะทำการส่งต่อข้อมูลไปยังเครื่องคอมพิวเตอร์และแสดงผลข้อความบนหน้าต่างเทอร์มินัล

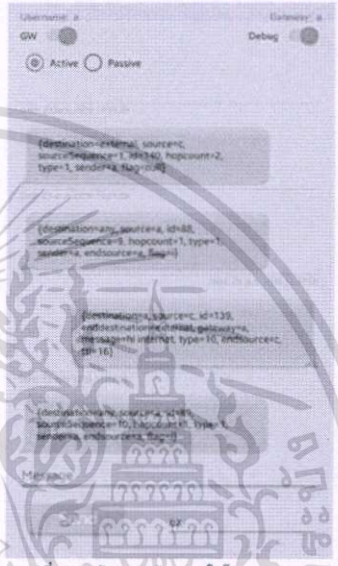


รูปที่ 14 ตัวอย่างการส่งข้อความหาเกตเวย์



รูปที่ 15 หน้าต่างเทอร์มินัลแสดงผลข้อความที่ได้รับ

ในส่วนของหน้า Debug mode (รูปที่ 16) จะเป็นการแสดงข้อความทั้งหมดที่ถูกส่งและรับเข้ามาในเครื่องของผู้ใช้ พร้อมทั้งแสดงข้อมูลทั้งหมดในแพ็คเกต ยกตัวอย่างเช่นข้อความที่ใช้ค้นหาเส้นทางสู่ปลายทางและเส้นทางสู่เกตเวย์ เป็นต้น Debug mode นั้นนอกจากจะถูกใช้งานเพื่อตรวจสอบว่าเครื่องของผู้ใช้กำลังรับและส่งข้อความใดบ้าง ในช่วงการทดสอบการทำงานของแอพพลิเคชัน Debug mode ก็เป็นส่วนสำคัญที่ใช้ในการตรวจสอบความถูกต้องของระบบอีกด้วย



รูปที่ 16 หน้าแสดงข้อความในโหมด Debug

5. สรุปผลการพัฒนาแอพพลิเคชัน

5.1 สรุปผลการพัฒนา

ในงานวิจัยฉบับนี้ได้มีการพัฒนาแอพพลิเคชันเกี่ยวกับกลไกเกตเวย์เพื่อเชื่อมต่อเครือข่ายเฉพาะกิจสมาร์ทโฟนกับเครือข่ายอินเทอร์เน็ต ซึ่งทำให้สมาร์ทโฟนที่อยู่ภายในเครือข่ายเฉพาะกิจสามารถติดต่อสื่อสารกับอุปกรณ์อื่นที่อยู่ในเครือข่ายอินเทอร์เน็ตผ่านทางสมาร์ทโฟนที่ทำหน้าที่เป็นเกตเวย์เชื่อมต่อทั้งสองเครือข่ายเข้าด้วยกันได้ ในการพัฒนาระบบผู้วิจัยได้ทำการศึกษาเกี่ยวกับโพรโท

ประยุกต์ใช้โพรโทคอล Ad hoc On-Demand Distance Vector Routing (AODV) กับแอพพลิเคชันที่พัฒนาขึ้น นอกจากนี้ยังได้ศึกษาทฤษฎีเกี่ยวกับการค้นหาเกตเวย์เพื่อนำมาพัฒนาให้แอพพลิเคชันมีการค้นหาเกตเวย์ได้สองรูปแบบคือ แบบ Active ที่อ้างอิงมาจากการค้นหาเกตเวย์แบบโพรแอกทีฟ และ แบบ Passive ที่อ้างอิงมาจากการค้นหาเกตเวย์แบบรีแอกทีฟ ซึ่งได้นำกลไกการประกาศเซอร์วิสที่เรียกว่าโพรโทคอล Generic Advertisement Service (GAS) ของ Wi-Fi Direct มาดัดแปลงเพื่อใช้ในการรับส่งข้อความ และได้มีการพัฒนาให้เครื่องคอมพิวเตอร์สามารถที่จะรับข้อความจากสมาร์ทโฟนที่อยู่ภายในเครือข่ายเฉพาะกิจได้

อย่างไรก็ตามในการพัฒนาระบบยังคงมีข้อจำกัดอยู่บางประการ โดยแอพพลิเคชั่นสามารถใช้งานได้บนสมาร์ทโฟนที่มีระบบปฏิบัติการแอนดรอยด์รุ่น 4.0 ขึ้นไป ประการต่อมาคือในการรับส่งข้อความนั้นใช้เวลาค่อนข้างนานเนื่องจากต้องคอยค้นหาข้อความตลอดเวลา และการส่งข้อความจากเครือข่ายเฉพาะกิจไปยังเครือข่ายอินเทอร์เน็ตสามารถส่งไปได้เฉพาะเครื่องที่กำหนดหมายเลข IP ไว้เท่านั้น

กิตติกรรมประกาศ

งานวิจัยนี้ได้รับการสนับสนุนทุนวิจัยงบประมาณแผ่นดิน ปีงบประมาณ 2559 จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารอ้างอิง

[1] T. Lin. " Mobile Ad-hoc Network Routing Protocols: Methodologies and Applications. " Ph.D. in Computer Engineering. 2004.

[2] Ning Zhang. "Wi-Fi Direct based Smart Set-up (WDSS) in Lighting Systems." , Delft University of Technology. 2012

[3] Wi-Fi Alliance. " Mobile Ad-Hoc Networking: Wi-Fi CERTIFIED™ IBSS with Wi-Fi Protected Setup™" 2012

[4] Jimmy Tieu and Sihan Ye. " Wi-Fi Direct Services" Information Technology,Faculty of Engineering, LTH, Lund University. June 2014.

[5] ARUBA network (2011). Wi-Fi Certified Passpoint Architecture for Public Access Aruba

[6] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, " Ad hoc On-Demand Distance Vector (AODV) Routing." Internet Engineering Task Force (IETF) draft, July 2003.

[7] H. El-Moshrify, M. A. Mangoud and M. Rizk, " Gateway Discovery in Ad hoc On-Demand Distance Vector (AODV) Routing for Internet Connectivity," Radio Science Conference, 2007. NRSC 2007. National, Cairo, 2007, pp. 1-8.

[8] Matthias Rosenschon. " Internet Gateway Discovery for Mobile Ad Hoc Networks" CITY University London. 2 October 2007.

[9] P. Ratanchandani and R. Kravets, "A hybrid approach to Internet connectivity for mobile Ad hoc networks," Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE, New Orleans, LA, USA, 2003, pp. 1522-1527 vol.3.

ภาคผนวก ข.

รายงานผลการทดสอบระบบเครือข่ายไร้สายเฉพาะกิจแบบแมเน็ตบนสมาร์ทโฟนสำหรับโครงการวิจัย
การพัฒนาเครือข่ายเฉพาะกิจสมาร์ทโฟนเพื่อการสื่อสารแบบไม่พึ่งพาโครงสร้างพื้นฐาน

รายงานผลการทดสอบ

ระบบเครือข่ายไร้สายเฉพาะกิจแบบแมเน็ตบนสมาร์ทโฟนสำหรับโครงการวิจัยการพัฒนาเครือข่ายเฉพาะกิจสมาร์ทโฟนเพื่อการสื่อสารแบบไม่พึ่งพาโครงสร้างพื้นฐาน

นาย ธนานพ ทองถาวร

ได้รับทุนสนับสนุนงานวิจัยจากเงินงบประมาณแผ่นดิน ประจำปีงบประมาณ 2559

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ละต้อง

สารบัญ

	หน้า
สารบัญ	ก
สารบัญตาราง	ข
สารบัญรูป	ค
บทที่ 1 บทนำ	1
บทที่ 2 งานวิจัยที่เกี่ยวข้อง	2
2.1 Ad Hoc On-Demand Distance Vector Routing protocol (AODV)	3
บทที่ 3 รายละเอียดการทำงานของแอพพลิเคชันต้นแบบ	5
บทที่ 4 การทดสอบและประเมินผล	9
4.1 การทดสอบความสามารถเบื้องต้นในการเชื่อมต่อบนระบบเครือข่ายเฉพาะกิจแบบแมเน็ตบนสมาร์ทโฟนสำหรับการสื่อสารในสถานการณ์ภัยพิบัติ	9
4.2 การทดสอบความสามารถของโปรโตคอลบนระบบเครือข่ายเฉพาะกิจแบบแมเน็ตบนสมาร์ทโฟนสำหรับการสื่อสารในสถานการณ์ภัยพิบัติในสถานการณ์จำลองภายในอาคาร	12
บทที่ 5 สรุปผลการทดสอบ	15

สารบัญตาราง

ตารางที่ หน้า

ตารางที่ 4.1 ข้อมูลโดยสรุปจากโปรแกรมไวร์ชาร์คที่โหนดต้นทาง 14

สารบัญภาพ

รูปที่	หน้า
ภาพที่ 2.1 แสดงตัวอย่างการสื่อสารบนเครือข่ายแบบ Ad Hoc	2
ภาพที่ 2.2 แสดงฟิลด์ข้อมูลของชุดข้อความ RREQ	4
ภาพที่ 2.3 แสดงการวิธีการกระจายข้อความ RREQ	4
ภาพที่ 3.1 หน้าหลักแอพพลิเคชัน	5
ภาพที่ 3.2 หน้าต่างแสดงผลการเรียกใช้งานแอพพลิเคชันด้วย username "A"	6
ภาพที่ 3.3 หน้าจอแสดงผลการรับส่งข้อความ	7
ภาพที่ 3.4 หน้าจอ Debug Mode	8
ภาพที่ 4.1 แสดงการส่งข้อความจากเครื่อง c ใน Debug Mode	10
ภาพที่ 4.2 แสดงการรับและส่งต่อข้อความของเครื่อง b	11
ภาพที่ 4.3 แสดงหน้าจอเครื่อง a ที่ได้รับข้อความจากเครื่อง c	11
ภาพที่ 4.4 แสดงหน้า Debug Mode ของเครื่อง a เมื่อได้รับข้อความ	12

บทที่ 1

บทนำ

เนื่องจากห้องปฏิบัติการวิจัยและพัฒนาระบบเครือข่ายอุปกรณ์โมบายล์และเซ็นเซอร์เน็ตเวิร์ค ประจำคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ได้มีการศึกษาและทำการวิจัยเกี่ยวกับระบบเครือข่ายไร้สายเฉพาะกิจแบบแมเน็ต (Mobile Ad Hoc Network - MANET) และ เครือข่ายเฉพาะกิจแบบดีทีเอ็น (Delay-Tolerant Network - DTN) มาระยะหนึ่ง โดยมีผลงานวิจัยที่ถูกตีพิมพ์และเผยแพร่ออกไปทั้งในและต่างประเทศ ซึ่งส่วนใหญ่แล้วงานวิจัยนั้นจะถูกจำลองการทำงานบนโปรแกรมจำลองเหตุการณ์ (Simulator) เพื่อเปรียบเทียบหรือวัดประสิทธิภาพของงานวิจัยซึ่งสามารถกำหนดสภาพแวดล้อม องค์ประกอบและตัวแปรต่างๆได้อย่างอิสระ

อย่างไรก็ตามเพื่อให้งานวิจัยที่ถูกพัฒนาขึ้นในห้องทดลองสามารถนำออกมาใช้งานได้จริงและสามารถเปรียบเทียบประสิทธิภาพของงานวิจัยภายใต้สภาพแวดล้อมจริง ทางห้องปฏิบัติการวิจัยได้พัฒนาระบบเครือข่ายเฉพาะกิจแบบดีทีเอ็นบนสมาร์ทโฟนในชื่อ "DTN CHAT" ซึ่งได้ถูกทดสอบประสิทธิภาพผ่านรายงาน "ระบบเครือข่ายเฉพาะกิจแบบดีทีเอ็นบนสมาร์ทโฟนสำหรับการสื่อสารในสถานการณ์ภัยพิบัติ" ในปี 2558

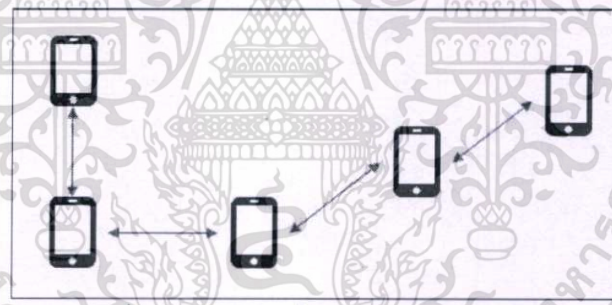
เพื่อให้การทดสอบระบบเครือข่ายไร้สายเฉพาะกิจแบบแมเน็ตสามารถถูกทดสอบภายใต้สภาพแวดล้อมจริงได้เช่นเดียวกับระบบเครือข่ายไร้สายเฉพาะกิจแบบดีทีเอ็น ทางห้องปฏิบัติการวิจัยจึงมีแนวคิดที่จะพัฒนาระบบเครือข่ายไร้สายเฉพาะกิจแบบแมเน็ตในรูปแบบของแอพพลิเคชันบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์โดยแอพพลิเคชันดังกล่าวถูกออกแบบมาให้ส่งข้อความอักษรผ่านระบบเครือข่ายไร้สายเฉพาะกิจแบบแมเน็ต ภายใต้โพรโทคอล Ad Hoc On-Demand Distance Vector Routing (AODV)

รายงานการทดสอบฉบับนี้ จะอธิบายถึงองค์ประกอบของแอพพลิเคชัน วิธีการใช้งานโดยรวม โดยมีจุดประสงค์หลักเพื่อทดสอบว่าแอพพลิเคชันดังกล่าวสามารถทำงานและส่งข้อความได้จริง รวมทั้งค้นหาข้อจำกัดหรือข้อผิดพลาดต่างๆของระบบเพื่อเป็นประโยชน์ต่อทางผู้พัฒนาต่อไป

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เครือข่ายเฉพาะกิจ (Ad Hoc Network) คือ ระบบพึ่งพาตัวเองของโหนดเคลื่อนที่แบบไร้สาย (Wireless Mobile Node) ที่สามารถเคลื่อนที่และจัดการตัวเองในขณะกำลังสื่อสารได้อย่างอิสระ ดังนั้น การเชื่อมต่อระหว่างกันของโหนดจึงมีการเปลี่ยนแปลงตลอดเวลา ด้วยเหตุนี้จึงเกิดการรวมกลุ่มของโหนด เพื่อสร้างเครือข่ายสื่อสารชั่วคราวที่ไม่จำเป็นต้องมีโครงสร้างพื้นฐาน เช่น เสากระจายสัญญาณ (Base Station) และเปลี่ยนการควบคุมของศูนย์กลางการควบคุม (Centralized Administration) เป็นการอาศัยความร่วมมือกันระหว่างโหนดในการส่งต่อข้อมูล โดยโหนดจะปฏิบัติตัวเองเป็นเสมือน เราเตอร์ (Router) ส่งต่อข้อมูลระหว่างจากโหนดหนึ่งไปยังอีกโหนดหนึ่ง ต่อเนื่องไปจนถึงโหนดปลายทาง เรียกการส่งข้อมูลลักษณะนี้ว่า มัลติฮอป (Multi-Hop) เพราะฉะนั้นเครือข่ายไร้สายเฉพาะกิจจึงมีความยืดหยุ่น เพราะสามารถสร้างได้ทุกพื้นที่และทุกเวลาที่ต้องการ



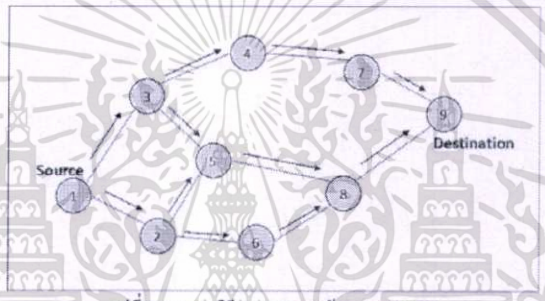
รูปที่ 2.1 แสดงตัวอย่างการสื่อสารบนเครือข่ายแบบ Ad Hoc

2.1 Ad Hoc On-Demand Distance Vector Routing (AODV)

Ad hoc On-Demand Distance Vector Routing (AODV) เป็นโพรโทคอลค้นหาเส้นทางบนเครือข่ายเฉพาะกิจแบบแมเน็ต ซึ่งเป็นโพรโทคอลดังกล่าวจะทำการค้นหาเส้นทางในการเชื่อมต่อจากต้นทางไปถึงปลายทางก็ต่อเมื่อโหนดต้องการจะส่งข้อความเท่านั้น ในการทำงานของโพรโทคอล AODV เมื่อโหนดต้นทางต้องการที่จะส่งข้อความไปยังโหนดปลายทางจะทำการค้นหาข้อมูลเส้นทาง (Routing Information) จากในตารางเส้นทาง (Routing Table) ของตนเองก่อน ถ้าหากไม่พบข้อมูลโหนดจะเริ่มต้นกระบวนการค้นหาเส้นทาง โดยโหนดต้นทางทำการกระจายข้อความร้องขอเส้นทาง (Route Request Message: RREQ) ไปให้โหนดข้างเคียงอื่น ๆ ซึ่งในข้อความ RREQ จะประกอบไปด้วย source address, destination address, source sequence number, destination sequence number และ broadcast ID ดังรูปที่ 2.2

โดยในทุก ๆ ครั้งที่โหนดต้นทางสร้างข้อความ RREQ ใหม่ ค่า broadcast ID จะมีค่าเพิ่มขึ้น ข้อมูลดังกล่าวเมื่อประกอบกับค่า destination sequence number แล้วจะมีความสำคัญมากเนื่องจากถูกใช้เพื่อกำหนดความใหม่ของเส้นทาง ทำให้ AODV เป็นโพรโทคอลค้นหาเส้นทางที่แน่ใจได้ว่าจะไม่มีการเกิดลูปขึ้นในเครือข่าย (Loop Free) เมื่อโหนดต้นทางทำการกระจายข้อความ RREQ ออกไป โหนดอื่นๆ ที่ไม่ใช่โหนดปลายทางได้รับข้อความ RREQ แล้ว จะทำการตรวจสอบ broadcast ID เพื่อตรวจสอบว่าเคยได้รับข้อความนี้หรือไม่ ถ้าโหนดเคยได้รับข้อความนี้แล้วจะทำการละทิ้งข้อความ แต่ถ้าไม่เคยได้รับโหนดจะตรวจสอบ destination sequence number ที่ได้รับมากับข้อความ RREQ ถ้าข้อความ RREQ ที่ได้รับมามี destination sequence number ที่เท่ากับหรือน้อยกว่าเมื่อเปรียบเทียบกับเส้นทางที่ถูกเก็บไว้โหนดจะทำการสร้าง RREP ตอบกลับไปเพื่ออัพเดทเส้นทางใหม่ให้แก่โหนดต้นทาง แต่ถ้าข้อความ RREQ ที่ได้รับมามี destination sequence number ที่มากกว่าก็จะทำการกระจายข้อความต่อไปจนถึงปลายทาง เมื่อโหนดปลายทางได้รับข้อความ RREQ ก็จะตอบ RREP กลับไป เมื่อโหนดต้นทางได้รับข้อความ RREP ก็จะทราบเส้นทางไปยังโหนดปลายทางและสามารถส่งข้อความหากันได้

รูปที่ 2.2 แสดงฟิลด์ข้อมูลของชุดข้อความ RREQ



รูปที่ 2.3 แสดงวิธีการกระจายข้อความ RREQ

บทที่ 3

รายละเอียดการทำงานของแอพพลิเคชันต้นแบบ

แอพพลิเคชันที่ถูกพัฒนาขึ้นมีแนวคิดพื้นฐานการออกแบบจากโปรแกรมแชททั่วไปเพียงแต่ไม่จำเป็นต้องใช้เทคโนโลยีไวไฟ (WIFI) หรือเทคโนโลยีสามจี (3G) เป็นตัวกลางในการติดต่อสื่อสาร แต่จะใช้เทคโนโลยีเครือข่ายไร้สายเฉพาะกิจแบบแมเน็ตเข้ามาแทนที่ ซึ่งการทำงานของเทคโนโลยีดังกล่าวได้มีการกล่าวถึงในบทที่ 2

รูปที่ 3.1 หน้าหลักแอพพลิเคชัน